

# PACHETUL STANDARD

## ECO AVENTURA MOBILE

Ghidul utilizatorului



Română

Ghid de utilizare ce cuprinde indicații atât pentru aplicația Server, cât și pentru aplicația mobilă.

# INTRODUCERE

Într-o societate puternic tehnologizată, avantajele furnizate de către aparatura de înaltă performanță au reușit să pătrundă în categoria de necesități ale omului contemporan. Viteza de accesare a informațiilor aproape inimaginabilă pentru generațiile anterioare a devenit în ziua de astăzi un fenomen cât se poate de firesc, ceva care nici n-ar trebui să se întâmple altfel.

Ajunsă la un grad înalt de dezvoltare, sfera tehnologiei s-a extins și asupra taberelor tematice, în care fiecare dintre noi, cel mai probabil, a participat măcar o dată.

EcoAventura Desktop&Mobile reprezintă un pachet alcătuit din două aplicații (Server și Mobilă) construite pe șablonul aplicațiilor de tip "Organizator" și mulate perfect pe necesitățile taberei.

Îmbinând utilul cu plăcutul, structura aplicației oferă numeroase facilități ușor de accesat datorită interfeței grafice prietenoase, prin intermediul căroră distribuirea și îndeplinirea sarcinilor ajunge să fie mult mai simplă.

## De unde a început totul?

Fiind voluntar în tabere sau alte activități înrudite de 2 ani, m-am familiarizat destul de bine cu demersul acestora. Cum peste tot mai există și unele probleme, am sesizat și în contextul actual anumite inconveniente. Marea majoritate din ele se rezumau la o singură problemă principală: lipsa timpului. În mai mică sau mai mare măsură, aceasta era cauzată de minuțiosul proces de planificare a evenimentelor unei zile. S-a dovedit a nu fi tocmai ușoara sarcină de a te asigura că fiecare voluntar se află la locul lui și că știe exact ce are de făcut.

De aceea, după un proces laborios de concepere a arhitecturii aplicației și aproape un an de dezvoltare și depanare, am reușit să ajung la un produs finit, menit să rezolve problemele menționate mai sus, care acum este prezentat sub denumirea de EcoAventura Application.

## De ce EcoAventura Application?

Simplu. Deoarece cuprinde o gamă largă de facilități ambalate într-un mod cât mai simplu și mai ușor de folosit. Mai mult, aplicația este construită în așa fel încât să poată administra o varietate largă de tabere, indiferent de tematica acestora.



### ATENȚIE

Modulul Mobile al aplicației poate fi distribuit fără restricții.  
Este strict interzisă distribuția în mai mult de un exemplar a  
modului Desktop.

Încălcarea normelor precizate poate aduce de la sine întreruperea  
funcționării aplicației.

# FACILITĂȚI

În scopul unei experiențe de utilizator cât mai plăcută, aplicația vine cu o suită de facilități



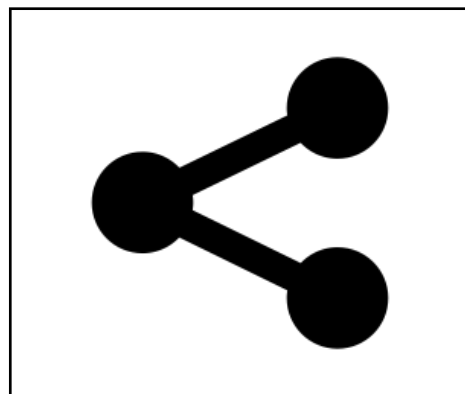
## Securitate

Sistemul de securitate încorporat va asigura siguranța datelor



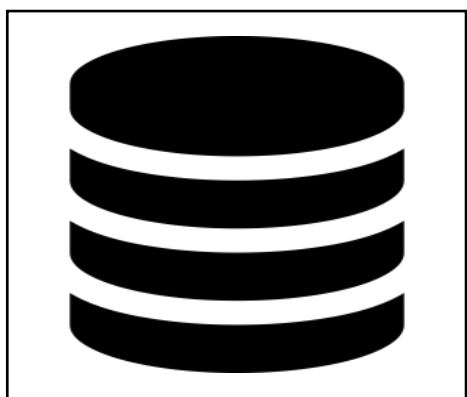
## Interfața grafică prietenoasă

Conferă un aspect plăcut aplicației și accesibil oricui



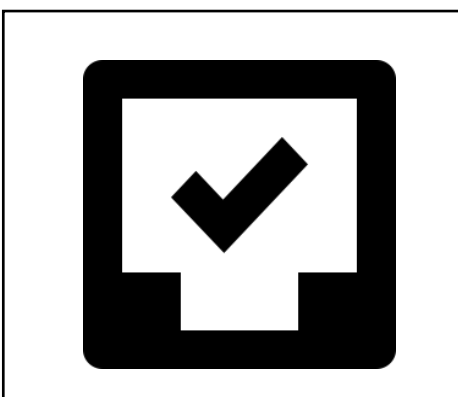
## Aplicație distribuită

Aplicația Mobilă poate comunica cu aplicația Desktop



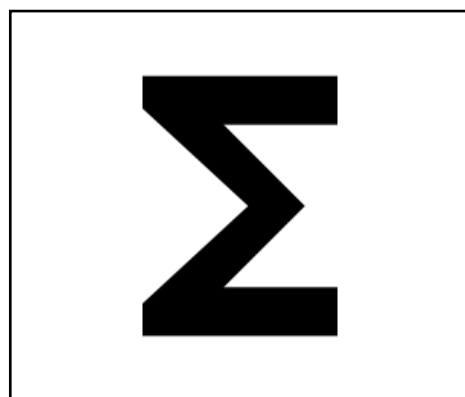
## Acces la baza de date

Aplicația stochează datele într-o bază de date dedicată



## Organizator

Se pot crea, edita, șterge sau distribui task-uri.



## Algoritmă

Pentru o funcționare fluentă, aplicația este construită pe baza unei suite de algoritmi matematici

# CONȚINUTUL PACHETULUI

Pachetul este reprezentat de arhiva "EcoAventuraBundle.rar". Aceasta trebuie să conțină două foldere : "EcoAventuraServer" și "EcoAventuraMobile". În folderul "EcoAventuraServer" trebuie să existe fișierul principal "EcoAventura.jar", de unde se va porni aplicația. Folderul "EcoAventuraMobile" va conține numai fișierul "EcoAventura.apk", care va instalat pe smart-phone sau tabletă.

În cazul lipsei unuia sau mai multor fișiere din cele menționate mai sus, contactați producătorul.

# CERINȚE ȘI RECOMANDĂRI TEHNICE

## Modulul Desktop (Server)

Cerințe :

Platformă : Windows 7 (recomandată), Windows 8/8.1/10/10+

- Procesor : cel puțin Single Core cu frecvența minimă de 2 GHz, sau Dual Core cu frecvența minimă de 1,5 GHz
- Memorie RAM : cel puțin 1,5 GHz
- Memorie ROM : 200 MB pentru instalare și încă 100 MB pentru date suplimentare

Recomandări :

În timpul utilizării modulului Server, este recomandat a nu se folosi și alte aplicații ce nu țin de EcoAventura Application.

Înainte de pornirea Serverului, este indicată oprirea aplicației Skype (în cazul în care există). Altfel, este posibil ca Serverul să nu poată porni.

## Modulul Mobile (Client)

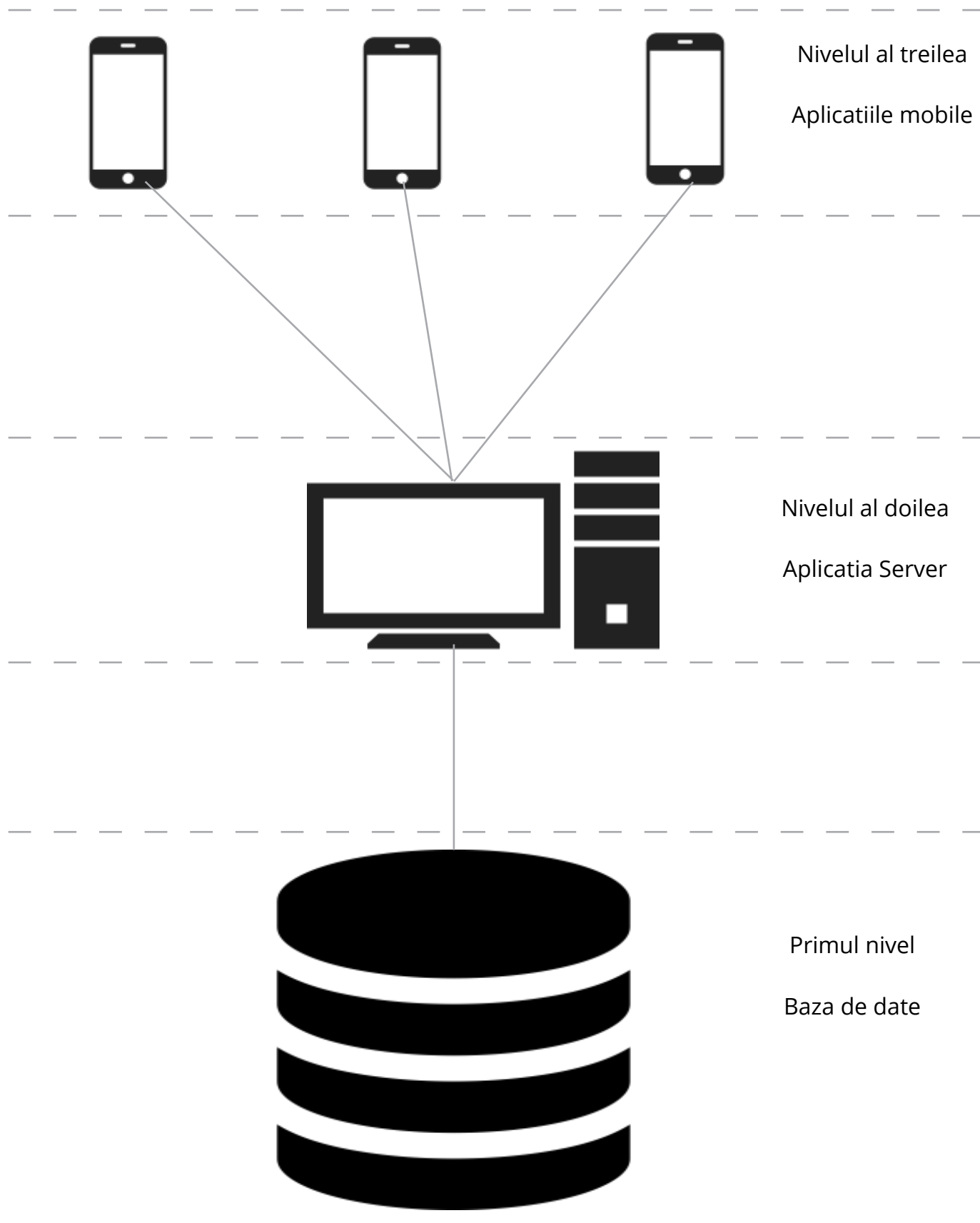
Cerințe

- Dispozitiv dedicat : Smartphone sau Tabletă
- Sistem de operare : Android
- Versiune a sistemului : 5.0+
- Procesor : cel puțin Single Core cu frecvența minimă de 1,2 GHz
- Memorie RAM : minim 750 MB
- Memorie ROM : minim 1 GB

Recomandări :

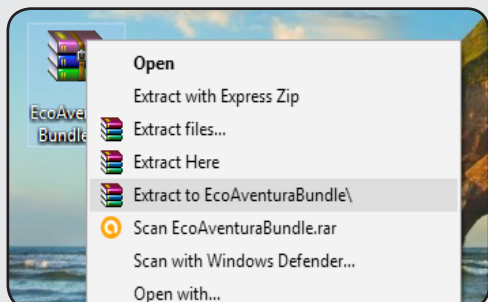
Pentru o experiență cât mai plăcută, se recomandă dispozitivele cu un ecran de cel puțin 4 inch.

# ARHITECTURA APLICAȚIEI



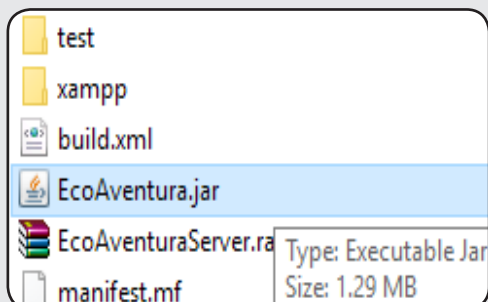
<b>1.</b>	<b>Modulul Desktop (EcoAventura Server)</b>	
	Instalarea Serverului .....	7
	Descrierea ferestrei principale .....	8
	Inițializarea aplicației .....	9
	Fereastra Seriei Active .....	10
	Crearea și postarea unui program .....	11
	Fereastra Serverului .....	12
	Fereastra pentru Ajutor .....	15
<b>2.</b>	<b>Întrebări și răspunsuri utile</b>	
	Cum aflu adresa Serverului în rețea? .....	13
	Cum schimb PORTul Serverului? .....	13
	Cum blochez accesul utilizatorilor la Server? .....	13
	Cum pot dezactiva/șterge seria activă? .....	14
<b>3.</b>	<b>Modulul Android (EcoAventura Mobile)</b>	
	Inițializarea aplicației mobile .....	16
	Utilizarea aplicației mobile .....	17
<b>4.</b>	<b>Secvențe de cod cheie</b>	
	Acceptarea clienților de către Server .....	18
	Crearea bazei de date și conectarea la aceasta .....	18
	Descărcarea bazei de date .....	19
	Descărcarea unui tabel într-un fișier de format .excel .....	19
	Arhivarea aplicației într-un fișier de format .zip .....	20
	Implementarea "Observer Pattern"-ului .....	21
	Implementarea "Singleton"-ului .....	21
<b>5.</b>	<b>Precauție</b>	
	Utilizarea responsabilă a aplicației .....	21

# GHID DE INSTALARE A MODULULUI SERVER



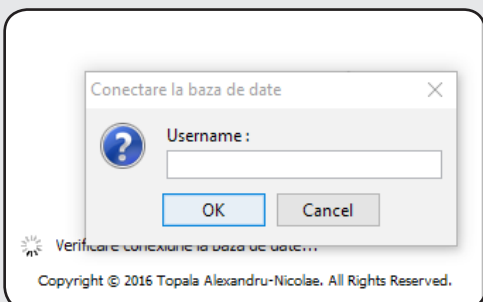
## 1. Dezarhivare

Folosind un program dedicat, se va efectua dezarhivarea celor doua foldere din arhiva "EcoAventuraBundle.rar" ca în imaginea alăturată.



## 2. Lansarea aplicației

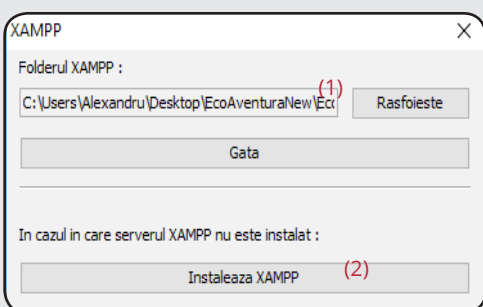
Se va efectua dând dublu-click pe fișierul "EcoAventura.jar" din folderul dezarhivat "EcoAventuraServer".



## 3. Conectarea la baza de date

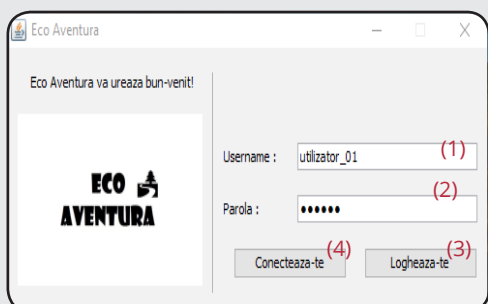
În ordinea în care sunt cerute, următoarele date vor fi introduse astfel :

1. username : "root"
2. parola : "" (adică nimic, se apasă Enter pur și simplu)
3. port : "3306"



## 4. Instalarea XAMPP\*

Aplicația va cere în mod automat specificarea folderului rădăcină al XAMPP. Dacă acesta este instalat deja, se specifică drumul către director prin câmpul (1). Altfel, se va folosi butonul (2) pentru instalare.



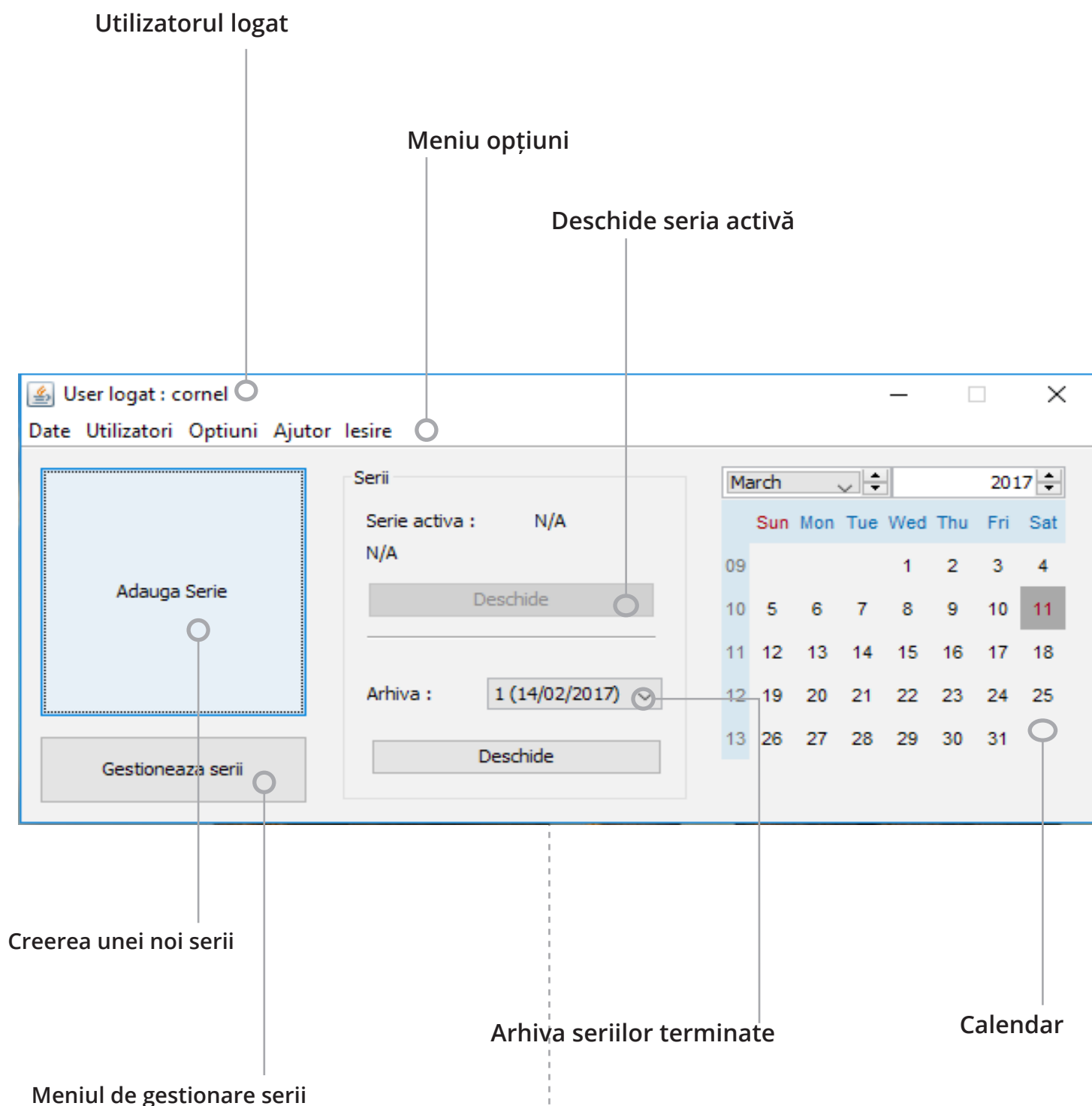
## 5. Crearea unui cont

După completarea câmpurilor (1) și (2) se va apăsa pe buton (3) pentru înregistrare, iar apoi pe butonul (4) pentru conectare.

\* - XAMPP reprezintă un software care asigură un domeniu în care să poată fi creată și gestionată baza de date

# FEREAȘTRA PRINCIPALĂ

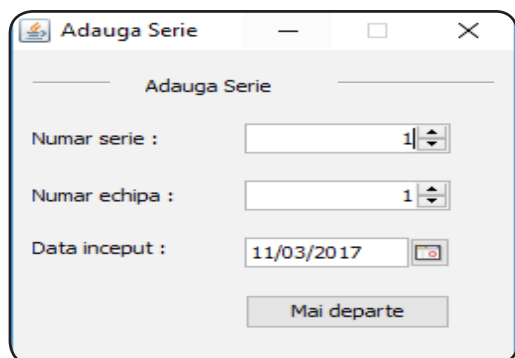
Aceasta este fereaștra principală a aplicației. Din opțiunile acestei ferestre pot fi gestionate seriile, utilizatorii aplicației sau baza de date.





# INIȚIALIZAREA APLICAȚIEI

## Adaugarea unei noi serii

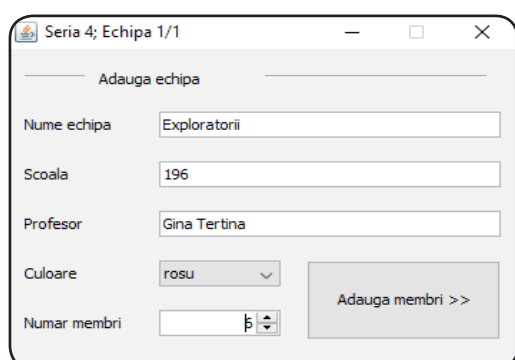


The 'Adauga Serie' dialog box contains the following fields and controls:

- Numar serie :** A text input field with the value '1' and a spinner control.
- Numar echipa :** A text input field with the value '1' and a spinner control.
- Data inceput :** A date picker control showing '11/03/2017'.
- Mai departe** button.

### 1. Apasă opțiunea “Adaugă Serie”

O fereastră ca cea din stânga va apărea. Selectează numărul seriei, câte echipe va conține aceasta și din ce dată va începe. Ai grijă! Seria nu va putea fi accesată mai devreme de data începerii acesteia.

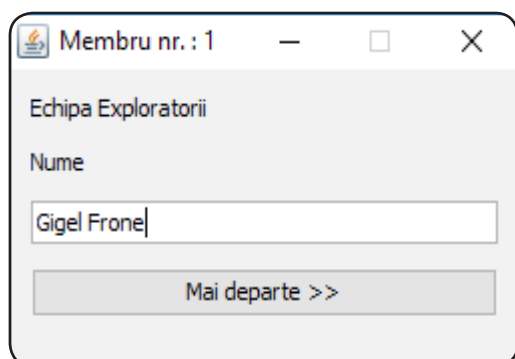


The 'Seria 4; Echipa 1/1' dialog box contains the following fields and controls:

- Adauga echipa** header.
- Nume echipa** text input field with the value 'Exploratorii'.
- Scoala** text input field with the value '196'.
- Profesor** text input field with the value 'Gina Tertina'.
- Culoare** dropdown menu with the value 'rosu'.
- Numar membri** spinner control with the value '1'.
- Adauga membri >>** button.

### 2. Adaugă echipele

După ce seria a fost creată, programul va cere adăugarea unui număr de echipe echivalent cu cel precizat anterior. Completează câmpurile text după instrucțiunile afișate, și asigură-te că ai o listă cu numele tuturor membrilor echipei care se creează.

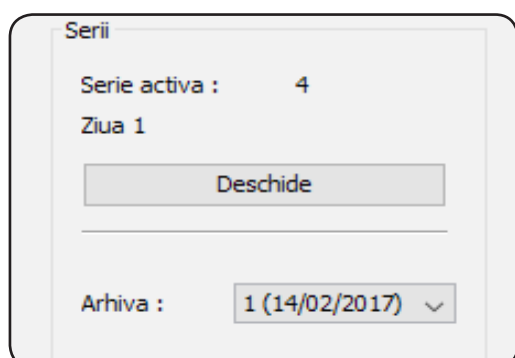


The 'Membru nr. : 1' dialog box contains the following fields and controls:

- Echipa Exploratorii** header.
- Nume** text input field with the value 'Gigel Frone'.
- Mai departe >>** button.

### 3. Adaugă membrii echipei

În cadrul echipei adăugate anterior, aplicația va cere adăugarea atâtor membri precum a fost specificat în pasul anterior. Pentru creerea unui membru, e suficient numele acestuia. După ce ai completat câmpul text, poți folosi tasta "Enter" pentru a trece rapid la următorul membru.



The 'Serii' dialog box contains the following fields and controls:

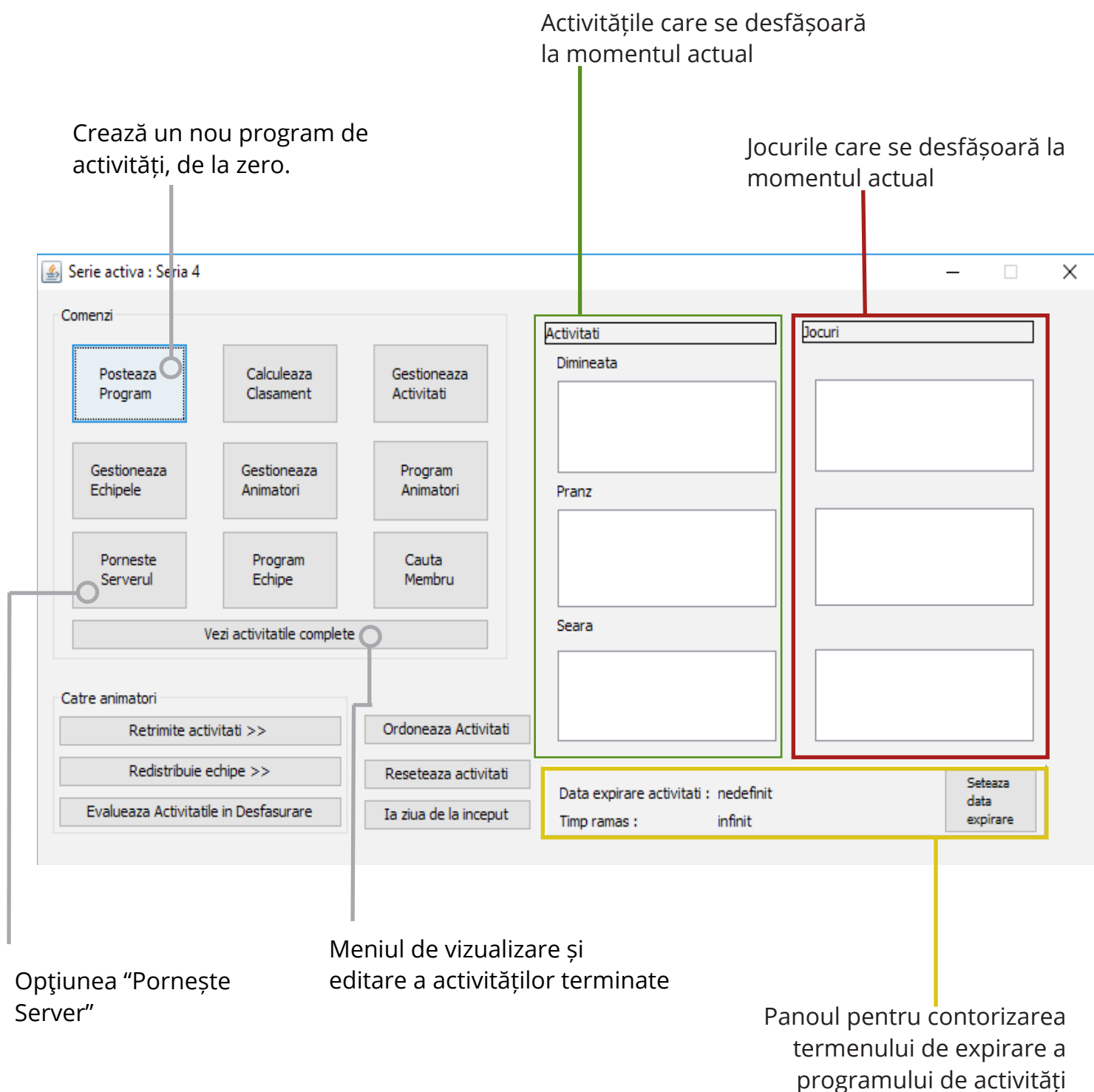
- Serie activa :** A label with the value '4'.
- Ziua 1** label.
- Deschide** button.
- Arhiva :** A dropdown menu with the value '1 (14/02/2017)'.

### 4. Accesarea seriei create

După ce au fost adăugate echipele seriei, aplicația se va întoarce la fereastra principală, unde panoul “Serie Activă” este acum activat. Pentru accesarea seriei, apasă butonul “Deschide”.

# FEREAȘTRA SERIEI ACTIVE

Fereaștra Seriei Active este punctul central gestiune al seriei actuale și al echipelor acesteia, alături de membri ei, activitățile realizate și cele în desfășurare.



# CREAREA ȘI POSTAREA UNUI PROGRAM

Apasă pe opțiunea "Postează Program" al ferestrei "Serie Activă"



Alege în ce perioadă se va desfășura fiecare activitate



Stabilește ce animator/ animatori se vor ocupa de fiecare activitate în parte



Pune la punct detalii legate de locul unde se va desfășura activitate și nu numai (Nu uita să salvezi după ce ai terminat!)



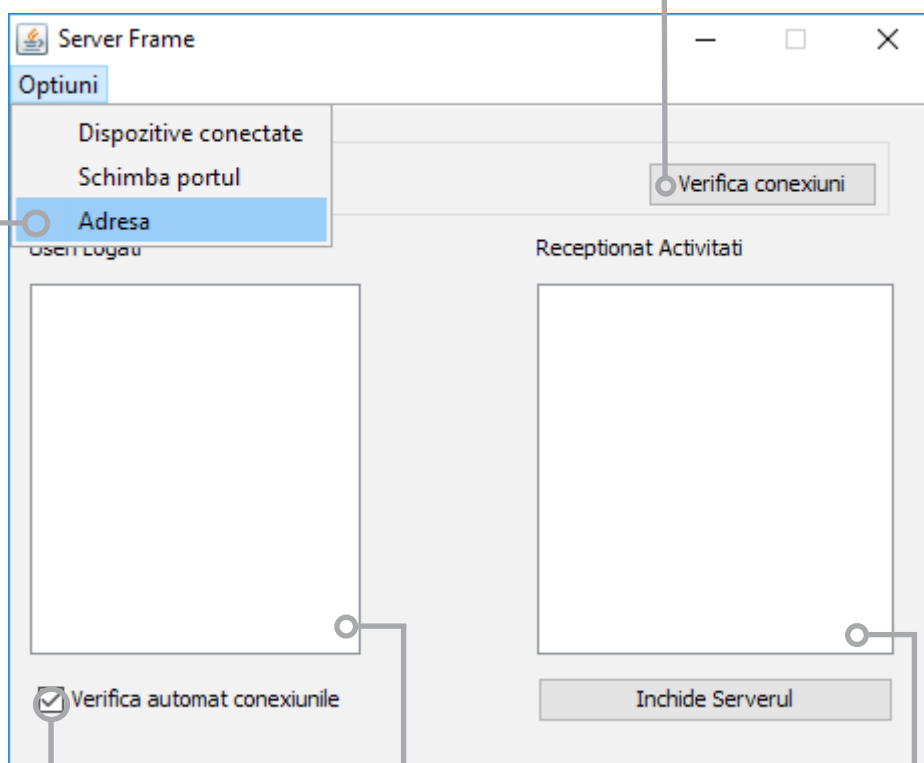
Selectează ce echipe vrei să participe la fiecare activitate în parte și suntem gata de acțiune!

# FEREASTRA SERVERULUI

În această fereastră sunt centralizate toate informațiile privitoare la distribuirea activităților în rețea. Pot fi urmăriți animatorii care sunt conectați, cei care au primit planul de activități și cel mai important, ADRESA la care poate fi găsit Serverul în rețea.

Opțiunea care afișează adresa la care se află Serverul în rețea

Opțiunea de verificare manuală a utilizatorilor care mai sunt conectați



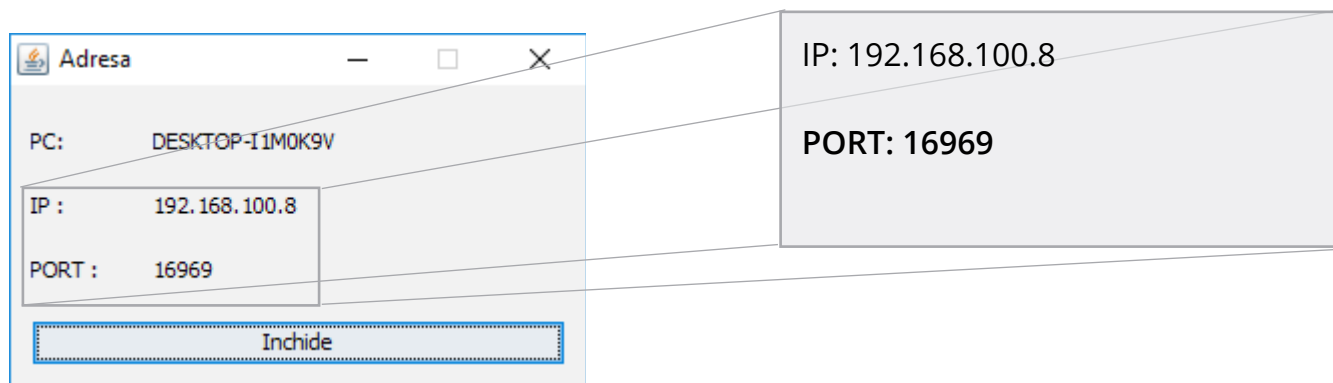
Opțiunea de a verifica automat utilizatorii care mai sunt conectați

Lista utilizatorilor conectați la server

Lista utilizatorilor, conectați sau deconectați, care și-au actualizat planul de activități

# CUM AFLU ADRESA SERVERULUI ÎN REȚEA?

1. Din fereastra Serverului, se accesează meniul "Opțiuni", și se selectează opțiunea "Adresă".
2. În fereastra care se va deschide, se localizează cele doua câmpuri: IP și PORT, și se comunică utilizatorilor valorile respective.



# CUM POT SCHIMBA PORTUL SERVERULUI?

Din fereastra Serverului, din meniul "Opțiuni", se alege opțiunea "Schimbă portul", și se alege un număr întreg din intervalul [1025, 65534].

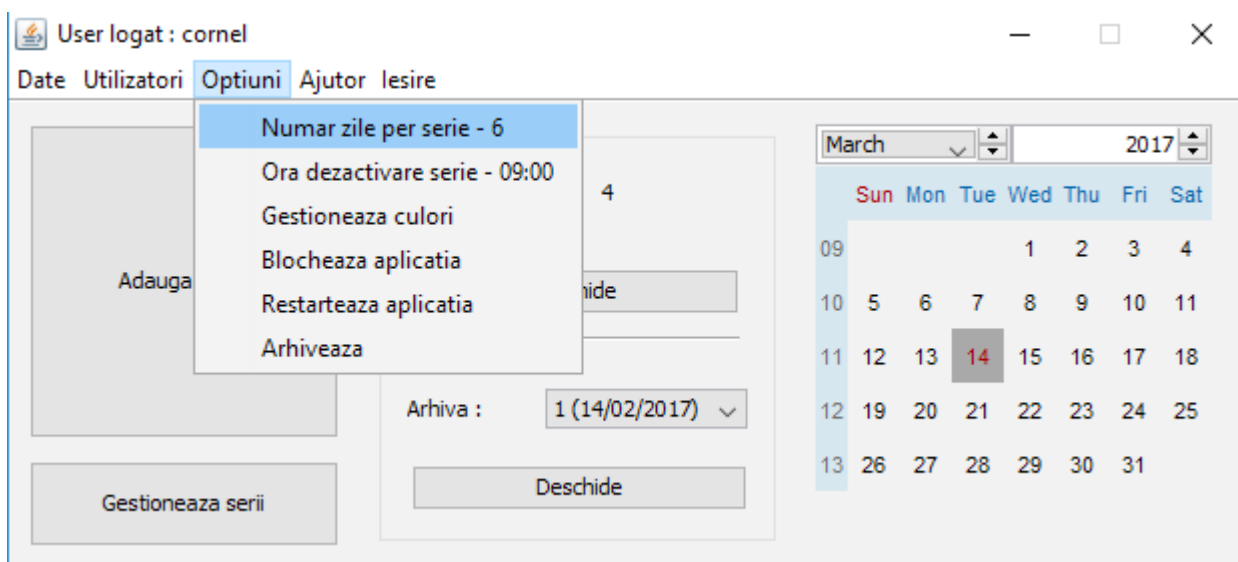
# CUM INTERZIC UTILIZATORILOR SĂ COMUNICE CU SERVERUL?

Utilizatorii pot primi sau trimite date către Server numai dacă acesta este deschis. Pentru a bloca accesul utilizatorilor la datele aplicației, este suficientă închiderea Serverului. Această operațiune se poate efectua folosind opțiunea "Închide Server", sau și mai simplu, de la "x".

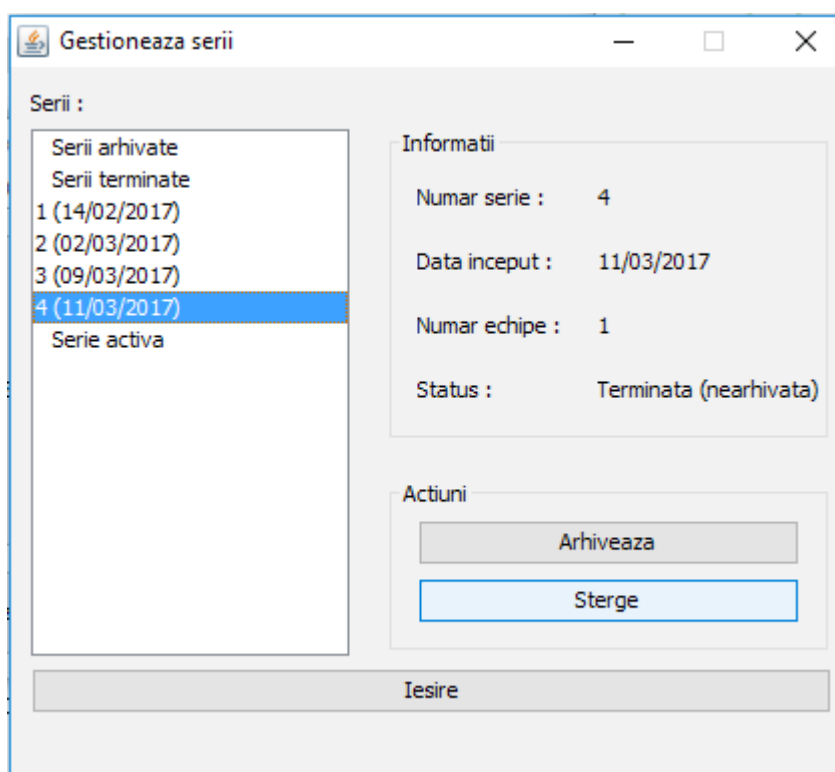
# CUM POT DEZACTIVA/ȘTERGE SERIA ACTIVĂ?

Deoarece se opune logicii aplicației, nu există o opțiune definită pentru dezactivarea seriei active. Chiar și așa, există totuși o soluție, care va fi descrisă în pașii de mai jos:

1. Se deschide fereastra pentru setarea zilei de dezactivare a seriei, găsită după calea : "Fereastră Principală -> Opțiuni (din bara de meniu) -> Număr zile per serie"



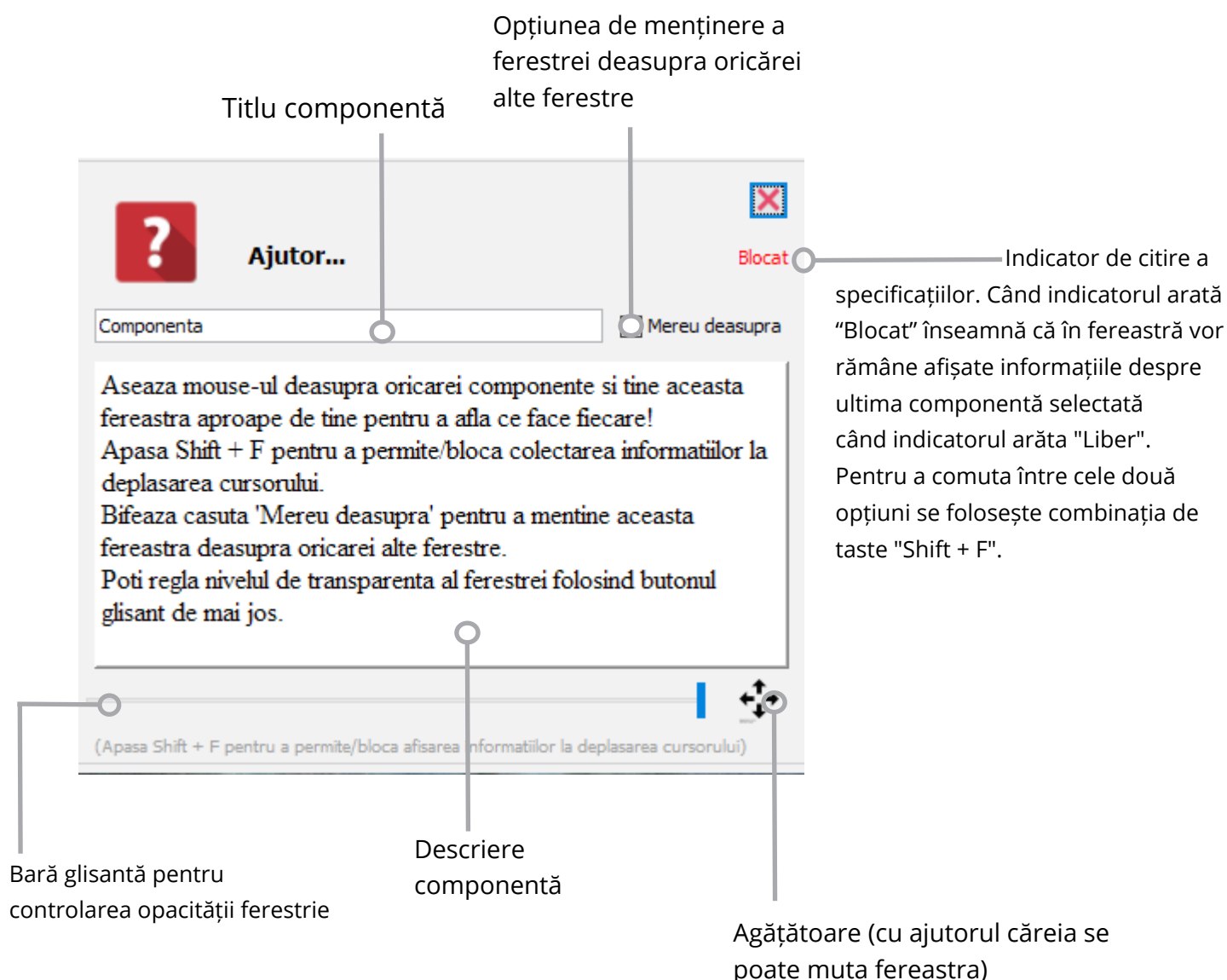
2. Se setează un număr de zile astfel încât aplicația să fi expirat deja. Dacă așa ceva nu este posibil, atunci se setează convenabil și ora de dezactivare.
3. La revenire, seria mai devreme activă va fi acum considerată terminată. După adăugarea unei noi serii, ziua și ora dezactivării trebuie setate înapoi la valorile inițiale.
4. Dacă în plus dorim să ștergem seria activă, după dezactivare, intrăm în meniul "Gestionează Serii" din fereastra principală, unde putem identifica și șterge seria (ne)dorită.



# FEREAȘTRA PENTRU AJUTOR

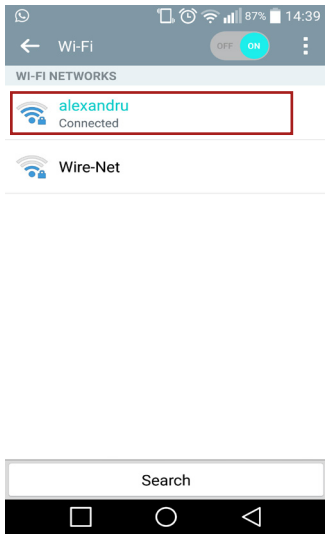
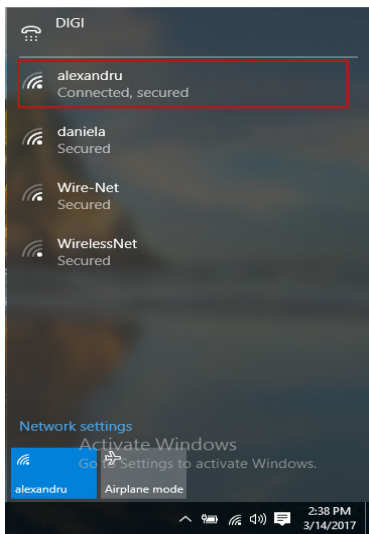
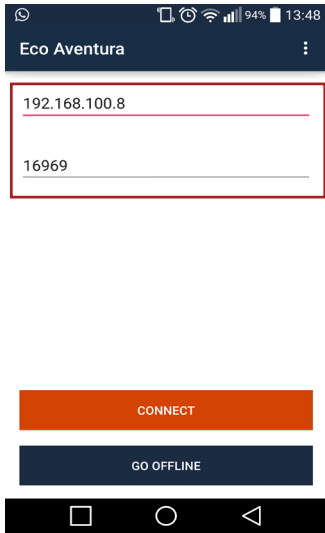
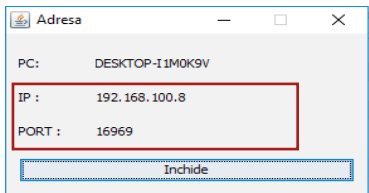
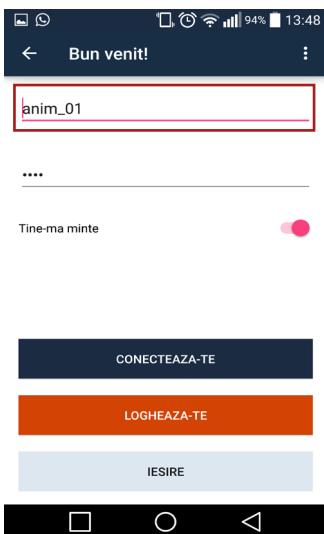
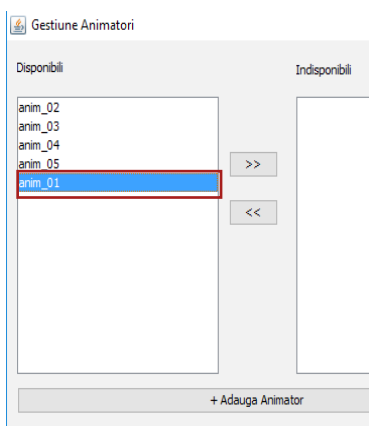
Pentru a afla în detaliu ce face fiecare componentă de oriunde din aplicație, fără să fie o avalanșă de informații, Fereaștra pentru Ajutor poate fi instrumentul perfect.

Ca să aflăm ce se va întâmpla dacă vom apăsa un buton anume, e suficient să poziționăm cursorul deasupra elementului dorit. În acel moment, Fereaștra pentru Ajutor va fi instant populată cu informații privitoare la acel buton.



Pentru lansarea Ferestrei pentru Ajutor (sau Îndrumător), din Fereaștra Principală, din bara de meniuri, se accesează meniul "Ajutor", iar apoi opțiunea "Lansează Îndrumător".

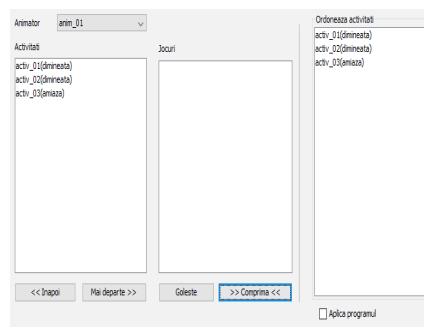
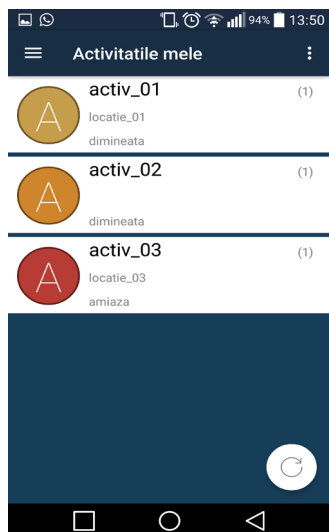
# INIȚIALIZAREA APLICAȚIEI MOBILE

Pas	Aplcația mobilă	Server	Descriere
1.			<p>Atât dispozitivul mobil, cât și calculatorul pe care rulează Serverul trebuie să fie conectate la aceeași rețea Wireless.</p> <p>(În cazul de față, dispozitivele sunt conectate la rețeaua "Alexandru")</p>
2.			<p>Pentru a conecta aplicația mobilă la Server trebuie să completăm datele necesare (IP și PORT) exact ca cele furnizate de Server.</p> <p>Aplicația mobilă poate funcționa și în modul Offline, singurul impediment fiind cel că aplicația nu va mai putea trimite sau primi date de la Server. În rest, activitățile se pot desfășura în mod normal.</p>
3.			<p>Pentru a se crea un nou utilizator, este necesar să existe pe Server un animator cu nume identic, deja creat.</p> <p>După logarea utilizatorului (folosind opțiunea "Loghează-te"), acesta își poate accesa contul folosind opțiunea "Conectează-te".</p>



# UTILIZAREA APLICAȚIEI MOBILE

4.



În fereastra principală a aplicației mobile va apărea lista tuturor activităților care trebuie realizate. Ordinea activităților este pe telefon este aceeași cu cea de pe Server. Ordinea este importantă în acest program deoarece activitățile vor fi realizate în ordine, de sus în jos.

Panoul unde este afișat termenul limită până la care pot fi trimise activitățile. După expirarea programului nu va mai putea fi trimisă nicio activitate.

Scurtătură către activitatea care se desfășoară Acum!

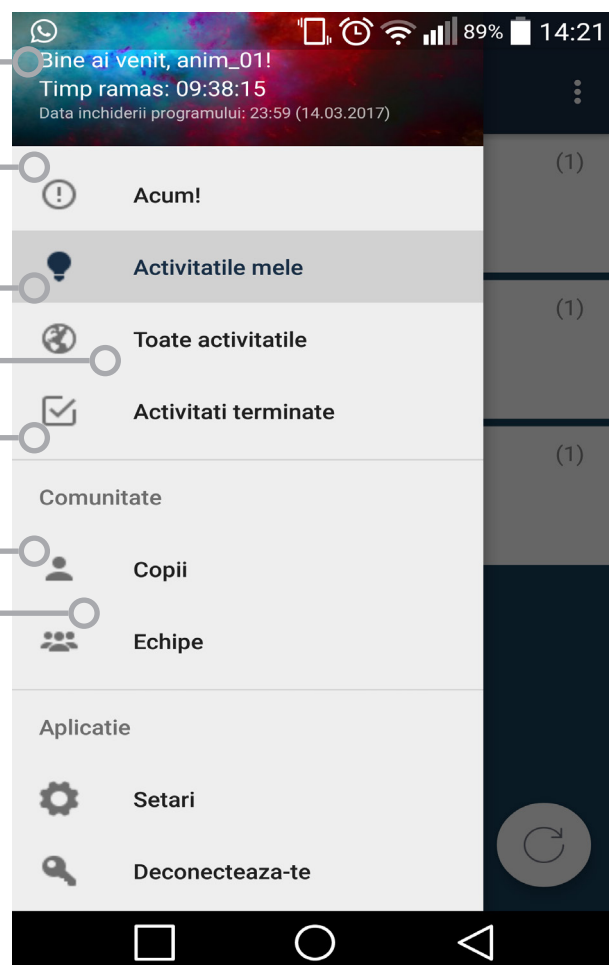
Trimitere către pagina principală, unde sunt listate activitățile mele.

Conduce la pagina unde sunt listați toți animatorii implicați în programul actual, al căror program disponibil.

Trimite la lista activităților terminate de către utilizatorul logat.

Copii seriei actuale

Echipele seriei actuale



# SECVENȚE DE COD CHEIE

În următoarele pagini, vor fi prezentate secvențe de cod esențiale în funcționarea corectă a aplicației.



## ATENȚIE

Capitolele ce urmează sunt dedicate în particular dezvoltatorilor și evaluatorilor.

## Acceptarea clienților de către Server

```
77     try {
78         socket = new ServerSocket(PORT);
79         running = true;
80         new Thread () {
81             @Override
82             public void run () {
83                 try {
84                     try {
85                         SerieActivaFrame.jButton7.setBackground(Color.GREEN);
86                     } catch (Exception e) {
87                     }
88                     while (running) {
89                         ServerThread st = new ServerThread(socket.accept(), receiver);
90                         clients.add(st);
91                         st.start();
92                         if (devices != null) devices.addDevice(st.getSocket());
93                     }
94                     socket.close();
95                     try {SerieActivaFrame.jButton7.setBackground(null);} catch (Exception e) {}
96                 } catch (Exception e) {
97                     ////
98                     System.out.println("Socket inchis");
99                 }
100             }
101         }.start();
102     } catch (IOException ex) {
103         Logger.getLogger(ServerFrame.class.getName()).log(Level.SEVERE, null, ex);
104     }
```

În secvența de cod alăturată este creat un fir de execuție dedicat Serverului. Invocând metoda "accept()", Serverul așteaptă conectarea unor noi clienți.

La momentul conectării, metoda returnează un obiect de tip Socket, corespunzător clientului conectat.

Este creat un nou fir de execuție prin instanțierea obiectului de tip ServerThread, care va răspunde comenzilor trimise de către client.

## Crearea bazei de date și conectarea la aceasta

În aplicație sunt implementate comenzi native ale limbajului SQL. Procedura de mai jos crează baza de date locală "EcoAventuraDB", la care se conectează folosind datele introduse de utilizator la inițierea aplicației. După conectare, prin intermediul unui obiect de tip ScriptRunner, este încărcată o configurație a bazei de date parsată prin parametrul "canonicalPath".

```
907     Statement st = sqlConnection.createStatement();
908     st.execute("CREATE DATABASE ecoaventuradb");
909     Class.forName("com.mysql.jdbc.Driver");
910
911     String data [] = StartFrame.deserializeaza();
912
913     Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:" + data[2] + "/ecoaventuradb", data[0], data[1]);
914
915     ScriptRunner runner = new ScriptRunner(conn, false, false);
916     runner.runScript(new BufferedReader(new FileReader(canonicalPath)));
917
```

## Descărcarea bazei de date

Folosind executabilul "mysqldump.exe", aplicația poate descărca, prin execuția unei instrucțiuni scrise în limbaj de comandă, întreaga bază de date într-un fișier, din care poate fi încărcată ulterior (vezi pasul anterior).

```
928 File f = new File (DATE_SALVATE_PATH);
929
930 if (!f.exists()) {
931     if (!f.mkdir()) {
932         throw new Exception("Fișierul nu poate fi scris");
933     }
934 }
935 SimpleDateFormat sdf = new SimpleDateFormat("dd_MM_yyyy-HH:mm:ss");
936 String fileName = sdf.format(Calendar.getInstance().getTime());
937 String [] data = StartFrame.deserializeaza();
938 int code;
939 Process p = null;
940 if (data[1] == "") {
941     p = Runtime.getRuntime().exec("./db/mysqldump -u" + data[0] +
942         " ecoaventuradb -r " + f.getCanonicalPath() + "\\\" + fileName + ".sql");
943 } else {
944     p = Runtime.getRuntime().exec("./db/mysqldump -u" + data[0] +
945         " -p" + data[1] + " ecoaventuradb -r " + f.getCanonicalPath() + "\\\" + fileName + ".sql");
946 }
947 code = p.waitFor();
948
949 if (code != 0) {
950     BufferedReader reader = new BufferedReader(new InputStreamReader(p.getErrorStream()));
951     throw new Exception (reader.readLine());
952 }
953 }
```

## Descărcarea unui tabel într-un fișier de format .excel

Secvența de mai jos surprinde formatarea unei celule a unui tabel Excel, specificându-i: valoarea, stilul, tipul de dată și formula.

```
592
593 int i = 1;
594 for (EchipaDB e : punctaje.keySet()){
595     cellId = 0;
596     row = ws.createRow(rowId++);
597     Cell count = row.createCell(cellId++);
598
599     count.setCellValue(i);
600     count.setCellStyle(numericBorderedStyle);
601
602     Cell team = row.createCell(cellId++);
603     team.setCellValue(e.toString());
604     team.setCellStyle(numericBorderedStyle);
605
606     for (int j = 0; j < matrix[i-1].length; ++j) {
607         Cell cell = row.createCell(cellId++);
608         cell.setCellValue(matrix[i-1][j]);
609
610         if (j == matrix[i-1].length - 1) {
611             cell.setCellStyle(numericBorderedStyle);
612         } else {
613             cell.setCellStyle(numericStyle);
614         }
615     }
616
617     Cell total = row.createCell(cellId++);
618     total.setCellType(CellType.FORMULA);
619     total.setCellFormula("SUM(" + columns[2] + (rowId) + ":" + columns[punctaje.get(e).size()+1] + (rowId) + ")");
620     total.setCellStyle(style);
621
622     ++i;
623 }
```

## Arhivarea aplicației într-un fișier de format .zip

Anticipând necesitatea de a muta ușor și rapid aplicația de pe un calculator pe altul, a fost implementată procedura de arhivare, care creează o arhivă cu toate fișierele aplicației din momentul actual.

În codul lipit mai jos, s-a implementat o coadă în care se află fișierele care trebuie să fie arhivate.

Folosind tabloul tipul byte "buffer", citim o secvență de bytes din fișierul care trebuie arhivat, și o scriem în fișierul .zip care se crează, în dreptul obiectului reprezentat de variabila ze, de tip ZipEntry.

```
81     fos = new FileOutputStream(outPath);
82     zos = new ZipOutputStream(fos);
83
84     System.out.println("Start zipping to : " + outPath);
85
86     for (File f : queue) {
87         System.out.println("Adding : " + f.getName());
88         frame.updateList("Se adauga : " + f.getName());
89         ZipEntry ze = new ZipEntry(f.getCanonicalPath().substring(source.length
90         zos.putNextEntry(ze);
91
92         try {
93             in = new FileInputStream(f);
94             int len;
95             while ((len = in.read(buffer)) > 0) {
96                 zos.write(buffer, 0, len);
97                 actualSize += len;
98                 frame.updateBar((int) (actualSize * 100 / totalSize));
99             }
100         } catch (Exception e) {}
101         finally {
102             in.close();
103         }
104         zos.closeEntry();
105     }
106
107     zos.close();
108     System.out.println("Done");
```

## Implementarea "Observer Pattern"-ului

Acest model care stă la baza programării orientată pe obiect a fost implementat în aplicație pe partea de Server pentru a notifica o listă de "ascultători" la momentul primirii unui nou pachet dedate.

Modelul este alcătuit din două interfețe principale: DataReceiver și DataListener.

```
14 public interface DataListener {
    public void update(Object... objs);
}
```

Această interfață este implementată de orice clasă care trebuie notificată la sosirea unui pachet de date. Implementarea metodei "update()", care primește ca parametru pachetul de date, trebuie să fie, în mod obligatoriu, suprascrisă de orice clasă care implementează aceasta interfață.

```

15 public interface DataReceiver {
16     public void addListener (DataListener dataListener);
17     public void removeListener (DataListener dataListener);
18     public void notifyListeners(Object... objs);
19 }

```

Interfața DataReceiver este cea care gestionează "ascultătorii". Această clasă este prima notificată la sosirea unui nou pachet de date, prin apelul metodei "notifyListeners()", prin care transmite informația mai departe.

Implementarea acestei clase este următoarea :

```

15 public class Receiver implements DataReceiver, Serializable {
16     private ArrayList<DataListener> listeners;
17
18     public Receiver () {
19         listeners = new ArrayList<>();
20     }
21
22     @Override
23     public void addListener(DataListener dataListener) {
24         listeners.add(dataListener);
25     }
26
27     @Override
28     public void removeListener(DataListener dataListener) {
29         listeners.remove(dataListener);
30     }
31
32     @Override
33     public void notifyListeners(Object... objs) {
34         for (DataListener dl : listeners) {
35             dl.update(objs);
36         }
37     }
38 }
39

```

## Implementarea "Singleton"-ului

Există situații în care avem nevoie de o instanță a unei clase care să fie unică în program, și accesibilă de oricunde. Acea clasă poartă denumirea de "Singleton", și se referă la unica instanță a acesteia. În cazul de față, avem clasa "ControllerDB", care este managerul obiectelor de acces la baza de date (DAO).

```

102 private ControllerDB() { }

```

Se suprascrie constructorul default cu unul privat.

```

132 public static ControllerDB getInstance() {
133     return ControllerDBHolder.INSTANCE;
134 }
135
136 private static class ControllerDBHolder {
137     private static final ControllerDB INSTANCE = new ControllerDB();
138 }

```

Se declară o variabilă statică ce va "ține" unica instanță, și o metodă statică și publică, ce va returna unica instanță a clasei.

## FOLOSIREA RESPONSABILĂ A APLICAȚIEI

În vederea unei experiențe cât mai plăcute, se recomandă a nu se modifica fișierele aplicației, sau a codului sursă. Este indicat ca utilizatorul să nu steargă niciun fișier asociat programului, în afara celor create chiar de el prin intermediul acestuia. În cazul lipsei unei componente, aplicația poate să aibă un comportament anormal, sau chiar să devină invalidă. În cazul unui comportament suspicios, apălați la asistența tehnică dedicată.

MULȚUMIM PENTRU ALEGEREA FĂCUTĂ!

# Suport tehnic

- Tel. : 0730072214
- Email : alexandru.top98@gmail.com

