# PROIECT

## Termen predare:
## 21-24.05.2024

**Obiectiv:**

Obiectivul acestui proiect este familiarizarea cu șablonul architectural ***Client/Server***, cu șabloanele arhitecturale orientate pe servicii (**SOA**) și cu șabloanele de proiectare.

Pentru persistența informației se va utiliza o bază de date relațională (SQL Server, MySQL, etc.).

**Cerințe:**

Transformați aplicația implementată la ***tema 3*** într-o aplicație **client/server** astfel încât să utilizați minim **5** șabloane de proiectare (minim un șablon de proiectare creațional, un șablon de proiectare comportamental și un șablon de proiectare structural) și o arhitectură orientată pe servicii **SOA** pentru comunicare între aplicația server și aplicația client.

- În ***faza de analiză*** se va realiza **diagrama cazurilor de utilizare** și **diagramele de activități** pentru fiecare caz de utilizare (***Observație***: numărul diagramelor de activități trebuie să fie ***egal*** cu numărul de cazuri de utilizare din diagrama cazurilor de utilizare).
- În ***faza de proiectare*** se vor realiza:
  - **2 diagrame de clase** corespunzătoare aplicației soft ***server*** și aplicației soft ***client*** respectând principiile **DDD** și folosind o arhitectură orientată pe servicii (**SOA**) și minim ***5*** șabloane de proiectare;
  - **diagrama entitate-relație** corespunzătoare bazei de date;
  - **diagrame de secvență** corespunzătoare tuturor cazurilor de utilizare (***Observație***: numărul diagramelor de secvență trebuie să fie cel puțin ***egal*** cu numărul de cazuri de utilizare din diagrama cazurilor de utilizare).
- În ***faza de implementare*** se va scrie cod pentru îndeplinirea tuturor funcționalităților precizate de diagrama cazurilor de utilizare utilizând:
  - proiectarea dată de diagramele de clase și diagramele de secvență;
  - unul dintre următoarele limbaje de programare: C#, C++, Java, Python.
- Finalizarea temei va consta în predarea unui director ce va cuprinde:
  - Un fișier cu diagramele UML realizate;
  - Baza de date;
  - Aplicația soft;
  - Documentația (minim 20 pagini) - un fișier care cuprinde:
    - numele studentului, grupa;
    - enunțul problemei;
    - instrumente utilizate;
    - justificarea limbajului de programare ales;
    - descrierea diagramelor UML (inclusiv figuri cu diagramele UML realizate);
    - descrierea aplicației (inclusiv figuri reprezentând interfețele grafice ale aplicației client).
- Arhiva se va trimite la adresa de email: anca.iordan@cs.utcluj.ro. Denumirea arhivei va respecta următoarea structură: **Proiect_NumePrenumeStudent**.

### *Problema 1*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-un **muzeu**. Aplicația va avea 3 tipuri de utilizatori: vizitator al muzeului, angajat al muzeului și administrator.

Utilizatorii de tip **vizitator** pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei tuturor operelor de artă expuse în muzeu sortată după artist (vizualizarea include și redarea unor imagini cu operele de artă; între 1 și 3 imagini pentru fiecare operă de artă);
- ❖ Vizualizarea listei tuturor artiștilor care au expuse opere de artă în muzeu (pentru fiecare artist se va afișa numele, data nașterii, locul nașterii, naționalitatea, o fotografie și lista tuturor operelor de artă realizate de artist și expuse în acest muzeu);
- ❖ Filtrarea listei operelor de artă plastică după următoarele criterii: artist, tipul operei de artă, etc.;
- ❖ Căutarea unei opere de artă după titlu,
- ❖ Căutarea unui artist după nume.

Utilizatorii de tip **angajat** pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip vizitator;
- ❖ Operații CRUD în ceea ce privește persistența operelor de artă expuse și a artiștilor;
- ❖ Salvare liste cu operele de artă expuse în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de operele de artă din muzeu 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 după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 2*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-un **lanț de farmacii**. Aplicația va avea 3 tipuri de utilizatori: angajat al unei farmacii, manager al lanțului de farmacii și administrator.

Utilizatorii de tip **angajat** al unei farmacii pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor medicamentelor din farmacia unde lucrează sortată după următoarele criterii: denumire, preț, dată valabilitate (vizualizarea include și redarea unor imagini cu medicamentele; între 1 și 3 imagini pentru fiecare medicament);
- ❖ Filtrarea medicamentelor din farmacia unde lucrează după următoarele criterii: disponibilitate, valabilitate, preț, producător;
- ❖ Căutarea unui medicament după denumire astfel:
  - ➢ În farmacia unde lucrează afișându-se toate informațiile despre medicament în cazul în care acesta este disponibil în acea farmacie;
  - ➢ Dacă nu este disponibil, să se caute acel medicament și în celelalte farmacii aparținând acelui lanț de farmacii, afișându-se doar lista farmaciilor unde este disponibil.
- ❖ Vânzarea unui medicament;
- ❖ Adăugarea și actualizarea informațiilor despre medicamente din farmacia la care lucrează acel angajat (***doar*** stoc, data valabilitate);
- ❖ Salvarea listei medicamentelor din farmacia unde este angajat în mai multe formate: csv, json, xml, doc.

Utilizatorii de tip **manager** pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor medicamentelor sortată după următoarele criterii: denumire, preț, dată valabilitate (vizualizarea include și redarea unor imagini cu medicamentele);
- ❖ Filtrarea produselor după următoarele criterii: farmacie, disponibilitate, valabilitate, preț, producător;
- ❖ Căutarea unui medicament după denumire, afișându-se doar lista farmaciilor unde este disponibil;
- ❖ Operații CRUD în ceea ce privește persistența medicamentelor (adăugare sau actualizare informații referitoare la denumire, producător, imagini și preț);
- ❖ Salvare liste cu situația medicamentelor din lanțul de farmacii în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de medicamentele din lanțul de farmacii utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatori;
- ❖ Vizualizarea listei tuturor utilizatorilor;
- ❖ Filtrarea listei utilizatorilor după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 3*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-un **lanț de magazine de parfumuri**. Aplicația va avea 3 tipuri de utilizatori: angajat al unui magazin de parfumuri, manager al lanțului de magazine de parfumuri și administrator.

Utilizatorii de tip **angajat** al unui magazin de parfumuri pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor parfumurilor din parfumeria la care este angajat sortată după următoarele criterii: denumire și preț (vizualizarea include și redarea unor imagini cu parfumurile; între 1 și 3 imagini pentru fiecare parfum);
- ❖ Filtrarea parfumurilor după următoarele criterii: producător, disponibilitate, preț;
- ❖ Căutarea unui parfum după denumire astfel:
  - ➢ În parfumeria unde lucrează afișându-se toate informațiile despre parfum în cazul în care acesta este disponibil în acea parfumerie;
  - ➢ Dacă nu este disponibil, să se caute acel parfum și în celelalte parfumerii aparținând acelui lanț de parfumerii, afișându-se doar lista parfumeriilor unde este disponibil.
- ❖ Vânzarea unui parfum;
- ❖ Adăugarea și actualizarea informațiilor despre parfumurile din parfumeria la care lucrează acel angajat (***doar*** stoc);
- ❖ Salvare liste cu parfumuri din parfumeria la care este angajat în mai multe formate: csv, json, xml, doc.

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

- ❖ Vizualizarea listei tuturor parfumurilor dintr-o parfumerie selectată sortată după următoarele criterii: denumire și preț (vizualizarea include și redarea unor imagini cu parfumurile);
- ❖ Filtrarea parfumurilor după următoarele criterii: parfumerie, producător, disponibilitate;
- ❖ Căutarea unui parfum după denumire, afișându-se doar lista parfumeriilor unde este disponibil;
- ❖ Operații CRUD în ceea ce privește persistența parfumurilor (adăugare sau actualizare informații referitoare la denumire, producător, imagini, preț, volum sticlă);
- ❖ Salvare liste cu situația parfumurilor din lanțul de parfumerii în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de parfumurile din lanțul de parfumerii utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatori;
- ❖ Vizualizarea listei tuturor utilizatorilor;
- ❖ Filtrarea listei utilizatorilor după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 4*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-un **lanț de florării**. Aplicația va avea 3 tipuri de utilizatori: angajat al unei florării, manager al lanțului de florării și administrator.

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

- ❖ Vizualizarea listei tuturor florilor din florăria unde lucrează sortată după culoare și preț (vizualizarea include și redarea unor imagini cu florile; între 1 și 3 imagini pentru fiecare floare);
- ❖ Filtrarea florilor din florăria unde lucrează după următoarele criterii: preț, culoare, stoc;
- ❖ Căutarea unei flori după denumire astfel:
  - ➢ În florăria unde lucrează afișându-se toate informațiile despre floare în cazul în care aceasta este disponibilă în acea florărie;
  - ➢ Dacă nu este disponibilă, să se caute acea floare și în celelalte florării aparținând acelui lanț de florării, afișându-se doar lista florăriilor unde este disponibilă, iar pentru fiecare florărie culorile în care este disponibilă.
- ❖ Vânzarea unei flori;
- ❖ Adăugarea și actualizarea informațiilor despre florile din florăria la care lucrează acel angajat (***doar*** stocul corespunzător fiecărei culori);
- ❖ Salvare liste cu situația florilor din florăria unde este angajat în mai multe formate: csv, json, xml, doc.

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

- ❖ Vizualizarea listei tuturor florilor din lanțul de florării selectată sortată după culoare și preț (vizualizarea include și redarea unor imagini cu florile; între 1 și 3 imagini pentru fiecare floare);
- ❖ Filtrarea florilor după următoarele criterii: florarie, preț, culoare, stoc;
- ❖ Căutarea unei flori după denumire, afișându-se doar lista florăriilor unde este disponibilă, iar pentru fiecare florărie culorile în care este disponibilă;
- ❖ Operații CRUD în ceea ce privește persistența florilor (adăugare sau actualizare informații referitoare la denumire, imagini, preț și culori aferente);
- ❖ Salvare liste cu situația florilor din lanțul de florării în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de florile din lanțul de florării utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatori;
- ❖ Vizualizarea listei tuturor utilizatorilor;
- ❖ Filtrarea listei utilizatorilor după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 5*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată în **bibliotecile unei universități**. Aplicația va avea 3 tipuri de utilizatori: abonat al bibliotecilor universității, bibliotecar și administrator.

Utilizatorii de tip **abonat** pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei tuturor cărților dintr-o bibliotecă selectată sortată după domeniu și autor (vizualizarea include și redarea unei imagini cu fiecare ediție a cărții);
- ❖ Filtrarea listei cărților după următoarele criterii: domeniu, editura, autor;
- ❖ Căutarea unei cărți după titlu, afișându-se lista bibliotecilor unde este disponibilă.

Utilizatorii de tip **bibliotecar** pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise utilizatorilor de tip abonat;
- ❖ Împrumutarea unei cărți unui abonat și returnarea unei cărți de către un abonat;
- ❖ Operații CRUD în ceea ce privește persistența cărților din biblioteca unde lucrează;
- ❖ Salvare liste cu situația cărților din biblioteca unde lucrează în mai multe formate: csv, json, xml, doc;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii de tip abonat și fișele acestor abonați;
- ❖ Salvarea fișei unui abonat în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de cărțile din biblioteca selectată 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 abonat;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare;
- ❖ Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- ❖ Filtrarea listei utilizatorilor după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 6*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-un **lanț de librării**. Aplicația va avea 3 tipuri de utilizatori: angajat al unei librării, manager al lanțului de librării și administrator.

Utilizatorii de tip **angajat** pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor cărților din librăria unde lucrează sortată după autor și titlu (vizualizarea include și redarea unei imagini cu fiecare ediție a cărții);
- ❖ Filtrarea listei cărților din librăria unde lucrează după următoarele criterii: domeniu, editura, autor, preț;
- ❖ Căutarea unei cărți după titlu astfel:
  - ➢ În librăria unde lucrează afișându-se toate informațiile despre carte în cazul în care aceasta este disponibilă în acea librărie;
  - ➢ Dacă nu este disponibilă, să se caute acea carte și în celelalte librării aparținând acelui lanț de librării, afișându-se doar lista librăriilor unde este disponibilă, iar pentru fiecare librărie edițiile disponibile.
- ❖ Vânzarea unei cărți din librăria la care lucrează acel angajat;
- ❖ Adăugarea și actualizarea informațiilor despre cărțile din librăria la care lucrează acel angajat (***doar*** stocul corespunzător fiecărei ediții);
- ❖ Salvare liste cu cărțile din librăria la care este angajat în mai multe formate: csv, json, xml, doc.

Utilizatorii de tip **manager** pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor cărților dintr-o librărie selectată sortată după autor și titlu (vizualizarea include și redarea unei imagini cu fiecare ediție a cărții);
- ❖ Filtrarea listei cărților după următoarele criterii: librărie, domeniu, disponibilitate, editura, autor, preț;
- ❖ Căutarea unei cărți după titlu, afișându-se doar lista librăriilor unde este disponibilă, iar pentru fiecare carte edițiile în care este disponibilă;
- ❖ Operații CRUD în ceea ce privește persistența cărților (adăugare sau actualizare informații referitoare la titlu, autor, domeniu) și a edițiilor acestora(isbn, an publicare, imagine copertă și prețul unui exemplar);
- ❖ Salvare liste cu situația cărților din toate librăriile în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de cărți utilizând grafice (structură radială, structură inelară, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatori;
- ❖ Vizualizarea listei tuturor utilizatorilor;
- ❖ Filtrarea listei utilizatorilor după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 7*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-un **lanț de galerii de artă**. Aplicația va avea 3 tipuri de utilizatori: vizitator al galeriei de artă, angajat al lanțului de galerii de artă și administrator.

Utilizatorii de tip **vizitator** pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei tuturor operelor de artă expuse în galeriile de artă sortată după anul realizării (vizualizarea include și redarea unor imagini cu operele de artă; între 1 și 3 imagini pentru fiecare operă de artă);
- ❖ Vizualizarea listei tuturor artiștilor care au expuse opere de artă în galeriile de artă (pentru fiecare artist se va afișa numele, data nașterii, locul nașterii, naționalitatea, o fotografie și lista tuturor operelor de artă realizate de artist și expuse în galeriile lanțului de galerii de artă);
- ❖ Filtrarea listei operelor de artă plastică după următoarele criterii: artist, tipul operei de artă;
- ❖ Căutarea unei opere de artă după titlu;
- ❖ Căutarea unui artist după nume.

Utilizatorii de tip **angajat** al unei galerii 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 și a artiștilor;
- ❖ Vânzarea unei opere de artă din galeria de artă la care lucrează acel angajat;
- ❖ Salvare liste cu situația operelor de artă în mai multe formate: csv, json, xml, doc.
- ❖ Vizualizarea unor statistici legate de operele de artă 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 după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 8*

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

Fără autentificare, aplicația va permite:

- Vizualizarea listei specializărilor la care se oferă consultații în acel cabinet medical, iar pentru fiecare specializare a listei medicilor;
  - La selectarea unui medic se vor afișa informații despre medicul selectat (CV, fotografie, program de lucru);
- Căutarea unui medic după nume și vizualizarea de informații despre acel medic (CV, fotografie, program de lucru).

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 din lista propriilor pacienți;
- 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 actualizari legate de simptome, diagnostic, tratament);
- Filtrarea pacienților după următoarele criterii: medic, diagnostic, vârstă;
- Căutarea unui pacient după nume;
- Planificarea pacienților pentru consultații;
- Operații CRUD în ceea ce privește persistența medicilor;
- Salvare liste cu pacienți sau medici în mai multe formate: csv, json, xml, doc:
- Vizualizarea unor statistici legate de pacienți utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatori;
- Vizualizarea listei tuturor utilizatorilor;
- Filtrarea listei utilizatorilor după tipul utilizatorilor;
- Notificarea fiecărui utilizator prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 9*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-un **lanț de magazine de încălțăminte**. Aplicația va avea 3 tipuri de utilizatori: angajat al unui magazin de încălțăminte, manager al lanțului de magazine de încălțăminte și administrator.

Utilizatorii de tip **angajat** pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor produselor de încălțăminte din magazinul unde lucrează sortată după preț (vizualizarea include și redarea unor imagini cu produsele de încălțăminte; între 1 și 3 imagini pentru fiecare produs de încălțăminte);
- ❖ Filtrarea produselor de încălțăminte din magazinul unde lucrează după următoarele criterii: producător, disponibilitate, preț;
- ❖ Căutarea unui produs de încălțăminte după model și mărime astfel:
  - ➢ În magazinul unde lucrează afișându-se toate informațiile despre produs în cazul în care acest model este disponibil în acel magazin;
  - ➢ Dacă nu este disponibil, să se caute acel model și în celelalte magazine aparținând acelui lanț de magazine de încălțăminte, afișându-se doar lista magazinelor unde este disponibil modelul căutat, iar pentru fiecare magazin culorile în care este disponibil modelul de mărimea căutată.
- ❖ Vânzarea unei perechi de încălțăminte;
- ❖ Adăugarea și actualizarea informațiilor despre modelele din magazinul la care lucrează acel angajat (***doar*** stocul, mărimea și culoarea pentru fiecărui model);
- ❖ Salvare liste cu situația produselor de încălțăminte din magazinul unde lucrează în mai multe formate: csv, json, xml, doc.

Utilizatorii de tip **manager** pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor produselor de încălțăminte sortată după modelul produselor (vizualizarea include și redarea unor imagini cu produsele de încălțăminte);
- ❖ Filtrarea produselor de încălțăminte din toate magazinele lanțului de magazine de încălțăminte după următoarele criterii: producător, disponibilitate, preț;
- ❖ Căutarea unui produs de încălțăminte după model, afișându-se doar lista magazinelor unde este disponibil modelul, iar pentru fiecare model mărimile și culorile în care este disponibil;
- ❖ Operații CRUD în ceea ce privește persistența produselor de încălțăminte (adăugare sau actualizare informații referitoare la model, producător, imagini, preț și culori aferente);
- ❖ Salvare liste cu situația produselor de încălțăminte din toate magazinele lanțului de magazine de încălțăminte în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de produsele de încălțăminte utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatori;
- ❖ Vizualizarea listei tuturor utilizatorilor și filtrarea acesteia după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 10*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-o **grădină zoologică**. Aplicația client va avea 3 tipuri de utilizatori: vizitator al grădinii zoologice, angajat al grădinii zoologice și administrator.

Utilizatorii de tip **vizitator** pot efectua următoarele operații fără autentificare:

- ❖ Vizualizarea listei tuturor animalelor din grădina zoologică sortată după specie și tipul de alimentație (vizualizarea include și redarea unor imagini cu toate animalele din grădina zoologică; între 1 și 3 imagini pentru fiecare animal);
- ❖ Filtrarea listei animalelor după următoarele criterii: categorie (pasăre, reptilă, insectă, pește, etc.), specie, tipul de alimentație, habitat;
- ❖ Căutarea unui animal după denumire.

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

- ❖ Toate operațiile permise utilizatorilor de tip vizitator;
- ❖ Operații CRUD în ceea ce privește persistența animalelor (categorie, specie, tipul de alimentație, habitat, greutate medie, vârsta medie), dar și a exemplarelor din fiecare specie (id, imagini, locație în grădina zoologică, vârstă, gerutate) din grădina zoologică;
- ❖ Salvare liste cu informațiile despre animale în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de animalele din grădina zoologică 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 și filtrarea acesteia după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 11*

Dezvoltați 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 listei tuturor plantelor din grădina botanică sortată după tip și specie (vizualizarea include și redarea unor imagini cu toate plantele din grădina botanică; între 1 și 3 imagini pentru fiecare plantă);
- ❖ 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 și filtrarea acesteia după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 12*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-un lanț de **magazine cu produse cosmetice**. Aplicația va avea 3 tipuri de utilizatori: angajat al unui magazin, manager al lanțului de magazine cu produse cosmetice și administrator.

Utilizatorii de tip **angajat** pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor produselor cosmetice din magazinul unde lucrează sortată după denumire și preț (vizualizarea include și redarea unor imagini cu produsele cosmetice; între 1 și 3 imagini pentru fiecare produs cosmetic);
- ❖ Filtrarea produselor cosmetice din magazinul unde lucrează după următoarele criterii: producător, disponibilitate, preț;
- ❖ Căutarea unui produs cosmetic după denumire astfel:
  - ➢ În magazinul unde lucrează afișându-se toate informațiile despre produs în cazul în care produsul cosmetic căutat este disponibil în acel magazin;
  - ➢ Dacă nu este disponibil, să se caute acel produs cosmetic și în celelalte magazine aparținând acelui lanț de magazine cu produse cosmetice, afișându-se doar lista magazinelor unde este disponibil produsul căutat.
- ❖ Vânzarea unui produs cosmetic;
- ❖ Adăugarea și actualizarea informațiilor despre produsele cosmetice din magazinul la care lucrează acel angajat (***doar*** stocul)
- ❖ Salvare liste cu situația produselor cosmetice din magazinul unde lucrează în mai multe formate: csv, json, xml, doc.

Utilizatorii de tip **manager** pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor produselor cosmetice sortată după denumire și preț (vizualizarea include și redarea unor imagini cu produsele cosmetice);
- ❖ Filtrarea produselor cosmetice după următoarele criterii: magazin, producător, disponibilitate;
- ❖ Căutarea unui produs cosmetic după denumire, afișându-se doar lista magazinelor unde este disponibil;
- ❖ Operații CRUD în ceea ce privește persistența produselor cosmetice (adăugare sau actualizare informații referitoare la denumire, producător, imagini și preț);
- ❖ Salvare liste cu situația produselor cosmetice în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de produsele cosmetice utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatori;
- ❖ Vizualizarea listei tuturor utilizatorilor și filtrarea acesteia după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 13*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată într-un **service auto**. Aplicația client va avea 3 tipuri de utilizatori: angajat, manager și administrator.

Utilizatorii de tip **angajat** pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor autototurismelor din service sortată după marcă și combustibil (vizualizarea include și redarea unor imagini cu autoturismele; între 1 și 3 imagini pentru fiecare autoturism);
- ❖ Filtrarea autototurismelor după anumite criterii: proprietar, marcă, culoare, combustibil, an fabricație;
- ❖ Căutarea unui autototurism după numărul de înmatriculare sau seria motorului;
- ❖ Operații CRUD în ceea ce privește persistența autototurismelor și a proprietarilor acestora; inclusiv specificarea defecțiunilor fiecărui autototurism și a costului reparării;
- ❖ Notificarea proprietarului (prin email, SMS, etc.) când autoturismul este reparat;
- ❖ Salvare liste cu informații despre autototurisme în mai multe formate: csv, json, xml, doc.

Utilizatorii de tip **manager** pot efectua următoarele operații după autentificare:

- ❖ Vizualizarea listei tuturor autototurismelor existente în service sortată după marcă și combustibil (vizualizarea include și redarea unor imagini cu autototurisme);
- ❖ Filtrarea autototurismelor după anumite criterii: proprietar, marcă, culoare, combustibil;
- ❖ Căutarea unui autototurism după numărul de înmatriculare sau seria motorului;
- ❖ Salvare liste cu informații despre autototurisme în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de autototurisme utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii aplicației;
- ❖ Vizualizarea listei tuturor utilizatorilor și filtrarea acesteia după tipul utilizatorilor;
- ❖ Notificarea fiecărui utilizator prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 14*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată de către o **agenție turistică**. Aplicația client 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 sortată după destinație și perioadă (vizualizarea include și redarea unor imagini cu pachetele turistice; între 1 și 3 imagini pentru fiecare pachet turistic);
- Filtrarea listei pachetelor după destinație, perioadă, preț.

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;
- Vizualizarea listei pachetelor rezervate într-o perioadă specificată;
- Salvare liste cu informații despre pachetele oferite de agenție în mai multe formate: csv, json, xml, doc;
- Vizualizarea unor statistici legate de pachetele turistice 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 client;
- Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare;
- Vizualizarea listei utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip;
- Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 15*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată de către o **agenție de rezervare bilete de tren**. Aplicația va avea 3 tipuri de utilizatori care necesită autentificare: angajat, manager și administrator.

Fără autentificare aplicația va permite:

- Vizualizarea listei tuturor trenurilor care trec printr-o gară (sosiri și plecări) într-o zi selectată;
- Vizualizarea listei trenurilor dintre 2 stații, inclusiv preț și disponibilitate locuri libere pentru o zi selectată ;
- Vizualizarea traseului optim după specificarea gării de plecare și a gării destinație (reprezentare grafică);
- Căutarea unui tren după număr.

Utilizatorii de tip **angajat** pot efectua următoarele operații după autentificare:

- Toate operațiile permise fără autentificare;
- Vânzarea unui bilet către un călător;
- Operații CRUD în ceea ce privește biletele vândute;
- Salvarea listelor cu informații despre biletele vândute în formatele csv, json, xml, doc.

Utilizatorii de tip **manager** pot efectua următoarele operații după autentificare:

- Toate operațiile permise fără autentificare;
- Operații CRUD în ceea ce privește persistența gărilor și trenurilor (inclusiv zilele când circulă și numărul de locuri disponibile);
- Salvarea listelor cu informații despre trenuri în formatele csv, json, xml, doc;
- Vizualizarea unor statistici legate de trenuri și biletele vândute 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 călător;
- Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare;
- Vizualizarea listei utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip;
- Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 16*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată de către un **lanț hotelier**. Aplicația client 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 sortată după locație și număr (vizualizarea include și redarea unor imagini cu camerele din hotel; între 1 și 3 imagini pentru fiecare cameră);
- Filtrarea listei camerelor după locație, disponibilitate, preț, poziție, facilități.

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 liste cu informații despre camerele rezervate în mai multe formate: csv, json, xml, doc;
- Vizualizarea unor statistici legate de camere 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 client;
- Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare;
- Vizualizarea listei utilizatorilor care necesită autentificare și filtrarea acesteia după tipul utilizatorilor;
- Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 17*

Dezvoltați 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 locuințelor disponibile pentru închiriat sortată după locație și preț (vizualizarea include și redarea unor imagini cu locuințele de închiriat; între 1 și 3 imagini pentru fiecare locuință);
- ❖ Filtrarea listei locuințelor disponibile pentru închiriat după locație, preț, tip locuință, număr camere.

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 disponibile pentru închiriat, locuințelor închiriate și a clienților;
- ❖ Închirierea unei locuințe de către un client pentru o perioadă (specificată/nespecificată);
- ❖ Vizualizarea listei locuințelor închiriate într-o perioadă selectată sortată după locație (vizualizarea include și redarea unor imagini cu locuințele de închiriat; între 1 și 3 imagini pentru fiecare locuință);
- ❖ Salvare liste cu informații despre locuințele de închiriat în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de locuințe 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 client;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare;
- ❖ Vizualizarea listei utilizatorilor care necesită autentificare;
- ❖ Filtrarea listei utilizatorilor care necesită autentificare după tip;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 18*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată de către o **agenție de rezervare bilete de avion**. Aplicația va avea 3 tipuri de utilizatori care necesită autentificare: angajat, manager și administrator.

Fără autentificare aplicația va permite:

- ❖ Vizualizarea listei tuturor zborurilor care aterizează/decolează dintr-un aeroport (sosiri și plecări) într-o zi selectată;
- ❖ Vizualizarea listei zborurilor (nu neapărat directe) dintre 2 aeroporturi, inclusiv preț și disponibilitate locuri libere pentru o zi selectată;
- ❖ Vizualizarea traseului optim după specificarea aeroportului de plecare și a gării destinație (reprezentare grafică);
- ❖ Căutarea unui zbor după număr.

Utilizatorii de tip **angajat** pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise fără autentificare;
- ❖ Operații CRUD în ceea ce privește persistența biletelor vândute (inclusiv a clienților);
- ❖ Vizualizarea grafică a locurilor (ocupate/libere) dintr-un avion, alegerea unui loc și vânzarea unui bilet către un călător pentru o dată specificată;
- ❖ Salvare liste cu informații despre zboruri în mai multe formate: csv, json, xml, doc.

Utilizatorii de tip **manager** pot efectua următoarele operații după autentificare:

- ❖ Toate operațiile permise fără autentificare;
- ❖ Operații CRUD în ceea ce privește persistența aeroporturilor și zborurilor (inclusiv numărul de locuri disponibile);
- ❖ Salvarea listelor cu informații despre zboruri în formatele csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de zboruri și biletele vândute 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 călător;
- ❖ Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare;
- ❖ Vizualizarea listei utilizatorilor care necesită autentificare;
- ❖ Filtrarea listei utilizatorilor care necesită autentificare după tip;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 19*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată de către o **casă de producție filme**. Aplicația va avea 3 tipuri de utilizatori: angajat, manager și administrator.

Utilizatorii de tip **angajat** pot efectua următoarele operații după autentificare:

- Vizualizarea listei filmelor sortată după tip și anul realizării; pentru fiecare film, vizualizarea include:
  - redarea unor imagini din film; între 1 și 3 imagini pentru fiecare film;
  - date despre regizorul, scenaristul și producătorul fimului;
  - lista actorilor care fac parte din distribuția acelui film;
- Filtrarea filmelor după anumite criterii: tip film (artistic sau serial), categorie film, anul realizării;
- Operații CRUD în ceea ce privește persistența filmelor;
- Căutarea unui film după titlu;
- Salvare liste cu informații despre filme în mai multe formate: csv, json, xml, doc.

Utilizatorii de tip **manager** pot efectua următoarele operații după autentificare:

- Vizualizarea listei filmelor sortată după tip și anul ; pentru fiecare film, vizualizarea include:
  - redarea unor imagini din film; între 1 și 3 imagini pentru fiecare film;
  - date despre regizorul, scenaristul și producătorul fimului;
  - lista actorilor care fac parte din distribuția acelui film;
- Filtrarea filmelor după anumite criterii: tip film (artistic sau serial), categorie film, anul realizării;
- Căutarea unui film după titlu;
- Vizualizarea listei regizorilor (pentru fiecare regizor se afișează o fotografie și lista filmelor regizate);
- Vizualizarea listei scenaristilor (pentru fiecare scenarist se afișează o fotografie și lista filmelor pentru care a scris scenariul);
- Vizualizarea listei actorilor (pentru fiecare actor se afișează o fotografie și lista filmelor în care a jucat împreună cu rolul interpretat în filmul respectiv);
- Salvare liste cu informații despre filme în mai multe formate: csv, json, xml, doc;
- Vizualizarea unor statistici legate de filme utilizând grafice (structură radială, structură inelară, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii aplicației;
- Vizualizarea listei utilizatorilor și filtrarea acesteia după tipul utilizatorilor;
- Notificarea fiecărui utilizator prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 20*

Dezvoltați 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 listei tuturor liniilor de transport în comun sortată după număr;
- Vizualizarea traseului optim (din punct de vedere al distanței) după specificarea stației de plecare și a stației de sosire (reprezentare grafică);
- Vizualizare listei tuturor liniilor de transport care trec printr-o stație selectată;
- 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 stațiilor și a liniilor de transport în comun;
- Salvare liste cu liniile de transport în comun în mai multe formate: csv, json, xml, doc;
- Vizualizarea unor statistici legate de liniile de transport în comun utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare;
- Vizualizarea listei utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip;
- Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 21*

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

Utilizatorii de tip **poștaș** pot efectua următoarele operații după autentificare:

- Vizualizarea listei coletelor de distribuit, inclusiv locațiile unde trebuie distribuite sortată după locație pentru o zi specificată;
- Vizualizarea traseului optim în funcție de lista coletelor (reprezentare grafică);
- Căutarea unui colet alocat poștașului după număr și setarea stării acestuia în livrat după livrarea coletului (inclusiv ora la care s-a realizat livrarea).

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

- Vizualizarea listei tuturor coletelor sortată după adresa de livrare;
- Filtrarea coletelor după starea acestora:
  - Nealocat unui poștaș;
  - Alocat unui poștaș, dar nelivrat;
  - Livrat;
- Operații CRUD în ceea ce privește persistența coletelor;
- Alocarea unui colet către un poștaș;
- Salvare liste cu informații despre colete în mai multe formate: csv, json, xml, doc;
- Vizualizarea unor statistici legate de colete utilizând grafice (structură radială, structură inelară, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatori;
- Vizualizarea listei tuturor utilizatorilor;
- Filtrarea listei utilizatorilor după tipul utilizatorilor;
- Vizualizarea unei statistici cu utilizatorii în funcție de tipul acestora;
- Notificarea fiecărui utilizator prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 22*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată de către o **firmă de colectare deșeuri**. 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 adresele deșeurilor ce urmează a fi colectate de angajat sortată după adresă;
- ❖ Vizualizarea traseului optim în funcție de adresa deșeurilor (reprezentare grafică);
- ❖ Căutarea adresei unui deșeu alocat angajatului după număr și setarea stării acestuia în colectat după colectarea deșeului.

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

- ❖ Vizualizarea listei tuturor deșeurilor sortată după adresa de colectare;
- ❖ Filtrarea deșeurilor după starea acestora:
  - ➢ Nealocat unui angajat;
  - ➢ Alocat unui angajat, dar necolectat;
  - ➢ Colectat;
- ❖ Operații CRUD în ceea ce privește persistența adreselor cu deșeuri;
- ❖ Alocarea unei adrese cu deșeuri către un un angajat în vederea colectării;
- ❖ Salvare liste cu informații despre listele cu adresele deșeurilor alocate angajaților în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici legate de deșeuri (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatori;
- ❖ Vizualizarea listei tuturor utilizatorilor;
- ❖ Filtrarea listei utilizatorilor după tipul utilizatorilor;
- ❖ Vizualizarea unei statistici cu utilizatorii în funcție de tipul acestora;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 23*

Dezvoltați 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 sortată după scop și locație (vizualizarea include și redarea unor imagini cu evenimente anterioare; între 1 și 3 imagini pentru fiecare eveniment).

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 rezervate după locație, număr persoane, scop, dată;
- Salvare liste cu informații despre evenimente rezervate în mai multe formate: csv, json, xml, doc;
- Vizualizarea unor statistici legate de evenimente 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 client;
- Operații CRUD pentru informațiile legate de utilizatorii aplicației care necesită autentificare;
- Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip;
- Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 24*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată pentru **organizarea de conferințe științifice**. Aplicația va avea 3 tipuri de utilizatori: participant conferință, organizator conferință și administrator.

Utilizatorii de tip **participant** pot efectua următoarele operații fără autentificare:

- Înscrierea la conferință (introducere date personale, inclusiv CV-ul și o fotografie);
- Vizualizarea programului conferinței pe secțiuni (la fiecare secțiune, pentru fiecare participant să fie vizibilă și o fotografie a acestuia, dar și accesarea CV-ului).

Utilizatorii de tip **participant** pot efectua următoarele operații după autentificare:

- Accesarea volumului conferinței.

Utilizatorii de tip **organizator conferință** pot efectua următoarele operații după autentificare:

- Acceptarea sau respingerea unui participant (notificare prin email);
- Operații CRUD în ceea ce privește persistența participanților și a lucrărilor prezentate;
- Generearea programului conferinței pe secțiuni;
- Filtrarea listei de participanți după secțiunea la care participă;
- Salvare liste filtrate cu informații despre lucrările prezentate în mai multe formate: csv, json, xml, doc;
- Vizualizarea unor statistici utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii aplicației care necesită autentificare;
- Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip;
- Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 25*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată pentru organizarea unui **turneu de tenis**. Aplicația va avea 4 tipuri de utilizatori: tenismen, arbitru, organizator turneu și administrator.

Utilizatorii de tip **tenismen** pot efectua următoarele operații fără autentificare:

- Înscrierea la turneu (introducere date personale, inclusiv CV-ul și o fotografie);
- Vizualizarea programului turneului pe probe; pentru fiecare partidă să fie disponibile mai multe informații despre cei 2 jucători și arbitru (inclusiv o fotografie pentru fiecare);
- Vizualizarea scorului fiecărei partide de tenis.

Utilizatorii de tip **arbitru** pot efectua următoarele operații după autentificare:

- Vizualizarea programului propriu;
- Gestionarea scorului partidei pe care o arbitrează.

Utilizatorii de tip **organizator turneu** pot efectua următoarele operații după autentificare:

- Acceptarea sau respingerea unui tenismen (notificare prin email);
- Operații CRUD în ceea ce privește persistența tenismenilor, arbitrilor și a partidelor de tenis;
- Generarea programului turneului;
- Filtrarea listei de tenismeni după diferite criterii;
- Salvare liste filtrate cu informații despre partide în mai multe formate: csv, json, xml, doc.;
- Vizualizarea unor statistici utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii aplicației care necesită autentificare;
- Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip;
- Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 26*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată ca soft educațional pentru **studiul cercului**. Aplicația va avea 2 tipuri de utilizatori: elev și administrator.

Utilizatorii de tip **elev** pot efectua următoarele operații fără autentificare:

- desenarea interactivă a cercurilor prin înlocuirea creionului și a riglei cu mouse-ul și alegerea stilului de desenare (culoare, stil linie, grosime linie);
- calcularea și afișarea unor proprietăți: aria unui cerc, lungimea unui cerc, aria unui sector de cerc, lungimea unui arc de cerc;
- vizualizarea unor cercuri particulare:
  - cercul circumscris unui poligon (dacă poligonul este inscriptibil),
  - cercul înscris (dacă poligonul este circumscriptibil);
  - cercuri specifice unui triunghi: cercul lui Tucker, cercurile lui Lucas, cercul ortocentroidal, cercurile lui Neuberg, cercul lui Adams, cercul lui Brocard.
- salvarea/încărcarea unui cerc într-un/dintr-un fișier xml, json și csv;
- solicitarea unui cont pentru testarea cunoștințelor.

Utilizatorii de tip **elev** pot efectua următoarele operații după autentificare:

- verificarea cunoștințelor prin efectuarea unui test de 10 întrebări (alese aleator dintr-un set de 50 de întrebări) și vizualizarea punctajului obținut după finalizarea testului. Dintre cele 10 întrebări, 3 trebuie să aibă un grad mic de dificultate, 4 întrebări trebuie să aibă un grad mediu de dificultat, iar 3 întrebări trebuie să aibă un grad mare de dificultate. De asemenea, cel puțin 5 întrebări din test trebuie să fie însoțite de o imagine ce reprezintă figura geometrică atașată întrebării respective.
- vizualizarea unei statistici după punctaj utilizând grafice (structură radială, structură inelară, etc.) care să raporteze punctajul elevului la punctajele celorlalți elevi.

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii aplicației care necesită autentificare;
- Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 27*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată ca soft educațional pentru **studiul suprafețelor de rotație**. Aplicația va avea 2 tipuri de utilizatori: elev și administrator.

Utilizatorii de tip **elev** pot efectua următoarele operații fără autentificare:

- desenarea interactivă a suprafețelor de rotație (sferă, pseudosferă, tor, elicoid, etc.) prin înlocuirea creionului și a riglei cu *mouse*-ul și alegerea stilului de desenare (culoare, stil linie, grosime linie);
- determinarea și desenarea intersecției suprafeței de rotație cu un plan;
- determinarea și desenarea suprafeței de rotație obținute prin simetrie față de un plan;
- determinarea și desenarea suprafeței de rotație obținute prin rotație în jurul axelor Ox, Oy și Oz;
- salvarea/încărcarea unei suprafețe de rotație într-un/dintr-un fișier xml, json și csv;
- solicitarea unui cont pentru testarea cunoștințelor.

Utilizatorii de tip **elev** pot efectua următoarele operații după autentificare:

- verificarea cunoștințelor prin efectuarea unui test de 10 întrebări (alese aleator dintr-un set de 50 de întrebări) și vizualizarea punctajului obținut după finalizarea testului. Dintre cele 10 întrebări, 3 trebuie să aibă un grad mic de dificultate, 4 întrebări trebuie să aibă un grad mediu de dificultat, iar 3 întrebări trebuie să aibă un grad mare de dificultate. De asemenea, cel puțin 5 întrebări din test trebuie să fie însoțite de o imagine ce reprezintă figura geometrică atașată întrebării respective.
- vizualizarea unei statistici după punctaj utilizând grafice (structură radială, structură inelară, etc.) care să raporteze punctajul elevului la punctajele celorlalți elevi.

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii aplicației care necesită autentificare;
- Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 28*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată ca soft educațional pentru **studiul geometriei patrulaterului**. Aplicația va avea 2 tipuri de utilizatori: elev și administrator.

Utilizatorii de tip **elev** pot 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 înscriptibil, 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 inscriptibil);
- 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 inscriptibil;
  - 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, cercul înscris;
- salvarea / încărcare unui patrulater într-un/dintr-un fișier xml, json și csv;
- solicitarea unui cont pentru testarea cunoștințelor.

Utilizatorii de tip **elev** pot efectua următoarele operații după autentificare:

- verificarea cunoștințelor prin efectuarea unui test de 10 întrebări (alese aleator dintr-un set de 50 de întrebări) și vizualizarea punctajului obținut după finalizarea testului. Dintre cele 10 întrebări, 3 trebuie să aibă un grad mic de dificultate, 4 întrebări trebuie să aibă un grad mediu de dificultat, iar 3 întrebări trebuie să aibă un grad mare de dificultate. De asemenea, cel puțin 5 întrebări din test trebuie să fie însoțite de o imagine ce reprezintă figura geometrică atașată întrebării respective.
- vizualizarea unei statistici după punctaj utilizând grafice (structură radială, structură inelară, etc.) care să raporteze punctajul elevului la punctajele celorlalți elevi.

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii aplicației care necesită autentificare;
- Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 29*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată ca soft educațional pentru **studiul conicelor**. Aplicația va avea 2 tipuri de utilizatori: elev și administrator.

Utilizatorii de tip **elev** pot 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, json și csv;
- solicitarea unui cont pentru testarea cunoștințelor.

Utilizatorii de tip **elev** pot efectua următoarele operații după autentificare:

- verificarea cunoștințelor prin efectuarea unui test de 10 întrebări (alese aleator dintr-un set de 50 de întrebări) și vizualizarea punctajului obținut după finalizarea testului. Dintre cele 10 întrebări, 3 trebuie să aibă un grad mic de dificultate, 4 întrebări trebuie să aibă un grad mediu de dificultat, iar 3 întrebări trebuie să aibă un grad mare de dificultate. De asemenea, cel puțin 5 întrebări din test trebuie să fie însoțite de o imagine ce reprezintă figura geometrică atașată întrebării respective.
- vizualizarea unei statistici după punctaj utilizând grafice (structură radială, structură inelară, etc.) care să raporteze punctajul elevului la punctajele celorlalți elevi.

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii aplicației care necesită autentificare;
- Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip;
- Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 30*

Dezvoltați o ***aplicație client/server*** care poate fi utilizată ca soft educațional pentru **studiul poliedrelor**. Aplicația va avea 2 tipuri de utilizatori: elev și administrator.

Utilizatorii de tip **elev** pot 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, json și csv;
- solicitarea unui cont pentru testarea cunoștințelor.

Utilizatorii de tip **elev** pot efectua următoarele operații după autentificare:

- verificarea cunoștințelor prin efectuarea unui test de 10 întrebări (alese aleator dintr-un set de 50 de întrebări) și vizualizarea punctajului obținut după finalizarea testului. Dintre cele 10 întrebări, 3 trebuie să aibă un grad mic de dificultate, 4 întrebări trebuie să aibă un grad mediu de dificultat, iar 3 întrebări trebuie să aibă un grad mare de dificultate. De asemenea, cel puțin 5 întrebări din test trebuie să fie însoțite de o imagine ce reprezintă figura geometrică atașată întrebării respective.
- vizualizarea unei statistici după punctaj utilizând grafice (structură radială, structură inelară, etc.) care să raporteze punctajul elevului la punctajele celorlalți elevi.

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii aplicației care necesită autentificare;
- Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- Filtrarea listei utilizatorilor care necesită autentificare după tip;
- Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 31*

Dezvoltați o ***aplicație client/server*** pentru următorul joc: se dă un labirint sub formă de poligon regulat având în centru o minge de fotbal. Se dorește determinarea drumului minim pentru recuperarea mingii.

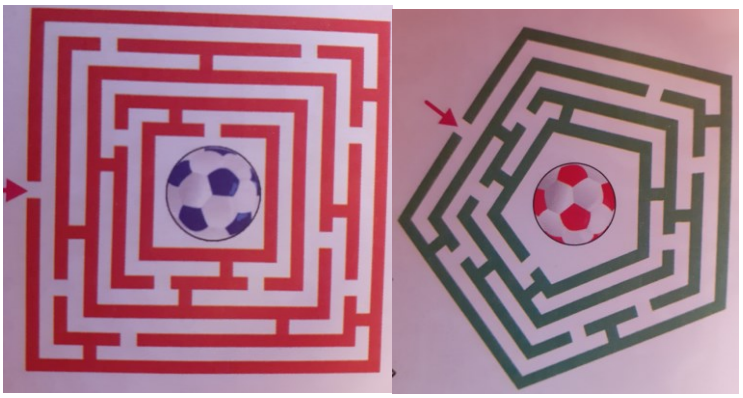
Jocul va fi dezvoltat pe mai multe niveluri (triunghi echilateral, pătrat, pentagon regulat, hexagon regulat, octogon regulat). După ce un **jucător** a obținut soluția, se va afișa numărul de deplasări utilizat de către acesta în soluția furnizată și i se va comunica dacă a obținut soluția optimă. Dacă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

Jucătorii vor putea să-și creze cont, să se autentifice și să-și vizualizeze punctajul. Jucătorii vor avea posibilitatea, de asemenea, să vizualizeze un grafic cu puntajul propriu raportat la punctajelor celorlați jucători (cum este punctajul său față de punctajul minim, punctajul maxim și punctajul mediu pentru nivelul respectiv).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru utilizatori;
- ❖ Vizualizarea listei tuturor jucătorilor care și-au creat cont;
- ❖ Salvare liste cu informații despre utilizatori în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici referitoare la punctajele jucătorilor pe nivele de dificultate utilizând grafice (structură radială, structură inelară, etc.);
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.



### *Problema 32*

Dezvoltați o ***aplicație client/server*** pentru un joc de tip sudoku bazat pe imagini. Jocul va fi dezvoltat pe mai multe niveluri (între 5 și 8). Pentru a realiza o interfață mai atractivă pentru copiii ce vor rezolva jocul, acesta va fi prezentat în trei variante (conform celor 3 imagini).

**Jucătorii** vor putea să-și creze cont, să se autentifice și să-și vizualizeze punctajul. Jucătorii vor avea posibilitatea, de asemenea, să vizualizeze un grafic cu puntajul propriu raportat la punctajelor celorlați jucători (cum este punctajul său față de punctajul minim, punctajul maxim și punctajul mediu pentru nivelul respectiv).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru utilizatori;
- ❖ Vizualizarea listei tuturor jucătorilor care și-au creat cont;
- ❖ Salvare liste cu informații despre utilizatori în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici referitoare la punctajele jucătorilor pe nivele de dificultate utilizând grafice (structură radială, structură inelară, etc.);
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

### *Problema 33*

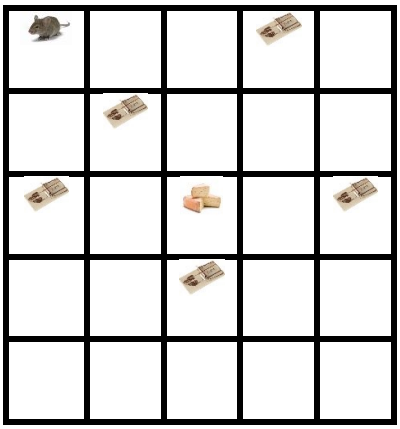
Dezvoltați o ***aplicație client/server*** pentru următorul joc: se consideră un labirint de dimensiune **nxn** căsuţe în care pot să existe curse pentru şoricei şi o "comoară" din brânză. Într-un colţ al labirintului se află un şoricel flămând care simte mirosul brânzei, dar nu ştie unde este poziţionată aceasta. Ajutaţi-l pe şoricel să ajungă la "comoara" de brânză astfel încât să nu fie prins într-una din curse, ştiind 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,…). 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*).

**Jucătorii** vor putea să-și creze cont, să se autentifice și să-și vizualizeze punctajul. Jucătorii vor avea posibilitatea, de asemenea, să vizualizeze un grafic cu puntajul propriu raportat la punctajelor celorlați jucători (cum este punctajul său față de punctajul minim, punctajul maxim și punctajul mediu pentru nivelul respectiv).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru utilizatori;
- ❖ Vizualizarea listei tuturor jucătorilor care și-au creat cont;
- ❖ Salvare liste cu informații despre utilizatori în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici referitoare la punctajele jucătorilor pe nivele de dificultate utilizând grafice (structură radială, structură inelară, etc.);
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.



### *Problema 34*

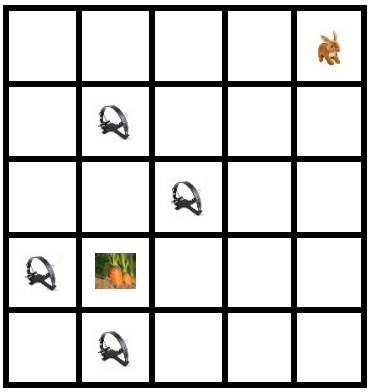
Dezvoltați o ***aplicație client/server*** pentru următorul joc: se un labirint de dimensiune **n**x**n** căsuţe în care pot să existe capcane pentru animale sălbatice şi o "comoară" de morcovi. Într-un colţ al labirintului se află un iepuraș flămând. Ajutaţi-l pe iepuraș să ajungă la "comoara" de morcovi astfel încât să nu fie prins într-una din capcane, ştiind 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,…). 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*).

**Jucătorii** vor putea să-și creze cont, să se autentifice și să-și vizualizeze punctajul. Jucătorii vor avea posibilitatea, de asemenea, să vizualizeze un grafic cu puntajul propriu raportat la punctajelor celorlați jucători (cum este punctajul său față de punctajul minim, punctajul maxim și punctajul mediu pentru nivelul respectiv).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru utilizatori;
- ❖ Vizualizarea listei tuturor jucătorilor care și-au creat cont;
- ❖ Salvare liste cu informații despre utilizatori în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici referitoare la punctajele jucătorilor pe nivele de dificultate utilizând grafice (structură radială, structură inelară, etc.);
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.



### *Problema 35*

Dezvoltați 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.

**Jucătorii** vor putea să-și creze cont, să se autentifice și să-și vizualizeze punctajul. Jucătorii vor avea posibilitatea, de asemenea, să vizualizeze un grafic cu puntajul propriu raportat la punctajelor celorlați jucători (cum este punctajul său față de punctajul minim, punctajul maxim și punctajul mediu pentru nivelul respectiv).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru utilizatori;
- ❖ Vizualizarea listei tuturor jucătorilor care și-au creat cont;
- ❖ Salvare liste cu informații despre utilizatori în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici referitoare la punctajele jucătorilor pe nivele de dificultate utilizând grafice (structură radială, structură inelară, etc.);
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

**Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională**.

Nivel 1

Nivel 2