

**LICEUL TEORETIC „STEPHAN LUDWIG ROTH”
MEDIAȘ**

PROIECT
pentru obținerea
Atestatului Profesional la Informatică

Îndrumător:
prof. Enea Constantin

Autor:
Veliciu Vlad

Mediaș
2020

**LICEUL TEORETIC „STEPHAN LUDWIG ROTH”
MEDIAȘ**

Învățarea și aplicarea elementelor de curent continuu

**Mediaș
2020**

Cuprins

INTRODUCERE.....	2
CAPITOLUL 1. CURENTUL CONTINUU - CONCEPTE.....	4
CAPITOLUL 2. PRINCIPII DE DEZVOLTARE ALE PROIECTULUI.....	5
CAPITOLUL 3. PREZENTAREA APLICAȚIEI.....	8
3.1 Modalitatea de dezvoltare a aplicației pornind de la date empirice.....	8
3.2 Modelul fizic al programului.....	9
CONCLUZII.....	16
BIBLIOGRAFIE.....	17
ANEXE.....	18
Anexa nr. 1 - Codul sursă.....	18
Anexa nr. 2 - Modelul conceptual.....	74

Introducere

Tema proiectului este legată de învățarea și aplicarea elementelor de curent continuu. Cu ajutorul acestui proiect elevii de gimnaziu, dar și cei de liceu pot învăța mai ușor materia la fizică despre curentul continuu, existând un modul teoretic și unul practic. Accesând modulul teoretic, elevul (utilizatorul) poate învăța și totodată aprofunda teoria legată de curentul continuu, precum și formulele. Cu ajutorul modulului practic elementele teoretice pot fi concretizate prin câteva exemple de circuite pe care le pot forma elevii, dar și profesorii, astfel încât aplicabilitatea cunoștințelor acumulate în modulul teoretic să poată fi înțeleasă de orice elev. Rezultatele așteptate în urma utilizării aplicației sunt legate de atragerea elevilor spre acest domeniu al electricității, dar, de asemenea, poate fi utilizată în cadrul orelor de curs de la școală, aducând o simplificare a modului de învățare și aplicare. Informațiile folosite în cadrul proiectului le-am obținut din diferite cărți despre curent continuu (mai ales pentru partea teoretică), dar și din cunoștințe proprii. Pe baza acestor informații s-au putut realiza capitolele din modulul teoretic.

Obiectivele acestui proiect sunt așadar ușurarea învățării și predării elementelor de curent continuu, proiectul constituind o modalitate eficientă de prezentarea a acestora. De asemenea, aplicația dezvoltată este ușor de folosit de către orice utilizator, meniul fiind construit într-un mod sintetic în așa fel încât utilizatorul poate alege opțiunile dorite foarte simplu.

Pentru dezvoltarea aplicației au fost folosite diferite metode cu ajutorul cărora se pot realiza afișările datelor circuitelor de utilizator, modificarea acestora, adăugarea, ștergerea sau modificarea datelor unui utilizator de către un administrator al bazei de date (de exemplu profesorul de la clasă), modificarea parolei unui utilizator, calculul unor caracteristici ale circuitului (de exemplu: puterea dezvoltată, intensitatea curentului ș.a.), precum și afișarea circuitului pe ecran astfel încât utilizatorul poate înțelege informațiile prezentate de aplicație într-un mod mai eficient. Sunt, de asemenea, folosite metode pentru calculul rezistențelor echivalente din circuit, tensiunilor electromotoare și pentru afișarea capitolelor teoretice. Toate aceste metode sunt prevăzute cu anumite condiții pentru a fi salvate astfel încât utilizatorul nu poate introduce date care să nu fie conforme cu cele deja existente în baza de date. Acest aspect permite lucrul doar cu tip de date prevăzut între paranteze în opțiunile de introducere de date.

Proiectul este format din două părți care reprezintă statutul persoanei care accesează aplicația: administrator și utilizator. Administratorul trebuie să introducă o parolă înainte de a avea acces la opțiunile prevăzute pentru acest statut: afișarea utilizatorilor (codul utilizatorului, username, parolă, număr de circuite), adăugarea, modificarea și ștergerea utilizatorilor și modificarea parolei de acces. Utilizatorul trebuie să introducă username-ul și parola pentru a putea accesa opțiunile aferente. El va trebui să aleagă între cele două module: teoretic și practic.

Modulul teoretic cuprinde nouă capitole despre curentul continuu care pot fi accesate, iar modulul practic opțiuni care facilitează afișarea unui circuit, a tuturor circuitelor, modificarea unui circuit, adăugarea și ștergerea unei baterii, a unei rezistențe sau a unui circuit, precum și modificarea parolei de acces. La închiderea aplicației utilizatorul va avea opțiunea de a oferi un feedback despre aplicație. De asemenea, toate meniurile cu opțiuni au și posibilitatea de a reveni la meniul anterior.

Capitolul 1. Curentul continuu - concepte

Curentul continuu face parte din capitolul electricitate, acest capitol având o importanță semnificativă în viața noastră de zi cu zi.

Electricitatea este o formă de energie, deci nu a fost descoperită. Electricitatea statică a fost descoperită de grecii antici în jurul anului 600 î.Hr., observând că la frecarea dintre blană și chihlimbar există o atracție între cele două elemente. Au existat mai multe descoperiri în istorie despre electricitate, una semnificativă fiind cea a fizicianului italian Alessandro Volta, el construind prima baterie electrică în anul 1800. El a creat și primul circuit electric, legând între ele bornele pozitive și negative, rezultând o tensiune electrică. În 1831 Michael Faraday a creat un generator de curent continuu. Generatorul a constatat dintr-un magnet care putea să se deplaseze într-o bobină din sârmă de cupru, generând astfel un curent electric care curgea prin firul bobinei. Această descoperire a stat la baza invențiilor lui Thomas Edison și Joseph Swan care au creat becul cu filament. La sfârșitul secolului 19 și începutul secolului 20, inventatorul sârbo-american Nikola Tesla a avut o contribuție semnificativă la aplicațiile comerciale ale electricității. El a obținut progrese importante în electromagnetism, bazele pentru inventarea radioului. El este recunoscut și pentru cercetările sale despre aplicațiile curentului alternativ, motoarele de curent alternativ. Printre cei a căror activitate a condus la utilizarea energiei electrice așa cum este în prezent se numără inventatorul scoțian James Watt, Andre Ampere, un matematician francez, matematicianul și fizicianul german George Ohm. Electricitatea a fost deci descoperită de mai mulți oameni de știință, fiecare având o contribuție esențială în înțelegerea acestui fenomen din natură.

Înțelegerea fenomenului acesta este așadar esențială pentru a putea fi capabil de a percepe ceea ce ne înconjoară. Un prim pas pentru elevi pentru a-și dezvolta cunoștințele despre lumea modernă și tehnologizată de astăzi este învățarea și înțelegerea fenomenelor legate de curentul continuu. Acestea reprezintă baza lumii de astăzi, căci fără electricitate, deci fără curent continuu și alternativ, toate mijloacele tehnice pe care oamenii le folosesc astăzi nu ar putea funcționa.

Această aplicație vine în ajutorul celor care vor să învețe și să aprofundeze cunoștințe legate de curentul continuu, fiind foarte ușor de utilizat, cu o interfață simplă astfel încât orice utilizator poate valorifica la maximum opțiunile oferite de aplicație și cu informații și aplicații menite să explice utilizarea elementelor de curent continuu. (informații preluate și prelucrate de pe site-ul <https://www.stiintaonline.ro/cine-a-descoperit-electricitatea/>)

Capitolul 2. Principii de dezvoltare ale proiectului

Stadiul inițial al aplicației înfățișează un meniu în care sunt expuse opțiunile pe care persoana care deschide aplicația le poate alege. Aceste opțiuni reprezintă statutul persoanei care folosește aplicația: administrator sau utilizator.

Statutul de administrator (care poate fi profesorul de fizică de la o clasă) se referă la persoana care poate vizualiza, modifica, adăuga și șterge orice utilizator. Acesta poate accesa opțiunile prescrise pentru el doar introducând o parolă. Administratorul este deci unic și din acest motiv există doar un fișier în care este memorată parola de acces pentru acest statut. Această parolă poate fi oricând modificată de administrator. Există totuși niște limite în legătură cu dimensiunea parolei, și anume maximum 20 de caractere (doar cifre, litere și unele caractere speciale ('.', ';', '_', ' ')). De asemenea, trebuie să se aibă în vedere faptul că username-ul și parola utilizatorului pot avea o lungime de maximum 20 de caractere (aceleași ca la parola administratorului). De asemenea, codul de identificare al unui utilizator poate avea maximum 3 cifre. Numărul maxim de circuite care pot fi atribuite unui utilizator este de 3 (minimum 1), iar codul pentru fiecare circuit poate fi format din maximum 3 cifre. Numărul maxim de baterii dintr-un circuit este 5 (minimum 1), iar cel de rezistențe exterioare 5. Codurile pentru baterii și rezistențe pot fi formate din maximum 3 cifre. Valoarea maximă a tensiunii electromotoare a bateriei este 24 (minimum 1), cea a rezistenței interioare 15 și cea a rezistenței exterioare 20. De asemenea, în cazul în care există mai mult de două baterii sau rezistențe, trebuie să fie aleasă modalitatea de grupare a acestora în circuit (serie sau paralel). În cazul afișării utilizatorilor administratorul poate vedea doar codul, username-ul, parola și numărul de circuite ale acestora. La adăugarea utilizatorilor administratorul trebuie să ia în calcul faptul că numărul maxim de utilizatori permiși este de 40.

Statutul de utilizator (care poate fi un elev) se referă la persoana care accesează aplicația pentru a învăța și a pune în practică elemente legate de curentul continuu. Utilizatorul poate accesa meniul conceput pentru el introducând username-ul său și parola aferentă. Odată introduse corect, utilizatorul are de ales între accesarea modulului teoretic și a modulului practic. Modulul teoretic cuprinde nouă capitole despre curentul continuu: "Ce este curentul electric?", "Tensiunea electrică", "Intensitatea curentului electric", "Rezistența electrică", "Legea lui Ohm și legea lui Kirchhof", "Legarea rezistentelor", "Legarea generatoarelor", "Energia curentului electric", "Puterea electrică". Aceste capitole sunt formate din fișe cu teorie și formule menite să prezinte elementele de curent continuu. Modulul practic facilitează afișarea unui circuit, a tuturor circuitelor, adăugarea, modificarea și ștergerea unei baterii, a unei rezistențe sau a unui circuit. În cazul afișării unui circuit, utilizatorul trebuie să introducă codul circuitului și astfel vor fi afișați

parametrii circuitului (codul și valoarea tensiunii electromotoare a bateriei/a bateriilor, modul de legare a acestora în circuit (serie/paralel), codul și valoarea rezistenței interioare a bateriei/bateriilor, valoarea rezistenței/rezistențelor exterioare, modul de legare a acestora (serie/paralel) precum și valorile echivalente ale acestora (în cazul grupării în serie/paralel), intensitatea curentului în ramura principală, tensiunea pe circuitul exterior, energia degajată pe circuitul exterior într-un minut, puterea dezvoltată pe circuitul exterior și puterea maximă). De asemenea, pe tot parcursul acestor afișări, în partea dreaptă a ecranului se va putea vedea o posibilă schemă a circuitului descris. În cazul afișării tuturor circuitelor, utilizatorul va vedea codul circuitului, codul și valoarea tensiunii electromotoare a bateriei/a bateriilor, modul de legare a acestora în circuit (serie/paralel), codul și valoarea rezistenței interioare a bateriei/bateriilor, valoarea rezistenței/rezistențelor exterioare, modul de legare a acestora (serie/paralel). Alegând opțiunea modificării unui circuit, utilizatorul trebuie să introducă codul circuitului, iar apoi va trebui să aleagă între modificarea unei baterii, a unei rezistențe sau a legării acestora, respectând limitele indicate în cazul statutului de administrator. Opțiunile referitoare la adăugare și ștergere se referă la baterie, rezistență exterioară și la circuit, în cazul ștergerii fiind necesară introducerea codului circuitului. Adăugarea are și ea loc cu respectarea normelor prezentate la statutul de administrator. Utilizatorul poate modifica parola proprie, respectând aceleași norme ca și administratorul. Datele despre utilizatori, circuite, baterii și rezistențe sunt salvate în fișiere diferite, iar legătura între aceste fișiere este realizată prin codul prin care este caracterizat fiecare element din fișier. De asemenea, înainte de închiderea aplicației, utilizatorul poate oferi un feedback cu privire la eficiența aplicației care va fi salvat într-un fișier separat și va fi accesibil doar dezvoltatorului aplicației.

Formulele folosite în cadrul proiectului sunt preluate din manualul de fizică pentru clasa a 10-a, capitolul curent continuu:

1. Tensiunea electromotoare echivalentă la gruparea în serie a bateriilor:

$$E_{es} = E_1 + E_2 + \dots + E_n$$

2. Tensiunea electromotoare echivalentă la gruparea în paralel a bateriilor:

$$E_{ep} = \frac{\frac{E_1}{r_1} + \frac{E_2}{r_2} + \dots + \frac{E_n}{r_n}}{\frac{1}{r_1} + \frac{1}{r_2} + \dots + \frac{1}{r_n}}$$

3. Valoarea rezistenței electrice la gruparea în serie: $R_{es} = R_1 + R_2 + \dots + R_n$; $r_{es} = r_1 + r_2 + \dots + r_n$

4. Valoarea rezistenței electrice la gruparea în paralel:

$$\frac{1}{R_{ep}} = \frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_n} \quad \frac{1}{r_{ep}} = \frac{1}{r_1} + \frac{1}{r_2} + \dots + \frac{1}{r_n}$$

5. Intensitatea curentului: $I = \frac{U}{R}$

6. Tensiunea pe circuitul exterior: $U = I \cdot R$

7. Energia electrică pe circuitul exterior: $W = U \cdot I \cdot \Delta t$

8. Puterea electrică pe circuitul exterior: $P = U \cdot I$

9. Puterea maximă: $P = \frac{E^2}{4 \cdot r}$

Capitolul 3. Prezentarea aplicației

3.1 Modalitatea de dezvoltare a aplicației pornind de la date empirice

Datele empirice folosite în aplicație sunt valoarea tensiunii electromotoare a bateriei (int E), valoarea rezistenței interioare a bateriei (int r), valoarea rezistenței exterioare (int R), precum și codul bateriei (int codB) și al rezistenței exterioare (int codR). Bateria și rezistența exterioară au fost declarate cu ajutorul unor structuri, fiind incorporate în interior unei alte structuri, "curent". Această structură cuprinde caracteristicile unui circuit: codul (int code), numărul de baterii (int nr_B), numărul de rezistențe exterioare (int nr_R), rezistențele exterioare (rezistenta_exterioara R[100]) și bateriile (baterie B[100]) (legate de structurile aferente), legarea bateriilor (int legare_B) și rezistențelor (int legare_Re). Structura "manager" se referă la utilizator și cuprinde attributele acestuia: codul (int codm), username (char username[100]) și parolă (char password[100]), apelarea structurii "curent" printr-un vector (curent c[10]) și numărul de circuite (int nr_c). De asemenea, structura "manager" este apelată în funcția principală printr-un vector (manager a[50]). Aceste date sunt prelucrate cu ajutorul mai multor funcții care facilitează calculul unor aspecte referitoare la circuitul electric (de exemplu intensitatea curentului), permit accesul utilizatorului, respectiv a administratorului, în aplicație, afișarea, ștergerea, modificarea circuitelor sau a parolei unui utilizator/a administratorului.

Există așadar două entități care reprezintă baza aplicației: "baterie", "rezistenta_exterioara". Attributele care caracterizează entitatea "baterie" sunt codB, E și r, iar cele care caracterizează entitatea "rezistenta_exterioara" sunt codR și R. Aceste entități transmise date entității "curent" prin codul care caracterizează fiecare baterie, respectiv rezistență exterioară care corespunde unui circuit. De asemenea, această entitate are și ea attribute specifice: code, nr_B, nr_R, legare_B, legare_Re. Această entitate transmite la rândul ei date către entitatea "manager" prin codul specific circuitului care corespunde unui utilizator. Entitatea aceasta are și ea attribute specifice: codm, username, password, nr_c. Așadar, toate subprogramele folosite se bazează pe aceste attribute preluate de la toate entitățile, existând o relație ierarhică între ele, sau pe informații obținute prin prelucrarea datelor oferite de aceste attribute. Pentru fiecare entitate în parte există câte un fișier care conține attributele specifice entității respective, iar legătura ierarhică între ele este realizată prin codurile fiecărei instanțe. Entitățile inițiale sunt deci "baterie" și "rezistenta_exterioara" care sunt subordonate entității "curent", aceasta la rândul ei fiind subordonată entității "manager".

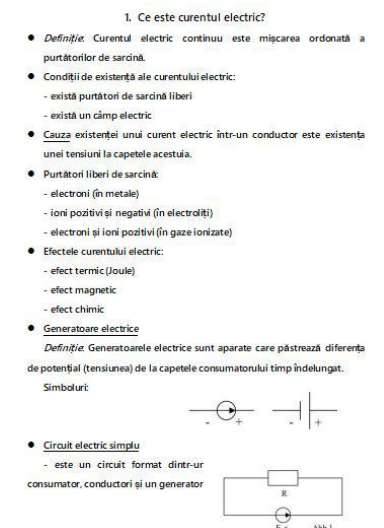
Relația între entități este structurată și în modelul conceptual din anexa nr. 2.

3.2 Modelul fizic al programului

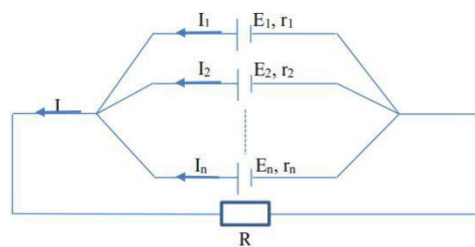
Proiectul se bazează pe modelul conceptual prezentat mai sus. Toate funcțiile și subprogramele folosite se bazează pe atributele din fiecare entitate. Subprogramul principal stabilește statutul persoanei care folosește aplicația și apare sub formă de meniu. Toate meniurile folosite în această aplicație obligă utilizatorul să introducă numărul aferent opțiunii dorite astfel încât aplicația nu va executa nimic până nu este introdus un număr al unei opțiuni prezentate. Ultima opțiune este mereu cea de întoarcere la meniul anterior. De asemenea, în funcția principală a aplicației este inițializat modul grafic, având culoarea fundalului RGB (75, 35, 255) și dimensiunile ferestrei sunt maxime; denumirea ferestrei este "Curent continuu". Sunt inițializați și parametri referitori la fereastra în care se deschide aplicația, și anume lungimea și lățimea, care vor apărea ca parametrii în fiecare subprogram. Lungimea (int lu) a fost inițializată cu `getmaxheight() / 8`, iar lățimea (int la) cu `getmaxwidth() / 16`. Afișarea textelor pe tot parcursul aplicației se va face cu ajutorul acestor parametrii, fiind folosite rapoarte în așa fel încât textul să fie încadrat cât mai bine pe ecran. Culoarea scrisului este albă, pentru a oferi un contrast plăcut cu fundalul, iar stilul este inițializat prin `settextstyle(1, 0, 18)`. Sunt folosite și funcții care facilitează citirea textului, respectiv a numerelor în modul grafic: `void citire_nr (int &nrt, int nr, int x, int y)` și `void citire_text (char t[], int nr, int x, int y)`, unde x și y sunt parametrii de lungime și lățime a ecranului și nr este lungimea maximă pe care o poate avea textul, respectiv numărul citit, vectorul de caractere t este textul introdus, iar referința nrt este numărul introdus. O altă funcție importantă pentru utilizarea modului grafic este funcția `itoa`, folosită în cadrul aplicației pentru a converti un număr într-un șir de caractere pentru a putea fi afișat pe ecran. Funcția de afișare în modul grafic este `outtextxy`, pentru folosirea căreia sunt esențiali parametrii de lungime și lățime a ecranului (textul fiind încadrat pe ecran cu ajutorul acestora). Exemplu: `outtextxy(la, lu, "Username: (max 20 de caractere) ")`. În componența acestei funcții sunt folosite deseori și funcțiile `textheight` și `textwidth` cu ajutorul cărora sunt așezate textele pe ecran într-un mod cât mai eficient, locul lor pe ecran fiind dependent de mărimea celorlalte texte și astfel este imposibil ca textele să se intercaleze. După fiecare afișare de text (în cazul în care nu trebuie introduse date), este afișat un mesaj ("Apasati orice tasta!"), urmat de funcția `getch()`, astfel încât utilizatorul să aibă mereu parte de o interfață care îl ghidează. Importante în modul grafic, pentru a avea o interfață cât mai prietenoasă cu utilizatorul, sunt imaginile. Imaginile sunt folosite atât la modulul teoretic, unde pentru fiecare

capitol există o imagine cu informații, cât și la afișarea unui circuit de către utilizator. Exemple:

img. 3.3.1 - fișa de
teorie a unui capitol
din modulul
teoretic



img. 3.3.2 - o schemă a
unui circuit



Pentru a accesa meniul administratorului, trebuie să fie introdusă o parolă care este preluată dintr-un fișier dedicat exclusiv acesteia. Odată cu introducerea corectă a parolei, se deschide meniul care presupune mai multe opțiuni: afișare, adăugare, modificare, ștergere utilizatori și modificare parolă. Opțiunea de afișare utilizatori, va afișa sub formă de tabel următoarele date ale tuturor utilizatorilor: cod, username, parolă, număr circuite. Pentru ca această afișare să fie făcută într-un mod cât mai atractiv, au fost folosiți parametri specifici modului grafic (lungime, respectiv lățimea ecranului) cu ajutorul cărora au fost formate rapoarte în așa fel încât afișarea să apară sub formă de tabel.

Exemplu: (capul de tabel)

```
outtextxy(la, lu, "COD ");
outtextxy(la*2, lu, "USERNAME ");
outtextxy(la*7, lu, "PAROLA ");
outtextxy(la*11, lu, "NR CIRCUITE");
```

Subprogramul de adăugare a unui utilizator este construit cu rigurozitate în așa fel încât introducerea de date să fie conformă cu normele prezentate în capitolul 2. În acest subprogram sunt folosite mai multe structuri repetitive de tip while care permit utilizatorului introducerea greșită a unor date doar o singură dată. Eficiența acestei metode constă în faptul că introducerea datelor nu este anulată imediat ce a fost introdusă o valoare greșită, ci oferă persoanei care introduce valori să greșească o singură dată. Evidența numărului de greșeli este ținută cu ajutorul unui contor (int ct). Un exemplu este introducerea codului unui nou utilizator (nou se referă la utilizator):

```
while (ct<2)
{
    cleardevice();
    outtextxy(la, lu, "Tastati codul utilizatorului: (max 3 cifre)");
    citire_nr(nou.codm, 100, la, lu+textheight("Tastati codul
    utilizatorului: (max 3 cifre) "));
    bool e=cautare_cod_manager(a, m, nou.codm);
```

```

        if (nou.codm<0||nou.codm>999)
        {
            outtextxy(la,lu+2*textheight("Tastati codul utilizatorului:
(max 3 cifre) "), "Cod incorect! ");
            outtextxy(la,lu+3*textheight("Tastati codul utilizatorului:
(max 3 cifre) "), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        if (e)
        {
            outtextxy(la,lu+2*textheight("Tastati codul utilizatorului:
(max 3 cifre) "), "Acest utilizator exista deja!");
            outtextxy(la,lu+3*textheight("Tastati codul utilizatorului:
(max 3 cifre) "), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        if (nou.codm>=0&&nou.codm<=999&&!e)
        {
            ct=2;
            sem=1;
        }
    }
}

```

După finalizarea introducerii datelor, administratorul poate vedea o parte din datele introduse (username și parolă) și poate opta între salvarea noului utilizator sau ștergerea datelor introduse. În acest mod administratorul are mereu controlul asupra datelor introduse, evitându-se astfel salvarea unor date nedorite.

Opțiunea de modificare a unui utilizator permite administratorului să modifice username-ul sau parola (respectând normele) sau componente dintr-un circuit al utilizatorului, referitoare la baterie, rezistență exterioară, legarea bateriilor și legarea rezistențelor. Pentru a modifica elementele unui circuit, trebuie introdus codul circuitului. În cazul modificării unei baterii sau a unei rezistențe, trebuie să fie introdus codul corespunzător, iar dacă acesta este existent în cadrul circuitului corespunzător, pot fi modificate valorile atributelor specifice (nu codul), respectând normele referitoare la limitele valorilor. Sunt, de asemenea, folosite subprograme pentru a determina poziția utilizatorului, a circuitului, a bateriei și a rezistenței în vectorii corespunzători pe baza codului unic. Aceste subprograme returnează deci poziția din vector dacă a fost găsit codul căutat și valoarea -1 în caz contrar, o valoare imposibilă din punct de vedere al indicilor vectorului (care încep de la 0). Aceste subprograme sunt apelate în mai multe funcții CRUD din aplicație, oferindu-se astfel o eficiență destul de mare acestor funcții.

Alte opțiuni la care administratorul poate opta sunt ștergerea unui utilizator și modificarea parolei de acces. Alegând opțiunea de ștergere a unui utilizator, administratorul trebuie să introducă codul utilizatorului, în cazul introducerii corecte afișându-se username-ul, parola și numărul de circuite ale acestuia. Ca și în cazul funcțiilor

de adăugare sau modificare, administratorul are posibilitatea să anuleze ștergerea, astfel încât controlul său asupra datelor este maximizat. Se poate observa și simplitatea acestei secvențe prin faptul că administratorul este obligat să aleagă o opțiune (să șteargă utilizatorul sau să anuleze operațiunea):

```
int x=0;
outtextxy(la,lu+4*textheight("Informatiile userului "), "Doriti sa
eliminati userul (apasati 1) sau sa anulati operatiunea (apasati 2)?");
citire_nr(x,10,la,lu+5*textheight("Informatiile userului
"));
int ata=6;//folosita pentru afisarea pe ecran a textului
fara a exista intercalari
while ((x!=1)&&(x!=2))
{
outtextxy(la,lu+lu+ata*textheight("Informatiile
userului "), "Alegeti o optiune intre 1 si 2!");
citire_nr(x,10,la,lu+(ata+1)*textheight("Informatiile
userului "));
ata++;
}
```

Opțiunea de modificare a parolei obligă administratorul să introducă parola curentă pentru a o putea modifica. În cazul în care noua parolă corespunde normelor specifice, ea este salvată într-un fișier prin suprascrierea parolei vechi.

Accesarea statutului de utilizatori presupune introducerea unui username și a parolei corespunzătoare. Identificarea utilizatorului este făcută printr-un subprogram în care sunt folosite funcția specifică șirurilor de caractere `strcmp` pentru a compara username-ul și parola introduse cu cele existente. Odată introduse corect, utilizatorul va trebui să aleagă între modulul teoretic și cel practic.

Modulul teoretic cuprinde imagini cu fișe de teorie despre curentul continuu, grupate în 9 capitole. La alegerea unui capitol este citită imaginea corespunzătoare, utilizând funcția `readimagefile`. Pentru folosirea acestei funcții sunt importanți parametri de lungime și lățime a ecranului.

Modulul practic cuprinde mai multe opțiuni. Alegând opțiunea de afișare a unui circuit, utilizatorul va trebui să introducă codul unui circuit. În cazul introducerii corecte vor fi afișate informații despre baterii și legarea lor în circuit, despre rezistențele exterioare și legarea lor, echivalența legării bateriilor și rezistențelor, parametrii circuitului (intensitate, tensiune, energie, putere și putere maximă). Există pentru fiecare dintre parametrii circuitului o formulă de calcul preluată din manualul de fizică. De asemenea, pentru calculul echivalențelor bateriilor și rezistențelor sunt concepute subprograme pe baza formulelor aferente. Un exemplu este subprogramul care calculează echivalența bateriilor grupate în paralel pe baza formulei din capitolul 2 (m este numărul de utilizatori, codu este codul utilizatorului, codc este codul circuitului, referința ee este valoarea tensiunii electromotoare echivalente, referința er este valoarea rezistenței interioare echivalente, se este suma raporturilor dintre tensiunea

electromotoare și rezistență interioară a unei baterii, sr este suma rezistențelor interioare, pr este produsul rezistențelor interioare):

```
void echivalent_paralel_baterie (manager a[], int m, curent c[], int
codu, int codc, int &ee, int &er)
{
    int se=0, sr=0, pr=1;
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    for (int i=0;i<a[o].c[p].nr_B;i++)
    {
        se+=a[o].c[p].B[i].E/a[o].c[p].B[i].r;
        sr+=a[o].c[p].B[i].r;
        pr*=a[o].c[p].B[i].r;
    }
    er=pr/sr;
    ee=se/(1/er); }
```

De asemenea, în cazul afișării unui circuit în partea dreaptă a ecranului va fi afișată schema aferentă circuitului. A fost conceput un subprogram eficient care va determina ce poză trebuie afișată în funcție de elementele din circuit (de menționat că legarea în serie a rezistențelor și bateriilor este memorată cu 1, iar legarea în paralel a acestora cu 2):

```
void poze_circuit(int m, manager a[], curent c[], int o, int p)
{
    if (a[o].c[p].nr_B==1&&a[o].c[p].nr_R==1)

readimagefile("elr1.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwid
th(),getmaxheight()/2);
        else
            if (a[o].c[p].legare_B==1)
                if (a[o].c[p].legare_Re==1)

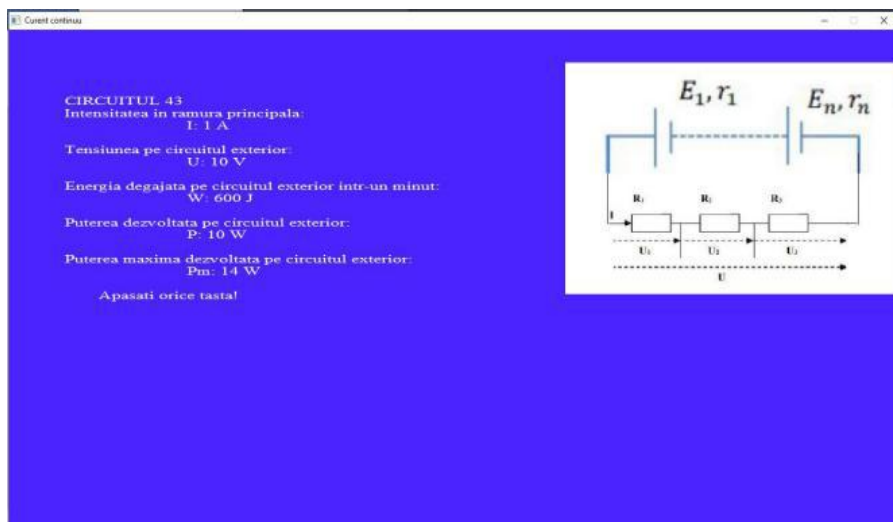
readimagefile("esrs.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwid
th(),getmaxheight()/2);
                    else

readimagefile("esrp.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwid
th(),getmaxheight()/2);
                        if (a[o].c[p].legare_B==2)
                            if (a[o].c[p].legare_Re==1)

readimagefile("eprs.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwid
th(),getmaxheight()/2);
                                    else

readimagefile("eprp.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwid
th(),getmaxheight()/2);
}
```

În imaginea de mai jos se poate observa cum arată ecranul în cazul afișării unui circuit:



img. 3.3.3 -
schema
circuitului și
informațiile
aferente

Opțiunea de adăugare permite adăugarea unei baterii, a unei rezistențe sau a unui circuit. Sunt folosite, ca și în funcțiile administratorului, interogări specifice pentru a se asigura respectarea normelor (de exemplu structuri de tip if, while). Se observă însă și eficiența construirii aplicației prin faptul că sunt apelate des subprograme care facilitează CRUD pentru mai multe funcții. Opțiunile de ștergere și modificare sunt construite în același mod, permițând managementul de baterii, rezistențe și circuit, și având numeroase interogări pentru a împiedica salvarea unor date nepermise.

De asemenea, utilizatorul își poate modifica propria parolă de acces, noua parolă fiind salvată, în cazul în care sunt respectate normele, tot prin suprascriere în fișierul dedicat utilizatorilor. Înainte de a ieși din aplicație, utilizatorul este rugat să lase un comentariu despre utilizarea aplicației (nu este obligatoriu). Prin acest feedback dezvoltatorul aplicației va lua la cunoștință părerile utilizatorilor și va putea îmbunătăți aplicația pe viitor. Feedback-ul este salvat într-un fișier separat, fiind accesibil exclusiv dezvoltatorului.

Așadar, aplicația folosește diferite subprograme cu ajutorul cărora sunt realizate citiri, identificări, interogări, funcții de tip CRUD, funcții care facilitează afișarea grafică și funcția de feedback. Variabilele esențiale care au rolul de a bloca introducerea de valori nepermise apar aproape în fiecare subprogram. Ele sunt bool sem, int ct (deci un semafor și un contor). Semaforul este folosit pentru a determina dacă o valoare corespunde normelor stabilite, iar contorul conține numărul de valori greșite introduse. Contorul este în strânsă legătură cu structura repetitivă while, iar semaforul apare în structuri de tip if. Aceste două variabile sunt deci importante, oferind eficiență programelor, dar și certitudinea că nicio valoare nepermisă nu va putea fi salvată sau prelucrată de aplicație.

Funcția de citire este și ea importantă, preluând în același timp date din toate fișierele necesare (date referitoare la baterii, rezistențe, circuite, utilizatori) și incluzându-le în vectori

specifici. După ieșirea din aplicație, toate schimbările datelor sunt salvate în fișierele corespunzătoare, fiind salvate în aceeași ordine în care au fost citite.

Citirea datelor (se observă citirea simultană a tuturor datelor care sunt caracteristice unui utilizator, precum și a celor preluate prin relația de ierarhie):

```
void citire_c (curent c[], char nf1[], char nf2[], manager a[], int &m,
char nf3[], char nf4[], rezistenta_exterioara R[], baterie B[])
{
    ifstream f(nf1);
    ifstream g(nf2);
    ifstream h(nf3);
    ifstream k(nf4);
    f>>m;
    for (int i=0;i<m;i++)
    {
        f>>a[i].codm>>a[i].username>>a[i].password>>a[i].nr_c;
        for (int j=0;j<a[i].nr_c;j++)
        {
            g>>a[i].c[j].codc>>a[i].c[j].nr_B>>a[i].c[j].legare_B>>a[i].c[j].nr_R>
            >a[i].c[j].legare_Re;
            for (int l=0;l<a[i].c[j].nr_R;l++)
                h>>a[i].c[j].R[l].codR>>a[i].c[j].R[l].Re;
            for (int l=0;l<a[i].c[j].nr_B;l++)
                k>>a[i].c[j].B[l].codB>>a[i].c[j].B[l].E>>a[i].c[j].B[l].r;
        }
    }
    f.close();
    g.close();
    h.close();
    k.close();
}
```

Această citire este una eficientă prin faptul că toate datele necesare sunt citite în același timp, iar funcția este construită într-o formă generală, numele fișierelor apărând doar în funcția principală unde este apelată citirea.

Proiectul este deci unul eficient și cuprinzător prin opțiunile diverse pe care le oferă administratorului și utilizatorului. Este tratată fiecare variantă posibilă de introducere a datelor, iar cu ajutorul interogărilor specifice date nepermise nu pot fi prelucrate. De asemenea, funcțiile care realizează CRUD-ul sunt eficiente prin structurile repetitive (for, while) folosite în așa fel încât afișarea, modificarea, ștergerea și adăugarea date să se realizeze într-un mod cât mai simplu. Întregul cod sursă poate fi consultat în Anexa nr. 1.

Concluzii

Aplicația are o aplicabilitate ridicată în ziua de azi, ea putând fi folosită nu doar în sistemul de învățământ de către elevi și profesori, ci și în spațiul public de către oricine. Ea oferă o imagine cuprinzătoare despre teoria și practica elementelor de curent continuu, având o interfață accesibilă pentru oricine, putând fi înțeleasă cu ușurință. De asemenea, imaginile folosite în aplicație sunt folositoare pentru aprofundarea cunoștințelor despre curentul continuu. Eficiența aplicației se observă în faptul că diferite subprograme sunt folosite pentru a executa acțiuni CRUD cu ușurință și într-un timp scurt. Aplicația asigură și siguranța datelor, fiind utilizate parole de acces atât pentru administratorul bazei de date, cât și pentru utilizatorii acesteia. Datele introduse, cât și cele salvate, sunt interogate în timp real de către subprogramele și structurile specifice de tip while și if ale aplicației în așa fel încât o violare a normelor valorilor este imposibilă. Aceste interogări sunt realizate în mod eficient într-un timp foarte scurt astfel că persoana care folosește aplicația poate conștientiza repede dacă datele introduse de el sunt corecte sau nu. Textele afișate pe ecran sunt folositoare în ghidarea utilizatorului și a administratorului, astfel că aceștia pot găsi opțiunea dorită într-un timp scurt. Eu cred că această aplicație este extrem de utilă în ziua de astăzi, mai ales în învățământ, dar nu numai, pentru că elementele de curent continuu sunt explicate foarte simplu și concis și sunt prezentate în cadrul unor mici aplicații practice, astfel că însușirea cunoștințelor în acest domeniu poate fi făcută într-un mod plăcut și interactiv. Aplicația poate fi dezvoltată mai departe. În această formă sunt tratate doar circuitele cu un singur ochi de rețea, dar pe viitor aplicația poate fi extinsă spre abordarea circuitelor cu mai multe ochiuri de rețea. De asemenea, pot fi adăugate mai multe imagini, de exemplu în afișarea unui circuit. Aplicația oferă în momentul de față imagini cu scheme ale circuitului în care sunt prezentate circuite cu un număr n de baterii și rezistențe legate în serie sau în paralel (sau 1 în cazul în care există o baterie sau o rezistență). S-ar putea realiza imagini specifice în așa fel încât numărul de baterii și rezistențe să fie cel care este precizat în fișierul cu circuite. Aplicația este deci una complexă, oferind multe opțiuni, și în viitor ea poate fi dezvoltată mai departe spre mai multe opțiuni și spre o abordare mai complexă a circuitelor electrice de curent continuu.

Bibliografie

1. <https://www.stiintaonline.ro/cine-a-descoperit-electricitatea/> (informațiile din capitolul 1)
2. Manual de Informatică - Clasa a 11-a - Editura Didactică și Pedagogică R.A.
3. <https://profu.info/algoritmi-fundamentali-in-c/>
4. Manual de Fizică - Clasa a 10-a - Editura Corint (formulele folosite)

Anexe

Anexa nr. 1 - Codul sursă

```
#include <iostream>
#include <cstring>
#include <iomanip>
#include <fstream>
#include <conio.h>
#include <windows.h>
#include <graphics.h>
using namespace std;
struct rezistenta_exterioara
{
    int codR;
    int Re;//rezistenta exterioara
};
struct baterie
{
    int codB;
    int E;//tensiunea electromotoare
    int r;//rezistenta interioara
};
struct curent
{
    int codc;//codul circuitului
    int nr_R;//numar rezistente exterioare
    int nr_B;//numar baterii
    rezistenta_exterioara R[100];
    baterie B[100];
    int legare_Re;
    int legare_B;
};
struct manager
{
    int codm;//codul managerului
    char username[100];//numele
    char password[100];//parola
    curent c[10];
    int nr_c;
};
void iniit()
{
    int md, dr, i;
    detectgraph(&dr, &md);
    initgraph(&dr, &md, "");
}
void citire_parola_admin (char nf[], char x[]);
void citire_text (char t[], int nr, int x, int y)
{
    char lit;
    int l=1;
    t[0]='-';
    t[1]='\0';
    do
    {
        outtextxy(x, y, t);
        lit=getch();
        if ((isalpha(lit)||isdigit(lit)|| (lit=='.'))||(lit==','))||(lit=='
')||(lit=='_'))&& lit !=8&&lit!=13&&l<=nr)
        {
```

```

        t[l-1]=lit;
        t[l]='-';
        t[l+1]='\0';
        l++;
    }
    else
        if (lit==8&& l>1)
        {
            outtextxy(x,y,t);
            t[l-2]=t[l-1];
            t[l-1]='\0';
            l--;
            outtextxy(x,y,t);
        }
    while (lit!=13);
    outtextxy(x,y,t);
    t[l-1]='\0';
    outtextxy(x,y,t);
}

void citire_nr (int &nrt, int nr, int x, int y)
{
    char lit, t[1000];
    int l=1;
    t[0]='-';
    t[1]='\0';
    do
    {
        outtextxy(x,y,t);
        lit=getch();
        if (isdigit(lit)&& lit!=8 && lit!=13 && l<=nr)
        {
            t[l-1]=lit;
            t[l]='-';
            t[l+1]='\0';
            l++;
        }
        else
            if (lit==8 && l>1)
            {
                outtextxy(x,y,t);
                t[l-2]=t[l-1];
                t[l-1]='\0';
                l--;
                outtextxy(x,y,t);
            }

    }while (lit!=13);
    nrt=atoi(t);
    outtextxy(x,y,t);
    t[l-1]='\0';
    outtextxy(x,y,t);
}

void citire_c (curent c[], char nf1[], char nf2[], manager a[], int &m, char
nf3[], char nf4[], rezistenta_exterioara R[], baterie B[])
{
    ifstream f(nf1);
    ifstream g(nf2);
    ifstream h(nf3);
    ifstream k(nf4);
    f>>m;
    for (int i=0;i<m;i++)
    {
        f>>a[i].codm>>a[i].username>>a[i].password>>a[i].nr_c;
        for (int j=0;j<a[i].nr_c;j++)

```

```

        {
g>>a[i].c[j].codc>>a[i].c[j].nr_B>>a[i].c[j].legare_B>>a[i].c[j].nr_R>>a[i].c
[j].legare_Re;
        for (int l=0;l<a[i].c[j].nr_R;l++)
            h>>a[i].c[j].R[l].codR>>a[i].c[j].R[l].Re;
        for (int l=0;l<a[i].c[j].nr_B;l++)

k>>a[i].c[j].B[l].codB>>a[i].c[j].B[l].E>>a[i].c[j].B[l].r;
        }
    }
    f.close();
    g.close();
    h.close();
    k.close();
}

void salvare_c (int m, curent c[], char nf1[], char nf2[], manager a[], char
nf3[], char nf4[], rezistenta_exterioara R[], baterie B[])
{
    ofstream f(nf1);
    ofstream g(nf2);
    ofstream h(nf3);
    ofstream k(nf4);
    f<<m;
    f<<'\n';
    for (int i=0;i<m;i++)
    {
        f<<a[i].codm<<"          "<<a[i].username<<"          "<<a[i].password<<"
"<<a[i].nr_c<<'\n';
        for (int j=0;j<a[i].nr_c;j++)
        {
            g<<"          "<<a[i].c[j].codc<<"          "<<a[i].c[j].nr_B<<"
"<<a[i].c[j].legare_B<<" "<<a[i].c[j].nr_R<<" "<<a[i].c[j].legare_Re<<'\n';
            for (int l=0;l<a[i].c[j].nr_R;l++)
                h<<a[i].c[j].R[l].codR<<" "<<a[i].c[j].R[l].Re<<'\n';
            for (int l=0;l<a[i].c[j].nr_B;l++)
                k<<a[i].c[j].B[l].codB<<"          "<<a[i].c[j].B[l].E<<"
"<<a[i].c[j].B[l].r<<'\n';
        }}
        f.close();
        g.close();
        h.close();
        k.close();
    }
}

bool cautare_cod_circuit1 (char nf[], int codc)
{
    ifstream f (nf);
    int v[100];
    int n=0;
    while (f>>v[n])
        n++;
    for (int i=0;i<n;i+=5)
        if (v[i]==codc)
            return 1;
    return 0;
    f.close();
}

bool cautare_cod_baterie (char nf[], int cod)
{
    ifstream f(nf);
    int v[100];
    int n=0;
    while (f>>v[n])
        n++;

```

```

        for (int i=0;i<n;i+=3)
            if (v[i]==cod)
                return 1;
        return 0;
        f.close();
    }
bool cautare_cod_rezistenta (char nf[], int cod)
{
    ifstream f(nf);
    int v[100];
    int n=0;
    while (!f.eof())
    {
        f>>v[n];
        n++;
    }
    for (int i=0;i<n;i+=2)
        if (v[i]==cod)
            return 1;
    return 0;
    f.close();
}
void afisare_manageri(manager a[], int m, int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    char t[100];
    outtextxy(la,lu,"COD ");
    outtextxy(la*2,lu,"USERNAME ");
    outtextxy (la*7,lu,"PAROLA ");
    outtextxy(la*11,lu,"NR CIRCUITE");
    int h;
    for (int i=0;i<m;i++)
    {
        itoa(a[i].codm,t,10);
        outtextxy(la,lu+(i+2)*textheight("COD "),t);
        outtextxy(la*2,lu+(i+2)*textheight("COD "),a[i].username);
        outtextxy (la*7,lu+(i+2)*textheight("COD "),a[i].password);
        itoa(a[i].nr_c,t,10);
        outtextxy(la*11,lu+(i+2)*textheight("COD "),t);
        h=i+2;
    }
    outtextxy(la,lu+(h+2)*textheight("Cod "), "Apasati orice tasta!");
    getch();
}
bool cautare_cod_manager (manager a[], int m, int cod)
{
    bool sem=0;
    for (int i=0;i<m;i++)
        if (a[i].codm==cod)
        {
            sem=1;
            break;
        }
    return sem;
}
bool cautare_username (int m, manager a[], char nume[100])
{
    for (int i=0;i<m;i++)
        if (strcmp(nume,a[i].username)==0)
            return 1;
        else
            return 0;
}

```

```

}
void stergere_manageri (manager a[], int &m, curent c[], int la, int lu)
{
    setcolor(WHITE);
    settxtstyle(1,0,18);
    char t[100];
    manager sterg;
    int u=1, ct=0;
    while ((u)&&(ct<2))
    {
        cleardevice();
        outtextxy(la,lu,"Va rog introduceti codul managerului (max 3
cifre):");
        citire_nr(sterg.codm,100,la,lu+textheight("Va rog introduceti codul
managerului (max 3 cifre):"));
        if (cautare_cod_manager(a,m,sterg.codm))
            u=0;
        else
        {
            outtextxy(la,lu+2*textheight("Va rog introduceti codul
managerului (max 3 cifre):"),"Va rog introduceti un cod existent!");
            outtextxy(la,lu+3*textheight("Va rog introduceti codul
managerului (max 3 cifre):"),"Apasati orice tasta!");
            getch();
            ct++;
        }
    }
    if (ct<2)
    {
        int q;
        for (int i=0;i<m;i++)
            if (sterg.codm==a[i].codm)
            {
                q=i;
                break;
            }
        cleardevice();
        outtextxy(la,lu,"Informatiile userului ");
        itoa(a[q].codm,t,10);
        outtextxy(la+textwidth("Informatiile userului "),lu,t);
        outtextxy(la,lu+textheight("Informatiile userului "), "Username:
");
        outtextxy(la+textwidth("Username: "),lu+textheight("Informatiile
userului "),a[q].username);
        outtextxy(la,lu+2*textheight("Informatiile userului "), "Parola:
");
        outtextxy(la+textwidth("Parola: "),lu+2*textheight("Informatiile
userului "),a[q].password);
        outtextxy(la,lu+3*textheight("Informatiile userului "), "Nr
circuite: ");
        itoa(a[q].nr_c,t,10);
        outtextxy(la+textwidth("Nr circuite:
"),lu+3*textheight("Informatiile userului "),t);
        int x=0;
        outtextxy(la,lu+4*textheight("Informatiile userului "), "Doriti sa
eliminati userul (apasati 1) sau sa anulati operatiunea (apasati 2)?");
        citire_nr(x,10,la,lu+5*textheight("Informatiile userului "));
        int ata=6;
        while ((x!=1)&&(x!=2))
        {
            outtextxy(la,lu+lu+ata*textheight("Informatiile userului
"), "Alegeti o optiune intre 1 si 2!");
            citire_nr(x,10,la,lu+(ata+1)*textheight("Informatiile
userului "));

```



```

        ata++;
    }
    cleardevice();
    if (x==1)
    {
        m--;
        for (int i=q;i<m;i++)
            a[i]=a[i+1];
        outtextxy(la,lu,"S-a sters userul!");
        outtextxy(la,lu+textheight("S-a sters userul!"),"Apasati
orice tasta!");
        getch();
    }
    else
        if (x==2)
        {
            outtextxy(la,lu,"Operatiune anulata!");
            outtextxy(la,lu+textheight("Operatiune
anulata!"),"Apasati orice tasta!");
            getch();
        }
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Va rog vizualizati utilizatorii existenti!");
        outtextxy(la,lu+textheight("Va rog vizualizati utilizatorii
existenti!"),"Apasati orice tasta!");
        getch();
    }
}
void adaugare_manager (manager a[], int &m, int la, int lu)
{
    setcolor(WHITE);
    setttextstyle(1,0,18);
    cleardevice();
    char t[100];
    manager nou;
    bool sem=0;
    int ct=0;
    int ata;
    if (m==40)
        outtextxy(la,lu,"Nu se mai pot adauga utilizatori!");
    else
    {
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu,"Tastati codul utilizatorului: (max 3 cifre) ");
            citire_nr(nou.codm,100,la,lu+textheight("Tastati codul utilizatorului:
(max 3 cifre) "));
            bool e=cautare_cod_manager(a,m,nou.codm);
            if (nou.codm<0||nou.codm>999)
            {
                outtextxy(la,lu+2*textheight("Tastati codul utilizatorului: (max
3 cifre) "),"Cod incorect! ");
                outtextxy(la,lu+3*textheight("Tastati codul utilizatorului: (max
3 cifre) "),"Apasati orice tasta!");
                getch();
                ct++;
            }
        }
        else
            if (e)
            {

```

```

        outtextxy(la,lu+2*textheight("Tastati codul utilizatorului: (max
3 cifre) "), "Acest utilizator exista deja!");
        outtextxy(la,lu+3*textheight("Tastati codul utilizatorului: (max
3 cifre) "), "Apasati orice tasta!");
        getch();
        ct++;
    }
    else
    if (nou.codm>=0&&nou.codm<=999&&!e)
    {
        ct=2;
        sem=1;
    }
}
cleardevice();
if(sem)
{
    ct=0;
    sem=0;
    while (ct<2)
    {
        cleardevice();
        outtextxy(la,lu,"Username: (max 20 de caractere) ");
        citire_text(nou.username,200,la,lu+textheight("Username: (max
20 de caractere) "));
        bool e=cautare_username(m,a,nou.username);
        if (strlen(nou.username)>20||strlen(nou.username)==0)
        {
            outtextxy(la,lu+2*textheight("Username: (max 20 de
caractere) "), "Username incorect!");
            outtextxy(la,lu+3*textheight("Username: (max 20 de
caractere) "), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        if (e)
        {
            outtextxy(la,lu+2*textheight("Username: (max 20 de
caractere) "), "Acest username exista deja!");
            outtextxy(la,lu+3*textheight("Username: (max 20 de
caractere) "), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        if (strlen(nou.username)<=20&&strlen(nou.username)!=0&&!e)
        {
            ct=2;
            sem=1;
        }
    }
    cleardevice();
    if (sem)
    {
        ct=0;
        sem=0;
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu,"Parola: (max 20 de caractere) ");
            citire_text(nou.password,200,la,lu+textheight("Parola:
(max 20 de caractere) "));
            if (strlen(nou.password)>20||strlen(nou.password)==0)

```

```

        {
            outtextxy(la,lu+2*textheight("Parola: (max 20 de
caractere) "), "Parola incorecta!");
            outtextxy(la,lu+3*textheight("Parola: (max 20 de
caractere) "), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        {
            ct=2;
            sem=1;
        }
    }
    cleardevice();
    if (sem)
    {
        ct=0;
        sem=0;
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu, "Nr circuite (max 3): ");
            citire_nr(nou.nr_c,10,la+textwidth("Nr circuite (max
3): "),lu);
            if (nou.nr_c==0||nou.nr_c>3)
            {
                outtextxy(la,lu+textheight("Nr circuite (max 3):
"), "Numar incorect!");
                outtextxy(la,lu+2*textheight("Nr circuite (max 3):
"), "Apasati orice tasta!");
                getch();
                ct++;
            }
            else
            {
                ct=2;
                sem=1;
            }
        }
        cleardevice();
        if (sem)
        {
            for (int i=0;i<nou.nr_c;i++)
            {
                ct=0;
                sem=0;
                while (ct<2)
                {
                    cleardevice();
                    outtextxy(la,lu, "Codul circuitului: (max
3 cifre) ");
                    citire_nr(nou.c[i].codc,10,la+textwidth("Codul circuitului: (max 3 cifre)
"),lu);
                    if (nou.c[i].codc<0||nou.c[i].codc>999)
                    {
                        outtextxy(la,lu+textheight("Codul
circuitului: (max 3 cifre) "), "Cod incorect!");
                        outtextxy(la,lu+2*textheight("Codul
circuitului: (max 3 cifre) "), "Apasati orice tasta!");
                        getch();
                        ct++;
                    }
                }
            }
        }
    }
}

```

```

else
    if
(cautare_cod_circuit1("curent.in",nou.c[i].codc))
    {
        outtextxy(la,lu+textheight("Codul
circuitului: (max 3 cifre) "), "Acest circuit exista!");
        outtextxy(la,lu+2*textheight("Codul
circuitului: (max 3 cifre) "), "Apasati orice tasta!");
        getch();
        ct++;
    }
    else
    {
        ct=2;
        sem=1;
    }
}
cleardevice();
if (sem)
{
    ct=0;
    sem=0;
    while (ct<2)
    {
        cleardevice();
        outtextxy(la,lu,"Numarul de baterii: (max
5)");
        citire_nr(nou.c[i].nr_B,10,la+textwidth("Numarul de baterii: (max 5)"),lu);
        if (nou.c[i].nr_B==0||nou.c[i].nr_B>5)
        {
            outtextxy(la,lu+textheight("Numarul
de baterii: (max 5)"), "Valoare incorecta!");
            outtextxy(la,lu+2*textheight("Numarul
de baterii: (max 5)"), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        {
            ct=2;
            sem=1;
        }
    }
    cleardevice();
    if (sem)
    {
        ct=0; sem=0;
        if (nou.c[i].nr_B==1) nou.c[i].legare_B=1;
        else
        {
            outtextxy(la,lu,"Cum doriti sa fie legate
bateriile: ");
            outtextxy(la,lu+textheight("Cum doriti sa
fie legate bateriile: "), "In serie (apasati 1) sau in paralel (apasati 2)?");
            citire_nr(nou.c[i].legare_B,10,la,lu+2*textheight("Cum doriti sa fie legate
bateriile: "));
            ata=3;
            while
((nou.c[i].legare_B!=1)&&(nou.c[i].legare_B!=2))
            {
                outtextxy(la,lu+ata*textheight("Cum
doriti sa fie legate bateriile: "), "Alegeti o optiune intre 1 si 2!");

```

```

citire_nr(nou.c[i].legare_B,10,la,lu+(ata+1)*textheight("Cum doriti sa fie
legate bateriile: "));

                                ata++;
                                }}
                                cleardevice();
                                while (ct<2)
                                {
                                cleardevice();
                                outtextxy(la,lu,"Numarul de rezistente
exterioare: (max 5)");

citire_nr(nou.c[i].nr_R,10,la+textwidth("Numarul de rezistente exterioare:
(max 5)"),lu);

                                if (nou.c[i].nr_R==0||nou.c[i].nr_R>5)
                                {
                                outtextxy(la,lu+textheight("Numarul
de rezistente exterioare: (max 5)"),"Valoare incorecta!");
                                outtextxy(la,lu+2*textheight("Numarul
de rezistente exterioare: (max 5)"),"Apasati orice tasta!");
                                getch();
                                ct++;
                                }
                                else
                                {
                                ct=2;
                                sem=1;
                                }
                                }
                                cleardevice();
                                if (sem)
                                {
                                ct=0; sem=0;
                                if
                                (nou.c[i].nr_R==1)

nou.c[i].legare_Re=1;

                                else
                                {
                                outtextxy(la,lu,"Cum doriti sa fie legate
rezistentele: ");

                                outtextxy(la,lu+textheight("Cum doriti sa
fie legate rezistentele: "), "In serie (apasati 1) sau in paralel (apasati
2)?");

citire_nr(nou.c[i].legare_Re,10,la,lu+2*textheight("Cum doriti sa fie legate
rezistentele: "));

                                ata=3;
                                while
                                ((nou.c[i].legare_Re!=1)&&(nou.c[i].legare_Re!=2))
                                {
                                outtextxy(la,lu+ata*textheight("Cum
doriti sa fie legate rezistentele: "), "Alegeti o optiune intre 1 si 2!");

citire_nr(nou.c[i].legare_Re,10,la,lu+(ata+1)*textheight("Cum doriti sa fie
legate rezistentele: "));

                                ata++;
                                }}
                                if
                                ((nou.c[i].legare_Re==1)|| (nou.c[i].legare_Re==2))
                                {
                                ct=2;
                                sem=1;
                                }
                                }
                                cleardevice();

```

```

        if (sem)
        {
            for (int x=0;x<nou.c[i].nr_B;x++)
            {
                ct=0;
                sem=0;
                while (ct<2)
                {
                    cleardevice();
                    outtextxy(la,lu,"Codul
bateriei ");

                    itoa(x+1,t,10);
                    outtextxy(la+textwidth("Codul
bateriei "),lu,t);

                    outtextxy(la+textwidth("Codul
bateriei")+textwidth(t),lu," : (max 3 cifre)");

                    citire_nr(nou.c[i].B[x].codB,100,la+textwidth("Codul
bateriei
")+textwidth(t)+textwidth(" : (max 3 cifre)"),lu);
                    if
                    (nou.c[i].B[x].codB<0||nou.c[i].B[x].codB>999)
                    {

                        outtextxy(la,lu+textheight("Codul bateriei "), "Cod incorect!");

                        outtextxy(la,lu+2*textheight("Codul bateriei "), "Apasati orice tasta!");
                        getch();
                        ct++;
                    }
                    else
                    if
                    (cautare_cod_baterie("baterie.in",nou.c[i].B[x].codB))
                    {

                        outtextxy(la,lu+textheight("Codul bateriei "), "Acest cod exista deja!");

                        outtextxy(la,lu+2*textheight("Codul bateriei "), "Apasati orice tasta!");
                        getch();
                        ct++;
                    }
                    else
                    {
                        ct=2;
                        sem=1;
                    }
                }
                cleardevice();
                if (sem)
                {
                    ct=0;
                    sem=0;
                    while (ct<2)
                    {
                        cleardevice();

                        outtextxy(la,lu,"Tensiunea electromotoare a bateriei: (max 24 V) ");

                        citire_nr(nou.c[i].B[x].E,10,la+textwidth("Tensiunea
                        electromotoare
                        a
bateriei: (max 24 V) "),lu);

                        if
                        (nou.c[i].B[x].E==0||nou.c[i].B[x].E>24)
                        {

```

```

outtextxy(la,lu+textheight("Tensiunea electromotoare a bateriei: (max 24 V)
"),"Valoare incorecta!");

outtextxy(la,lu+2*textheight("Tensiunea electromotoare a bateriei: (max 24 V)
"),"Apasati orice tasta!");

                                getch();
                                ct++;
                                }
                                else
                                {
                                    ct=2;
                                    sem=1;
                                }
                                }
                                cleardevice();
                                if (sem)
                                {
                                    sem=0;
                                    ct=0;
                                    while (ct<2)
                                    {
                                        cleardevice();

outtextxy(la,lu,"Rezistenta interioara a bateriei: (max 15 Ohm) ");

citire_nr(nou.c[i].B[x].r,10,la+textwidth("Rezistenta interioara a bateriei:
(max 15 Ohm) "),lu);

                                if (nou.c[i].B[x].r>15)
                                {

outtextxy(la,lu+textheight("Rezistenta interioara a bateriei: (max 15 Ohm)
"),"Valoare incorecta!");

outtextxy(la,lu+2*textheight("Rezistenta interioara a bateriei: (max 15 Ohm)
"),"Apasati orice tasta!");

                                getch();
                                ct++;
                                }
                                else
                                {
                                    ct=2;
                                    sem=1;
                                }
                                }
                                cleardevice();
                                }}
                                }

                                if (sem)
                                {
                                    for (int

x=0;x<nou.c[i].nr_R;x++)

                                {
                                    sem=0;
                                    ct=0;
                                    while (ct<2)
                                    {
                                        cleardevice();

outtextxy(la,lu,"Codul rezistentei: (max 3 cifre) ");

citire_nr(nou.c[i].R[x].codR,100,la+textwidth("Codul rezistentei: (max 3
cifre) "),lu);

```

```

                                                                    if
(nou.c[i].R[x].codR<0||nou.c[i].R[x].codR>999)
                                                                    {
outtextxy(la,lu+textheight("Codul rezistentei: (max 3 cifre) "), "Cod
incorect!");

outtextxy(la,lu+2*textheight("Codul rezistentei: (max 3 cifre) "), "Apasati
orice tasta!");

                                                                    getch();
                                                                    ct++;
                                                                    }
                                                                    else
                                                                    if
(cautare_cod_rezistenta("rezistenta.in",nou.c[i].R[x].codR))
                                                                    {

outtextxy(la,lu+textheight("Codul rezistentei: (max 3 cifre) "), "Acest cod
exista deja!");

outtextxy(la,lu+2*textheight("Codul rezistentei: (max 3 cifre) "), "Apasati
orice tasta!");

                                                                    getch();
                                                                    ct++;
                                                                    }
                                                                    else
                                                                    {
                                                                    ct=2;
                                                                    sem=1;
                                                                    }
                                                                    }
cleardevice();
if (sem)
{
    sem=0;
    ct=0;
    while (ct<2)
    {
        cleardevice();

outtextxy(la,lu,"Valoarea rezistentei (max 20 Ohm, diferita de 0) ");

citire_nr(nou.c[i].R[x].Re,10,la+textwidth("Valoarea rezistentei (max 20 Ohm,
diferita de 0) "),lu);

                                                                    if
(nou.c[i].R[x].Re==0||nou.c[i].R[x].Re>20)
                                                                    {

outtextxy(la,lu+textheight("Valoarea rezistentei (max 20 Ohm, diferita de 0)
"), "Valoare incorecta!");

outtextxy(la,lu+2*textheight("Valoarea rezistentei (max 20 Ohm, diferita de 0)
"), "Apasati orice tasta!");

                                                                    getch();
                                                                    ct++;
                                                                    }
                                                                    else
                                                                    {
                                                                    ct=2;
                                                                    sem=1;
                                                                    }
                                                                    }
cleardevice();
}

```



```

cleardevice();
char t[100];
int h;
int p=determinare_cod_manager(a,m,codu);
for (int k=0;k<a[p].nr_c;k++)
{
    outtextxy(la,lu,"CIRCUITUL ");          itoa(a[p].c[k].codc,t,10);
outtextxy(la+textwidth("CIRCUITUL "),lu,t);
    outtextxy(la,lu+textheight("CIRCUITUL"),"Nr baterii: ");
    itoa(a[p].c[k].nr_B,t,10);          outtextxy(la+textwidth("Nr baterii:
"),lu+textheight("CIRCUITUL"),t);
    int z=textheight("CIRCUITUL");
    int j=textwidth("serie");
    if (a[p].c[k].nr_B!=1)
    {outtextxy(la,lu+2*textheight("CIRCUITUL"),"Legate in ");
    if (a[p].c[k].legare_B==1) outtextxy(la+textwidth("Legate in
"),lu+2*z,"serie");
    else outtextxy(la+textwidth("Legate in "),lu+2*z,"paralel");}
    h=0;
    for (int i=0;i<a[p].c[k].nr_B;i++)
    {
        outtextxy(la+j,lu+(h+3)*z,"Bateria ");
        itoa (a[p].c[k].B[i].codB,t,10);
        outtextxy(la+j+textwidth("Bateria "),lu+(h+3)*z,t);
        outtextxy(la+j+textwidth("Bateria "),lu+(h+4)*z,"E: ");
        itoa (a[p].c[k].B[i].E,t,10);
        outtextxy(la+j+textwidth("Bateria")+textwidth("E: "),lu+(h+4)*z,t);
        outtextxy(la+j+textwidth("Bateria ") +textwidth("E:
")+textwidth(t),lu+(h+4)*z," V");
        outtextxy(la+j+textwidth("Bateria "),lu+(h+5)*z,"r: ");
        itoa(a[p].c[k].B[i].r,t,10);
        outtextxy(la+j+textwidth("Bateria")+textwidth("r: "),lu+(h+5)*z,t);
        outtextxy(la+j+textwidth("Bateria ") +textwidth("r:
")+textwidth(t),lu+(h+5)*z," Ohm");
        h=i+3;
    }
    outtextxy(la+j,lu+(h+7)*z,"Apasati orice tasta pentru a vedea
rezistentele exterioare!");
    getch();
    cleardevice();
    outtextxy(la,lu,"CIRCUITUL ");          itoa(a[p].c[k].codc,t,10);
outtextxy(la+textwidth("CIRCUITUL "),lu,t);
    outtextxy(la,lu+textheight("CIRCUITUL"),"Nr rezistente: ");
    itoa(a[p].c[k].nr_R,t,10);
    outtextxy(la+textwidth("Nr rezistente:
"),lu+textheight("CIRCUITUL"),t);
    if (a[p].c[k].nr_R!=1)
    {outtextxy(la,lu+2*textheight("CIRCUITUL"),"Legate in ");
    if (a[p].c[k].legare_Re==1) outtextxy(la+textwidth("Legate in
"),lu+2*z,"serie");
    else outtextxy(la+textwidth("Legate in "),lu+2*z,"paralel");}
    h=0;
    for (int i=0;i<a[p].c[k].nr_R;i++)
    {
        outtextxy(la+j,lu+(h+3)*z,"Rezistenta ");
        itoa (a[p].c[k].R[i].codR,t,10);
        outtextxy(la+j+textwidth("Rezistenta "),lu+(h+3)*z,t);
        outtextxy(la+j+textwidth("Rezistenta "),lu+(h+4)*z,"R: ");
        itoa (a[p].c[k].R[i].Re,t,10);
        outtextxy(la+j+textwidth("Rezistenta")+textwidth("R:
"),lu+(h+4)*z,t);
        outtextxy(la+j+textwidth("Rezistenta ") +textwidth("R:
")+textwidth(t),lu+(h+4)*z," Ohm");
        h=i+2;
    }
}

```

```

    }
    outtextxy(la,lu+(h+6)*z,"Apasati orice tasta ");
    if (k==a[p].nr_c-1)
    {
        outtextxy(la+textwidth("Apasati orice tasta "),lu+(h+6)*z,"pentru
a iesi!");
        getch();
    }
    else
    {
        outtextxy(la+textwidth("Apasati orice tasta "),lu+(h+6)*z,"pentru
urmatorul circuit!");
        getch();
        cleardevice();
    }
}
}
bool cautare_cod_circuit (int m, manager a[], int codu, int x, curent c[])
{
    for (int i=0;i<m;i++)
        if (a[i].codm==codu)
        {
            for (int j=0;j<a[i].nr_c;j++)
                if (x==a[i].c[j].codc)
                    return 1;
        }
    return 0;
}
int determinare_cod_manager (manager a[], int m, int codu)
{
    for (int i=0;i<m;i++)
        if (a[i].codm==codu)
            return i;
    return -1;
}
int determinare_cod_circuit (manager a[], int m, int codu, int codci)
{
    for (int i=0;i<m;i++)
    {
        if (a[i].codm==codu)
            for (int j=0;j<a[i].nr_c;j++)
                if (a[i].c[j].codc==codci)
                    return j;
    }
    return -1;
}
int determinare_cod_baterie (manager a[], int m, int codu, int codci, int
codb)
{
    for (int i=0;i<m;i++)
    {
        if (a[i].codm==codu)
            for (int j=0;j<a[i].nr_c;j++)
            {
                if (a[i].c[j].codc==codci)
                    for (int z=0;z<a[i].c[j].nr_B;z++)
                        if (a[i].c[j].B[z].codB==codb)
                            return z;
            }
    }
    return -1;
}
int determinare_cod_rezistenta (manager a[], int m, int codu, int codci, int
codr)

```

```

{
    for (int i=0;i<m;i++)
    {
        if (a[i].codm==codu)
        for (int j=0;j<a[i].nr_c;j++)
        {
            if (a[i].c[j].codc==codci)
            for (int z=0;z<a[i].c[j].nr_R;z++)
            if (a[i].c[j].R[z].codR==codr)
            return z;
        }
    }
    return -1;
}

void modificare_baterie (int codu, int codc, baterie B[], manager a[], int m,
curent c[], int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    char t[100];
    baterie b;
    int codb;
    outtextxy(la,lu,"Introduceti codul bateriei: ");
    citire_nr(codb,100,la+textwidth("Introduceti codul bateriei: "),lu);
    b.codB=codb;
    int ct; bool sem;
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    int l=determinare_cod_baterie(a,m,codu,codc,codb);
    if (l!=-1)
    {
        outtextxy(la,lu+textheight("Introduceti codul bateriei: "), "Tensiunea
electromotoare: ");
        itoa(a[o].c[p].B[l].E,t,10);
        outtextxy(la+textwidth("Tensiunea electromotoare:
"),lu+textheight("Introduceti codul bateriei: "),t);
        outtextxy(la+textwidth("Tensiunea electromotoare:
")+textwidth(t),lu+textheight("Introduceti codul bateriei: "), " V");
        outtextxy(la,lu+2*textheight("Introduceti codul bateriei: "), "Rezistenta
interioara: ");
        itoa(a[o].c[p].B[l].r,t,10);
        outtextxy(la+textwidth("Rezistenta interioara:
"),lu+2*textheight("Introduceti codul bateriei: "),t);
        outtextxy(la+textwidth("Rezistenta interioara:
")+textwidth(t),lu+2*textheight("Introduceti codul bateriei: "), " Ohm");
        ct=0;
        sem=0;
        outtextxy(la,lu+3*textheight("Introduceti codul bateriei: "), "Apasati
orice tasta!");
        getch();
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu,"Noua tensiune electromotoare a bateriei: (max 24 V)
");
            citire_nr(b.E,100,la+textwidth("Noua tensiune electromotoare a
bateriei: (max 24 V) "),lu);
            if (b.E==0||b.E>24)
            {
                outtextxy(la,lu+textheight("Noua tensiune electromotoare a
bateriei: (max 24 V) "), "Valoare incorecta!");
                outtextxy(la,lu+2*textheight("Noua tensiune electromotoare a
bateriei: (max 24 V) "), "Apasati orice tasta!");
            }
        }
    }
}

```

```

        getch();
        ct++;
    }
    else
    {
        ct=2;
        sem=1;
    }
}
cleardevice();
if (sem)
{
    sem=0;
    ct=0;
    while (ct<2)
    {
        cleardevice();
        outtextxy(la,lu,"Noua rezistenta interioara a bateriei: (max 15
Ohm) ");
        citire_nr(b.r,100,la+textwidth("Noua rezistenta interioara a
bateriei: (max 15 Ohm) "),lu);
        if (b.r>15)
        {
            outtextxy(la,lu+textheight("Noua rezistenta interioara a
bateriei: (max 15 Ohm) "), "Valoare incorecta!");
            outtextxy(la,lu+2*textheight("Noua rezistenta interioara a
bateriei: (max 15 Ohm) "), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        {
            ct=2;
            sem=1;
        }
    }
}
cleardevice();
if (sem)
{
    int x;
    outtextxy(la,lu,"Tensiunea electromotoare noua: ");
    itoa(b.E,t,10);
    outtextxy(la+textwidth("Tensiunea electromotoare noua: "),lu,t);
    outtextxy(la+textwidth("Tensiunea electromotoare noua:
")+textwidth(t),lu, " V");
    outtextxy(la,lu+textheight("Tensiunea electromotoare noua:
"), "Rezistenta interioara: ");
    itoa(b.r,t,10);
    outtextxy(la+textwidth("Rezistenta interioara:
"),lu+textheight("Tensiunea electromotoare noua: "),t);
    outtextxy(la+textwidth("Rezistenta interioara:
")+textwidth(t),lu+textheight("Tensiunea electromotoare noua: "), " Ohm");
    outtextxy(la,lu+2*textheight("Tensiunea electromotoare noua:
"), "Doriti sa salvati modificarile (apasati 1) sau nu (apasati 2)?");
    citire_nr(x,10,la,lu+3*textheight("Tensiunea electromotoare noua:
"));
    int ata=4;
    while (x!=1&& x!=2)
    {
        outtextxy(la,lu+ata*textheight("Tensiunea electromotoare noua:
"), "Alegeti o optiune intre 1 si 2!");
        citire_nr(x,10,la,lu+(ata+1)*textheight("Tensiunea
electromotoare noua: "));
        ata++;
    }
}

```

```

    }
    cleardevice();
    if (x==1)
    {
        a[o].c[p].B[l]=b;
        outtextxy(la,lu,"Operatiune finalizata cu succes!");
        outtextxy(la,lu+textheight("Operatiune finalizata cu
succes!"),"Apasati orice tasta!");
        getch();
    }
    else
    {
        if (x==2)
        {
            outtextxy(la,lu,"Operatiune anulata!");
            outtextxy(la,lu+textheight("Operatiune anulata!"),"Apasati orice
tasta!");
            getch();
        }
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Operatiune esuata!");
        outtextxy(la,lu+textheight("Operatiune esuata!"),"Apasati orice
tasta!");
        getch();
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Acesta baterie nu exista!");
        outtextxy(la,lu+textheight("Aceasta baterie nu exista!"),"Apasati
orice tasta!");
        getch();
    }
}

void modificare_rezistenta (int codu, int codc, rezistenta_exterioara R[],
manager a[], int m, curent c[], int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    char t[100];
    rezistenta_exterioara r;
    int codr;
    outtextxy(la,lu,"Introduceti codul rezistentei: ");
    citire_nr(codr,100,la+textwidth("Introduceti codul rezistentei: "),lu);
    r.codR=codr;
    int ct; bool sem;
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    int l=determinare_cod_rezistenta(a,m,codu,codc,codr);
    if (l!=-1)
    {
        outtextxy(la,lu+textheight("Introduceti codul rezistentei: "),
"Valoarea rezistentei: ");
        itoa(a[o].c[p].R[l].Re,t,10);
        outtextxy(la+textwidth("Valoarea rezistentei:
"),lu+textheight("Introduceti codul rezistentei: "),t);
        outtextxy(la+textwidth("Valoarea rezistentei:
")+textwidth(t),lu+textheight("Introduceti codul rezistentei: "),
" Ohm");
        ct=0;
    }
}

```

```

sem=0;
outtextxy(la,lu+2*textheight("Introduceti codul rezistentei: "), "Apasati
orice tasta!");
getch();
while (ct<2)
{
    cleardevice();
    outtextxy(la,lu,"Noua valoare a rezistentei (max 20 Ohm, diferita de
0) ");
    citire_nr(r.Re,100,la,lu+textheight("Noua valoare a rezistentei (max
20 Ohm, diferita de 0) "));
    if (r.Re==0||r.Re>20)
    {
        outtextxy(la,lu+2*textheight("Noua valoare a rezistentei (max 20
Ohm, diferita de 0) "), "Valoare incorecta!");
        outtextxy(la,lu+3*textheight("Noua valoare a rezistentei (max 20
Ohm, diferita de 0) "), "Apasati orice tasta!");
        getch();
        ct++;
    }
    else
    {
        ct=2;
        sem=1;
    }
}
cleardevice();
if (sem)
{
    int x;
    outtextxy(la,lu,"Noua rezistenta: ");
    itoa(r.Re,t,10);
    outtextxy(la+textwidth("Noua rezistenta: "),lu,t);
    outtextxy(la+textwidth("Noua rezistenta:")+textwidth(t),lu," Ohm");
    outtextxy(la,lu+textheight("Noua rezistenta: "), "Doriti sa salvati
modificarile (apasati 1) sau nu (apasati 2)?");
    citire_nr(x,10,la,lu+2*textheight("Noua rezistenta: "));
    int ata=3;
    while (x!=1&&x!=2)
    {
        outtextxy(la,lu+ata*textheight("Noua rezistenta: "), "Alegeti o
optiune intre 1 si 2!");
        citire_nr(x,10,la,lu+(ata+1)*textheight("Noua rezistenta: "));
        ata++;
    }
    cleardevice();
    if (x==1)
    {
        a[o].c[p].R[1]=r;
        outtextxy(la,lu,"Operatiune finalizata cu succes!");
        outtextxy(la,lu+textheight("Operatiune finalizata cu
succes!"), "Apasati orice tasta!");
        getch();
    }
    else
    {
        if (x==2)
        {
            outtextxy(la,lu,"Operatiune anulata!");
            outtextxy(la,lu+textheight("Operatiune anulata!"), "Apasati orice
tasta!");
            getch();
        }
    }
}
else

```

```

        {
            cleardevice();
            outtextxy(la,lu,"Operatiune esuata!");
            outtextxy(la,lu+textheight("Operatiune esuata!"),"Apasati orice
tasta!");
            getch();
        }
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Aceasta rezistenta nu exista!");
        outtextxy(la,lu+textheight("Aceasta rezistenta nu
exista!"),"Apasati orice tasta!");
        getch();
    }
}

void meniu_modificari_circuit(manager a[], baterie B[],
rezistenta_exterioara R[], int m, int codu, curent c[],int la, int lu);
void modificare_manageri_user_parola(manager a[], int m, baterie B[],
rezistenta_exterioara R[], curent c[], int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    char s[100];
    manager t;
    outtextxy(la,lu,"Introduceti codul utilizatorului: ");
    citire_nr(t.codm,100,la,lu+textheight("Introduceti codul utilizatorului:
"));
    if (cautare_cod_manager(a,m,t.codm))
    {
        int x, ct=0;
        bool sem=0;
        outtextxy(la,lu+2*textheight("Introduceti codul utilizatorului:
"),"Doriti sa modificati doar username si parola (apasati 1) sau sa
modificati circuitele (apasati 2)?");
        citire_nr(x,10,la,lu+3*textheight("Introduceti codul utilizatorului:
"));
        int ata=4;
        while (x!=1&&x!=2)
        {
            outtextxy(la,lu+ata*textheight("Introduceti codul utilizatorului:
"),"Alegeti o optiune intre 1 si 2!");
            citire_nr(x,10,la,lu+(ata+1)*textheight("Introduceti codul
utilizatorului: "));
            ata++;
        }
        cleardevice();
        if (x==1)
        {
            while (ct<2)
            {
                cleardevice();
                outtextxy(la,lu,"Introduceti noul username (max 20 caractere):
");
                citire_text(t.username,200,la,lu+textheight("Introduceti noul
username (max 20 caractere): "));
                if (strlen(t.username)==0||strlen(t.username)>20)
                {
                    outtextxy(la,lu+2*textheight("Introduceti noul username
(max 20 caractere):"),"Username incorect!");
                    outtextxy(la,lu+3*textheight("Introduceti noul username
(max 20 caractere):"),"Apasati orice tasta!");
                }
            }
        }
    }
}

```



```

        getch();
        ct++;
    }
    else
        if (cautare_username(m,a,t.username))
        {
            outtextxy(la,lu+2*textheight("Introduceti noua
username (max 20 caractere): "), "Acest username exista deja!");
            outtextxy(la,lu+3*textheight("Introduceti noua
username (max 20 caractere): "), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        {
            ct=2;
            sem=1;
        }
    }
    cleardevice();
    if (sem)
    {
        ct=0;
        sem=0;
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu, "Introduceti noua parola: (max 20 de
caractere) ");
            citire_text(t.password,200,la,lu+textheight("Introduceti
noua parola: (max 20 de caractere) "));
            if (strlen(t.password)==0||strlen(t.password)>20)
            {
                outtextxy(la,lu+2*textheight("Introduceti noua parola:
(max 20 de caractere) "), "Parola incorecta!");
                outtextxy(la,lu+3*textheight("Introduceti noua parola:
(max 20 de caractere) "), "Apasati orice tasta!");
                getch();
                ct++;
            }
            else
            {
                ct=2;
                sem=1;
            }
        }
        cleardevice();
        if (sem)
        {
            outtextxy(la,lu, "Utilizatorul ");
            itoa(t.codm,s,10);
            outtextxy(la+textwidth("Utilizatorul "),lu,s);
            outtextxy(la,lu+textheight("Utilizatorul "), "Username: ");
            outtextxy(la+textwidth("Username:
"),lu+textheight("Utilizatorul "),t.username);
            outtextxy(la,lu+2*textheight("Utilizatorul "), "Parola: ");
            outtextxy(la+textwidth("Parola:
"),lu+2*textheight("Utilizatorul "),t.password);
            outtextxy(la,lu+3*textheight("Utilizatorul "), "Doriti sa
salvati modificarile (apasati 1) sau nu (apasati 2)?");
            int y;
            citire_nr(y,10,la,lu+4*textheight("Utilizatorul "));
            int atal=5;
            while (y!=1&&y!=2)

```

```

        {
            outtextxy(la,lu+atal*textheight("Utilizatorul
"),"Alegeti o optiune intre 1 si 2!");
citire_nr(y,10,la,lu+(atal+1)*textheight("Utilizatorul "));
            atal++;
        }
        cleardevice();
        if (y==1)
        {
            int p=determinare_cod_manager(a,m,t.codm);
            strcpy(a[p].username,t.username);
            strcpy(a[p].password,t.password);
            outtextxy(la,lu,"Modificari salvate!");
            outtextxy(la,lu+textheight("Modificari
salvate!"),"Apasati orice tasta!");
            getch();
        }
        else
            if (y==2)
            {
                outtextxy(la,lu,"Actiune anulata!");
                outtextxy(la,lu+textheight("Actiune
anulata!"),"Apasati orice tasta!");
                getch();
            }
        }
    }
    else
        if (x==2)
            meniu_modificari_circuit(a,B,R,m,t.codm,c,la,lu);
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Acest utilizator nu exista!");
        outtextxy(la,lu+textheight("Acest utilizator nu exista!"),"Apasati
orice tasta!");
        getch();
    }
}
void stergere_rezistenta (int codu, int codc, rezistenta_exterioara R[],
manager a[], int m, curent c[], int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    char t[100];
    rezistenta_exterioara r; int x;
    int codr;
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    outtextxy(la,lu,"Introduceti codul rezistentei: ");
    citire_nr(codr,100,la+textwidth("Introduceti codul rezistentei: "),lu);
    r.codR=codr;
    int ct; bool sem;
    int l=determinare_cod_rezistenta(a,m,codu,codc,codr);
    cleardevice();
    if (l!=-1)
    {
        outtextxy(la,lu,"Valoarea rezistentei: ");
        itoa(a[o].c[p].R[l].Re,t,10);
        outtextxy(la+textwidth("Valoarea rezistentei: "),lu,t);
    }
}

```

```

        outtextxy(la+textwidth("Valoarea rezistentei: ") + textwidth(t), lu, "
Ohm");
        outtextxy(la, lu+textheight("Valoarea rezistentei: "), "Doriti sa
stergeti rezistenta (apasati 1) sau nu (apasati 2)?");
        citire_nr(x, 10, la, lu+2*textheight("Valoarea rezistentei: "));
        int ata=3;
        while (x!=1&&x!=2)
        {
            outtextxy(la, lu+ata*textheight("Valoarea rezistentei: "), "Alegeti o
optiune intre 1 si 2!");
            citire_nr(x, 10, la, lu+(ata+1)*textheight("Valoarea rezistentei: "));
            ata++;
        }
        cleardevice();
        if (x==1)
        {
            a[o].c[p].nr_R--;
            for (int i=p; i<c[p].nr_R; i++)
                a[o].c[p].R[i]=a[o].c[p].R[i+1];
            outtextxy(la, lu, "Operatiune finalizata cu succes!");
            outtextxy(la, lu+textheight("Operatiune finalizata cu
succes!"), "Apasati orice tasta!");
            getch();
        }
        else
            if (x==2)
            {
                outtextxy(la, lu, "Operatiune anulata!");
                outtextxy(la, lu+textheight("Operatiune anulata!"), "Apasati orice
tasta!");
                getch();
            }
            else
            {
                cleardevice();
                outtextxy(la, lu, "Aceasta rezistenta nu exista!");
                outtextxy(la, lu+textheight("Aceasta rezistenta nu
exista!"), "Apasati orice tasta!");
                getch();
            }
    }
}

void stergere_circuit (int m, manager a[], curent c[], int codc, int codu,
int la, int lu)
{
    setcolor(WHITE);
    setttextstyle(1, 0, 18);
    cleardevice();
    char t[100];
    outtextxy(la, lu, "CIRCUITUL: ");
    outtextxy(la, lu+textheight("CIRCUITUL: "), "Nr baterii: ");
    itoa(a[codu].c[codc].nr_B, t, 10);
    outtextxy(la+textwidth("Nr baterii: "), lu+textheight("CIRCUITUL: "), t);
    if (a[codu].c[codc].nr_B!=1)
    {
        outtextxy(la+textwidth("Nr baterii: "), lu+textheight("CIRCUITUL: "), "legate in ");
        if (a[codu].c[codc].legare_B==1)
            outtextxy(la+textwidth("Nr baterii: ") + textwidth(t) + textwidth("
legate in "), lu+textheight("CIRCUITUL: "), "serie");
        else
            outtextxy(la+textwidth("Nr baterii: ") + textwidth(t) + textwidth("
legate in "), lu+textheight("CIRCUITUL: "), "paralel");
    }
    outtextxy(la, lu+2*textheight("CIRCUITUL: "), "Nr rezistente: ");
    itoa(a[codu].c[codc].nr_R, t, 10);
}

```

```

        outtextxy(la+textwidth("Nr rezistente: "),lu+2*textheight("CIRCUITUL:
"),t);
        if (a[codu].c[codc].nr_R!=1)
        {outtextxy(la+textwidth("Nr rezistente: ")+" legate in ");
        if (a[codu].c[codc].legare_Re==1)
            outtextxy(la+textwidth("Nr rezistente: ") +textwidth(t)+textwidth("
legate in "),lu+2*textheight("CIRCUITUL: "), "serie");
        else
            outtextxy(la+textwidth("Nr rezistente: ") +textwidth(t)+textwidth("
legate in "),lu+2*textheight("CIRCUITUL: "), "paralel");}
        int x;
        outtextxy(la,lu+3*textheight("CIRCUITUL: "), "Doriti sa continuati
stergera? (apasati 1) sau nu? (apasati 0)");
        citire_nr(x,10,la,lu+4*textheight("CIRCUITUL: "));
        int ata=5;
        while (x!=1&&x!=0)
        {
            outtextxy(la,lu+ata*textheight("CIRCUITUL: "), "Alegeti o optiune
intre 0 si 1!");
            citire_nr(x,10,la,lu+(ata+1)*textheight("CIRCUITUL: "));
            ata++;
        }
        cleardevice();
        if(x==1)
        {
            a[codu].nr_c--;
            for (int i=codc;i<a[codu].nr_c;i++)
                a[codu].c[i]=a[codu].c[i+1];
            outtextxy(la,lu,"Stergere realizata cu succes!");
            outtextxy(la,lu+textheight("Stergere realizata cu succes!"),"Apasati
orice tasta!");
            getch();
        }
        else
        {
            outtextxy(la,lu,"Operatiune anulata!");
            outtextxy(la,lu+textheight("Operatiune anulata!"),"Apasati orice
tasta!");
            getch();
        }
    }
}

void stergere_baterie (int codu, int codc, baterie B[], manager a[], int m,
curent c[], int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    baterie b;
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    int codb;
    outtextxy(la,lu,"Introduceti codul bateriei: ");
    citire_nr(codb,100,la+textwidth("Introduceti codul bateriei: "),lu); int
x;
    b.codB=codb;
    int ct; bool sem;
    int l=determinare_cod_baterie(a,m,codu,codc,codb);
    if (l!=-1)
    {
        outtextxy(la,lu+textheight("Introduceti codul bateriei: "), "Tensiunea
electromotoare: ");
        char t[100];
        itoa(a[o].c[p].B[l].E,t,10);
    }
}

```



```

int ct=0;
bool sem=0;
if (p!=-1)
{
    if (a[o].c[p].nr_B==5)
    {
        outtextxy(la,lu,"Nu se mai pot adauga baterii!");
        outtextxy(la,lu+textheight("Nu se mai pot adauga
baterii!"),"Apasati orice tasta!");
        getch();
    }
    else
    {
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu,"Codul noii bateriii: (max 3 cifre)");
            citire_nr(nou.codB,100,la+textwidth("Codul noii baterii: (max
3 cifre)"),lu);
            if (nou.codB<0||nou.codB>999)
            {
                outtextxy(la,lu+textheight("Codul noii baterii: (max 3
cifre)"),"Cod incorect!");
                outtextxy(la,lu+2*textheight("Cod incorect!"),"Apasati
orice tasta!");
                getch();
                ct++;
            }
            else
            if (cautare_cod_baterie("baterie.in",nou.codB))
            {
                outtextxy(la,lu+textheight("Codul noii baterii: (max
3 cifre)"),"Acest cod exista deja!");
                outtextxy(la,lu+textheight("Acest cod exista
deja!"),"Apasati orice tasta!");
                getch();
                ct++;
            }
            else
            {
                ct=2;
                sem=1;
            }
        }
        cleardevice();
        if (sem)
        {
            ct=0;
            sem=0;
            while (ct<2)
            {
                outtextxy(la,lu,"Tensiunea electromotoare a bateriei: (max 24 V) ");
                citire_nr(nou.E,100,la,lu+textheight("Tensiunea electromotoare a
bateriei: (max 24 V) "));
                if (nou.E==0||nou.E>24)
                {
                    outtextxy(la,lu+2*textheight("Tensiunea electromotoare a
bateriei: (max 24 V)"),"Valoare incorecta!");
                    outtextxy(la,lu+3*textheight("Valoare incorecta!"),"Apasati
orice tasta!");
                    getch();
                    ct++;
                }
            }
            else

```

```

        {
            ct=2;
            sem=1;
        }
    }
cleardevice();
if (sem)
{
    sem=0;
    ct=0;
    while (ct<2)
    {
        cleardevice();
        outtextxy(la,lu,"Rezistenta interioara a bateriei: (max 15
Ohm) ");
        citire_nr(nou.r,100,la,lu+textheight("Rezistenta interioara a
bateriei: (max 15 Ohm) "));
        if (nou.r>15)
        {
            outtextxy(la,lu+3*textheight("Rezistenta interioara a
bateriei: (max 15 Ohm) "), "Valoare incorecta!");
            outtextxy(la,lu+4*textheight("Valoare incorecta!"), "Apasati
orice tasta!");
            getch();
            ct++;
        }
        else
        {
            ct=2;
            sem=1;
        }
    }
cleardevice();
if (sem)
{
    outtextxy(la,lu,"Noua baterie: "); char t[100];
    outtextxy(la,lu+textheight("Noua baterie: "), "Bateria ");
    itoa(nou.codB,t,10);
    outtextxy(la+textwidth("Bateria "),lu+textheight("Noua baterie:
"),t);
    outtextxy(la+textwidth("Bateria "),lu+2*textheight("Noua baterie:
"), "E: ");
    itoa(nou.E,t,10);
    outtextxy(la+textwidth("Bateria ") + textwidth("E:
"),lu+2*textheight("Noua baterie: "),t);
    outtextxy(la+textwidth("Bateria ") + textwidth("E:
") + textwidth(t),lu+2*textheight("Noua baterie: "), " V");
    outtextxy(la+textwidth("Bateria "),lu+3*textheight("Noua baterie:
"), "r: ");
    itoa(nou.r,t,10);
    outtextxy(la+textwidth("Bateria ") + textwidth("r:
"),lu+3*textheight("Noua baterie: "),t);
    outtextxy(la+textwidth("Bateria ") + textwidth("r:
") + textwidth(t),lu+3*textheight("Noua baterie: "), " Ohm");
    int y;
    outtextxy(la,lu+4*textheight("Noua baterie: "), "Doriti sa salvati
modificarile (apasati 1) sau nu (apasati 2)?");
    citire_nr(y,10,la,lu+5*textheight("Noua baterie: "));
    int ata=6;
    while ((y!=1)&&(y!=2))
    {
        outtextxy(la,lu+ata*textheight("Noua baterie: "), "Alegeti o
optiune intre 1 si 2!");
        citire_nr(y,10,la,lu+(ata+1)*textheight("Noua baterie: "));
    }
}
}

```

```

        ata++;
    }
    cleardevice();
    if (y==1)
    {
        a[o].c[p].nr_B++;
        a[o].c[p].B[a[o].c[p].nr_B-1]=nou;
        outtextxy(la,lu,"Operatiune finalizata cu succes!");
        outtextxy(la,lu+textheight("Operatiune          finalizata          cu
succes!"),"Apasati orice tasta!");
        getch();
    }
    else
        if (y==2)
        {
            outtextxy(la,lu,"Operatiune anulata!");
            outtextxy(la,lu+textheight("Operatiune
anulata!"),"Apasati orice tasta!");
            getch();
        }
    }
    }
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Operatiune esuata!");
        outtextxy(la,lu+textheight("Operatiune          esuata!"),"Apasati
orice tasta!");
        getch();
    }
}

void adaugare_rezistenta (manager a[], int m, curent c[],
rezistenta_exterioara R[], int codu, int la, int lu, int o)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    int codc;
    rezistenta_exterioara nou;
    outtextxy(la,lu,"Codul circuitului unde vreti sa adaugati rezistenta: ");
    citire_nr(codc,100,la,lu+textheight("Codul circuitului unde vreti sa
adaugati rezistenta: "));
    int p=determinare_cod_circuit(a,m,codu,codc);
    int ct=0;
    bool sem=0;
    if (p!=-1)
    {
        if (a[o].c[p].nr_R==5)
        {
            outtextxy(la,lu,"Nu se mai pot adauga rezistente!");
            outtextxy(la,lu+textheight("Nu          se          mai          pot          adauga
rezistente!"),"Apasati orice tasta!");
            getch();
        }
        else
        {
            while (ct<2)
            {
                cleardevice();
                outtextxy(la,lu,"Codul rezistentei: (max 3 cifre) ");
                citire_nr(nou.codR,100,la+textwidth("Codul rezistentei: (max 3
cifre) "),lu);

```



```

        if (nou.codR<0||nou.codR>999)
        {
            outtextxy(la,lu+textheight("Codul rezistentei: (max 3 cifre)
"), "Cod incorect!");
            outtextxy(la,lu+2*textheight("Codul rezistentei: (max 3 cifre)
"), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        {
            if (cautare_cod_rezistenta("rezistenta.in",nou.codR))
            {
                outtextxy(la,lu,"Acest cod exista deja!");
                outtextxy(la,lu+textheight("Acest cod exista
deja!"), "Apasati orice tasta!");
                ct++;
            }
            else
            {
                ct=2;
                sem=1;
            }
        }
        cleardevice();
        if (sem)
        {
            sem=0;
            ct=0;
            while (ct<2)
            {
                cleardevice();
                outtextxy(la,lu,"Valoarea rezistentei (max 20 Ohm, diferita
de 0) ");
                citire_nr(nou.Re,100,la,lu+textheight("Valoarea rezistentei
(max 20 Ohm, diferita de 0) "));
                if (nou.Re==0||nou.Re>20)
                {
                    outtextxy(la,lu+2*textheight("Valoarea rezistentei (max
20 Ohm, diferita de 0) "), "Valoare incorecta!");
                    outtextxy(la,lu+3*textheight("Valoarea rezistentei (max
20 Ohm, diferita de 0) "), "Apasati orice tasta!");
                    ct++;
                }
                else
                {
                    ct=2;
                    sem=1;
                }
            }
        }
        if (sem)
        {
            cleardevice();
            char t[100];
            outtextxy(la,lu,"Noua rezistenta: ");
            outtextxy(la,lu+textheight("Noua rezistenta: "), "Rezistenta ");
            itoa (nou.codR,t,10);
            outtextxy(la+textwidth("Rezistenta "),lu+textheight("Noua
rezistenta: "),t);
            itoa(nou.Re,t,10);
            outtextxy(la+textwidth("Rezistenta "),lu+2*textheight("Noua
rezistenta: "), "R: ");
            outtextxy(la+textwidth("Rezistenta "),lu+2*textheight("Noua
rezistenta: "),t);
        }
    }
}

```

```

        outtextxy(la+textwidth("Rezistentă
")+textwidth(t),lu+2*textheight("Nouă rezistentă: "), " Ohm");
        int y;
        outtextxy(la,lu+3*textheight("Nouă rezistentă: "), "Doriti să
salvati modificarile (apasati 1) sau nu (apasati 2)?");
        citire_nr(y,10,la,lu+4*textheight("Nouă rezistentă: "));
        int ata=5;
        while ((y!=1)&&(y!=2))
        {
            outtextxy(la,lu+ata*textheight("Nouă rezistentă: "), "Alegeti
o optiune între 1 și 2!");
            citire_nr(y,10,la,lu+(ata+1)*textheight("Nouă rezistentă: "));
        }
        cleardevice();
        if (y==1)
        {
            a[o].c[p].nr_R++;
            a[o].c[p].R[a[o].c[p].nr_R-1]=nou;
            outtextxy(la,lu, "Operatiune finalizată cu succes!");
            outtextxy(la,lu+textheight("Operatiune finalizată cu
succes!"), "Apasati orice tasta!");
            getch();
        }
        else
            if (y==2)
            {
                outtextxy(la,lu, "Operatiune anulată!");
                outtextxy(la,lu+textheight("Operatiune
anulată!"), "Apasati orice tasta!");
                getch();
            }
        }}}
    }
    else
    {
        cleardevice();
        outtextxy(la,lu, "Operatiune esuată!");
        outtextxy(la,lu+textheight("Operatiune esuată!"), "Apasati
orice tasta!");
        getch();
    }
}

void modificare_parola_admin (manager a[], int m, char nf[], int la, int lu)
{
    char s[21], q[21], t[21];
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    outtextxy(la,lu, "Introduceti parola actuală: ");
    citire_text(s,20,la,lu+textheight("Introduceti parola actuală: "));
    citire_parola_admin(nf,q);
    cleardevice();
    if (strcmp(q,s)==0)
    {
        int ct=0;
        bool sem=0;
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu, "Introduceti nouă parolă (max 20 de caractere):
");
            citire_text(t,20,la,lu+textheight("Introduceti nouă parolă (max
20 de caractere): "));

```

```

        if (strlen(t)>20||strlen(t)==0)
        {
            outtextxy(la,lu+textheight("Introduceti noua parola (max 20
de caractere): "), "Parola invalabila!");
            outtextxy(la,lu+2*textheight("Introduceti noua parola (max 20
de caractere): "), "Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        {
            ct=2;
            sem=1;
        }
    }
    cleardevice();
    if (sem)
    {
        ofstream f(nf);
        f<<t;
        f.close();
        outtextxy(la,lu,"Parola modificata!");
        outtextxy(la,lu+textheight("Parola    modificata!"),"Apasati orice
tasta!");
        getch();
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Parola incorecta!");
        outtextxy(la,lu+textheight("Parola    incorecta!"),"Apasati orice
tasta!");
        getch();
    }
}

void modificare_parola (manager a[], int m, int codu, int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    char s[21], q[100];
    outtextxy(la,lu,"Introduceti parola actuala: ");
    citire_text(s,100,la,lu+textheight("Introduceti parola actuala: "));
    int o=determinare_cod_manager(a,m,codu);
    if (strcmp(s,a[o].password)==0)
    {
        int ct=0;
        bool sem=0;
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu,"Introduceti noua parola (max 20 de caractere):
");
            citire_text(q,101,la,lu+textheight("Introduceti noua parola (max
20 de caractere): "));
            if (strlen(q)>20||strlen(q)==0)
            {
                outtextxy(la,lu+2*textheight("Introduceti noua parola (max 20
de caractere): "), "Parola invalabila!");
                outtextxy(la,lu+3*textheight("Introduceti noua parola (max 20
de caractere): "), "Apasati orice tasta!");
                getch();
                ct++;
            }
        }
    }
}

```

```

        }
        else
        {
            ct=2;
            sem=1;
        }
    }
    cleardevice();
    if (sem)
    {
        strcpy(a[o].password,q);
        outtextxy(la,lu,"Parola modificata!");
        outtextxy(la,lu+textheight("Parola  modificata!"),"Apasati  orice
tasta!");
        getch();
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Parola incorecta!");
        outtextxy(la,lu+textheight("Parola  incorecta!"),"Apasati  orice
tasta!");
        getch();
    }
}

void modificare_legare_rezistente (manager a[], curent c[], int codu, int m,
int codc, int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    int ct=0;
    bool sem=0;
    if (p!=-1)
    {
        if (a[o].c[p].legare_Re==1)
        {
            outtextxy(la,lu,"Exista o singura rezistenta!");
            outtextxy(la,lu+textheight("Exista o singura rezistenta!"),"Apasati
orice tasta!");
            getch();
        }
        else
        {
            if (a[o].c[p].legare_Re==1)
            {
                int x;
                outtextxy(la,lu,"Sunt legate in serie.");
                outtextxy(la,lu+textheight("Sunt legate in serie."),"Doriti sa le
legati in paralel (apasati 2) sau sa anulati operatiunea (apasati 0)?");
                citire_nr(x,10,la,lu+2*textheight("Sunt legate in serie."));
                int ata=3;
                while (x!=0&&x!=2)
                {
                    outtextxy(la,lu+ata*textheight("Sunt legate in serie."),"Alegeti o
optiune intre 2 si 0!");
                    citire_nr(x,10,la,lu+(ata+1)*textheight("Sunt legate in serie."));
                    ata++;
                }
                cleardevice();
                if (x==2)

```

```

    {
        a[o].c[p].legare_Re=2;
        outtextxy(la,lu,"Rezistentele sunt legate in paralel!");
        outtextxy(la,lu+textheight("Rezistentele sunt legate in
paralel!"),"Apasati orice tasta!");
        getch();
    }
    else
        if (x==0)
        {
            outtextxy(la,lu,"Operatiune anulata!");
            outtextxy(la,lu+textheight("Operatiune anulata!"),"Apasati orice
tasta!");
            getch();
        }
    }
    else
        if (a[o].c[p].legare_Re==2)
        {
            outtextxy(la,lu,"Sunt legate in paralel.");
            outtextxy(la,lu+textheight("Sunt legate in paralel."),"Doriti sa
le legati in serie (apasati 1) sau sa anulati operatiunea (apasati 0)?");
            int x;
            citire_nr(x,10,la,lu+2*textheight("Sunt legate in paralel.));
            int ata=3;
            while (x!=1&&x!=0)
            {
                outtextxy(la,lu+ata*textheight("Sunt legate in paralel."),"Alegeti o
optiune intre 1 si 0!");
                citire_nr(x,10,la,lu+(ata+1)*textheight("Sunt legate in paralel.));
                ata++;
            }
            cleardevice();
            if (x==1)
            {
                a[o].c[p].legare_Re=1;
                outtextxy(la,lu,"Rezistentele sunt legate in serie!");
                outtextxy(la,lu+textheight("Operatiune anulata!"),"Apasati orice
tasta!");
                getch();
            }
            else
                if (x==0)
                {
                    outtextxy(la,lu,"Operatiune anulata!");
                    outtextxy(la,lu+textheight("Operatiune anulata!"),"Apasati orice
tasta!");
                    getch();
                }
            }
        }
        else
        {
            cleardevice();
            outtextxy(la,lu,"Operatiune esuata!");
            outtextxy(la,lu+textheight("Operatiune anulata!"),"Apasati
orice tasta!");
            getch();
        }
    }
}
void modificare_legare_baterii (manager a[], curent c[], int codu, int m, int
codc, int la, int lu)
{
    setcolor(WHITE);

```

```

settextstyle(1,0,18);
cleardevice();
int o=determinare_cod_manager(a,m,codu);
int p=determinare_cod_circuit(a,m,codu,codc);
int ct=0;
bool sem=0;
if (p!=-1)
{
    if (a[o].c[p].nr_B==1)
    {
        outtextxy(la,lu,"Exista o singura baterie!");
        outtextxy(la,lu+textheight("Exista o singura
baterie!"),"Apasati orice tasta!");
        getch();
    }
    else
    {
        if (a[o].c[p].legare_B==1)
        {
            int x;
            outtextxy(la,lu,"Sunt legate in serie.");
            outtextxy(la,lu+textheight("Sunt legate in serie."),"Doriti sa le
legati in paralel (apasati 2) sau sa anulati operatiunea (apasati 0)?");
            citire_nr(x,10,la,lu+2*textheight("Sunt legate in serie.));
            int ata=3;
            while (x!=0&&x!=2)
            {
                outtextxy(la,lu+ata*textheight("Sunt legate in paralel"),"Alegeti o
optiune intre 2 si 0!");
                citire_nr(x,10,la,lu+(ata+1)*textheight("Sunt legate in paralel"));
                ata++;
            }
            cleardevice();
            if (x==2)
            {
                a[o].c[p].legare_B=2;
                outtextxy(la,lu,"Bateriile sunt legate in paralel!");
                outtextxy(la,lu+textheight("Bateriile sunt legate in
paralel!"),"Apasati orice tasta!");
                getch();
            }
            else
            {
                if (x==0)
                {
                    outtextxy(la,lu,"Operatiune anulata!");
                    outtextxy(la,lu+textheight("Bateriile sunt legate in
paralel!"),"Apasati orice tasta!");
                    getch();
                }
            }
        }
        else
        {
            if (a[o].c[p].legare_B==2)
            {
                outtextxy(la,lu,"Sunt legate in paralel");
                outtextxy(la,lu+textheight("Sunt legate in paralel"),"Doriti sa
le legati in serie (apasati 1) sau sa anulati operatiunea (apasati 0)?");
                int x;
                citire_nr(x,10,la,lu+2*textheight("Sunt legate in paralel"));
                int ata=3;
                while (x!=1&&x!=0)
                {
                    outtextxy(la,lu+ata*textheight("Sunt legate in paralel"),"Alegeti o
optiune intre 1 si 0!");
                    citire_nr(x,10,la,lu+(ata+1)*textheight("Sunt legate in paralel"));

```

```

        ata++;
    }
    cleardevice();
    if (x==1)
    {
        a[o].c[p].legare_B=1;
        outtextxy(la,lu,"Bateriile sunt legate in serie!");
        outtextxy(la,lu+textheight("Bateriile sunt legate in
paralel!"),"Apasati orice tasta!");
        getch();
    }
    else
        if (x==0)
        {
            outtextxy(la,lu,"Operatiune anulata!");
            outtextxy(la,lu+textheight("Bateriile sunt legate in
paralel!"),"Apasati orice tasta!");
            getch();
        }
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Operatiune esuata!");
        outtextxy(la,lu+textheight("Bateriile sunt legate in
paralel!"),"Apasati orice tasta!");
        getch();
    }
}

void echivalent_serie_baterie (manager a[], int m, curent c[], int codu, int
codc, int &sr, int &se)
{
    int sel=0;
    int srl=0;
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    for (int i=0;i<a[o].c[p].nr_B;i++)
    {
        sel+=a[o].c[p].B[i].E;
        srl+=a[o].c[p].B[i].r;
    }
    se=sel;
    sr=srl;
}

int echivalent_serie_rezistenta (manager a[], int m, curent c[], int codu,
int codc)
{
    int s=0;
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    for (int i=0;i<a[o].c[p].nr_R;i++)
        s+=a[o].c[p].R[i].Re;
    return s;
}

int echivalent_paralel_rezistenta (manager a[], int m, curent c[],
rezistenta_exterioara R[],int codu, int codc)
{
    int s=0, pr=1;
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    for (int i=0;i<a[o].c[p].nr_R;i++)
    {
        pr*=a[o].c[p].R[i].Re;
    }
}

```

```

        s+=a[o].c[p].R[i].Re;
    }
    return pr/s;
}
void echivalent_paralel_baterie (manager a[], int m, curent c[], int codu,
int codc, int &ee, int &er)
{
    int se=0, sr=0, pe=1, pr=1;
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    for (int i=0;i<a[o].c[p].nr_B;i++)
    {
        se+=a[o].c[p].B[i].E/a[o].c[p].B[i].r;
        sr+=a[o].c[p].B[i].r;
        pr*=a[o].c[p].B[i].r;
    }
    er=pr/sr;
    ee=se/(1/er);
}
void adaugare_circuit (manager a[], int m, curent c[], baterie B[],
rezistenta_exterioara R[], int codu, int la, int lu, int o)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    curent nou;
    int ct=0;
    bool sem=0;
    while (ct<2)
    {
        cleardevice();
        outtextxy(la,lu,"Codul circuitului: (max 3 cifre) ");
        citire_nr(nou.codc,100,la+textwidth("Codul circuitului: (max 3 cifre)
"),lu);
        if (nou.codc<0||nou.codc>999)
        {
            outtextxy(la,lu+textheight("Codul circuitului: (max 3 cifre)
"),"Cod incorect!");
            outtextxy(la,lu+2*textheight("Codul circuitului: (max 3 cifre)
"),"Apasati orice tasta!");
            getch();
            ct++;
        }
        else
            if (cautare_cod_circuit1("curent.in",nou.codc))
            {
                outtextxy(la,lu+textheight("Codul circuitului: (max 3
cifre) "), "Acest circuit exista deja!");
                outtextxy(la,lu+2*textheight("Codul circuitului: (max 3
cifre) "), "Apasati orice tasta!");
                getch();
                ct++;
            }
        else
        {
            ct=2;
            sem=1;
        }
    }
    cleardevice();
    if (sem)
    {
        ct=0;
        sem=0;
        while (ct<2)

```



```

{
    cleardevice();
    outtextxy(la,lu,"Numarul de baterii: (max 5)");
    citire_nr(nou.nr_B,100,la+textwidth("Numarul de baterii: (max
5)"),lu);
    if (nou.nr_B==0||nou.nr_B>5)
    {
        outtextxy(la,lu+textheight("Numarul de baterii: (max
5)"),"Valoare incorecta!");
        outtextxy(la,lu+2*textheight("Numarul de baterii: (max
5)"),"Apasati orice tasta!");
        getch();
        ct++;
    }
    else
    {
        ct=2;
        sem=1;
    }
}
cleardevice();
if (sem)
{
    ct=0; sem=0;
    if (nou.nr_B==1) {nou.legare_B=1; sem=1;}
    else
    {
        outtextxy(la,lu,"Cum doriti sa fie legate bateriile: ");
        outtextxy(la,lu+textheight("Cum doriti sa fie legate bateriile:
"),"In serie (apasati 1) sau in paralel (apasati 2)?");
        citire_nr(nou.legare_B,10,la,lu+2*textheight("Cum doriti sa fie
legate bateriile: "));
        int ata=3;
        while ((nou.legare_B!=1)&&(nou.legare_B!=2))
        {
            outtextxy(la,lu+ata*textheight("Cum doriti sa fie legate
bateriile: "), "Alegeti o optiune intre 1 si 2!");
            citire_nr(nou.legare_B,10,la,lu+(ata+1)*textheight("Cum
doriti sa fie legate bateriile: "));
            ata++;
        }
    }
    cleardevice();
    while (ct<2)
    {
        cleardevice();
        outtextxy(la,lu,"Numarul de rezistente exterioare: (max 5)");
        citire_nr(nou.nr_R,100,la+textwidth("Numarul de rezistente
exterioare: (max 5)"),lu);
        if (nou.nr_R==0||nou.nr_R>5)
        {
            outtextxy(la,lu+textheight("Numarul de rezistente
exterioare: (max 5)"),"Valoare incorecta!");
            outtextxy(la,lu+2*textheight("Numarul de rezistente
exterioare: (max 5)"),"Apasati orice tasta!");
            getch();
            ct++;
        }
        else
        {
            ct=2;
            sem=1;
        }
    }
}
cleardevice();

```

```

if (sem)
{
    ct=0; sem=0;
    if (nou.nr_R==1) {nou.legare_Re=1; sem=1;}
    else
    {
        outtextxy(la,lu,"Cum doriti sa fie legate rezistentele: ");
        outtextxy(la,lu+textheight("Cum doriti sa fie legate
rezistentele: "), "In serie (apasati 1) sau in paralel (apasati 2)?");
        citire_nr(nou.legare_Re,100,la,lu+2*textheight("Cum doriti sa
fie legate rezistentele: "));
        int ata2=3;
        while ((nou.legare_Re!=1)&&(nou.legare_Re!=2))
        {
            outtextxy(la,lu+ata2*textheight("Cum doriti sa fie legate
rezistentele: "), "Alegeti o optiune intre 1 si 2!");

citire_nr(nou.legare_Re,100,la,lu+(ata2+1)*textheight("Cum doriti sa fie
legate rezistentele: "));
        }
        if ((nou.legare_Re==1)|| (nou.legare_Re==2))
        {
            ct=2;
            sem=1;
        }
    }
cleardevice();
if (sem)
{
    for (int x=0;x<nou.nr_B;x++)
    {
        ct=0;
        sem=0;
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu,"Codul bateriei: (max 3 cifre)");
            citire_nr(nou.B[x].codB,100,la+textwidth("Codul
bateriei: (max 3 cifre)"),lu);
            if (nou.B[x].codB<0||nou.B[x].codB>999)
            {
                outtextxy(la,lu+textheight("Codul bateriei: (max
3 cifre)"), "Cod incorect!");
                outtextxy(la,lu+2*textheight("Codul bateriei:
(max 3 cifre)"), "Apasati orice tasta!");
                getch();
                ct++;
            }
            else
            if
(cautare_cod_baterie("baterie.in",nou.B[x].codB))
            {
                outtextxy(la,lu+textheight("Codul bateriei:
(max 3 cifre)"), "Acest cod exista deja!");
                outtextxy(la,lu+2*textheight("Codul bateriei:
(max 3 cifre)"), "Apasati orice tasta!");
                getch();
                ct++;
            }
            else
            {
                ct=2;
                sem=1;
            }
        }
    }
}

```

```

    }
    cleardevice();
    if (sem)
    {
        ct=0;
        sem=0;
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu,"Tensiunea      electromotoare      a
bateriei: (max 24 V, diferita de 0) ");
            citire_nr(nou.B[x].E,100,la+textwidth("Tensiunea
electromotoare a bateriei: (max 24 V, diferita de 0) "),lu);
            if (nou.B[x].E==0||nou.B[x].E>24)
            {
                outtextxy(la,lu+textheight("Tensiunea
electromotoare a bateriei: (max 24 V, diferita de 0) "), "Valoare incorecta!");
                outtextxy(la,lu+2*textheight("Tensiunea
electromotoare a bateriei: (max 24 V, diferita de 0) "), "Apasati orice
tasta!");
                getch();
                ct++;
            }
            else
            {
                ct=2;
                sem=1;
            }
        }
        cleardevice();
        if (sem)
        {
            sem=0;
            ct=0;
            while (ct<2)
            {
                cleardevice();
                outtextxy(la,lu,"Rezistenta      interioara      a
bateriei: (max 15 Ohm) ");
                citire_nr(nou.B[x].r,100,la+textwidth("Rezistenta interioara a bateriei: (max
15 Ohm) "),lu);
                if (nou.B[x].r>15)
                {
                    outtextxy(la,lu+textheight("Rezistenta
interioara a bateriei: (max 15 Ohm) "), "Valoare incorecta!");
                    outtextxy(la,lu+2*textheight("Rezistenta
interioara a bateriei: (max 15 Ohm) "), "Apasati orice tasta!");
                    getch();
                    ct++;
                }
                else
                {
                    ct=2;
                    sem=1;
                }
            }
            cleardevice();
        }
    }
    if (sem)
    {
        for (int x=0;x<nou.nr_R;x++)
        {

```

```

        sem=0;
        ct=0;
        while (ct<2)
        {
            cleardevice();
            outtextxy(la,lu,"Codul rezistentei
exterioare: (max 3 cifre) ");

citire_nr(nou.R[x].codR,100,la+textwidth("Codul rezistentei exterioare: (max
3 cifre) "),lu);

            if (nou.R[x].codR<0||nou.R[x].codR>999)
            {
                outtextxy(la,lu+textheight("Codul
rezistentei exterioare: (max 3 cifre) "), "Cod incorect!");
                outtextxy(la,lu+2*textheight("Codul
rezistentei exterioare: (max 3 cifre) "), "Apasati orice tasta!");
                getch();
                ct++;
            }
            else
            if
(cautare_cod_rezistenta("rezistenta.in",nou.R[x].codR))
            {
                outtextxy(la,lu+textheight("Codul
rezistentei exterioare: (max 3 cifre) "), "Acest cod exista deja!");
                outtextxy(la,lu+2*textheight("Codul
rezistentei exterioare: (max 3 cifre) "), "Apasati orice tasta!");
                getch();
                ct++;
            }
            else
            {
                ct=2;
                sem=1;
            }
        }
        cleardevice();
        if (sem)
        {
            sem=0;
            ct=0;
            while (ct<2)
            {
                cleardevice();
                outtextxy(la,lu,"Valoarea
rezistentei (max 20 Ohm, diferita de 0) ");

citire_nr(nou.R[x].Re,100,la+textwidth("Valoarea rezistentei (max 20 Ohm,
diferita de 0) "),lu);

                if
(nou.R[x].Re==0||nou.R[x].Re>20)
                {

outtextxy(la,lu+textheight("Valoarea rezistentei (max 20 Ohm, diferita de 0)
"), "Valoare incorecta!");

outtextxy(la,lu+2*textheight("Valoarea rezistentei (max 20 Ohm, diferita de 0)
"), "Apasati orice tasta!");

                                getch();
                                ct++;
                            }
                            else
                            {
                                ct=2;

```

```

        sem=1;
    }
}
cleardevice();
}}}}}}
if (sem)
{
    int y; char t[100];
    outtextxy(la,lu,"CIRCUITUL ");
    itoa(nou.codc,t,10); outtextxy(la+textwidth("CIRCUITUL "),lu,t);

    outtextxy(la,lu+textheight("CIRCUITUL"),"Nr baterii: "); itoa(nou.nr_B,t,10);
    outtextxy(la+textwidth("Nr baterii: "),lu+textheight("CIRCUITUL"),t);
    int z=textheight("CIRCUITUL");
    if (nou.nr_B!=1)

{outtextxy(la,lu+2*textheight("CIRCUITUL"),"Legate in ");
    if (nou.legare_B==1)
    outtextxy(la+textwidth("Legate in "),lu+2*z,"serie");
    else outtextxy(la+textwidth("Legate
in "),lu+2*z,"paralel");}

    int h=0;
    int j=textwidth("serie");
    for (int i=0;i<nou.nr_B;i++)
    {

        outtextxy(la+j,lu+(h+3)*z,"Bateria ");
        itoa (nou.B[i].codB,t,10);
        outtextxy(la+j+textwidth("Bateria
"),lu+(h+3)*z,t);
        outtextxy(la+j+textwidth("Bateria
"),lu+(h+4)*z,"E: ");
        itoa (nou.B[i].E,t,10);

        outtextxy(la+j+textwidth("Bateria")+textwidth("E: "),lu+(h+4)*z,t);
        outtextxy(la+j+textwidth("Bateria
")+textwidth("E: ") +textwidth(t),lu+(h+4)*z," V");
        outtextxy(la+j+textwidth("Bateria
"),lu+(h+5)*z,"r: ");
        itoa(nou.B[i].r,t,10);

        outtextxy(la+j+textwidth("Bateria")+textwidth("r: "),lu+(h+5)*z,t);
        outtextxy(la+j+textwidth("Bateria
")+textwidth("r: ") +textwidth(t),lu+(h+5)*z," Ohm");
        h=i+3;
    }
    outtextxy(la+j,lu+(h+7)*z,"Apasati
orice tasta pentru a vedea rezistentele exterioare!");
    getch();
    cleardevice();
    outtextxy(la,lu,"CIRCUITUL ");
    itoa(nou.codc,t,10); outtextxy(la+textwidth("CIRCUITUL "),lu,t);

    outtextxy(la,lu+textheight("CIRCUITUL"),"Nr rezistente: ");
    itoa(nou.nr_R,t,10);
    outtextxy(la+textwidth("Nr rezistente:
"),lu+textheight("CIRCUITUL"),t);
    if (nou.nr_R!=1)

{outtextxy(la,lu+2*textheight("CIRCUITUL"),"Legate in ");
    if (nou.legare_Re==1)
    outtextxy(la+textwidth("Legate in "),lu+2*z,"serie");
    else outtextxy(la+textwidth("Legate
in "),lu+2*z,"paralel");}

```

```

        h=0;
        for (int i=0;i<nou.nr_R;i++)
        {
            outtextxy(la+j,lu+(h+3)*z,"Rezistenta ");
            itoa (nou.R[i].codR,t,10);

            outtextxy(la+j+textwidth("Rezistenta "),lu+(h+3)*z,t);

            outtextxy(la+j+textwidth("Rezistenta "),lu+(h+4)*z,"R: ");
            itoa (nou.R[i].Re,t,10);

            outtextxy(la+j+textwidth("Rezistenta")+textwidth("R: "),lu+(h+4)*z,t);

            outtextxy(la+j+textwidth("Rezistenta ") +textwidth("R: ")
            "+textwidth(t),lu+(h+4)*z," Ohm");

            h=i+2;
        }
        outtextxy(la+j,lu+(h+6)*z,"Doriti sa salvati circuitul? (apasati 1)
sau nu? (apasati 0)?");
        citire_nr(y,100,la+j,lu+(h+7)*textheight("Doriti sa salvati circuitul?
(apasati 1) sau nu? (apasati 0)?"));
        while (y!=0&& y!=1)
        {
            cleardevice();
            outtextxy(la,lu,"Doriti sa salvati circuitul? (apasati 1) sau nu?
(apasati 0)?");
            outtextxy(la,lu+textheight("Doriti sa salvati circuitul? (apasati
1) sau nu? (apasati 0)?"),"Alegeti o optiune intre 0 si 1!");
            citire_nr(y,100,la,lu+2*textheight("Doriti sa salvati circuitul?
(apasati 1) sau nu? (apasati 0)?"));
        }
        if (y==1)
        {
            cleardevice();
            a[o].nr_c++;
            a[o].c[a[o].nr_c-1]=nou;
            outtextxy(la,lu,"Adaugare cu succes!");
            outtextxy(la,lu+textheight("Adaugare cu succes!"),"Apasati orice
tasta!");
            getch();
        }
        else
        {
            cleardevice();
            outtextxy(la,lu,"Adaugare anulata!");
            outtextxy(la,lu+textheight("Adaugare anulata!"),"Apasati
orice tasta!");
            getch();
        }
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Adaugare esuata!");
        outtextxy(la,lu+textheight("Adaugare esuata!"),"Apasati orice
tasta!");
        getch();
    }
}

void poze_circuit(int m, manager a[], curent c[], int o, int p)
{
    if (a[o].c[p].nr_B==1&&a[o].c[p].nr_R==1)

```

```

readimagefile("elr1.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwidth(),ge
tmaxheight()/2);
    else
        if (a[o].c[p].legare_B==1)
            if (a[o].c[p].legare_Re==1)

readimagefile("esrs.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwidth(),ge
tmaxheight()/2);
        else

readimagefile("esrp.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwidth(),ge
tmaxheight()/2);
        if (a[o].c[p].legare_B==2)
            if (a[o].c[p].legare_Re==1)

readimagefile("eprs.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwidth(),ge
tmaxheight()/2);
        else

readimagefile("eprp.JPG",5*getmaxwidth()/8,getmaxheight()/16,getmaxwidth(),ge
tmaxheight()/2);
}
void afisare_completa (manager a[], int m, curent c[], baterie B[],
rezistenta_exterioara R[], int codu,int codc, int la, int lu)
{
    char t[100];
    int o=determinare_cod_manager(a,m,codu);
    int p=determinare_cod_circuit(a,m,codu,codc);
    setcolor(WHITE);
    cleardevice();
    poze_circuit(m,a,c,o,p);
    setttextstyle(1,0,18);
    outtextxy(la,lu,"CIRCUITUL ");
    itoa(codc,t,10);
    outtextxy(la+textwidth("CIRCUITUL "),lu,t);
    outtextxy(la,lu+textheight("CIRCUITUL"),"Nr baterii: ");
    itoa(a[o].c[p].nr_B,t,10);
    outtextxy(la+textwidth("Nr baterii:
"),lu+textheight("CIRCUITUL"),t);
    int z=textheight("CIRCUITUL");
    if (a[o].c[p].nr_B!=1)
    {outtextxy(la,lu+2*textheight("CIRCUITUL"),"Legate in ");
    if (a[o].c[p].legare_B==1) outtextxy(la+textwidth("Legate in
"),lu+2*z,"serie");
    else outtextxy(la+textwidth("Legate in "),lu+2*z,"paralel");}
    int h=0;
    int j=textwidth("serie");
    for (int i=0;i<a[o].c[p].nr_B;i++)
    {

outtextxy(la+j,lu+(h+3)*z,"Bateria ");
    itoa (a[o].c[p].B[i].codB,t,10);
    outtextxy(la+j+textwidth("Bateria
"),lu+(h+3)*z,t);
    outtextxy(la+j+textwidth("Bateria
"),lu+(h+4)*z,"E: ");
    itoa (a[o].c[p].B[i].E,t,10);

outtextxy(la+j+textwidth("Bateria")+textwidth("E: "),lu+(h+4)*z,t);
    outtextxy(la+j+textwidth("Bateria
")+textwidth("E: ")+textwidth(t),lu+(h+4)*z," V");
    outtextxy(la+j+textwidth("Bateria
"),lu+(h+5)*z,"r: ");
    itoa(a[o].c[p].B[i].r,t,10);
    }
}

```

```

outtextxy(la+j+textwidth("Bateria")+textwidth("r: "),lu+(h+5)*z,t);
outtextxy(la+j+textwidth("Bateria
")+textwidth("r: ") +textwidth(t),lu+(h+5)*z," Ohm");
h=i+3;
}
outtextxy(la+j,lu+(h+7)*z,"Apasati
orice tasta pentru a vedea rezistentele exterioare!");
getch();
cleardevice();
poze_circuit(m,a,c,o,p);
outtextxy(la,lu,"CIRCUITUL ");
itoa(codc,t,10); outtextxy(la+textwidth("CIRCUITUL "),lu,t);

outtextxy(la,lu+textheight("CIRCUITUL"),"Nr rezistente: ");
itoa(a[o].c[p].nr_R,t,10); outtextxy(la+textwidth("Nr rezistente:
"),lu+textheight("CIRCUITUL"),t);

if (a[o].c[p].nr_R!=1)

{outtextxy(la,lu+2*textheight("CIRCUITUL"),"Legate in ");
if (a[o].c[p].legare_Re==1)
outtextxy(la+textwidth("Legate in "),lu+2*z,"serie");
else outtextxy(la+textwidth("Legate
in "),lu+2*z,"paralel");}

h=0;
for (int i=0;i<a[o].c[p].nr_R;i++)
{

outtextxy(la+j,lu+(h+3)*z,"Rezistenta ");
itoa (a[o].c[p].R[i].codR,t,10);

outtextxy(la+j+textwidth("Rezistenta "),lu+(h+3)*z,t);

outtextxy(la+j+textwidth("Rezistenta "),lu+(h+4)*z,"R: ");
itoa (a[o].c[p].R[i].Re,t,10);

outtextxy(la+j+textwidth("Rezistenta")+textwidth("R: "),lu+(h+4)*z,t);

outtextxy(la+j+textwidth("Rezistenta ") +textwidth("R:
")+textwidth(t),lu+(h+4)*z," Ohm");

h=i+2;
}
outtextxy(la+j,lu+(h+7)*z,"Apasati
orice tasta pentru a vedea parametrii circuitului!");
getch();
cleardevice();
poze_circuit(m,a,c,o,p);
int lb, lr, lR;
outtextxy(la,lu,"CIRCUITUL ");
itoa(codc,t,10); outtextxy(la+textwidth("CIRCUITUL "),lu,t);
if (a[o].c[p].legare_B==1)
{
lb, lr;

echivalent_serie_baterie(a,m,c,codu,codc,lr,lb);

outtextxy(la,lu+textheight("CIRCUITUL "), "Tensiunea electromotoare
echivalenta: ");

outtextxy(la+textwidth("Tensiunea
electromotoare echivalenta: ")/2,lu+2*textheight("CIRCUITUL "), "E: ");
itoa(lb,t,10);
outtextxy(la+textwidth("Tensiunea electromotoare echivalenta:
")/2+textwidth("E: "),lu+2*textheight("CIRCUITUL "),t);

```



```

                                outtextxy(la+textwidth("Tensiunea
electromotoare                echivalenta:                ") / 2 + textwidth("E:
")+textwidth(t), lu+2*textheight("CIRCUITUL "), " V");

outtextxy(la, lu+4*textheight("CIRCUITUL "), "Rezistenta interioara echivalenta:
");

                                outtextxy(la+textwidth("Tensiunea
electromotoare echivalenta: ") / 2, lu+5*textheight("CIRCUITUL "), "r: ");
                                itoa(lr, t, 10);
outtextxy(la+textwidth("Tensiunea                electromotoare                echivalenta:
") / 2 + textwidth("r: "), lu+5*textheight("CIRCUITUL "), t);
                                outtextxy(la+textwidth("Tensiunea
electromotoare                echivalenta:                ") / 2 + textwidth("r:
")+textwidth(t), lu+5*textheight("CIRCUITUL "), " Ohm");
                                if (a[o].c[p].legare_Re==1)
                                {

lR=echivalent_serie_rezistenta(a,m,c,codu,codc);

outtextxy(la, lu+7*textheight("CIRCUITUL "), "Rezistenta exterioara echivalenta:
");

outtextxy(la+textwidth("Tensiunea                electromotoare                echivalenta:
") / 2, lu+8*textheight("CIRCUITUL "), "R: ");
                                itoa(lR, t, 10);
outtextxy(la+textwidth("Tensiunea                electromotoare                echivalenta:
") / 2 + textwidth("R: "), lu+8*textheight("CIRCUITUL "), t);

outtextxy(la+textwidth("Tensiunea                electromotoare                echivalenta:
") / 2 + textwidth("R: ") + textwidth(t), lu+8*textheight("CIRCUITUL "), " Ohm");
                                }
                                else
                                {

lR=echivalent_paralel_rezistenta(a,m,c,R,codu,codc);

outtextxy(la, lu+7*textheight("CIRCUITUL "), "Rezistenta exterioara echivalenta:
");

outtextxy(la+textwidth("Tensiunea                electromotoare                echivalenta:
") / 2, lu+8*textheight("CIRCUITUL "), "R: ");
                                itoa(lR, t, 10);
outtextxy(la+textwidth("Tensiunea                electromotoare                echivalenta:
") / 2 + textwidth("R: "), lu+8*textheight("CIRCUITUL "), t);

outtextxy(la+textwidth("Tensiunea                electromotoare                echivalenta:
") / 2 + textwidth("R: ") + textwidth(t), lu+8*textheight("CIRCUITUL "), " Ohm");
                                }
                                }
                                else
                                {

echivalent_paralel_baterie(a,m,c,codu,codc,lr,lb);

outtextxy(la, lu+textheight("CIRCUITUL                "), "Tensiunea                electromotoare
echivalenta: ");
                                outtextxy(la+textwidth("Tensiunea
electromotoare echivalenta: ") / 2, lu+2*textheight("CIRCUITUL "), "E: ");
                                itoa(lb, t, 10);
outtextxy(la+textwidth("Tensiunea                electromotoare                echivalenta:
") / 2 + textwidth("E: "), lu+2*textheight("CIRCUITUL "), t);
                                outtextxy(la+textwidth("Tensiunea
electromotoare                echivalenta:                ") / 2 + textwidth("E:
")+textwidth(t), lu+2*textheight("CIRCUITUL "), " V");

```

```

outtextxy(la,lu+4*textheight("CIRCUITUL "), "Rezistenta interioara echivalenta:");
outtextxy(la+textwidth("Tensiunea electromotoare echivalenta: ")/2,lu+5*textheight("CIRCUITUL "), "r: ");
itoa(lr,t,10);
outtextxy(la+textwidth("Tensiunea electromotoare echivalenta: ")/2+textwidth("r: "),lu+5*textheight("CIRCUITUL "),t);
outtextxy(la+textwidth("Tensiunea electromotoare echivalenta: ")/2+textwidth("r: ") +textwidth(t),lu+5*textheight("CIRCUITUL "), " Ohm");
if (a[o].c[p].legare_Re==1)
{
lR=echivalent_serie_rezistenta(a,m,c,codu,codc);

outtextxy(la,lu+7*textheight("CIRCUITUL "), "Rezistenta exterioara echivalenta:");

outtextxy(la+textwidth("Tensiunea electromotoare echivalenta: ")/2,lu+8*textheight("CIRCUITUL "), "R: ");
itoa(lR,t,10);
outtextxy(la+textwidth("Tensiunea electromotoare echivalenta: ")/2+textwidth("R: "),lu+8*textheight("CIRCUITUL "),t);

outtextxy(la+textwidth("Tensiunea electromotoare echivalenta: ")/2+textwidth("R: ") +textwidth(t),lu+8*textheight("CIRCUITUL "), " Ohm");
}
else
{
lR=echivalent_paralel_rezistenta(a,m,c,R,codu,codc);

outtextxy(la,lu+7*textheight("CIRCUITUL "), "Rezistenta exterioara echivalenta:");

outtextxy(la+textwidth("Tensiunea electromotoare echivalenta: ")/2,lu+8*textheight("CIRCUITUL "), "R: ");
itoa(lR,t,10);
outtextxy(la+textwidth("Tensiunea electromotoare echivalenta: ")/2+textwidth("R: "),lu+8*textheight("CIRCUITUL "),t);

outtextxy(la+textwidth("Tensiunea electromotoare echivalenta: ")/2+textwidth("R: ") +textwidth(t),lu+8*textheight("CIRCUITUL "), " Ohm");
}
}

outtextxy(la,lu+10*textheight("CIRCUITUL "), "Apasati orice tasta pentru a vedea I, U, W, P");

getch();
cleardevice();
poze_circuit(m,a,c,o,p);
outtextxy(la,lu,"CIRCUITUL ");
itoa(codc,t,10); outtextxy(la+textwidth("CIRCUITUL "),lu,t);
outtextxy(la,lu+textheight("CIRCUITUL "), "Intensitatea in ramura principala: ");
int I=lb/(lr+lR);
itoa(I,t,10);
outtextxy(la+textwidth("Intensitatea in ramura principala: ")/2,lu+2*textheight("CIRCUITUL "), "I: ");
outtextxy(la+textwidth("Intensitatea in ramura principala: ")/2+textwidth("I: "),lu+2*textheight("CIRCUITUL "),t);

```

```

                                outtextxy(la+textwidth("Intensitatea
in          ramura          principala:          ") / 2 + textwidth("I:
")+textwidth(t), lu+2*textheight("CIRCUITUL "), " A");

outtextxy(la, lu+4*textheight("CIRCUITUL "), "Tensiunea pe circuitul exterior:
");

                                int U=I*I*R;
                                itoa(U, t, 10);
                                outtextxy(la+textwidth("Intensitatea
in ramura principala: ") / 2, lu+5*textheight("CIRCUITUL "), "U: ");
                                outtextxy(la+textwidth("Intensitatea
in ramura principala: ") / 2 + textwidth("U: "), lu+5*textheight("CIRCUITUL "), t);
                                outtextxy(la+textwidth("Intensitatea
in          ramura          principala:          ") / 2 + textwidth("U:
")+textwidth(t), lu+5*textheight("CIRCUITUL "), " V");

outtextxy(la, lu+7*textheight("CIRCUITUL "), "Energia degajata pe circuitul
exterior intr-un minut: ");

                                int W=U*I*60;
                                itoa(W, t, 10);
                                outtextxy(la+textwidth("Intensitatea
in ramura principala: ") / 2, lu+8*textheight("CIRCUITUL "), "W: ");
                                outtextxy(la+textwidth("Intensitatea
in ramura principala: ") / 2 + textwidth("W: "), lu+8*textheight("CIRCUITUL "), t);
                                outtextxy(la+textwidth("Intensitatea
in          ramura          principala:          ") / 2 + textwidth("W:
")+textwidth(t), lu+8*textheight("CIRCUITUL "), " J");

outtextxy(la, lu+10*textheight("CIRCUITUL "), "Puterea dezvoltata pe circuitul
exterior: ");

                                int P=U*I;
                                itoa(P, t, 10);
                                outtextxy(la+textwidth("Intensitatea
in ramura principala: ") / 2, lu+11*textheight("CIRCUITUL "), "P: ");
                                outtextxy(la+textwidth("Intensitatea
in ramura principala: ") / 2 + textwidth("P: "), lu+11*textheight("CIRCUITUL "), t);
                                outtextxy(la+textwidth("Intensitatea
in          ramura          principala:          ") / 2 + textwidth("P:
")+textwidth(t), lu+11*textheight("CIRCUITUL "), " W");

outtextxy(la, lu+13*textheight("CIRCUITUL "), "Puterea maxima dezvoltata pe
circuitul exterior: ");

                                int Pm=lb*lb/(4*I*R);
                                itoa(Pm, t, 10);
                                outtextxy(la+textwidth("Intensitatea
in ramura principala: ") / 2, lu+14*textheight("CIRCUITUL "), "Pm: ");
                                outtextxy(la+textwidth("Intensitatea
in ramura principala: ") / 2 + textwidth("Pm: "), lu+14*textheight("CIRCUITUL
"), t);

                                outtextxy(la+textwidth("Intensitatea
in          ramura          principala:          ") / 2 + textwidth("Pm:
")+textwidth(t), lu+14*textheight("CIRCUITUL "), " W");
                                outtextxy(la+j, lu+16*z, "Apasati orice tasta!");
                                getch();
}

void citire_parola_admin (char nf[], char x[])
{
    ifstream f(nf);
    f >> x;
    f.close();
}

void meniu_modificari_circuit (manager a[], baterie B[],
rezistenta_exterioara R[], int m, int codu, curent c[], int la, int lu)
{

```

```

setcolor(WHITE);
settextstyle(1,0,18);
cleardevice();
outtextxy(la,lu,"Introduceti codul circuitului: ");
int p;
citire_nr(p,100,la+textwidth("Introduceti codul circuitului: "),lu);
int d=determinare_cod_circuit(a,m,codu,p);
if (d!=-1)
{
char x;
do
{
cleardevice();
outtextxy(la,lu,"1.Modificare baterie");
outtextxy(la,lu+textheight("1.Modificare baterie"),"2.Modificare
rezistenta exterioara");
outtextxy(la,lu+2*textheight("1.Modificare baterie"),"3.Modificare
legare baterii");
outtextxy(la,lu+3*textheight("1.Modificare baterie"),"4.Modificare
legare rezistente");
outtextxy(la,lu+4*textheight("1.Modificare baterie"),"5.Inapoi");
outtextxy(la,lu+5*textheight("1.Modificare baterie"),"-----
Alegeti optiunea dorita!");
do
{
x=getch();
}while (x<='0'&&x>='6');
cleardevice();
switch (x)
{
case '1': modificare_baterie(codu,p,B,a,m,c,la,lu); break;
case '2': modificare_rezistenta(codu,p,R,a,m,c,la,lu); break;
case '3': modificare_legare_baterii(a,c,codu,m,p,la,lu); break;
case '4': modificare_legare_rezistente(a,c,codu,m,p,la,lu); break;
}
}while (x!='5');
outtextxy(la,lu,"Actiune terminata!"); outtextxy(la+textwidth("Actiune
terminata! "),lu,"Va multumim!");
outtextxy(la,lu+textheight("Actiune terminata!"),"Apasati orice tasta!");
getch();
}
else
{
cleardevice();
outtextxy(la,lu,"Acest circuit nu exista!");
outtextxy(la,lu+textheight("Acest circuit nu exista!"),"Apasati
orice tasta!"); getch();
}
}
void meniu_stergere (manager a[], int m, baterie B[], curent c[],
rezistenta_exterioara R[], int codu, int la, int lu)
{
setcolor(WHITE);
settextstyle(1,0,18);
cleardevice();
outtextxy(la,lu,"Introduceti codul circuitului: ");
int p;
citire_nr(p,100,la+textwidth("Introduceti codul circuitului: "),lu);
int f=determinare_cod_manager(a,m,codu);
int d=determinare_cod_circuit(a,m,codu,p);
if (d!=-1)
{
char x;
do

```

```

{
    setbkcolor( RGB(75,35,255) );
    cleardevice();
    setcolor(WHITE);
    settextstyle(1,0,18);
    outtextxy(la,lu,"1.Stergere baterie");
    outtextxy(la,lu+textheight("1.Stergere baterie"),"2.Stergere
rezistenta exterioara");
    outtextxy(la,lu+2*textheight("1.Stergere baterie"),"3.Stergere
circuit");
    outtextxy(la,lu+3*textheight("1.Stergere baterie"),"4.Inapoi");
    outtextxy(la,lu+4*textheight("1.Stergere baterie"),"----- Alegeti
optiunea dorita!");
    do
    {
        x=getch();
    }while (x<='0'&&x>='5');
    cleardevice();
    switch (x)
    {
        case '1': stergere_baterie(codu,p,B,a,m,c,la,lu); break;
        case '2': stergere_rezistenta(codu,p,R,a,m,c,la,lu); break;
        case '3': stergere_circuit(m,a,c,d,f,la,lu); break;
    }
    }while (x!='4');
    outtextxy(la,lu,"Actiune terminata!"); outtextxy(la+textwidth("Actiune
terminata! "),lu,"Va multumim!");
    outtextxy(la,lu+textheight("Actiune terminata!"),"Apasati orice tasta!");
    getch();
    }
    else
    {
        cleardevice();
        outtextxy(la,lu,"Acest circuit nu exista!");
        outtextxy(la,lu+textheight("Acest circuit nu exista!"),"Apasati
orice tasta!"); getch();
    }
}

void meniu_adaugare (manager a[], int m, baterie B[], curent c[],
rezistenta_exterioara R[], int codu, int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    char x;
    int o=determinare_cod_manager(a,m,codu);
    do
    {
        setbkcolor( RGB(75,35,255) );
        cleardevice();
        setcolor(WHITE);
        settextstyle(1,0,18);
        outtextxy(la,lu,"1.Adaugare baterii");
        outtextxy(la,lu+textheight("1.Adaugare baterii"),"2.Adaugare
rezistente");
        outtextxy(la,lu+2*textheight("1.Adaugare baterii"),"3.Adaugare
circuit");
        outtextxy(la,lu+3*textheight("1.Adaugare baterii"),"4.Inapoi");
        outtextxy(la,lu+4*textheight("1.Adaugare baterii"),"----- Alegeti
optiunea dorita!");
        do
        {
            x=getch();
        }while (x<='0'&&x>='5');
        cleardevice();
    }
}

```

```

        switch (x)
        {
            case '1': adaugare_baterie(m,a,c,B,codu,la,lu,o); break;
            case '2': adaugare_rezistenta(a,m,c,R,codu,la,lu,o); break;
            case '3': if (a[0].nr_c==3) {outtextxy(la,lu,"Nu se mai pot adauga
circuite!");
                outtextxy(la,lu+textheight("Nu se mai pot adauga circuite!"),"Apasati
orice tasta!");
                getch();} else adaugare_circuit(a,m,c,B,R,codu,la,lu,o); break;
        }
    }while (x!='4');
    outtextxy(la,lu,"Actiune terminata! "); outtextxy(la+textwidth("Actiune
terminata! "),lu,"Va multumim!");
    outtextxy(la,lu+textheight("Actiune terminata! "), "Apasati orice tasta!");
    getch();
}
void meniu_teoretic (manager a[], int m, baterie B[], curent c[],
rezistenta_exterioara R[], int codu, int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    char x;
    do
    {
        setbkcolor(RGB(75,35,255));
        cleardevice();
        setcolor(WHITE);
        settextstyle(1,0,18);
        outtextxy(getmaxwidth()/8,getmaxheight()/16,"TEORIE");
        outtextxy(la,lu,"1.Ce este curentul electric?");
        outtextxy(la,lu+textheight("1.Ce este curentul
continuu?"),"2.Tensiunea electrica");
        outtextxy(la,lu+2*textheight("1.Ce este curentul
continuu?"),"3.Intensitatea curentului electric");
        outtextxy(la,lu+3*textheight("1.Ce este curentul
continuu?"),"4.Rezistenta electrica");
        outtextxy(la,lu+4*textheight("1.Ce este curentul continuu?"),"5.Legea
lui Ohm si legea lui Kirchhoff");
        outtextxy(la,lu+5*textheight("1.Ce este curentul
continuu?"),"6.Legarea rezistentelor");
        outtextxy(la,lu+6*textheight("1.Ce este curentul
continuu?"),"7.Legarea generatoarelor");
        outtextxy(la,lu+7*textheight("1.Ce este curentul
continuu?"),"8.Energia curentului electric");
        outtextxy(la,lu+8*textheight("1.Ce este curentul
continuu?"),"9.Puterea electrica");
        outtextxy(la,lu+9*textheight("1.Ce este curentul
continuu?"),"0.Iesire");
        outtextxy(la,lu+10*textheight("1.Ce este curentul continuu?"),"-----
-- Alegeti optiunea dorita!");
        do
        {
            x=getch();
        }while (x<'0'&&x>='10');
        cleardevice();
        outtextxy(la,getmaxheight()/16,"TEORIE");
        int lul=getmaxheight()/16;
        switch (x)
        {
            case '1': {readimagefile("c1.JPG",la*5,lul,la*10,lul*14);
outtextxy(la,lul*15,"Apasati orice tasta!"); getch();}break;
            case '2': {readimagefile("c2.JPG",la*5,lul,la*10,lul*14);
outtextxy(la,lul*15,"Apasati orice tasta!"); getch();}break;

```

```

        case '3': {readimagefile("c3.JPG",la*5,lul,la*10,lul*14);
outtextxy(la,lul*15,"Apasati orice tasta!"); getch();}break;
        case '4': {readimagefile("c4.JPG",la*5,lul,la*10,lul*14);
outtextxy(la,lul*15,"Apasati orice tasta!"); getch();}break;
        case '5': {readimagefile("c5.JPG",la*5,lul,la*10,lul*14);
outtextxy(la,lul*15,"Apasati orice tasta!"); getch();}break;
        case '6': {readimagefile("c6.JPG",la*5,lul,la*10,lul*14);
outtextxy(la,lul*15,"Apasati orice tasta!"); getch();}break;
        case '7': {readimagefile("c7.JPG",la*5,lul,la*10,lul*14);
outtextxy(la,lul*15,"Apasati orice tasta!"); getch();}break;
        case '8': {readimagefile("c8.JPG",la*5,lul,la*10,lul*14);
outtextxy(la,lul*15,"Apasati orice tasta!"); getch();}break;
        case '9': {readimagefile("c9.JPG",la*5,lul,la*10,lul*14);
outtextxy(la,lul*15,"Apasati orice tasta!"); getch();}break;
    }
    }while (x!='0');
    outtextxy(la,lu,"Actiune terminata! "); outtextxy(la+textwidth("Actiune
terminata! "),lu,"Va multumim!");
    outtextxy(la,lu+textheight("Actiune terminata! "),"Apasati orice tasta!");
getch();
}
void meniu_practic_utilizatori (manager a[], int m, baterie B[], curent c[],
rezistenta_exterioara R[], int codu, int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    char x;
    do
    {
        setbkcolor(RGB(75,35,255));
        cleardevice();
        setcolor(WHITE);
        settextstyle(1,0,18);
        outtextxy(la,lu,"APLICARE");
        outtextxy(la,lu,"1.Afiseaza un circuit");
        outtextxy(la,lu+textheight("1.Afisare"),"2.Afiseaza toate
circuitele");
        outtextxy(la,lu+2*textheight("1.Afisare"),"3.Modificare circuit");
        outtextxy(la,lu+3*textheight("1.Afisare"),"4.Adaugare");
        outtextxy(la,lu+4*textheight("1.Afisare"),"5.Stergere");
        outtextxy(la,lu+5*textheight("1.Afisare"),"6.Modificare parola");
        outtextxy(la,lu+6*textheight("1.Afisare"),"7.Inapoi");
        outtextxy(la,lu+7*textheight("1.Afisare"),"----- Alegeti optiunea
dorita!");
        do
        {
            x=getch();
        }while (x<='0'&&x>='8');
        cleardevice();
        outtextxy(la,lu,"APLICARE");
        switch (x)
        {
            case '1': int cc; outtextxy(la,lu,"Introduceti codul circuitului: ");
citire_nr(cc,10,la+textwidth("Introduceti codul circuitului: "),lu);
            if (cautare_cod_circuit(m,a,codu,cc,c))
afisare_completa(a,m,c,B,R,codu,cc,la,lu); else {cleardevice();
            outtextxy(la,lu,"Cod inexistent! Apasati orice tasta!"); getch();}
break;
            case '2': afisare_circuit(a,m,c,R,B,codu,la,lu); break;
            case '3': meniu_modificari_circuit(a,B,R,m,codu,c,la,lu); break;
            case '4': meniu_adaugare(a,m,B,c,R,codu,la,lu); break;
            case '5': meniu_stergere(a,m,B,c,R,codu,la,lu); break;
            case '6': modificare_parola(a,m,codu,la,lu); break;
        }
    }
}

```

```

    }while (x!='7');
    outtextxy(la,lu,"Actiune terminata! "); outtextxy(la+textwidth("Actiune
terminata! "),lu,"Va multumim!");
    outtextxy(la,lu+textheight("Actiune terminata! "), "Apasati orice tasta!");
    getch();
}
void meniu_utilizatori (manager a[], int m, baterie B[], curent c[],
rezistenta_exterioara R[], int codu, int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    char x;
    int p=determinare_cod_manager(a,m,codu);
    do
    {
        setbkcolor( RGB(75,35,255) );
        cleardevice();
        setcolor(WHITE);
        settextstyle(1,0,18);
        outtextxy(getmaxwidth()/8,getmaxheight()/16,"BUN VENIT, ");
        outtextxy(getmaxwidth()/8+textwidth("BUN VENIT,
"),getmaxheight()/16,a[p].username);
        outtextxy(getmaxwidth()/8+textwidth("BUN VENIT,
")+textwidth(a[p].username),getmaxheight()/16,"!");
        outtextxy(la,lu,"1.Modul teoretic");
        outtextxy(la,lu+textheight("1.Modul teoretic"),"2.Modul practic");
        outtextxy(la,lu+2*textheight("1.Modul teoretic"),"3.Inapoi");
        outtextxy(la,lu+3*textheight("1.Modul teoretic"),"----- Alegeti
modulul dorit!");
        do
        {
            x=getch();
        }while (x<='0'&&x>='4');
        cleardevice();
        outtextxy(getmaxwidth()/8,getmaxheight()/16,"BUN VENIT, ");
        outtextxy(getmaxwidth()/8+textwidth("BUN VENIT,
"),getmaxheight()/16,a[p].username);
        outtextxy(getmaxwidth()/8+textwidth("BUN VENIT,
")+textwidth(a[p].username),getmaxheight()/16,"!");
        switch (x)
        {
            case '1': meniu_teoretic (a,m,B,c,R,codu,la,lu); break;
            case '2': meniu_practic_utilizatori(a,m,B,c,R,codu,la,lu); break;
        }
    }while (x!='3');
    outtextxy(la,lu,"Actiune terminata! "); outtextxy(la+textwidth("Actiune
terminata! "),lu,"Va multumim!");
    outtextxy(la,lu+textheight("Actiune terminata! "), "Parerea ta conteaza!
Da-ne un feedback pentru a imbunatati aplicatia!");
    int y;
    outtextxy(la,lu+2*textheight("Actiune terminata! "), "Apasa 1 daca doresti
sa ne lasi un comentariu sau 0 daca vrei sa iesi!");
    citire_nr(y,10,la,lu+3*textheight("Actiune terminata! "));
    int ata=4;
    while (y!=1&&y!=0)
    {
        outtextxy(la,lu+ata*textheight("Actiune terminata! "), "Alegeti o
optiune intre 1 si 0!");
        citire_nr(y,10,la,lu+(ata+1)*textheight("Actiune terminata! "));
        ata++;
    }
    cleardevice();
    if (y==0)
    {

```



```

        outtextxy(la,lu,"Actiune terminata! ");
outtextxy(la+textwidth("Actiune terminata! "),lu,"Va multumim!");
        outtextxy(la,lu+textheight("Actiune terminata! "), "Apasati orice
tasta!"); getch();
    }
    else
        if (y==1)
        {
            ifstream f("feedback.txt");
            char coment[500][250];
            int n=0;
            while (!f.eof())
                f.getline(coment[++n],250);
            f.close();
            char s[201];
            outtextxy(la,lu,"Scrie-ne un comentariu de maxim 200 de caractere!");
            citire_text(s,201,la,lu+textheight("Scrie-ne un comentariu de maxim
200 de caractere!"));
            ofstream g("feedback.txt");
            for (int i=1;i<=n;i++)
                g<<coment[i]<<'\n';
            g<<a[p].codm<<" "<<s;
            g.close();
            outtextxy(la,lu+2*textheight("Scrie-ne un comentariu de maxim 200 de
caractere!"),"Va multumim!");
            outtextxy(la,lu+3*textheight("Scrie-ne un comentariu de maxim 200 de
caractere!"),"Apasati orice tasta!");
            getch();
        }
    }
}

void identificare_user (manager a[], int m, curent c[], baterie B[],
rezistenta_exterioara R[], int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    cleardevice();
    char u[100], p[100];
    int ct=0;
    bool sem=0;
    int codut;
    while (ct<2)
    {
        cleardevice();
        outtextxy(la,lu,"Introducecti username si parola:");
        outtextxy(la,lu+textheight("Introducecti username si parola:"),"Username:
"); citire_text(u,200,la+textwidth("Username: "),lu+textheight("Introducecti
username si parola:"));
        outtextxy(la,lu+2*textheight("Username: "), "Parola: ");
        citire_text(p,200,la+textwidth("Parola: "),lu+2*textheight("Username: "));
        for (int i=0;i<m;i++)
            if (strcmp(a[i].username,u)==0&&strcmp(a[i].password,p)==0)
            {
                codut=a[i].codm;
                ct=2;
                sem=1;
                break;
            }
        if (sem==0)
        {
            cleardevice();
            outtextxy(la,lu,"Username si parola incorecte!");
            outtextxy(la,lu+textheight("Username si parola
incorecte!"),"Apasati orice tasta!");
            getch();
        }
    }
}

```

```

        ct++;
    }
}
if (sem)
    meniu_utilizatori(a,m,B,c,R,codut,la,lu);
}
void meniu_admin (manager a[], int &m, baterie B[], curent c[],
rezistenta_exterioara R[],int la, int lu)
{
    setcolor(WHITE);
    settextstyle(1,0,18);
    char x;
    do
    {
        setbkcolor( RGB(75,35,255) );
        cleardevice();
        setcolor(WHITE);
        settextstyle(1,0,18);
        outtextxy(getmaxwidth()/8,getmaxheight()/16,"ADMINISTRATOR");
        outtextxy(la,lu,"1.Afisarea utilizatori");
        outtextxy(la,lu+textheight("1.Afisarea          utilizatori"),"2.Adaugare
utilizator");
        outtextxy(la,lu+2*textheight("1.Afisarea          utilizatori"),"3.Modificare
utilizator");
        outtextxy(la,lu+3*textheight("1.Afisarea          utilizatori"),"4.Stergere
utilizator");
        outtextxy(la,lu+4*textheight("1.Afisarea          utilizatori"),"5.Modificare
parola");
        outtextxy(la,lu+5*textheight("1.Afisarea          utilizatori"),"6.Inapoi");
        outtextxy(la,lu+6*textheight("1.Afisarea          utilizatori"),"Alegeti
optiunea dorita!");
        do
        {
            x=getch();
        }while (x<='0'&&x>='7');
        cleardevice();
        outtextxy(getmaxwidth()/8,getmaxheight()/16,"ADMINISTRATOR");
        switch (x)
        {
            case '1': afisare_manageri(a,m,la,lu); break;
            case '2': adaugare_manager(a,m,la,lu); break;
            case '3': modificare_manageri_user_parola(a,m,B,R,c,la,lu); break;
            case '4': stergere_manageri(a,m,c,la,lu); break;
            case '5':   modificare_parola_admin(a,m,"parola_admin.txt",la,lu);
break;
        }
    }while (x!='6');
    outtextxy(la,lu,"Actiune terminata! "); outtextxy(la+textwidth("Actiune
terminata! "),lu,"Va multumim!");
    outtextxy(la,lu+textheight("Actiune terminata! "), "Apasati orice tasta!");
    getch();
}
void meniu_principal (manager a[], int &m, baterie B[], curent c[],
rezistenta_exterioara R[], int la, int lu)
{
    setbkcolor( RGB(75,35,255) );
    setcolor(WHITE);
    settextstyle(1,0,18);
    char x;
    char pas[100];
    char y[21];
    citire_parola_admin("parola_admin.txt",y);
    do
    {

```

```

cleardevice();
setbkcolor( RGB(75,35,255) );
setcolor(WHITE);
settextstyle(1,0,18);
outtextxy( getmaxwidth()/16, getmaxheight()/16, "INVATAREA SI APLICAREA
ELEMENTELOR DE CURENT CONTINUU");
outtextxy( la, lu, "1.Administrator");
outtextxy( la, lu+textheight("1.Administrator"), "2.Utilizator");
outtextxy( la, lu+2*textheight("1.Administrator"), "3.Iesire");
outtextxy( la, lu+3*textheight("1.Administrator"), "Alegeti          statutul
dorit!");
do
{
    x=getch();
}while (x<='0'&&x>='4');
cleardevice();
outtextxy( getmaxwidth()/16, getmaxheight()/16, "INVATAREA SI APLICAREA
ELEMENTELOR DE CURENT CONTINUU");
switch (x)
{
    case '1': outtextxy( la, lu, "Introduceti          parola:          ");
setbkcolor( RGB(75,35,255) ); setcolor(WHITE);
citire_text( pas, 200, la, lu+strlen("Introduceti          parola:          ")); if
(strcmp(y,pas)==0) meniu_admin(a,m,B,c,R,la,lu); else
{cleardevice(); outtextxy( la, lu, "Incorect! Apasati orice tasta!");
getch();} break;
    case '2': identificare_user (a,m,c,B,R,la,lu); break;
}
}while (x!='3');
outtextxy( la, lu, "Actiune terminata! "); outtextxy( la+textwidth("Actiune
terminata! "), lu, "Va multumim!");
outtextxy( la, lu+textheight("Actiune terminata! "), "Apasati orice tasta!");
getch();
}
int main()
{
    iniit();
    initwindow( getmaxwidth(), getmaxheight(), "Curent continuu", 0, 0);
    setbkcolor( RGB(75,35,255) );
    cleardevice();
    manager a[50];
    int m;
    curent c[10];
    baterie B[10];
    rezistenta_exterioara R[10];
    char t[100];
    int lu, la;
    lu=getmaxheight()/8;
    la=getmaxwidth()/16;
    citire_c(c, "manager.in", "curent.in", a, m, "rezistenta.in", "baterie.in", R, B);
    meniu_principal(a, m, B, c, R, la, lu);
    salvare_c(m, c, "manager.in", "curent.in", a, "rezistenta.in", "baterie.in", R, B);
    closegraph(); return 0; }

```

Anexa nr. 2 - Modelul conceptual

