Siemens Internship

de Cristea Codrin

Problema 1

Diagrama de clase

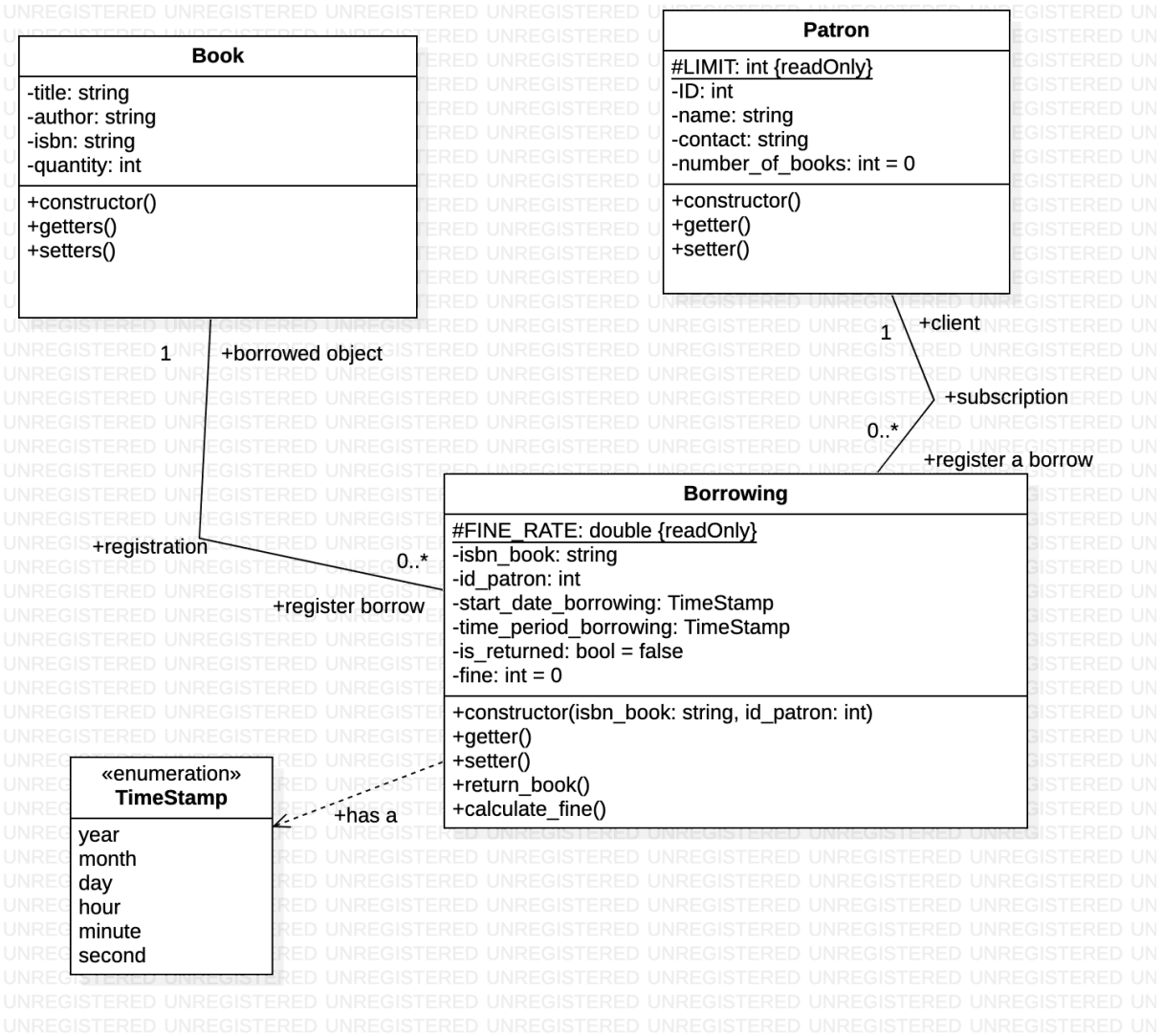
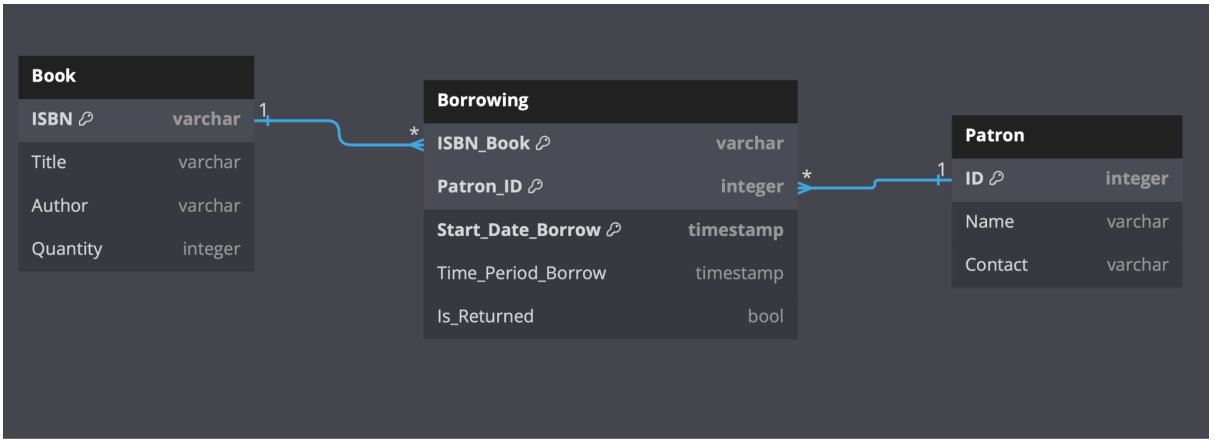


Diagrama de baze de date



Problema 2

1. Organizarea aplicatiei si a intrebarilor:

   Aplicatia se porneste accesand mai intai **login.html**. Trebuie mentionat ca aplicatia a fost dezolvata pe un MacBook si testata pe urmatoarele browser-e: Safari si Chrome.

   Intrebarile sunt impartite pe categorii. Deci fiecare categorie va avea o serie de intrebari, fiecare intrebare fiind formata din detalii, raspunsuri si raspunsul corect. Detaliile unei intrebari sunt prezentate sub urmatoarea forma [id] [text intrebare], cu diverse mici variatiuni. Fiecare intrebare va avea o serie de raspunsuri. Pentru o mai buna gestionare a acestora, fiecarui raspuns i se va fi atribuit o valoare, un fel de id, pentru a se putea face o diferentiere intre rasounsuri si pentru a se putea determina raspunsul corect. Fiecare raspuns poate sa fie de forma [id] [text raspuns]. Fiecare intrebare va avea o mica sectiune in care apare identificatorul raspunsului corect.

   Id-urile sunt unice pentru tot programul, indiferent de domeniu, adica nu exista 2 intrebari care sa aibe acelasi id in acelasi domeniu sau in oricare 2 domenii diferite. Acelasi lucru se poate spune despre id-urile raspunsurilor.

   Tinand cont ca este vorba de o aplicatie cu interfata web, arhitectura aplicatiei trebuie sa fie de tipul client server, cu majoritatea actiunior/ proceselor critice pe partea de server.

   La inceput utilizatorul va introduce un nume de utilizator. Dupa ce o serie de validari sunt trecute cu success, se poate incepe o noua sesiune de joc. Fiecare utilizator va trebui sa raspunda un anumit numar de intrebari. In functie de acest numar de intrebari:
   - Daca numarul de intrebari ce trebuie raspunse in quiz este mai mic decat numarul de domenii, atunci se va parcurge fiecare domeniu in oridinea in care sunt salvate si din fiecare domeniu se va alege aleator cate o intrebare.
   - Daca numarul de intrebari ce trebuie extrase este egal cu numarul de intrebari din domeniu, atunci iau toate intrebarile din domeniul respectiv.
   - Daca numarul de intrebari este mai mare decat numarul de domenii, atunci din fiecare domeniu se vor extrage in mod aleator un numar cat mai egal de intrebari.
   - In functie de numarul de intrebari ce trebuie extrase, se va calcula un fel de distributie pentru fiecare domeniu in parte, astfel incat extragerea per fiecare domeniu sa fie cat mai egal distribuita, adica dintr-un domeniu cu un numar mare de intrebari se vor extrage un numar mai mare de intrebari, iar dintr-un alt domeniu cu numar mic de intrebari se vor extrage mai putin intrebari, iar in cel mai rau caz se vor extrage toate intrebaril din acel domeniu. (un fel de “atenuare”).
   - Pentru extragerile din fiecare domeniu, se va tine o evidenta a intrebarilor deja selectate si atunci cand o noua intrebare este extrasa atunci se va verifica daca intrebarea respectiva a mai fost extrasa o data, caz in care se va extrage din nou.
   - Daca numarul de intrebari pentru quiz este egal cu numarul de intrebari din toata colectia disponibila, atunci se iau toate problemele disponibile.

- La finalul extragerilor se obtine o serie de intrebari pentru sesiunea de quiz a utilizatorului.
- Indiferent de cazurile mai sus prezentate, seria finala de intrebari este amestecata si dupa poate fi procesata.

Deci se stabilesc inca de la inceput la ce intrebari va trebui sa raspunda utilizatorul. Aplicatia va difuza o intrebare. Utilizatorul va selecta un raspuns si va confirma alegerea. Aplicatia va trimite catre server o cerere cu raspunsul selectat, iar acesta va decide validitatea raspunsului si va incrementa un contor de progres/scor (in cazul in care raspunsul trimis este cel corect). Apoi aplicatia va incarca urmatoarea intrebare (indifierent daca raspunsul este corect/gresit). Server ul pur si simplu va parcurge lista de intrebari deja prestabilita. Cererea trimisa catre server va contine si numele utilizatorul pentru a se putea face o diferentiere intre sesiuni.

Dupa ce toate intrebarile au fost raspunse, la final aplicatia va afisa scorul utilizatorului, care in cazul de fata reprezinta numarul de intrebari la care a raspuns corect, si ii va oferi obtinuea sa inceapa o noua sesiune fara a mai introduce un nou nume de utilizator.

2. Implementarea algoritmilor

In fisierului “server.js” am creat o mica tabela/lista de intrebari (50 in total) din 5 domenii diferite (Istorie, Matematica, Geografie, Astronomie si Chimie). Functia “shuffle_questions” este cea care determina intrebarile ce trebuie raspunse pentru fiecare sesiune in parte.

In prima instanta se decide cate intrebari sa fie extrase pentru fiecare domeniu in parte (“decide_how_many_questions_per_domain”) respectandu-se regulile mai sus mentionate. Aceasta decizie se foloseste de o functie (“distrubute_impact_per_columns”) care determina matematic cate intrebari sa se extraga pentru fiecare domeniu(sau coloana) in parte.

Dupa ce s-a determinat cate intrebari se vor extrage din fiecare domeniu, urmeaza partea de extragere efectiva a intrebarilor din fiecare domeniu in parte respectand regulile de mai sus (“extract_questions_from_domain”) si alipirea lor la lista/setul final de intrebari.
Dupa lista finala de intrebari este amestecata(“shuffle_array”) si returnata.

3. Reprezentarea entitatilor si a interactiunilor dintre ele

Entitatile principale identificare intr-o diagrama de clase sunt: Intrebare, Raspuns, Utilizator si Sesiune.
Clasa Intrebare are urmatoarele atribute (id, domeniu, text, raspunsuri, id_raspuns_corect).
Clasa Raspuns are urmatoarele atribute (id, text).
Clasa Utilizator are urmatoarele atribute(nume de utilizator).
Clasa Sesiune are urmatoarele atribute(utilizator, intrebari, numarul intrebarii curente, scor).

Intre Intrebare si Raspuns exista o asociere de compozitie (un Raspuns nu poate exista fara o Intrebare) de multiplicitate 1 la 4, adica fiecare Intrebare are 4 Raspunsuri, iar fiecare Raspuns ii apartine unei singure intrebari.
Intre Intrebare si Sesiune exista o asociere simpla de multiplicitate 1..* la 1..*. Adica o Intrebare poate fi raspunsa de mai multi Utilizatori in mai multe Sesiuni si o Sesiune poate contine mai multe Intrebari.
Intre Utilizator si Sesiune exista o asociere simpla de tip 0..* la 1, adica un Utilizator poate incepe mai multe Sesiuni, iar o Sesiune contine un singur Utilizator.

Entitatile principale intr-o diagrama de baza de date sunt: Domenii, Intrebari, Raspunsuri, Optiuni, Utilizatori, Sesiuni.

Tabela Domenii contine urmatoarele coloane: Id(cheie prima), Nume.
Tabela Intrebari contine urmatoarele coloane: Id(cheie primara), Text, Id_Domeniu(cheie straina cu referire catre Id din tabela Domenii).
Tabela Raspunsuri contine urmatoarele coloane: Id(cheie primara), Text.
Tabela Optiuni contine urmatoarele coloane: Id_intrebare(cheie primara, cheie straina cu referire la Id din tabla Intrebari), Id_raspuns(cheie primara, cheie straina cu referire la Id din tabla Raspunsuri), Este_Raspuns_Corect (adevarat/fals).
Tabela Utilizator contine urmatoarele coloane: Id(cheie primara), Nume_Utilizator, Scor_Curent.
Tabela Sesiuni contine urmatoarele atribute: Id(cheie primara), Id_Utilizator (cheie straina cu referire la Id din tabela Utilizator), Id_Intrebare(cheie straina cu referire la Id din tabela Intrebari).

Tabelele Intrebari, Raspunsuri si Utilizatori sunt tabele care contin doar date specifice quiz-ului. Tabelele Optiuni si Sesiuni sunt tabele de legatura corespunzatoare unei relatii de tipul many-to-many intre Intrebari si Raspunsuri, respectiv intre Utilizatori si Intrebari.