

1. Základný popis internetovej aplikácie

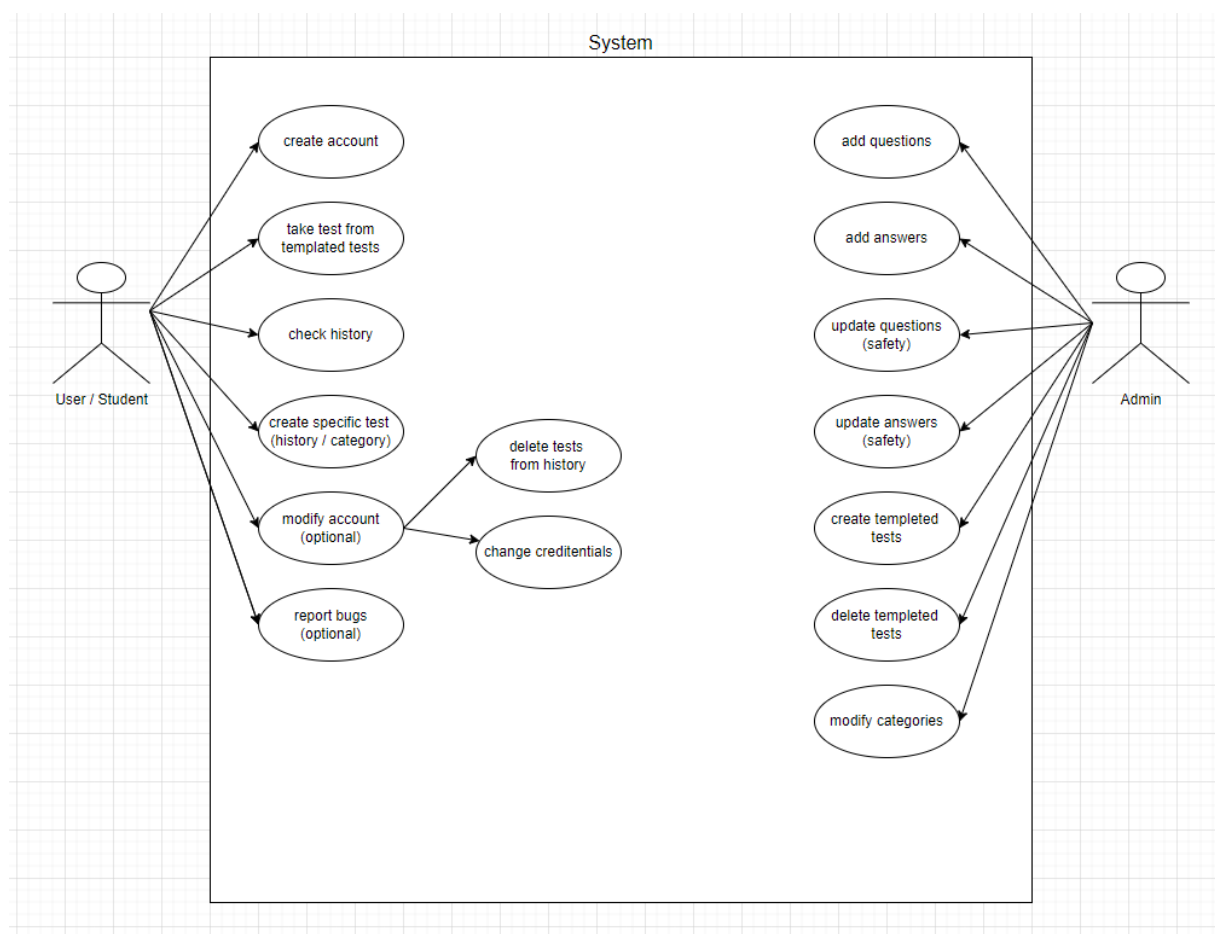
O akú aplikáciu sa jedná: Aplikácia na prípravu na prijímacie skúšky na medicínu.

Cieľová skupina používateľov: Študenti pripravujúci sa na prijímacie skúšky na medicínu.

2. Používateľské požiadavky

Študent: Môže generovať testy, prezerať historické výsledky, spravovať svoj profil.

Admin: Správa databázu otázok a kategórií, môže pridávať alebo updatovať otázky. Zatiaľ nepovažujem, za potrebné mazať otázky zo strany admina, z dôvodu štruktúry príjmačok. (tie otázky sa zídu vždy)



3. Dátový model

Popis dátového modelu:

je možnosť, že ešte do deadlinu 2 zmením mená tabuliek kvôli tomu, že niekde používam _ a niekde len medzeru

MedicineApplicants : Táto tabuľka obsahuje informácie o študentoch medicíny. Každý záznam predstavuje jedného študenta.

student_id: Unikátne identifikačné číslo pre každého študenta, slúži ako primárny kľúč

student_name: Meno študenta

student_email: email študenta, musí byť unikátny
registration_date: dátum a čas registrácie do systému
student_password: dobre zahashovane heslo

MedQuestions: Táto tabuľka bude uchovávať otázky ktoré sa môžu vyskytnúť v testoch.

question_id: Unikátne identifikačné číslo pre každú otázku, primárny kľúč.
question_text: Text otázky.
question_image: Odkaz na obrázok súvisiaci s otázkou (môže byť prázdny). Tu budú prevažne odkazy na obrázky pre chémiu, keďže sa tam nachádza veľa komplikovaných vzorcov
category_id: FK určujúci kategóriu

MedAnswers: Obsahuje odpovede na otázky. Každá odpoveď je spojená s konkrétnou otázkou.

answer_id: Unikátne identifikačné číslo pre každú odpoveď, primárny kľúč.
answer_text: Text odpovede.
answer_image: Odkaz na obrázok súvisiaci s odpoveďou (môže byť prázdny).
question_id: Referencia na otázku, k ktorej odpoveď patrí.
is_correct: Indikátor, či je odpoveď správna.
explanation: Vysvetlenie odpovede. (Optional pravdepodobne sa pridá v neskoršej fáze projektu)
selection_count: Počet výberov danej odpovede študentmi (predvolená hodnota je 0). (optional)

Med_Test_Templates: Definuje šablóny testov, ktoré môžu byť použité na generovanie konkrétnych testov pre študentov.

Cez Test Template sa pokúšame vytvoriť vopred definované štruktúry testov. Tieto testy by mohli / mali pri dodržaní cieľa práce byť dostupné študentom. Nechceme aby študent / používateľ musel veľa rozmyšľať a pre urýchlenie a comfort daného používateľa bude možnosť si vopred vybrať z preddefinovaných testov. Toto sa dalo vyriešiť aj programovo, cez pár ľahkých preddefinovaných selektov, ale pre budúci účel aplikácie sa viac hodia vopred preddefinované testy.

test_template_id: Unikátne identifikačné číslo pre každú šablónu testu, primárny kľúč.
template_name: Názov šablóny testu.
template_description: Popis šablóny testu.

Med_Test_Template_Questions: Spája otázky s konkrétnymi šablónami testov a určuje ich poradie v teste.

test_template_id: Referencia na šablónu testu.
question_id: Referencia na otázku.
sequence_number: Číslo určujúce poradie otázky v teste.
Kompozitný primárny kľúč je tvorený test_template_id a question_id.

Student_Tests: Uchováva informácie o testoch, ktoré študenti absolvovali.

test_id: Unikátne identifikačné číslo pre každý test, primárny kľúč.
student_id: Referencia na študenta, ktorý test absolvoval.
test_template_id: Referencia na šablónu testu použitú pre test.
test_date: Dátum a čas absolvovania testu.
score: Skóre študenta v teste.
total_possible_score: Maximálne možné skóre, ktoré mohol študent v teste dosiahnuť.

Student_Answers: Zaznamenáva, ktoré odpovede boli vybrané študentmi počas testov a či boli správne

student_answer_id: Unikátne identifikačné číslo pre každú vybranú odpoveď, primárny kľúč.

test_id: Referencia na test.

question_id: Referencia na otázku.

selected_answer_id: Referencia na vybranú odpoveď.

is_correct: Indikuje, či bola vybraná odpoveď správna.

Admin: tabuľka pre evidenciu adminov

admin_id: Id admina

admin_name: Meno admina

admin_password: Heslo, ktoré uložíme pomocou Hashingu + saltingu

admin_email: Email

Category: Zaznamenáva aké kategórie momentálne máme, možno sa rozšíri o informácie o škole atď.

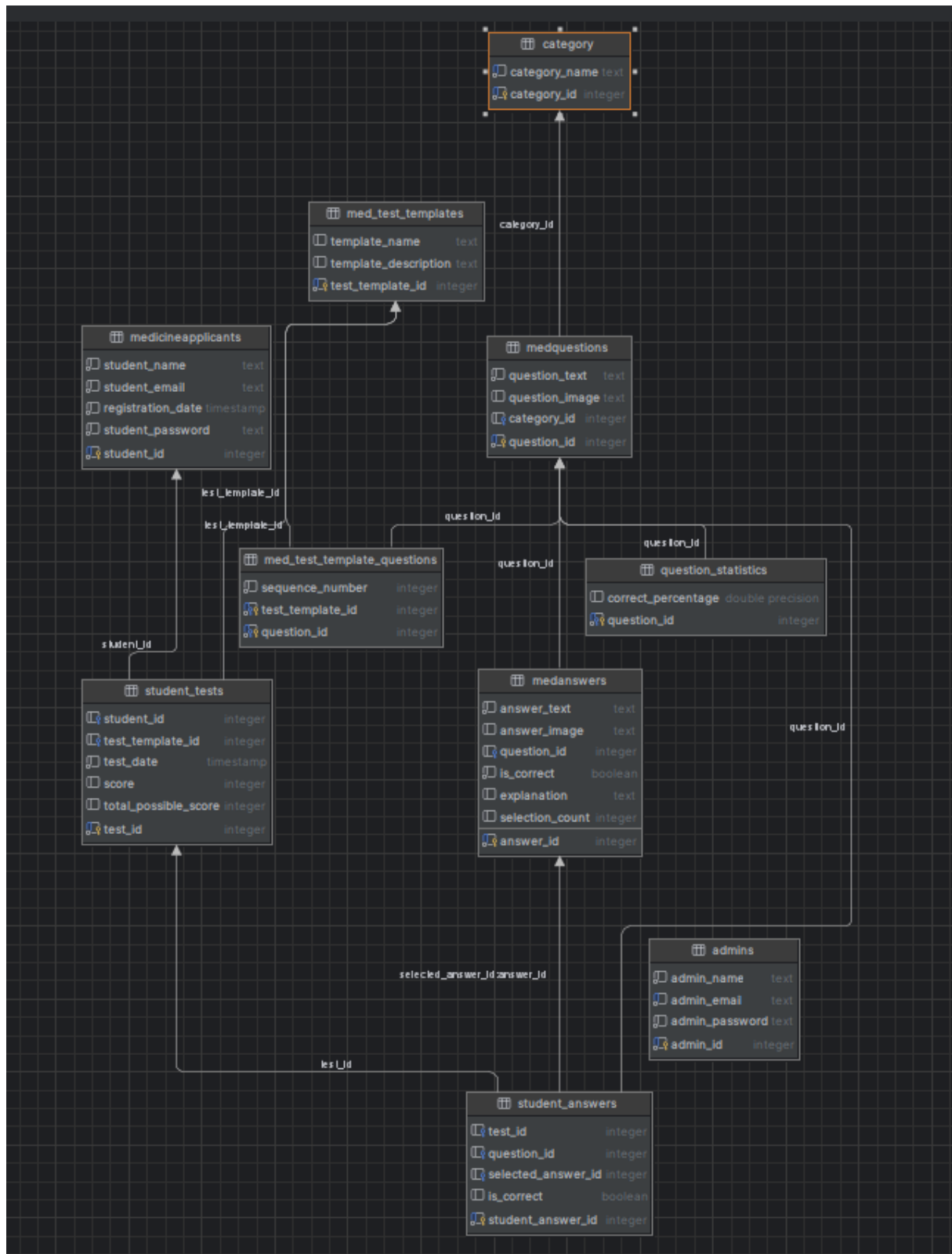
category_id: Unikátne identifikačné číslo pre kategórie

category_name: Text samotného názvu kategórie

Question_Statistics: (OPTIONAL) uchováva info o správnosti

question_id: Unikátne identifikačné číslo otázky

correct_percentage: Úspešnosť



4. Technologické požiadavky

Ak by AWS sa ukázalo ako príliš drahá voľba budem sa pozerať po free to use veciach, ale zatiaľ to vyzerá priateľné, ale to uvidím pri implementácii.

Client-side:

Jazyky a knižnice: JavaScript s využitím Vue.js a Nuxt.js, HTML, CSS.

Styling: TailwindCSS.

Server-side:

Hlavný framework: Django (Python), ktorý poskytuje robustný framework pre modely, pohľady, a šablóny, ako aj admin rozhranie pre rýchlu správu obsahu. (Tu rozmýšlam ešte nad zmenou na Node.js, Express.js ...)

Doplnkové technológie: AWS Lambda pre bezserverové funkcie, S3 pre ukladanie statického a mediálneho obsahu

Komunikácia medzi klientom a serverom:

Prednostne REST API, ale môže byť zvážené aj GraphQL pre komplexnejšie dopytovanie dát, možnosť implementácie cez AWS API Gateway.

Hosting a infraštruktúra:

Netlify a AWS.

Databáza:

Amazon RDS s PostgreSQL: Využitie Amazon Relational Database Service (RDS) pre hostovanie PostgreSQL database.

Podporované prehliadače:

Chrome, Firefox, Safari, (Edge optional)

5. Časový plán

Popri celej dobe trvania projektu bude trvať aj vývoj grafického návrhu frontendu v canve a zároveň nahadzovanie otázok do csv súboru.

Práca na projekte, priebežné (týždenné) reporty (5. - 9. týždeň)

5. týždeň: Základná implementácia a vývojové prostredie

Konfigurácia vývojového prostredia: Nastavenie servera, databázy, a klienta podľa zvolených technológií. Štúdia, čo všetko sa dá s AWS, poprípade zmena Technologických požiadaviek.

Základná štruktúra aplikácie: Implementácia základnej architektúry aplikácie, rozdelenie na moduly/komponenty.

6. týždeň: Backend a databáza

Vývoj backendu: Vytvorenie modelov, kontrolérov a endpointov pre základné funkcie aplikácie. Implementácia databázy: Definovanie schémy databázy podľa dátového modelu, migrácie. (do cloudovej formy)

7. týždeň: Autentifikácia a zabezpečenie

Autentifikácia používateľov: Implementácia registrácie, prihlásenia, a správy používateľských účtov.
Bezpečnostné opatrenia: Zabezpečenie komunikácie, ochrana pred bežnými útokmi.

8. týždeň: Frontend a integrácia

Vývoj frontendu: Implementácia užívateľského rozhrania podľa wireframe a designu.
Integrácia s backendom: Pripojenie klienta na server cez API, testovanie funkcionality.

9. týždeň: Funkcionality a testovanie

Rozšírenie funkcionality: Dokončenie prác na kľúčových funkcionalitách aplikácie.
Testovanie: Unit testy, integráciu testy, a manuálne testovanie celej aplikácie pre identifikáciu a opravu chýb.

10+ presentácia, bugfixes, oprava podľa feedbacku ...

6. Future work (optional)

Ak sa stihne všetko budem rád. Do budúcnosti sa dá vyhrať s „optional“ vecami spomenutými v tejto špecifikácii projektu. To znamená napríklad možnosť študenta reportovať bugs, vytvorenie a načítanie do databázy. Pravdepodobne prepracovanie kvalitnejšieho frontendu. Pridanie možnosti „sociálnej siete“ a komunikácii medzi študentmi (study groups). Pridanie okrem testov aj study kartičky, poznámky, implementácia databáz pre viaceré školy. Ak sa tam pridajú reálne otázky z „testovníč“ vybavenie autorského povolenia na prácu a manipuláciu s daným materiálom ...

Odpovede na pripomienky + updates:

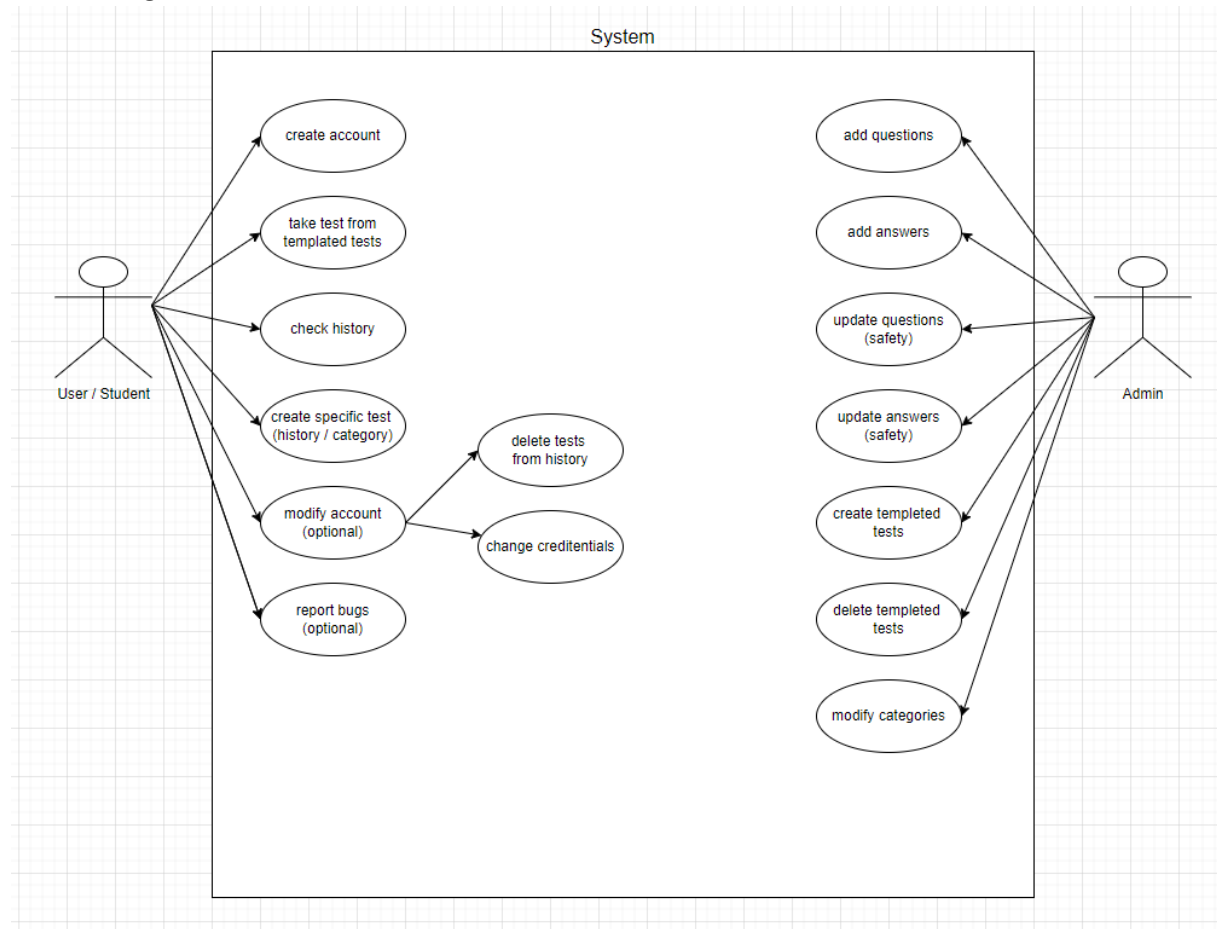
Vseobecne => fixed 😊 (myslím, že bude pohode nechať student_id a podobné parametre, keďže tam by to nemalo vyvolať zmätenie, v prípade potreby zmením na applicant_id, ale to sa mi úprimne páči menej)

Test template => Cez Test Template sa pokúšame vytvoriť vopred definované štruktúry testov. Tieto testy by mohli / mali pri dodržaní cieľa práce byť dostupné študentom. Nechceme aby študent / používateľ musel veľa rozmýšľať a pre urýchlenie a comfort daného používateľa bude možnosť si vopred vybrať z preddefinovaných testov. Toto sa dalo vyriešiť aj programovo, cez pár ľahkých preddefinovaných selektov, ale pre budúci účel aplikácie sa viac hodia vopred preddefinované testy. V skratke cez test template rozumieme vopred definovaný test. Možno nie najlepší pojem a je sanca že sa to v implementácii premenuje na niečo ako predefined_tests. Plus tým že otázok na jeden predmet je 1500 a to len za jednu školu, tak sa tým snažím dosiahnuť získanie väčšieho množstva dát na špecifické otázky.

Používateľské požiadavky

=>preklepy by už nemali existovať. (verím)

=> Uc changed



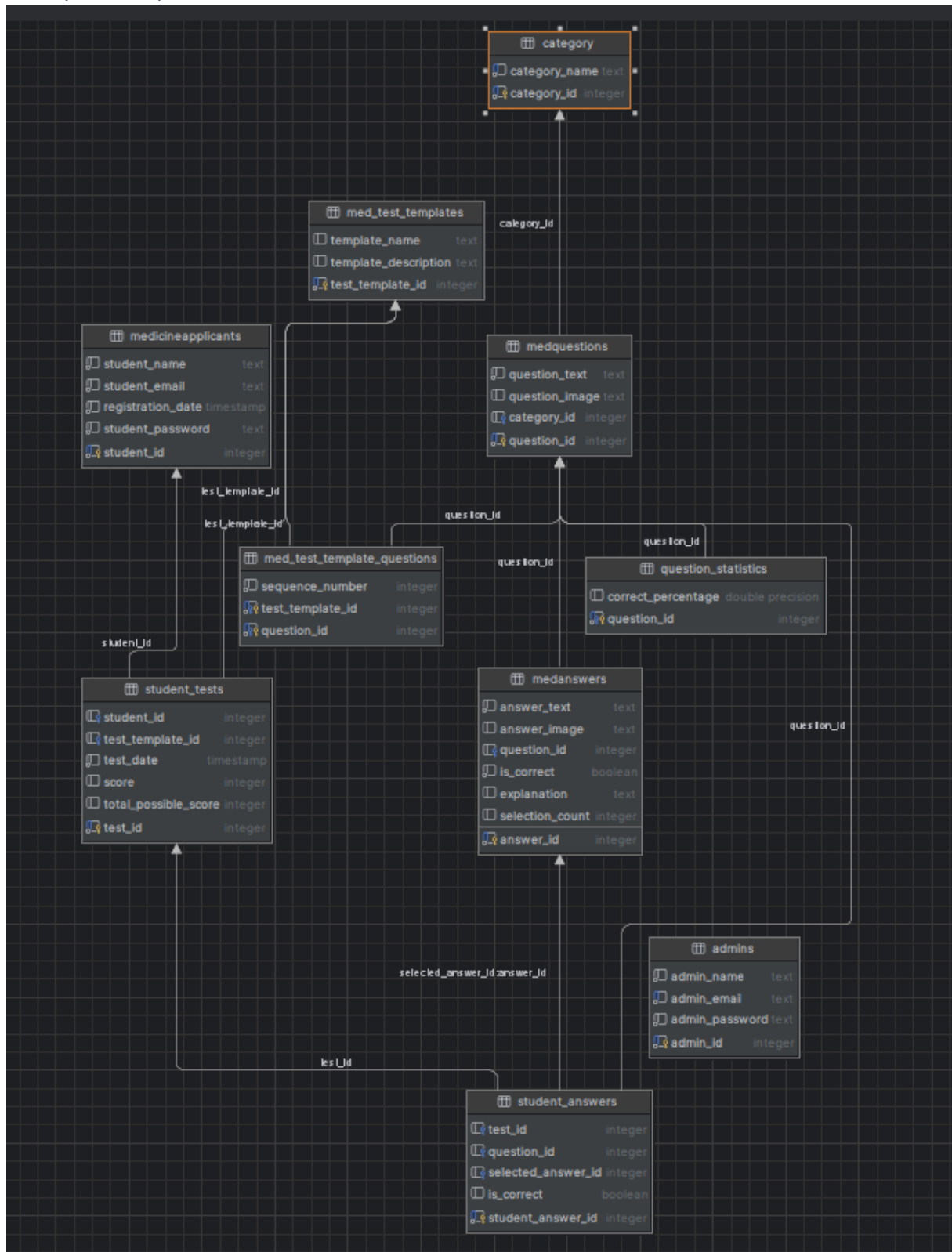
=> right now sa databaza uklada do velkeho excelu, z ktoreho sa spravi csv, a to sa cez python script hodi do db (complicated, not that effective, but working and easy to handle), priznam sa nemam do detailov premyslene ako by som to chcel riesit cez admina.

=> UC admin create templated tests -> vid hore -> predvytvori admin nejaky test

=>zmenene z create test from ... na take test from ..

pokracovanie na dalsej strane

Datovy Model updated



.statistiky budu optional aj ked odhadujem, ze sa to stihne (verim, myslim, dufam)

Časový plán -> deployment na server, prebehne keď bude všetko hotové teda pravdepodobne 9 – 10 týždeň, right now som sa pozrel na aws rds free tier a deployol som databázu, čo bolo celkom v poriadku následne ju stiahol keďže by tam len stála a nič nerobila 😊, vzhľadom ale na aws services verím, že nebude deployment problém, teda buď po každej hotovej časti deploynem danú časť (unlikely), ale najskôr všetko naraz, záleží či prototype projektu máme prezentovať ako localhost apku, alebo deploynutu apku.