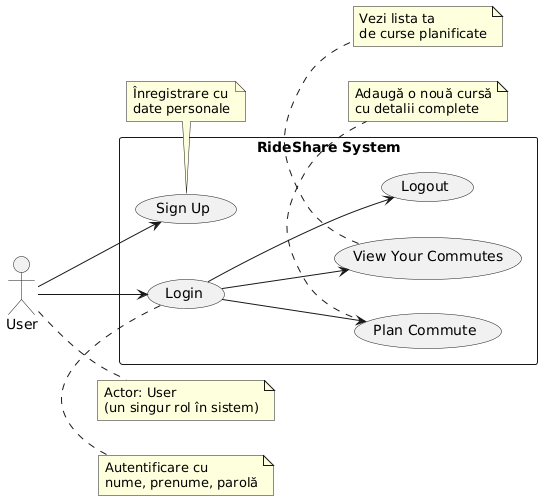
**Documentație Proiect RideShare**

**Moldovan Nicolae Florentin**

**1. Descrierea proiectului**

Proiectul RideShare este o aplicație pentru partajarea de curse între utilizatori, constând dintr-un server scris în C# și o aplicație mobilă pentru Android dezvoltată în Jav pa. Aplicația permite utilizatorilor să își planifice curse, să vizualizeze cursele existente și să gestioneze detaliile acestora.

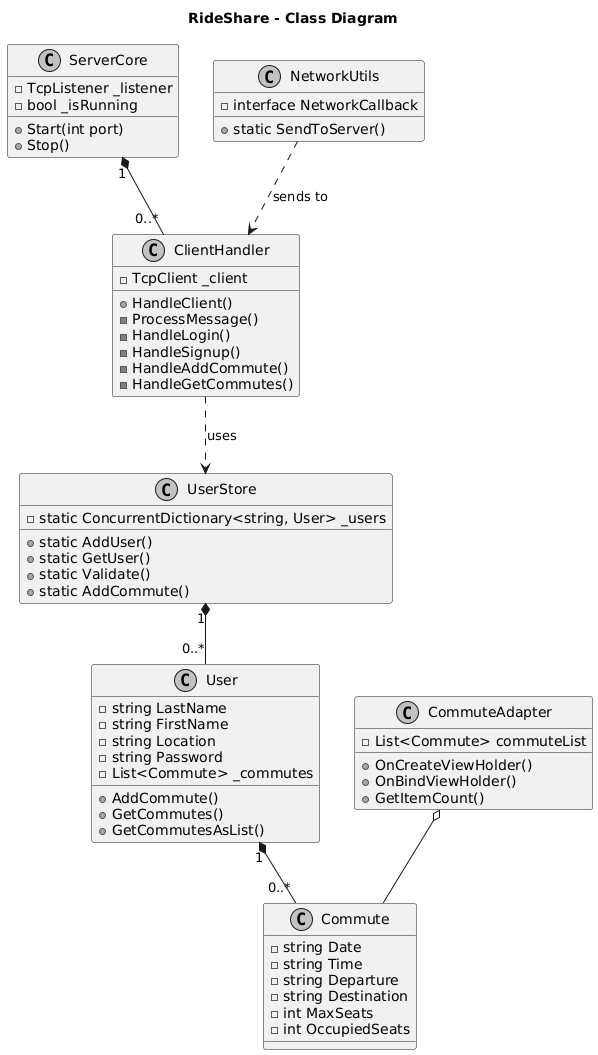
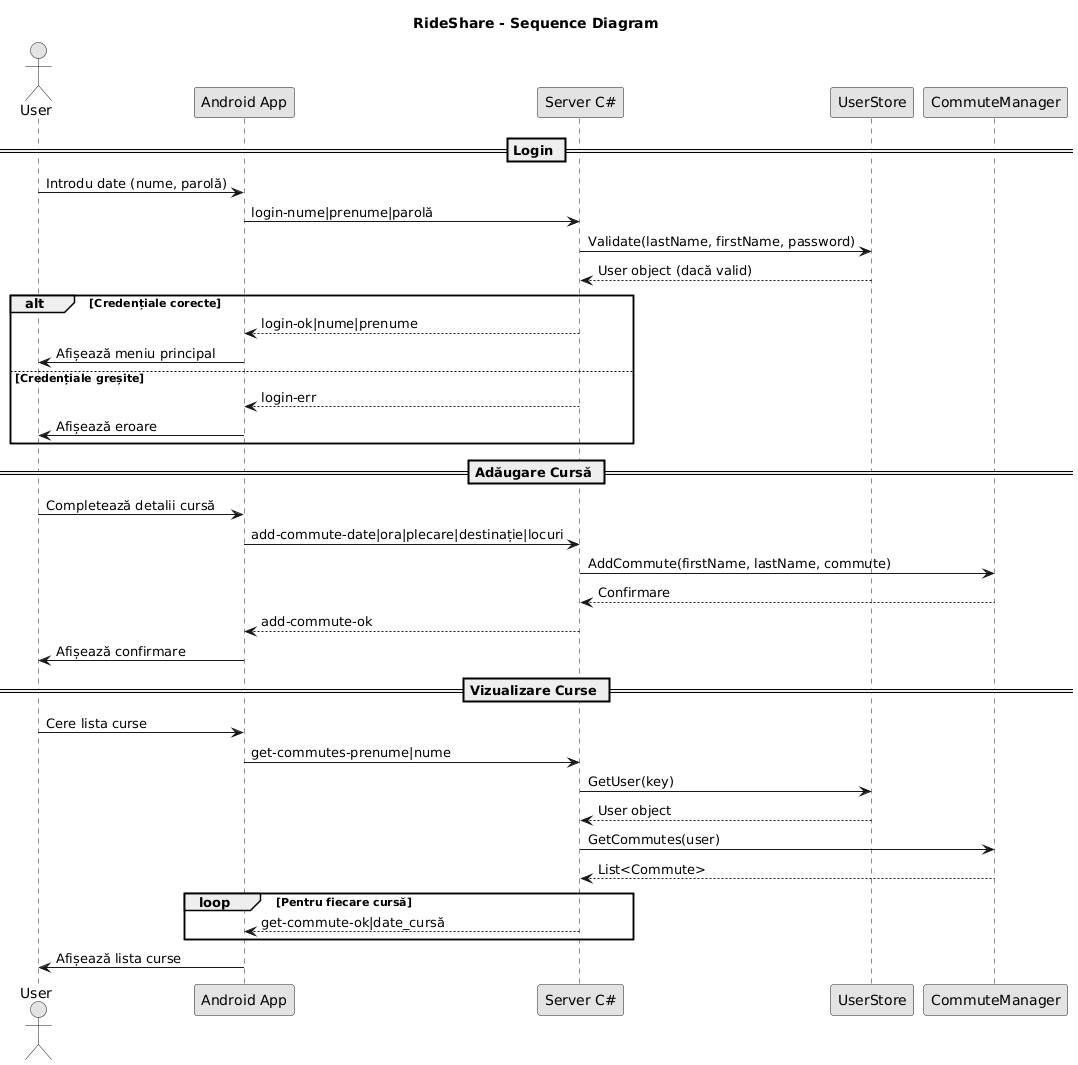
Scopul principal al aplicației este să faciliteze organizarea și partajarea de transporturi între persoane care călătoresc pe aceeași rută, reducând astfel costurile și impactul asupra mediului.

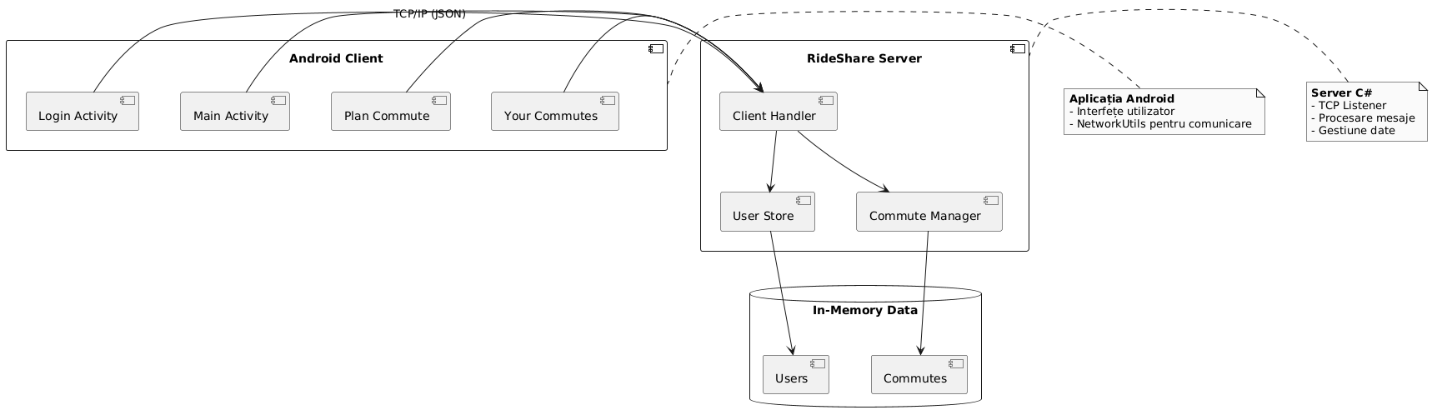
**2. Tehnologii și instrumente folosite**

**Platforme și limbaje:**

* **Server**: C# (.NET)
* **Client**: Java (Android)

**Framework-uri și biblioteci:**

* **Server**:
  + System.Net pentru networking
  + System.Net.Sockets pentru conexiuni TCP
  + System.Collections.Concurrent pentru gestionarea thread-safe a datelor
* **Client**:
  + Android SDK pentru dezvoltarea aplicației mobile
  + RecyclerView pentru afișarea listelor de curse
  + SharedPreferences pentru stocarea locală a datelor utilizatorului

****

**Instrumente de dezvoltare:**

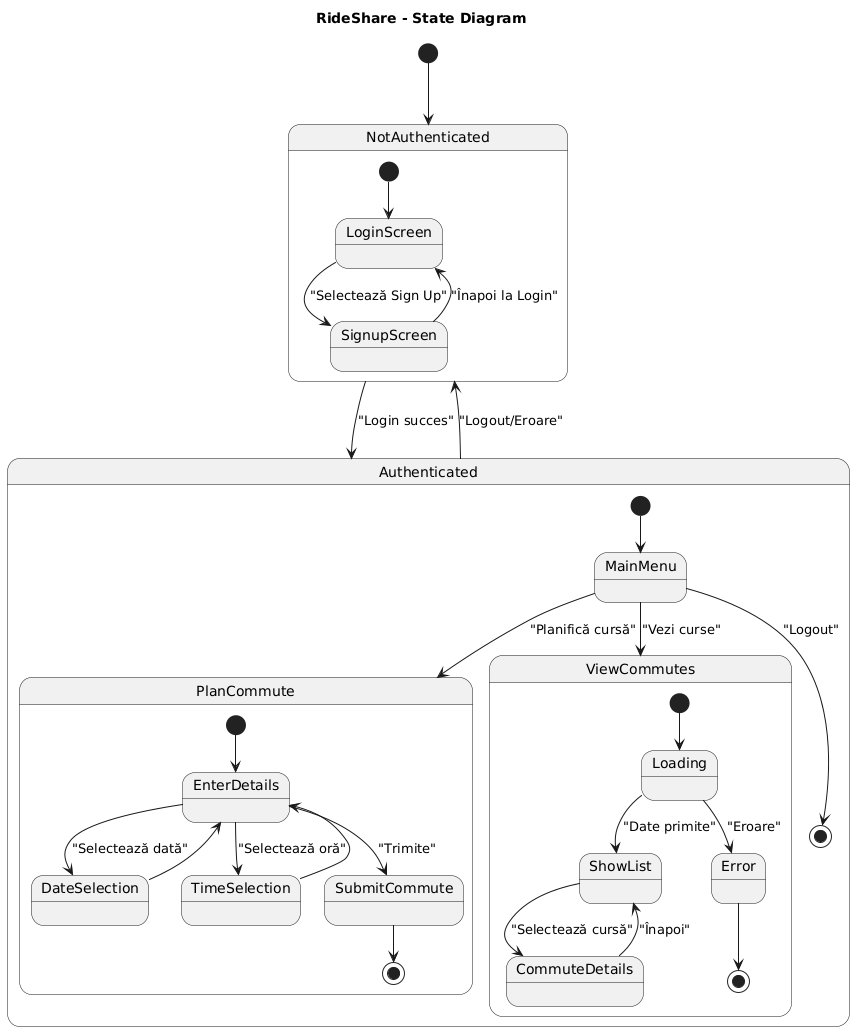
* Visual Studio pentru server
* Android Studio pentru client
* Emulator Android pentru testare

**Protocoale:**

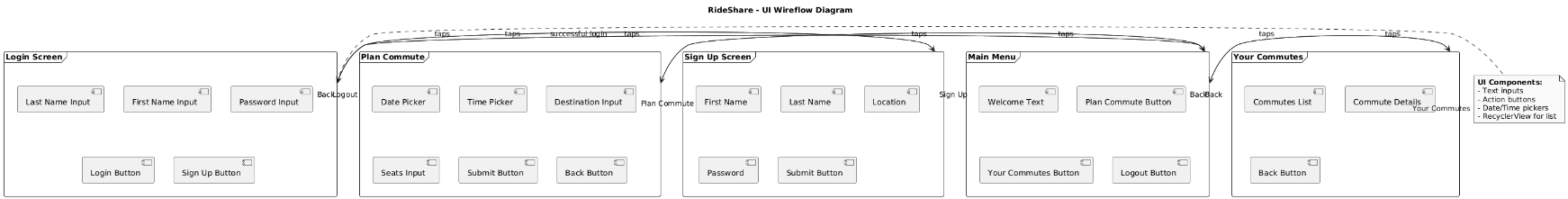
* TCP/IP pentru comunicarea între client și server
* Format de mesaje text personalizat pentru schimbul de date

**3. Structura proiectului**

**Server:**

* **ServerCore.cs**: Gestionează pornirea și oprirea serverului, precum și acceptarea conexiunilor de la clienți
* **ClientHandler.cs**: Procesează mesajele primite de la clienți și răspunde în consecință
* **UserStore.cs**: Stochează și gestionează utilizatorii și cursele acestora
* **User.cs**: Reprezintă un utilizator și metodele asociate
* **Commute.cs**: Model pentru o cursă
* **Program.cs**: Punctul de intrare al aplicației server

**Client (Android):**

* **LoginActivity**: Interfața de autentificare
* **SignupActivity**: Interfața de înregistrare a unui nou utilizator
* **MainActivity**: Meniul principal după autentificare
* **PlanCommuteActivity**: Interfața pentru planificarea unei curse noi
* **YourCommutesActivity**: Interfața pentru vizualizarea curselor existente
* **Commute**: Model pentru o cursă
* **CommuteAdapter**: Adaptor pentru afișarea curselor în RecyclerView
* **NetworkUtils**: Utilitar pentru comunicarea cu serverul

**4. Funcționalități**

**Autentificare și înregistrare:**

* Utilizatorii existent se pot autentifica cu nume, prenume și parolă
* Utilizatorii noi se pot înregistra furnizând nume, prenume, locație și parolă

**Gestiunea curselor:**

* Adăugare de curse noi cu detalii precum dată, oră, destinație, locuri disponibile etc.
* Vizualizarea tuturor curselor planificate de utilizator
* Afișarea detaliată a fiecărei curse (dată, oră, punct de plecare, destinație, locuri)

**Comunicare client-server:**

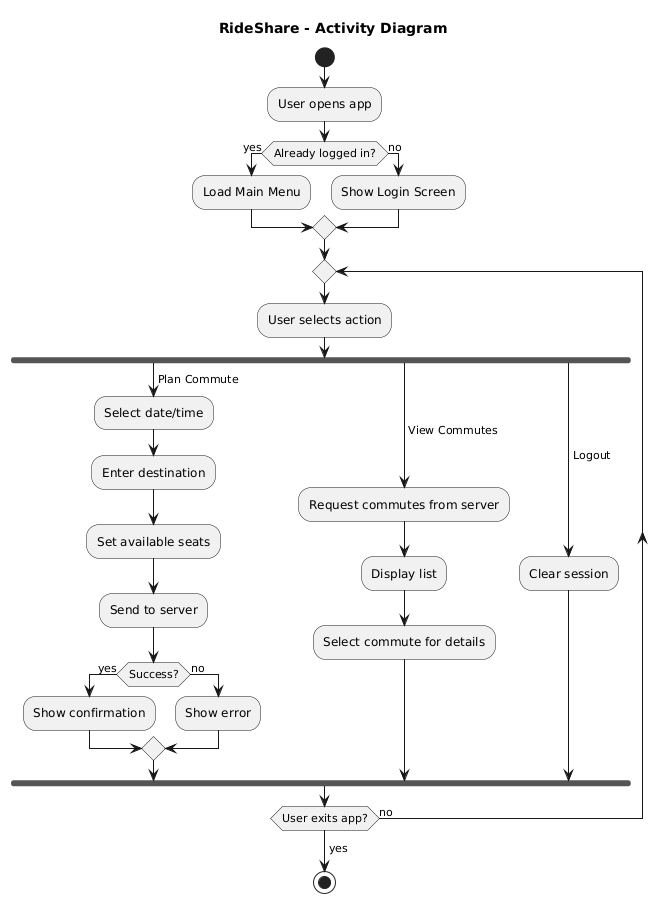
* Sistem de mesaje text pentru toate operațiile (autentificare, înregistrare, gestionare curse)
* Conexiuni TCP persistente pentru fiecare operație

**5. Server**

**Arhitectură:**

Serverul utilizează o arhitectură clasică client-server cu următoarele caracteristici:

* Ascultă pe un port specificat pentru conexiuni TCP
* Creează un fir de execuție separat pentru fiecare client conectat
* Utilizează un dicționar thread-safe (ConcurrentDictionary) pentru stocarea utilizatorilor



**Fluxuri de procesare:**

1. **Pornire server**:
   * Inițializează TCP listener pe portul specificat
   * Așteaptă conexiuni în bucla principală
2. **Gestionare client**:
   * Pentru fiecare client, creează un ClientHandler
   * Procesează mesajele primite și trimite răspunsuri
3. **Operații suportate**:
   * Autentificare (login)
   * Înregistrare (signup)
   * Adăugare cursă (add-commute)
   * Obținere curse (get-commutes)

**Stocare date:**

* Datele sunt stocate în memorie (nu există persistare pe disc)
* Utilizatorii sunt stocați într-un ConcurrentDictionary
* Fiecare utilizator are o listă de curse asociate

**Mesaje server-client:**

Serverul utilizează un format simplu de mesaje text cu următoarea structură:  
[tip-operatie]-[status]|[detalii]

Exemple:

* login-ok|Moldovan|Florentin|test123
* add-commute-err|Invalid seat numbers
* get-commute-ok|2025-03-25|08:00|Poienita|Targu-Mures|4|0

**Securitate:**

* Parolele sunt stocate în clar (necriptate)
* Nu există autentificare pe bază de token - sesiunile nu sunt persistente
* Comunicarea este în clar (fără criptare) - necesită HTTPS pentru mediul de producție

**Limitări:**

* Scalabilitate limitată (toate datele sunt în memorie)
* Nu există backup pentru date
* Nu există gestionare a concurenței pentru operații complexe
* Nu există mecanisme de recuperare a erorilor

Acest server oferă funcționalități de bază pentru o aplicație RideShare, dar ar necesita îmbunătățiri semnificative pentru a fi utilizat într-un mediu de producție.