**Scenarii Initiale de Testare**

**Alexandru Panainte**

**User pierde conexiunea la confirmarea alegerii unui tutore**

In cazul in care un elev intra in aplicatie, selecteaza un profesor, materia despre care vrea sa faca licenta, confirma cererea iar in stadiul final conexiunea cu aplicatia cade se pot intampla trei lucruri.

1. Cererea a fost trimisa de elev, a fost primita de profesor. (Caz ok)
2. Cererea nu a fost inregistrata ca trimisa de elev, nu a fost primita de profesor. (Caz ok)
3. Cererea a fost inregistrata ca trimisa de elev, nu a fost primita de profesor. (Caz gresit)

Modalitate test:

1. Elev acceseaza pagina aplicatiei.
2. Elevul intra in contul sau.
3. Elevul trimite cererea catre profesor(in acest moment este intrerupta brusc conexiunea la aplicatie).

3.1. Cererea nu a fost trimisa de elev si nu a fost primita de profesor.(Test trecut)

3.2. Cererea a fost trimisa de elev si a fost primita de profesor.(Test trecut)

3.3. Cererea a fost trimisa de eleve dar nu a fost primita de profesor.(Test esuat)

4. Incheiere test.

**Posibila problema:** componenta front end a inregistrat si afisat confirmarea actiunii, dar datorita pierderii conexiunii nu a ajuns la back end confirmarea.

**User pierde conexiunea in timpul transmiterii unui mesaj**

Un user poate alege sa transmita un mesaj altui user iar in momentul in care acesta trimite mesajul, conexiunea cu serverul va cadea. In acest scenariu pot aparea trei cazuri.

1. Mesajul este trimis de primul user si primit de al doilea.
2. Mesajul nu este trimis de primul user si nu este primit de al doilea.
3. Mesajul apare ca trimis de primul user dar nu a fost primit de al doilea user.

Modalitate test:

1. User-ul acceseaza pagina aplicatiei.
2. User-ul selecteaza alt user caruia vrea sa ii trimita un mesaj
3. User-ul completeaza campurile necesare si trimite mesajul(in acest moment este intrerupta brusc conexiunea la aplicatie).
   1. Mesajul nu a fost trimis de primul user si nu a fost primit de al doilea user.(Test trecut)
   2. Mesajul a fost trimis de primul user si a fost primit de al doilea user.(Test trecut)
   3. Mesajul a fost trimis de primul user dar nu a fost primit de al doilea user.(Test esuat)
4. Incheiere test.

**Posibila problema:** Utilizatorul apasa trimitere mesaj, dar conexiunea se pierde inainte ca mesajul sa fie salvat in baza de date, iar utilizatorului I se afiseaza confirmarea trimiterii mesajului in front end.

**User pierde conexiunea cu aplicatia dupa ce a apasat butonul de logare**

Un user poate intra pe aplicatie urmand sa isi introduca datele iar in momentul in care apasa pe butonul de logare conexiunea cu serverul va cadea. In acest scenariu sunt doua cazuri.

1. User-ul va intra din nou in aplicatie, isi va introduce datele si va apasa butonul de logare iar conectarea va avea success.
2. User-ul va intra din nou in aplicatie, isi va introduce datele si va apasa butonul de logare iar conectarea va esuat, aparand o eroare.(Useru-ul apare deja conectat pe aplicatie)

Modalitate test:

1. User-ul acceseaza pagina aplicatiei.
2. User-ul introduce datele de logare si apasa pe butonul de login.(In acest moment este intrerupta brusc conexiunea la aplicatie)
   1. Cand user-ul va reintra pe aplicatie si se va autentifica, login-ul va avea success. (Test trecut).
   2. Cand user-ul va reintra pe aplicatie si se va autentifica, login-ul nu va avea success si va aparea o eroare precum el este deja autentificat. (Test picat).
3. Incheiere test.

**Posibila problema:** Dupa trimiterea informatiilor la back end utilizatorul va fi inregistrat ca logat, intre timp acesta va fi inchis componenta client a aplicatiei si o va fi repornit cu intentia de a se reloga, dar aparand logat deja, nu va putea realiza operatiunea.

**Trimiterea fragmentata a mesajelor din cauza unei conexiuni deficiente**

In momentul in care conexiunea este un fragmentata este posibil ca atunci cand incercam sa trimitem un mesaj sa se inregistreze numai anumite parti din acesta(in momentele cand conexiunea este stabila)

Modalitate test:

1. User-ul acceseaza pagina aplicatiei.
2. Acceseaza sectiunea de mesaje si incepe scrierea unui mesaj catre alt utilizator. In acest moment vom scadea drastic viteza internetului pana la limita de a nu fi scosi de pe server.
3. User-ul va trimite mesajul in aceste conditii.
   1. Mesajul va fi trimis integral .(Test trecut)
   2. Mesajul va fi trimis fragmentat.(Test picat)
4. Incheiere test.

**Posibila problema:** Se trimite mesajul, dar conexiunea are variatii foarte mari, incluzand intreruperi intre client si server, la back end nu se confirma dimensiunea mesajului inainte de salvare, astfel inregistranduse in BD mesajul fragmentat.

**Cretu Bogdan**

**Utilizatorul poate introduce la inregistrare nume si parola care sa depaseasca bufferul de transmisie sau limita maxima din baza de date a aplicatiei,**

Etape test:

1. Accesare pagina aplicatiei,
2. Selectare Inregistrare
3. Introducere nume, parola sau nume si parola prea lungi

3.a Inregistrare refuzata cu mesaj corespunzator (test trecut)

3.b Inregistrare acceptata sau neprimire eroare la utilizator (test esuat)

4. Incheiere test.

**Posibila problema:** La verificarea parolei si a numelui de utilizator nu se verifica intr-una dintre componentele aplicatiei dimenisunea inputului utilizatorului, cauzand astfel transmisia de date invalide.

**Utilizatorul Incearca sa trimita aplicatiei comenzi nepermise clasei sale (student/profesor/tutore) prin intermediul modificarilor in adresa URL.**

Etape test

1. Logare in aplicatie
2. Copiere Url dupa executarea unei comenzi oarecare
3. Modificare si accesare URL

3.a Aplicatia returneaza eroare (test trecut)

3.b Aplicatia accpta comanda (test esuat)

1. Incheiere test

**Posibila problema:** Componenta back-end nu verifica permisiunile unui utilizator daca acestea sunt verificate in front end sau exista verificari omise.

**George Burghelea**

**Se pot trimite mesaje fara subiect si/sau continut:**

Utilizatorul X doreste sa trimita un mesaj utilizatorului Y. Din greseala acesta apasa butonul de expediere fara a complete spatiile obligatorii de subiect si continut.

Modalitate de testare:

1. Utilizatorul acceseaza sectiune de mesagerie
2. Utilizatorul isi alege un destinatar
3. Utilizatorul apasa butonul de expediare fara a completa campurile subiect/continut.
   1. Mesajul nu se expediaza. Se returneaza eroare campuri incomplete (test trecut)
   2. Mesajul se expediaza necompletat (test esuat)
4. Incheiere test

**Posibila problema:** La trimiterea unu mesaj nu se verifica corepsunzator in componenta back-end continutul mesajului. La componenta front-end se poate deasemenea face verificarea pentru evitarea trimiterii de date invalide.

**Aplicatia poate trimite mesaje la destinatari inexistenți**

Utilizatorul X doreste sa trimita un mesaj unui alt utilizator. In campul rezervat destinatarului, utilizatorul x, introduce un numele/o adresa inexistent(a).

Modalitate de testare:

1. Utilizatorul acceseaza sectiune de mesagerie
2. Utilizatorul isi alege un destinatar inexistent
3. Dupa completarea mesajului utilizatorul apasa butonul de expediere
   1. Mesajul nu se expediaza. Se returneaza eroarea utilizator inexistent (test trecut)
   2. Mesajul se expediaza catre un utilizator inexistent (test esuat)
4. Incheiere test

**Posibila problema:** La trimiterea unui mesaj in componenta back-end, nu este verificat destinatarul mesajului inainte de inserarea in baza de date.

**Se pot scrie mesaje care depasesc lungimea maxima de caractere**

Utilizatorul X doreste sa trimita un mesaj altui utilizator. Continutul mesajului compus de acesta depaseste lungimea maxima de caractere acceptata de aplicatie.

Modalitate de testare:

1. Utilizatorul acceseaza sectiune de mesagerie
2. Utilizatorul isi alege un destinatar
3. Utilizatorul compune un mesaj cu o lungime ce o depaseste pe cea maxim admisa de aplicatie
   1. Utilizatorul nu mai poate introduce caractere noi in mesaj dupa ce lungimea maxima a fost atinsa (test trecut)
   2. Utilizatorul nu poate expedia mesajul. Se returneaza eroarea lungime maxima depasita(test trecut)
   3. Mesajul se expediaza chiar daca depaseste lungimea maxima admisa (test esuat)
4. Incheiere test

**Posibila problema:** In componenta front-end nu exista limitare asupra numarului de caractere care poate fi scris in casuta de mesaj. In componenat back-end nu se verifica dimensiunea mesajului inainte de confirmarea primirii acestuia.

**Utilizatorii trimit fisiere prea mari iar continutul se pierde**

Utilizatorul doreste sa expedieze un mesaj ce are atasat un fisier media/txt. Dimensiunea fisierului o depaseste pe cea maxim admisa de aplicatie

Modalitate de testare:

1. Utilizatorul acceseaza sectiune de mesagerie
2. Utilizatorul isi alege un destinatar
3. Utilizatorul incearca sa ataseze un fisier care depaseste dimensiunea admisa
   1. Fisierul nu poate fi incarcat. Se returneaza eroarea dimensiune fisier prea mare (test trecut)
   2. Fisierul se incarca, dar mesaj nu poate fi expediat. Se returneaza eroarea dimensiune fisier prea mare (test trecut)
   3. Fisierul se incarca, iar mesajul se expediaza ( test esuat)
4. Incheiere test

**Posibila problema:** La trimiterea unui fisier, datorita latimii de banda mici sau a conexiunii instabile fisierul trimis este distorsionat cand ajunge la componenta back-end, iar aceasta nu face o verificare asupra integritatii fiserului inainte de salvarea acestuia.

**Codrin Donciu**

**Studentul se poate autentifica cu drepturi de Profesor sau vice-versa;**

Un user cu un cont creat de student se autentifica, folosind user-ul si parola sa, insa Baza de Date, ori la crearea contului,

ori in urma unor interogari efectuate eronat, asociaza drepturi de Profesor si/sau Tutore acestui user. Totdata, un user care ar

trebui sa aiba drepturi de Profesor si/sau Tutore, la autentificare, primeste drepturi de Student si functionalitatile asociate.

Proces de testare:

1. Login cu datele contului preluate din cache/introduse la deschiderea aplicatiei;

2. Se face apel la baza de date, se verifica data datele introduse exista:

A. Datele nu exista, login-ul se face (test esuat);

B. Datele exista, login-ul nu se face (test esuat);

C. Datele exista, login-ul se face (test trecut);

D. Datele nu exista, login-ul nu se face (test trecut);

3. Baza de date raspunde la cererea user-ului:

A. Baza de date returneaza corect drepturile (test trecut);

B. Baza de date returneaza un alt drept decat cel asociat (test esuat);

**Posibila problema:** In urma autentificarii nu se verifica statusul utilizatorului in componenta back-end. Prin manipularea fornt-end-ului utilizatorul poate acesa si rula comenzi specifice doar altei clase decat a sa (student – profesor de exemplu).

**Profesor poate sa aleaga si el un Profesor (+ are un tutore asociat)**

E o consecinta a bug-ului anterior. Profesor primeste optiunea de a alege un Profesor pentru licenta; desi e contraintuitiv pentru un user cu drept de Profesor sa

isi aleaga un coordonator, daca aceasta operatie este disponibila din interfata livrata la Profesor, avand in vedere ca aceasta

clasa nu are atribute si metode specifice functionalitatii, mutiple erori pot aparea si la nivel de aplicatie, si in baza de date.

Proces de testare:

1. Profesor acceseaza interfata livrata de aplicatie;

2. Profesor are posibilitatea de a face mai multe operatiuni;

3. Se incearca alegerea unui profesor coordonator:

A. Cererea este respinsa (test trecut);

B. Profesor primeste un Profesor coordonator (test esuat);

4. Se verifica asocierea cu un Tutore:

A. Profesor are asociat un Tutore (test esuat);

B. user-ului de tip Profesor nu i se poate asocia un Tutore;

(5. (daca e si Tutore) Se incearca a deveni Tutore pentru un Profesor - fail/succes)

**Posibila problema:** La nivel de back-end nu se verifica statutul unui utilizator la executarea de comenzi specifice.

**Profesorul accepta cereri dupa limita admisa;**

Profesor isi poate seta o limita pentru cate lucrari de licenta poate coordona. Un profesor a acceptat deja lucrari de licenta si s-a ajuns la limita,

insa la primirea de noi cereri, inca le poate accepta, iar acestea sunt stocate si afisate ca orice alte lucrari de licenta deja acceptate.

Proces de testare:

1. Mai multi Studenti trimit cereri catre Profesor;

2. Profesorul accepta/respinge cereri;

3. Exista o variabila Counter care retine cate astfel de cereri au fost acceptate, atribut a clasei Profesor;

4. Counter = Limit;

5. Cereri inca sunt trimise;

A. Cereri inca sunt acceptate (test esuat);

B. Cererile sunt respinse (test trecut) --> studentii nu mai pot trimite cereri - butonul de request poate fi greyed out la ei pe interfata, de ex;

**Posibila problema:** La nivel de back-end nu se verifica in timp real numarul de

Studenti de la un profesor inainte de a i se trimite noi cereri.

**Exista tutori/profesori multipli pentru un singur student:**

Un student poate avea un singur Profesor coordonator si un singur Tutore asociat. In mod normal, la crearea contului/autentificare, Studentul primeste automat un Tutore,

si nu are niciun Profesor coordonator asociat. Dupa ce isi alege un profesor coordonator, studentul nu mai poate alege un profesor coordonator.

Proces de testare:

1. Studentul alege un profesor coordonator si ii este acceptata cererea;

2. Studentul mai alege un profesor coordonator:

A. Cererea noua este acceptata (test esuat);

B. Cererea noua este respinsa (test trecut);

**Posibila problema:** La nivel de back-end nu este contorizat si verifcat numarul de relatii de tutoriat pentru un student, astfel putand fi inserate in basa de date campuri cu tutori diferiti pentru acelasi student.

**Sabina Petrisor**

**Utilizator se conecteaza introducand cod sql la nume/parola**

Din neatentie/intentii gresite, utilizatorul introduce, la autentificare, ca nume sau parola o secventa de cod.

1 - (Test) Utilizator acceseaza pagina aplicatiei

2 - (Test) Utilizator isi scrie numele si parola

3 - (Test) Utilizator trimite informatiile

3.A - Aplicatia returneaza eroare, nume/parola gresite (test reusit)

3.B - Se accepta numele/parola introduse, cu ignorare sau rularea codului (test esuat)

4 - Incheierea testului

**Posibila problema:** La nivel de back end nu se face verificare impotriva injectie de cod de orice tip la crearea unui cont sau la logarea unui utilizator.

**Utilizator schimba parola folosind parola anterioara**

Dupa ce se conecteaza, un utilizator doreste sa isi schimbe parola. Acesta isi introduce din greseala vechea parola.

1 - (Test) Utilizator acceseaza pagina aplicatiei

2 - (Test) Utilizator se conecteaza la propriul cont

3 - (Test) Utilizator acceseaza schimbarea parolei

4 - (Test) Utilizator introduce noua parola, asemanatoare cu cea anterioara

5 - (Test) Utilizator trimite informatiile

5.A - Aplicatia returneaza eroare, parola nou introdusa a mai fost introdusa anterior (test reusit)

5.B - Se accepta noua parola introdusa, aceasta fiind din nou adaugata in baza de date (test esuat)

6 - Incheierea testului

**Posibila problema:** La nivel de back end nu se face verificarea pentru a diferentia noua parola de cea veche.

**Utilizator creaza cont cu numele vid / din spatii/ caractere non alfanumerice**

Un nou utilizator doreste sa isi creeze cont. La numele de utilizator, acesta nu introduce nimic, doar spatii sau caractere non alfanumerice, fiindca acesta nu a fost instiintat asupra restrictiilor la crearea unui nume de utilizator.

1 - (Test) Utilizator acceseaza pagina aplicatiei

2 - (Test) Utilizator acceseaza pagina pentru creare cont

3 - (Test) Utilizator introduce ca nume: nimic, spatii sau caractere non alfanumerice

4 - (Test) Utilizator trimite informatiile

4.A - Aplicatia returneaza eroare, numele de utilizator trebuie sa contina cel putin un caracter, sa nu contina doar spatii sau caractere non alfanumerice (test reusit)

4.B - Se accepta numele de utilizator introdus la crearea contului, fiind adaugata in baza de date (test esuat)

5 - Incheierea testului

**Posibila problema:** la nivel de back-end nu se verifica toate caracterele din nume si parola utilizator. La nivel de front-end se permite trimiterea de caractere non-suported la server.

**Utilizator creaza un cont cu parola dintr-un caracter, un spatiu sau nimic**

Un nou utilizator doreste sa isi creeze cont. La parola, acesta introduce un singur caracter, un spatiu sau nimic, fiindca acesta nu a fost instiintat asupra restrictiilor la crearea unei parole.

1 - (Test) Utilizator acceseaza pagina aplicatiei

2 - (Test) Utilizator acceseaza pagina pentru creare cont

3 - (Test) Utilizator introduce ca parola: un singur caracter, un spatiu sau nimic

4 - (Test) Utilizator trimite informatiile

4.A - Aplicatia returneaza eroare, parola trebuie sa contina mai mult de un caracter (test reusit)

4.B - Se accepta parola introdusa la crearea contului, fiind adaugata in baza de date (test esuat)

5 - Incheierea testului

**Posibila problema:** La nivel de front end nu este afisat numarul minim de caractere pentru parola, respective nume. La nivel de back-end nu se reface verificarea numeui si a parolei.

**Utilizator se deconecteaza, insa acesta inca apare logat in baza de date**

Un utilizator doreste sa se deconecteze, insa pe parcurs se pierde conexiunea, iar in baza de date acesta ramane logat.

1 - (Test) Utilizator apasa butonul pentru deconectare

2 - (Test) Pe parcursul deconectarii se pierde conexiunea

2.A - Utilizator ramane logat in baza de date (test reusit)

2.B - Utilizator nu mai apare logat in baza de date (test esuat)

3 - Incheierea testului

**Posibila problema:** La nivel de back-end nu se primeste semnalul de deconectare deoarece utilizatorul a pierdut conexiunea.