**Examen Final**

**Java 1 Professional**

timp de lucru: 30 de zile

Se va realiza o aplicație ce are ca scop administrarea zborurilor din cadrul unui aeroport. Aplicația va avea următoarele elemente:

* (5p) o pagină de Start, care va fi afișată la pornirea aplicației. Această pagină va avea 2 butoane:
  + *Register* - care atunci când va fi apăsat, ne va direcționa către pagina de Register de mai jos.
  + *Login* - care atunci când va fi apăsat, ne va direcționa către pagina de Login de mai jos.
* (15p) o pagină de Register, care va avea următoarele câmpuri:
  + *username* - câmpul nu poate fi lăsat necompletat; nu trebuie să existe deja un utilizator cu același username
  + *password* - câmpul trebuie să fie completat cu o parolă care să aibă minim 6 caractere, minim o cifră, o literă mică și o literă mare
  + *password confirmation* - trebuie să fie introdusă aceeași parolă ca în câmpul “password”
  + *email address* - trebuie să fie un email valid (să aibă neapărat @ și ., în ordinea asta; exemplu minim de email valid: a@y.c)
  + Orice condiție nerespectată va face ca un pop-up să apară, cu un mesaj sugestiv (ex: username existent/parola nu respecta regulile de securitate, etc.)
  + dacă înregistrarea utilizatorului a fost făcută cu succes, vom fi redirecționați către pagina de Login
* (10p) o pagină de Login, cu 2 câmpuri: *username*/*email* și *parolă*. Câmpul de username/email va accepta atât username-ul, cât și email-ul utilizatorului. Dacă aceste câmpuri nu sunt completate cu datele de autentificare stocate în aplicație, va apărea un mesaj sugestiv (ex: “Date de autentificare invalide!”)
* (20p) o pagina principală, care va apărea doar după ce utilizatorul a fost autentificat cu succes, care să conțină:
  + data și timpul curent (**BONUS (5p) -** să se modifice timpul, în realtime)
  + un tabel - fiecare linie reprezintă un zbor
    - în dreptul fiecărei linii va fi un buton mic, roșu, care dacă va fi 3apăsat, va face ca un pop-up de confirmare să apară, care să întrebe “Sunteți siguri că vreți să ștergeți zborul?”. Dacă se apasă pe DA, informațiile corespunzătoare liniei vor fi șterse din baza de date, iar pagina va fi reîncărcată (linia va dispărea)
  + tabelul va avea următoarea structură:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sursa | Destinatie | Ora  Plecare | Ora  Sosire | Zile | Pret |
| Bucuresti | Londra | 7:40 | 10:25 | Luni, Miercuri, Vineri, Duminica | 1040 |
| Londra | Bucuresti | 13:05 | 15:40 | Marti, Joi, Sambata | 965 |

* + un buton ADAUGA\_ZBOR, care atunci când va fi apăsat, ne va duce către o pagină unde vor fi introduse informațiile necesare pentru un zbor nou:
    - SURSA - trebuie să aibă minim 3 litere
    - DESTINATIE - trebuie să aibă minim 3 litere, și trebuie să fie diferită de sursă (ex: nu pot exista zboruri Bucuresti - Bucuresti)
      * perechea nu trebuie să existe deja (dacă tabelul ar arăta fix ca cel de mai sus, nu putem să introducem încă un zbor Bucuresti - Londra)
    - ORA PLECARE - trebuie neapărat să aibă structura ORA:MINUT, altfel va apărea o eroare
    - DURATA - trebuie neapărat să aibă structura ORA:MINUT. Cu ajutorul acestui câmp, se va calcula ORA SOSIRE (atenție la modul de calcul)
    - ZILE - vor exista 7 check box-uri, pentru fiecare zi a săptămânii; se vor putea selecta oricâte zile
    - PRET - nu poate fi 0 sau negativ
  + pagina de adăugare va avea 2 butoane: ANULEAZA, care ne va duce înapoi la pagina principală, și ADAUGA ZBOR, care va realiza toate validările, iar dacă se va trece de ele, informațiile vor fi salvate în baza de date, și ne va duce înapoi la pagina principală, unde trebuie să apară și noul zbor în tabel.
* (20p) o pagină My Account:
  + vor fi afișate datele curente ale utilizatorului (username și email)
  + va exista un câmp pentru new\_email și un câmp pentru new\_username, fiecare cu propriul buton (CHANGE USERNAME si CHANGE EMAIL ADDRESS)
    - valorile din câmpul corespunzător trebuie să respecte aceleași regula de validare ca la Register. (ex: new\_username trebuie să nu fie gol)
    - la apăsarea butonului de CHANGE\_EMAIL, se va valida campul new\_email, iar dacă acesta este valid, va fi modificat email-ul stocat în baza de date, afișându-se un mesaj de succes. Dacă nu a respectat regulile validării, i se va afișa un mesaj corespunzător.
    - la apăsarea butonului de CHANGE\_USERNAME, se va valida câmpul new\_username, iar dacă acesta este valid, va fi modificat username-ul stocat în baza de date, afișându-se un mesaj de succes. Dacă nu au fost respectate regulile validării, se va afișa un mesaj corespunzător.
  + va avea un buton de CHANGE\_PASSWORD, care ne va duce pe o pagina nouă, unde va trebui introdusă o nouă parolă în 2 campuri (*password* și *confirm password*), acestea vor fi validate și, dacă îndeplinesc condițiile de la Register, se va modifica parola stocată în baza de date. Orice schimbare a parolei va deloga utilizatorul (va fi trimis la pagina de Login)
* (15p) o pagină cu istoricul acțiunilor utilizatorului curent; vor fi afișate toate acțiunile pe care le-a făcut, împreună cu data și ora la care acestea au avut loc, în ordine descrescătoare (ultima acțiune va fi afișată prima). Acest procedeu se numește “Audit”. Acțiunile care vor trebui auditate sunt:
  + register (ex: “contul a fost creat pe …”)
  + login/logout (ex: “login pe 25.04.2020, 20:14” / “logout pe ...”)
  + modificare de date (ex: “a fost modificată parola pe 25.04.2020, 20:14”)
  + navigare în pagina (ex: “a accesat pagina X pe ...”)
* (15p) un MenuBar, care să aibă opțiunile:
  + Home, care ne va duce la pagina principală
  + Account, care ne va duce la pagina My Account
  + Log Out, care ne va duce la pagina de Login
  + **(10p) BONUS -** Un buton de Back, care ne va duce, la fiecare apăsare, la pagina anterioară. Pentru acordarea punctajului, butonul ar trebui să rețină toate paginile accesate, în ordine inversă, și să funcționeze indiferent de numărul de apăsări (ideal ar fi ca ultima pagina la care ne duce butonul să fie cea de Login)
* **(15p) BONUS -** implementare de logout după o perioada de 15 minute. Exemplu: dacă se face login la ora 20:14, se va stoca în baza de date data și ora ultimului login pentru utilizatorul curent (Hint: vor trebui stocate data și ora ultimului login pentru fiecare utilizator). După ce acesta s-a autentificat, la fiecare acțiune pe care o face, se va verifica dacă au trecut 15 de minute (în cazul menționat, dacă ora e mai mica de 20:29); dacă au trecut mai mult de 15 minute, utilizatorul va fi redirecționat către pagina de Login.

**IMPORTANT:** va fi acordat un punctaj de 75% din punctajul cerinței dacă funcționalitatea merge pe cazurile obișnuite. Punctajul total este acordat doar dacă implementarea funcționează inclusiv pe cazurile limita.

Detalii de implementare:

* se vor respecta principiile SOLID, cât și standardele de scriere a codului (identificatori sugestivi, metode scurte, evitarea codului duplicat, etc.)
* toate informațiile vor fi salvate într-o bază de date (cel puțin 2 tabele)
* se va folosi Design Pattern-ul MVC
* se vor respecta cât mai multe Principii de Design