Fangli Andrei
grupa 231
5 mai 2014

Testarea programelor

Enunțul problemei

O instituție teatrală pune la dispoziția spectatorilor un sistem pentru rezervarea locurilor. În fiecare zi instituția are un singur spectacol iar spectatorii pot rezerva locuri urmând să plătească atunci când Se prezintă la spectacol. Locurile în sala sunt organizate în stal, loje și balcon. Stalul și balconul sunt organizate pe rânduri, fiecare rând are mai multe locuri, iar lojele sunt organizate doar în locuri datorită numărului redus al acestora. Sistemul este compus din tipuri de aplicații: server și client. Aplicația server deține responsabilitatea de a reține locațiile în sala de spectacol, spectacolele repartizate pe zile, pentru fiecare spectacol un registru cu locurile rezervate și de cine au fost rezervate. Aplicația client este responsabilă de a facilita informații utilizatorilor referitoare la spectacole (ce spectacol rulează și în ce dată) și la starea locurilor rezervate cu opțiunea de a rezerva unul sau mai multe bilete la unul sau mai multe spectacole pe bază de nume, adresă de e-mail și telefon (pentru a putea anunța spectatorii în cazul în care apar modificări). În urma unei rezervări toate aplicațiile client care rulau deja se actualizează.

Descriere generală a soluției și unit testing

Aplicația în mare este un exemplu de comunicare prin TCP/IP funcționalitatea fiind restrânsă la posibilitatea de a rezerva locuri și a adăuga sau șterge spectacole, iar terminalele clienților trebuie să fie în orice moment actualizate. Din acest motiv modulele care trebuie testate sunt într-un număr restrâns deoarece cea mai mare problema este cea de serializarea respectiv deserializarea datelor.

Aplicația modelează două entități: Spectacolul (Show) și Rezervarea la spectacol (ShowReservation). Pentru fiecare dintre entități se concepe câte un modul de testare pentru a verifica crearea corectă a instanțelor prevăzute.

Pentru serializare/deserializare este definită o interfață (INetworkSerializer<TItem>) cu două metode: Serialize(item : TItem) : string și Deserialize(serializedValue : string) : TItem. Orice tip care implementează această interfață are responsabilitatea de a serializa instanțe de tipul TItem și a le deserializa (funcția Deserialize este inversa funcției Serialize), aceasta ajută la transmiterea datelor peste Internet deoarece se folosesc instanțe de aceleași tipuri (care realizează INetworkSerializer) atât la client cât și la server pentru serializare respectiv deserializate iar logica pentru cele două operații este definit in cadrul aceleași definiți al unui tip (pentru a minimiza șansa de eroare) iar prin unit testing cel care scrie testele este forțat să verifice atât Serialize cât si Deserialize.

Pentru aplicația descrisă mai sus se realizează trei implementări: ShowSerializer, ShowReservationSerializer (pentru a putea trimite între server și client instanțe de tipul Show respectiv ShowReservation) și CommandNetworkSerializer.

CommandNetworkSerializer serializează o comandă (cu nume și listă de parametrii actuali), reprezintă o cerere de la un terminal la server (spre exemplu cererea de a rezerva 10 locuri la balcon) sau o actualizare trimisă de la server la un terminal (de exemple cum ca 10 locuri de la balcon au fost rezervate). Numele este folosit doar pentru a face distincție între cereri/actualizări. Parametrii actuali sunt toți de același tip care necesită o instanță de tipul INetworkSerializer pentru a putea serializa și deserializa valorile parametrilor actuali. Având în vedere că se vor trimite liste de instanțe de tipul ShowReservation această implementare se bazează pe alte componente ca fiind implementate corect.

Testarea de integrare

Având în vedere dimensiunea aplicației testarea de integrare se realizează pe modelul big-bang. Numărul de funcționalități este mic și relativ dependente (adăugarea de spectacole nu implică adăugarea de rezervări, dar adăugarea de rezervări depinde de adăugarea de spectacole) ceea ce face localizarea unei posibile erori destul de rapidă. De exemplu când un utilizator încearcă să rezerve un număr de locuri iar aplicația se închide neașteptat cauza poate fi conexiunea cu serverul a cedat (ceea ce poate fi depistat repede prin verificarea conexiunii la Internet), serializarea rezervărilor nu funcționează corespunzător sau serializarea comenzii nu se funcționează corespunzător.

Testarea prin incrementare nu este accesibilă deoarece dacă mai întâi se testează gestionarea spectacolelor atunci testarea acelui modul rămâne parțial completă deoarece în momentul în care se pot adăuga rezervări trebuie iar revenit deoarece ștergerea unui spectacol poate implica (daca este programat pentru ziua curentă sau una precedentă) ștergerea de rezervări. Iar ștergerea unui spectacol din ziua curentă implică împiedicarea realizării unei rezervări pentru acea zi (având în vedere faptul ca pot fi clienți conectați care în acel moment își rezervă un bilet).