TEMA 2

Termen predare: 6 aprilie 2022

Obiectiv:

Obiectivul acestei teme este familiarizarea cu șablonul architectural ***Model-View-ViewModel***, dar și cu șabloanele de proiectare creaționale și cu șablonul de proiectare comportamental ***Command.***

Restricție: Pentru persistență se va utiliza o bază de date relațională (SQL Server, MySQL, Oracle, etc.). Se va utiliza baza de date doar în aplicațiile unde este necesară persistența.

Cerințe:

- În ***faza de analiză*** se va realiza diagrama cazurilor de utilizare.
- În ***faza de proiectare*** se va realiza diagrama de relaționare a entităților și diagrama de clase folosind arhitectura **MVVM** (care utilizează șablonul de proiectare comportamental ***Command***) și un șablon de proiectare creațional (la alegere).
- În ***faza de implementare*** se va scrie cod pentru îndeplinirea tuturor funcționalităților precizate de diagrama cazurilor de utilizare utilizând:
  - proiectarea dată de diagrama de clase;
  - 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;
  - Directorul cu aplicația implementată;
  - Baza de date utilizată;
  - Documentația - un fișier word care cuprinde:
    - numele studentului, grupa;
    - enunțul problemei;
    - instrumente utilizate;
    - justificarea limbajului de programare ales;
    - descrierea diagramelor UML;
    - descrierea aplicației.

Nr_crt	Lista studentilor	Număr problemă
1	**Berea Roxana**	13
2	**Birlutiu Claudiu - Andrei**	23
3	**Blidea Tudorel - Alexandru**	9
4	**Coroian Razvan**	14
5	**Deto Sergiu - Alexandru**	24
6	**Dudas Octavian - Mateo**	28
7	**Duma Bianca Nicoleta**	18
8	**Fazacas Denisa Georgiana**	11

9	**Gavra Anamaria**	15
10	**Hagymas Alexandra - Maria**	21
11	**Hitu Octavian**	4
12	**Lese Doru - Calin**	1
13	**Lupulescu Teodor - Damian**	16
14	**Marian Gheorghe Theodor**	29
15	**Munteanu Cezar Lucian**	8
16	**Pastiu Georgiana - Maria**	22
17	**Pop Ruxandra Maria**	17
18	**Racz Milan**	19
19	**Rus Andrei - Nicolae**	20
20	**Stahie Dragos Marian**	25
21	**Suciu Andrei Ioan**	2
22	**Timis Iulia - Georgeana**	5
23	**Torzsa Gabriela**	10
24	**Viman Andrei Liviu**	3
25	**Zabulic Diana**	7
26	**Zaicenco Felix - Cristian**	6
27	**Zelenszky Bianca**	26
28	**Steko Daiana (gr. 10)**	6
29	**Nyamcz Simon Istvan**	12
30	**Fechete Rareș Cristian**	30

Problema 1

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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 urm. criterii: instituția de artă, artist, tipul operei de artă;
- Căutarea unei opere de artă după titlu.

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, xml;

❖ 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 administrator pot efectua următoarele operații după autentificare:

❖ Toate operațiile permise utilizatorilor de tip vizitator;
❖ Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare.

Problema 2

Descrierea aplicației:

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

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

❖ Vizualizarea tuturor produselor dintr-o farmacie selectată;
❖ Filtrarea produselor după următoarele criterii: disponibilitate, valabilitate, preț, producător;
❖ Căutarea unui produs după denumire;
❖ Operații CRUD în ceea ce privește persistența produselor din farmacia la care lucrează acel angajat;
❖ Salvare rapoarte/liste cu produse în mai multe formate: csv, json, xml;
❖ Vizualizarea unor statistici legate de produsele din farmacia la care este angajat: procente după disponibilitate, valabilitate, producător 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.

Problema 3

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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;
❖ Operații CRUD în ceea ce privește persistența parfumurilor din magazinul la care lucrează acel angajat;
❖ Salvare rapoarte/liste cu parfumuri în mai multe formate: csv, json, xml;
❖ Vizualizarea unor statistici legate de produsele din magazinul la care este angajat: procente după disponibilitate, preț, producător 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.

Problema 4

Descrierea aplicației:

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

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

- Vizualizarea tuturor produselor de încălțăminte disponibile într-un magazin selectat;
- Filtrarea produselor de încălțăminte după următoarele criterii: producător, disponibilitate, preț;
- Căutarea unui produs de încălțăminte după denumire și/sau producător;
- Operații CRUD în ceea ce privește persistența produselor de încălțăminte din magazinul la care lucrează acel angajat;
- Salvare rapoarte/liste cu produse de încălțăminte în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de produsele din magazinul la care este angajat: procente după disponibilitate, preț, producător 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.

Problema 5

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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;
- Operații CRUD în ceea ce privește persistența florilor din florăria la care lucrează acel angajat;
- Salvare rapoarte/liste cu flori în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de florile din florăria la care este angajat: procente după disponibilitate, preț, culoare 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.

Problema 6

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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;
- Operații CRUD în ceea ce privește persistența produselor cosmetice din magazinul la care lucrează acel angajat;
- Salvare rapoarte/liste cu produse cosmetice în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de produsele din magazinul la care este angajat: procente după disponibilitate, preț, producător 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.

Problema 7

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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 fără autentificare:

- Vizualizarea tuturor cărților dintr-o bibliotecă selectată;
- Filtrarea listei cărților după următoarele criterii: domeniu, disponibilitate, editura, autor;
- Căutarea unei cărți după titlu.

Utilizatorii de tip bibliotecar pot efectua următoarele operații după autentificare:

- Toate operațiile permise utilizatorilor de tip abonat;
- Împrumutarea unei cărți unui abonat;
- Returnarea unei cărți de către un abonat;
- Operații CRUD în ceea ce privește persistența cărților;
- Salvare rapoarte/liste cu cărțile în mai multe formate: csv, json, xml;
- Operații CRUD pentru informațiile legate de utilizatorii de tip abonat;
- Vizualizarea unor statistici legate de cărțile din biblioteca selectată: procente după disponibilitate, domeniu, editura 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.

Problema 8

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** care poate fi utilizată într-un lanț de librării. Aplicația va avea 2 tipuri de utilizatori: angajat al unei 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;
- Vânzarea unei cărți (decrementarea cu 1 a numărului de exemplare disponibile din aceea carte) din librăria la care lucrează acel angajat;
- Operații CRUD în ceea ce privește persistența cărților din librăria la care lucrează acel angajat;
- Salvare rapoarte/liste cu cărțile în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de cărțile din librăria la care este angajat: procente după disponibilitate, preț, domeniu 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.

Problema 9

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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;
- 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, varstă;
- Căutarea unui pacient după nume;
- Planificarea pacienților pentru consultații;
- Salvare rapoarte/liste cu pacienți sau medici în mai multe formate: csv, json, xml.
- Vizualizarea unor statistici legate de pacienți: procente după vârstă, diagnostic 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.

Problema 10

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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 animalor;
- Filtrarea animalelor după următoarele criterii: diagnostic, tratament;
- 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 în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de animale: procente după specie, diagnostic 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.

Problema 11

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** care poate fi utilizată într-o grădină zoologică. Aplicația 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 tuturor animalelor din grădina zoologică;
- Filtrarea listei animalelor după următoarele criterii: specie, 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 din grădina zoologică;
- Salvare rapoarte/liste cu informațiile despre animale în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de animalele din grădina zoologică: procente după specie, alimentație 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.

Problema 12

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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.

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, xml;
- Vizualizarea unor statistici legate de plantele din grădina botanică: procente după specie 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.

Problema 13

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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, xml.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

- Operații CRUD pentru informațiile legate de utilizatorii care necesită autentificare.

Problema 14

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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;
- Vizualizarea traseului optim în funcție de lista coletelor;
- Căutarea unui colet după număr.

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

- Toate operațiile permise utilizatorilor de tip poștaș;
- Operații CRUD în ceea ce privește persistența coletelor;
- Alocarea unui colet către un poștaș;
- Salvare rapoarte/liste cu informații despre colete în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de colete: procente după greutate, stare 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.

Problema 15

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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 pozițiile deșeurilor ce urmează a fi colecate;
- Vizualizarea traseului optim în funcție de poziția deșeurilor.

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 deșeuri;
- Alocarea unei locații cu deșeuri către un un angajat în vederea colectării;
- Salvare rapoarte/liste cu informații despre listele cu pozițiile deșeurilor alocate angajaților în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de deșeuri: procente după stare 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.

Problema 16

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** care poate fi utilizată într-un service auto. Aplicația va avea 2 tipuri de utilizatori: angajat și administrator.

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- Vizualizarea tuturor autovehiculelor existente în service;
- Filtrarea autovehiculelor după anumite criterii: marcă, culoare, combustibil;
- Căutarea unui autovehicul după proprietar;
- Operații CRUD în ceea ce privește persistența autovehiculelor și a proprietarilor acestora;
- Salvare rapoarte/liste cu informații despre autovehicule în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de autovehicule: procente după marcă, combustibil 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.

Problema 17

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** care poate de către o casă de producție filme. Aplicația va avea 2 tipuri de utilizatori: angajat și administrator.

Utilizatorii de tip angajat pot efectua următoarele operații după autentificare:

- Vizualizarea listei filmelor;
- 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 rapoarte/liste cu informații despre filme în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de filme: procente după tip, categorie 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.

Problema 18

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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.

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/liste cu informații despre pachetele oferite de agenție în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de pachetele turistice: procente după destinație, preț 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.

Problema 19

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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;
- Salvare rapoarte/liste cu informații despre trenuri în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de trenuri: procente după statie de plecare, destinație 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.

Problema 20

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** care poate fi utilizată de către o agenție de rezervare bilete de avion.

Aplicația va avea 3 tipuri de utilizatori: călător, angajat și administrator.

Utilizatorii de tip călător pot efectua următoarele operații fără autentificare:

- Vizualizarea listei zborurilor după aeroport de plecare, destinație, durată;
- Vizualizarea listei zborurilor dintre 2 locații, inclusiv preț și disponibilitate locuri libere;
- Căutarea unui zbor 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 zborurilor și biletelor vândute;
- Vânzarea unui bilet către un călător;
- Salvare rapoarte/liste cu informații despre zboruri în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de zboruri: procente după aeroport de plecare, destinație 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.

Problema 21

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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.

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, xml;
- Vizualizarea unor statistici legate de locuințe: procente după locație, număr 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.

Problema 22

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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.

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, xml.
- Vizualizarea unor statistici legate de camere: procente după disponibilitate, poziție 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.

Problema 23

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** care poate fi utilizată ca soft educațional pentru studiul cercului. Aplicația va avea un singur tip de utilizator care va putea 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.

Problema 24

Descrierea aplicației:

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

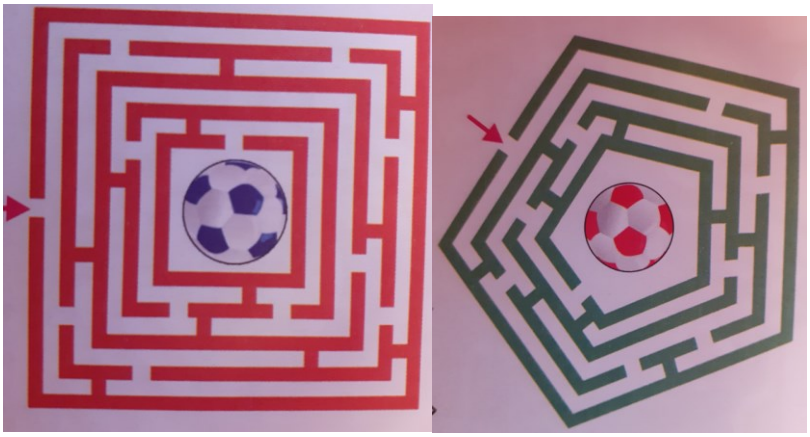
- desenarea interactivă a suprafețelor de rotație (sferă, pseudosferă, tor, elicoid, etc.) prin înlocuirea creionului și a riglei cu *mouse*-ul și alegerea stilului de desenare (culoare, stil linie, grosime linie);
- determinarea și desenarea intersecției suprafeței de rotație cu un plan;
- determinarea 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.

Problema 25

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** pentru următorul joc: se dă un labirint sub formă de poligon regulat având în centru o minge de fotbal. Ajutați un copil să ajungă la minge.

Jocul va fi dezvoltat pe mai multe niveluri (triunghi echilateral, pătrat, pentagon regulat și hexagon 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*).



Problema 26

Descrierea aplicației:

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

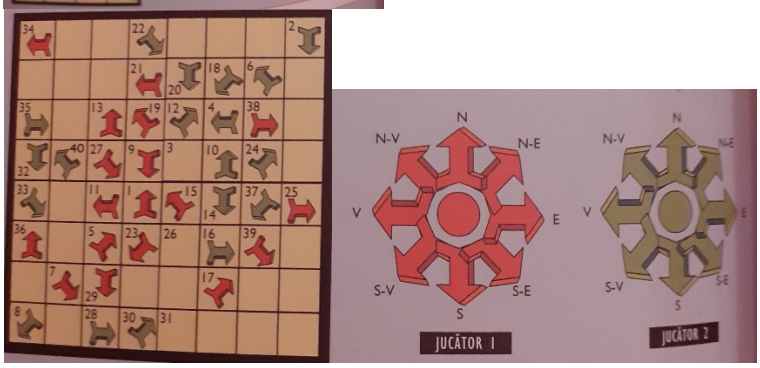
Problema 27

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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 28

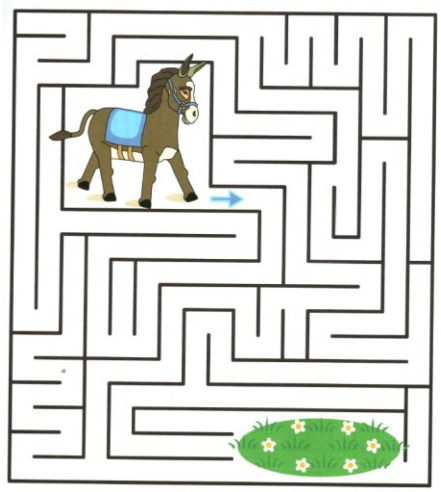
Descrierea aplicației:

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

Problema 29

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** pentru următorul joc: se consideră un labirint similar cu cel prezentat în imaginea următoare. Ajutaţi căluțul să ajungă la păşune, ştiind că el se poate deplasa fie pe orizontală, fie pe verticală.



Jocul va fi dezvoltat pe cel puțin 3 niveluri de dificultate. 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 30

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o ***aplicație desktop*** 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.

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 liste filtrate cu informații despre evenimentele contractate în mai multe formate: csv, json, xml;
- Vizualizarea unor statistici legate de evenimente: procente după scop, locatie 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.