

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Database Guru

propusă de

Alexandru Roșca

Sesiunea: *iunie, 2019*

Coordonator științific

Conf. Dr. Anca Vitcu

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele _____

Data _____ Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemnatul(a)

domiciliul în

născut(ă) la data de, identificat prin CNP,
absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de
..... specializarea, promoția
....., declar pe propria răspundere, cunoscând consecințele falsului în
declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr.
1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

_____elaborată sub îndrumarea dl. / d-na
_____, pe care urmează să o susțină în fața
comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin
orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea
conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări
științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei
lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere
că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi,

Semnătură student

ACORD PRIVIND PROPRIETATEA DREPTULUI DE AUTOR

Facultatea de Informatică este de acord ca drepturile de autor asupra programelor-calculator, în format executabil și sursă, să aparțină autorului prezentei lucrări, *Alexandru Roșca*.

Încheierea acestui acord este necesară din următoarele motive:

Doresc să dezvolt aplicația în continuare.

Iași, *data*

Decan *Adrian Iftene*

(semnătura în original)

Absolvent *Alexandru Roșca*

(semnătura în original)

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

Database Guru

Alexandru Roșca

Sesiunea: *iunie, 2019*

Coordonator științific

Conf. Dr. Anca Vitcu

Cuprins

| | |
|--|-----------|
| Introducere | 4 |
| 1.1 Obiective generale | 4 |
| 1.2 Motivație | 4 |
| 1.3 Aplicații similare | 4 |
| 1.4 Contribuții..... | 5 |
| 1.5 Aplicabilitate..... | 6 |
| 1.6 Structura lucrării | 6 |
| 1 Structura aplicației | 7 |
| 1.1 Structura proiectului..... | 7 |
| 1.2 Tehnologii utilizate..... | 8 |
| 1.3 Arhitectura bazei de date a aplicației..... | 9 |
| 1.4 Interfața..... | 9 |
| 1.5 Încărcarea informațiilor | 10 |
| 2 Module | 11 |
| 2.1 Modulul <i>accounts</i> | 11 |
| 2.1.1 Descriere | 11 |
| 2.1.2 Modelarea bazei de date:..... | 11 |
| 2.2 Modulul <i>exercises</i> | 13 |
| 2.2.1 Descriere | 13 |
| 2.2.2 Modelarea bazei de date:..... | 13 |
| 2.2.3 Baza de date pentru testarea exercițiilor | 15 |
| 3 Interacțiunea utilizatorului cu aplicația | 16 |
| 3.1 Conturi | 16 |

| | | |
|-------|---|----|
| 3.1.1 | Tipuri | 16 |
| 3.1.2 | Profil..... | 16 |
| 3.1.3 | Clasament | 16 |
| 3.1.4 | Cum creezi un cont? | 16 |
| 3.1.5 | Cum ajungi profesor? | 16 |
| 3.1.6 | Cum ajungi admin? | 16 |
| 3.2 | Exercitii | 16 |
| 3.2.1 | Cum răspund la întrebare? | 16 |
| 3.2.2 | Istoric | 16 |
| 3.2.3 | Verificarea | 16 |
| 3.2.4 | Punctarea utilizatorului..... | 16 |
| 3.2.5 | Cum postez o întrebare ca profesor? | 16 |
| 3.2.6 | Gestionarea întrebărilor postate | 16 |
| 3.2.7 | Statistici..... | 16 |

Introducere

1.1 Obiective generale

Aplicația „Database Guru” a fost realizată din dorința de a ajuta studenții sau oamenii pasionați în învățarea limbajului *SQL* dar și testarea cunoștințelor deja dobândite despre acesta, într-un mod cât mai interactiv, interesant și aparte. Aplicația poate fi utilizată atât de studenți, cât și de profesori sau alte persoane care sunt interesate de acest domeniu. Astfel, utilizatorii își pot îmbogăți bagajul de cunoștințe, iar profesorii pot găsi nivelul optim de dificultate în crearea unei întrebări, cât și modul cel mai atractiv de a o formula.

1.2 Motivație

Ceea ce am observat în decursul anilor la ceilalți studenți și mai ales la mine, a fost capacitatea scăzută de concentrare față de lucrurile ce nu reprezentau un interes nici prin aspect (*design*), cât nici prin metodele de interacțiune abordate.

Este foarte important pentru un tânăr, cu o capacitate crescută de învățare și memorare, că ceea ce reprezintă un punct de interes să fie abordat în diferite moduri astfel încât să îi stârnească curiozitatea și interesul. Procesul de învățare trebuie să fie unul simplu, interactiv, atractiv, fără presiuni asupra studentului și cu substraturi motivaționale.

Plecând de la aceste caracteristici și însușiri, am creat aplicația „Database Guru”, prin care am reușit să îmbin tot ceea ce mi-am propus, astfel încât învățarea să se facă într-un mod plăcut, relaxant și deschis.

Profesorii, studenții, chiar și firmele care au diferite *training*-uri, pot folosi această aplicație care combină utilul cu plăcutul și mai ales care le oferă tuturor șansa de a acumula informațiile necesare și de a crește încrederea în cunoștințele proprii.

1.3 Aplicații similare

În primă instanță m-a atras ideea aplicației de pe „<https://flukeout.github.io/>”. Aceasta este o platformă de învățare a limbajului CSS, printr-o modalitate mai inedită: reprezentarea întrebărilor

sub formă de text dar și sub formă de obiect 2D cu animație. Inițial am fost atras foarte mult de concept, dar și de *design*-ul minimalist ce are nuanțe închise la culoare.

O altă aplicație deja existentă cu o idee similară este cea de la obiectul Baze de Date din anul al II-lea, de la Facultatea de Informatică din Iași. Aceasta are rolul de a testa cunoștințele studenților prin răspunderea la întrebări și posibilitatea de a crea chiar ei altele noi pentru colegii lor.

1.4 Contribuții

Deși ideea este inspirată din cele două aplicații prezentate anterior (1.3 *Aplicații similare*), contribuția personală a constat în următoarele îmbunătățiri:

- ✓ Aplicația *Database Guru* are atât opțiune de rezolvare a exercițiilor fără înregistrare, cât și posibilitatea de a crea un cont și de a avea progresul salvat indiferent de *device*-ul de pe care se conectează utilizatorul.
- ✓ Nu există un număr limitat de întrebări și acestea nu sunt *hard coded*.
- ✓ Există o delimitare a nivelelor de acces asupra conturilor.
- ✓ Există o modalitate de a îți verifica răspunsul și de a-l compara cu cel corect
- ✓ Fiecare cont are un profil care poate fi editat
- ✓ Există o pagină de istoric în cazul în care utilizatorul dorește să își revadă răspunsul pentru reîmprospătarea memoriei atunci când are nevoie
- ✓ Profesorii pot să își personalizeze întrebările având posibilitatea de a modifica baremul de notare după necesitățile fiecăruia
- ✓ Profesorii au la dispoziție statistici pentru fiecare întrebare individuală, dar și pe toate întrebările postate, pentru o privire de ansamblu
- ✓ Profesorii pot ajuta utilizatorii, oferindu-le *hint*-uri
- ✓ Clasament în funcție de punctajul obținut în urma rezolvării exercițiilor
- ✓ Preluarea și afișarea dinamică a bazei de date pe care rulează exercițiile pentru a ajuta utilizatorii în a-și forma o privire de ansamblu asupra bazei de date

1.5 Aplicabilitate

Încă de la început am pornit cu ideea de a crea o aplicație care mai poate fi dezvoltată ulterior și care poate fi implementată atât în instituții precum licee și facultăți cât și în domeniul privat pentru *training*-uri sau *online* pentru publicul pasionat de subiectul abordat.

De asemenea, aplicația poate fi extinsă foarte mult și ușor datorită modularizării sale, astfel extinzându-se și aria de interes a publicului față de aceasta dar și categoriile de cunoștințe ce pot fi evaluate sau chiar aprofundate.

1.6 Structura lucrării

Structura aplicației – prezentarea în ansamblu a aplicației, a structurii ei și un mic rezumat despre felul cum funcționează

Module – prezentarea structurii bazei de date a fiecărui modul și explicarea pe scurt a câmpurilor din fiecare tabel

1 Structura aplicației

1.1 Structura proiectului

Datorită modului organizat de lucru în *framework*-ul *Django*, aplicația a fost împărțită în două module: *accounts* și *exercises* (Fig. 1 – Structura proiectului). Inițial aceste module au fost proiectate să funcționeze separat, în scopul de a fi cât mai ușor de modificat sau eliminat din aplicație. Totuși, acest lucru nu a fost posibil datorită dependenței modulului *exercises* față de *accounts*, ținând cont de faptul că este imposibil ca un exercițiu să fie rezolvat de către un cont inexistent. În schimb, modulul *accounts* este total dependent față de celălalt modul, utilizatorul având posibilitatea de a explora restul *site*-ului în cazul în care modulul *exercises* este șters.

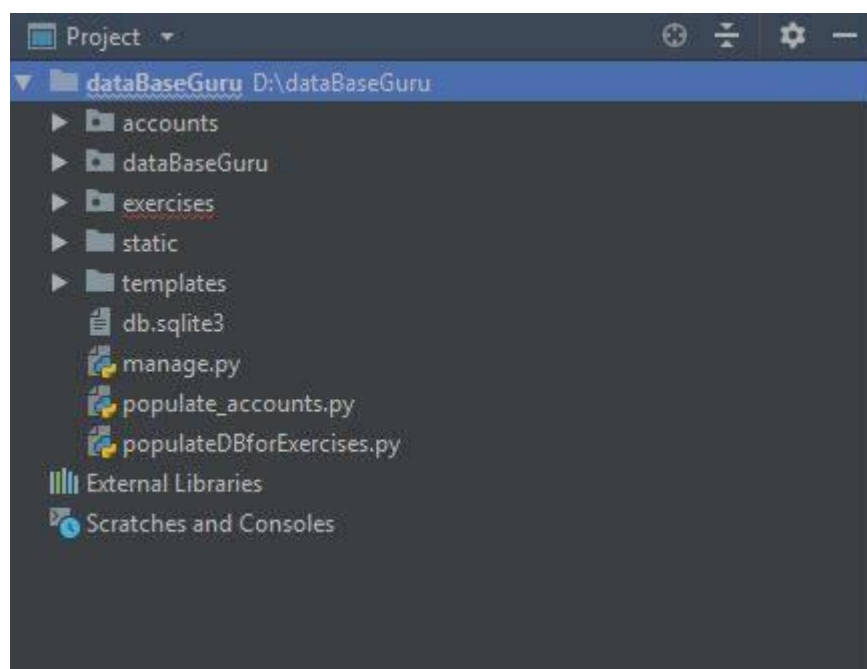


Fig. 1 – Structura proiectului

1.2 Tehnologii utilizate

Aplicația a fost în principal proiectată cu ajutorul *framework*-ului *Django*, un *framework* pentru limbajul de programare *Python* (am folosit versiunea 3.x).

Auxiliar, am folosit următoarele:

1. Pentru interfață:

- *Bootstrap* – utilizat în mare parte a *design*-ului, deoarece ajută la crearea unui *site* cât mai *responsive*, dar totodată ajută și la încadrarea mai ușoară și precisă a obiectelor în pagină
- *JQuery* – predominant pentru funcționalitatea butoanelor dar nu numai
- *Font Awesome* – preluarea iconițelor sub formă de text (mai exact *<i>*), util pentru manipularea lor: poziție, culoare, *padding*
- *Animate.CSS* – pentru animații
- *Chart.js* – pentru crearea de statistici într-un mod cât mai plăcut vizual, dar totodată și cât mai eficient

2. Pentru *server*:

- *Django* – *web development framework* pentru *Python*

3. Pentru comunicarea interfață – *server*:

- *AJAX*

1.3 Arhitectura bazei de date a aplicației

Baza de date este formată din opt tabele, șase sunt din modulul *accounts* și două sunt din modulul *exercises* (Fig. 2 – Arhitectura bazei de date). Pentru fiecare tabel, dacă nu este specificat un *primary key*, *Django* pune automat un *id* de tip *integer* ca și *primary key*

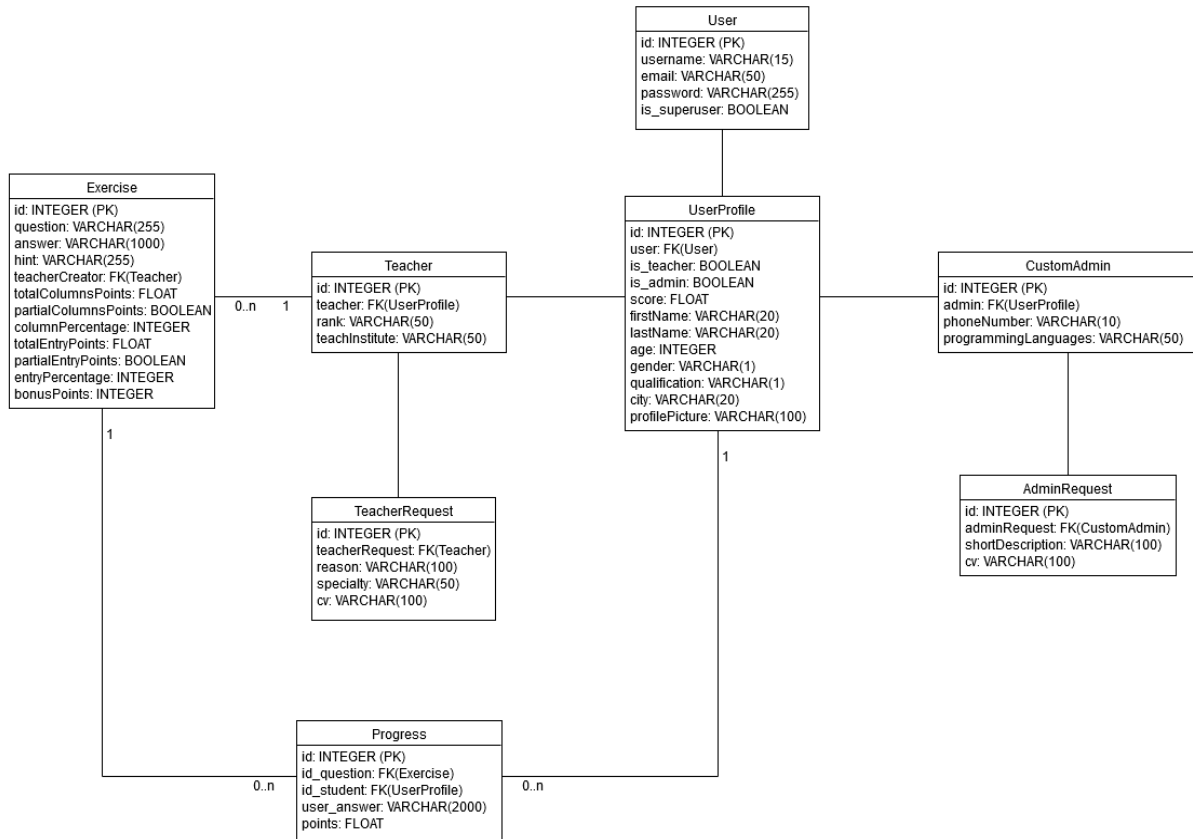


Fig. 2 – Arhitectura bazei de date

1.4 Interfața

În *Django*, o pagină *html* este denumită *template* deoarece, practic ea este un suport vizual pentru informațiile *view*-ului care o încarcă.

Pentru optimizare, am folosit moștenirea oferită de *Django* pentru *template*-uri (Fig. 3 - extends).

```
{% extends 'dataBaseGuru/baseDesign.html' %}
```

Fig. 3 - extends

Acest tag "extends" oferă posibilitatea de a extinde un *template* în alt *template*. Mai concret, majoritatea paginilor *html* din aplicație extind pagina "baseDesign.html". În această pagina se află toate elementele repetitive de pe *site* care trebuie încărcate mereu: *navbar*-ul, *background*-ul, culorile, *CDN*-urile pentru *Bootstrap*, *Font Awesome* și alte *script*-uri. Astfel, se îmbunătățește calitatea codului, având în vedere că nu trebuie scris același lucru pe fiecare pagină, se facilitează înțelegerea acestuia și crește viteza de încărcare a paginilor.

1.5 Încărcarea informațiilor

Pentru a încărca informațiile, *Django* folosește *view*-uri, acestea fiind funcții sau clase. Prin intermediul acestora sunt încărcate atât paginile cât și informațiile lor. În fiecare modul există un fișier "views.py" care conține toate funcțiile și clasele modulului respectiv, astfel existând o organizare mult mai bună a proiectului.

2 Module

2.1 Modulul *accounts*

2.1.1 Descriere

Acest modul se ocupă de partea administrativă a aplicației: *login*-ul, *logout*-ul, crearea și ștergerea conturilor și gestionarea permisiunilor acestora.

2.1.2 Modelarea bazei de date:

Baza de date a acestui modul este formată din 6 tabele (Fig. 4): *User*, *UserProfile*, *Teacher*, *CustomAdmin*, *TeacherRequest*, *AdminRequest*

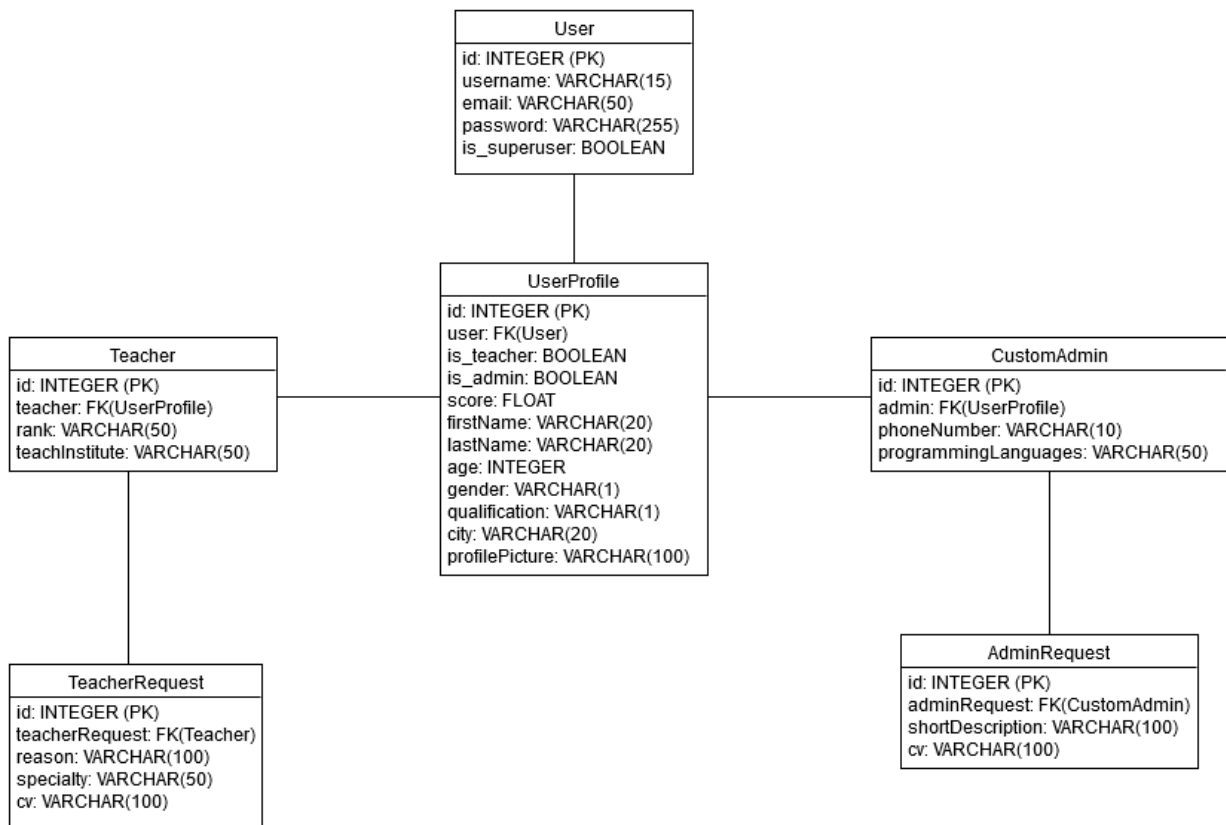


Fig. 4 – *accounts* – modelarea bazei de date

Django are predefinit tabelul *User* cu mai multe câmpuri, dintre care cele preluate sunt:

- *id* – *primary key-ul* generat automat
- *username* – ales de utilizator la înregistrare
- *password* – *Django* nu salvează niciodată parola așa cum o scrie utilizatorul, el generează un *hash* cu algoritmul *SHA256*
- *is_superuser* – un *boolean* care are valoarea 1 cand *user-ul* este administrator și 0 în caz contrar

Modelul *UserProfile* extinde modelul *User*, oferind:

- *id* – *primary key-ul* generat automat
- *user* – *foreign key* pentru modelul *User*
- *is_teacher* – *boolean* care are valoarea 1 atunci când utilizatorul are drepturi de profesor sau valoarea 0 în caz contrar
- *is_admin* – *boolean* care are valoarea 1 atunci când utilizatorul are drepturi de admin sau valoarea 0 în caz contrar
- *score* – coloana unde se înregistrează punctajul acumulat de la toate întrebările
- *firstName*
- *lastName*
- *age*
- *gender* – este reprezentat de un *varchar* de lungime maximă 1 care poate lua trei valori: N – *Neutral*; M – *Male*; F – *Female*
- *qualification* – este reprezentat de un *varchar* de lungime maximă 1 care poate lua patru valori: N – *Nothing*; P – *Primary School*; H – *High School*; U – *University*
- *city*
- *profilePicture*

Modelul *Teacher* extinde modelul *UserProfile*, oferind:

- *id* – *primary key-ul* generat automat
- *teacher* – *foreign key* pentru modelul *UserProfile*
- *rank* – gradul pe care îl are domnul profesor în cadrul de învățământ
- *teachInstitute* – locul unde predă domnul profesor

Modelul *TeacherRequest* extinde modelul *Teacher*, oferind:

- *id* – *primary key*-ul generat automat
- *teacherRequest* – *foreign key* pentru modelul *Teacher*
- *reason* – motivul pentru care utilizatorul dorește să aibă drepturi de profesor
- *specialty* – aria de specialitate al utilizatorului
- *cv*

Modelul *CustomAdmin* extinde modelul *UserProfile*, oferind:

- *id* – *primary key*-ul generat automat
- *admin* – *foreign key* pentru modelul *UserProfile*
- *phoneNumber*
- *programmingLanguages* – câteva limbaje de programare pe care *admin*-ul le stăpânește

Modelul *AdminRequest* extinde modelul *CustomAdmin*, oferind:

- *id* – *primary key*-ul generat automat
- *adminRequest* – *foreign key* pentru modelul *CustomAdmin*
- *shortDescription* – o scurtă descriere a personalității utilizatorului
- *cv*

2.2 Modulul *exercises*

2.2.1 Descriere

Modulul *exercises* se ocupă cu înregistrarea exercițiilor și salvarea progresului fiecărui utilizator

2.2.2 Modelarea bazei de date:

Baza de date a acestui modul este formată din două tabele: *Exercise* și *Progress* (Fig. 5).



Fig. 5 – exercises – modelarea bazei de date

Modelul *Exercise* conține:

- *id* – *primary key* generat automat
- *question* – cerința întrebării
- *answer* – răspunsul corect al întrebării
- *hint* – o sugestie pentru cel ce rezolvă, dacă este nevoie
- *teacherCreator* – *foreign key* pentru modelul *Teacher*
- *totalColumnsPoints* – numărul total de puncte disponibile pentru selectarea tuturor coloanelor cerute
- *partialColumnsPoints* – *boolean* cu valoarea 1 în cazul în care profesorul dorește să puncteze parțial coloanele selectate de utilizator sau 0 în cazul contrat
- *columnPercentage* – dacă *partialColumnsPoints* are valoarea 1, atunci aceasta opțiune se deblochează și numărul introdus reprezintă $x\%$ din totalul de puncte puse disponibile pentru coloane
- *totalEntryPoints* – numărul total de puncte disponibile pentru selectarea tuturor înregistrărilor cerute
- *partialEntryPoints* – *boolean* cu valoarea 1 în cazul în care profesorul dorește să puncteze parțial înregistrările selectate de utilizator sau 0 în cazul contrat
- *entryPercentage* – dacă *partialEntryPoints* are valoarea 1, atunci aceasta opțiune se deblochează și numărul introdus reprezintă $x\%$ din totalul de puncte puse disponibile pentru înregistrări

- *bonusPoints* – puncte bonus în cazul în care profesorul dorește să le ofere utilizatorilor dacă înregistrările și coloanele au fost corecte și în ordine

Modelul *Progress* conține:

- *id* – *primary key* generat automat
- *id_question* – *foreign key* pentru modelul Exercises
- *id_student* – *foreign key* pentru modelul UserProfile
- *user_answer* – răspunsul utilizatorului
- *points* – punctele obținute la acea întrebare

2.2.3 Baza de date pentru testarea exercițiilor

3 Interacțiunea utilizatorului cu aplicația

3.1 Conturi

3.1.1 Tipuri

3.1.2 Profil

3.1.3 Clasament

3.1.4 Cum creezi un cont?

3.1.5 Cum ajungi profesor?

3.1.6 Cum ajungi admin?

3.2 Exerciții

3.2.1 Cum răspund la întrebare?

3.2.2 Istoric

3.2.3 Verificarea

3.2.4 Punctarea utilizatorului

3.2.5 Cum postez o întrebare ca profesor?

3.2.6 Gestionarea întrebărilor postate

3.2.7 Statistici