

1. Základný popis internetovej aplikácie

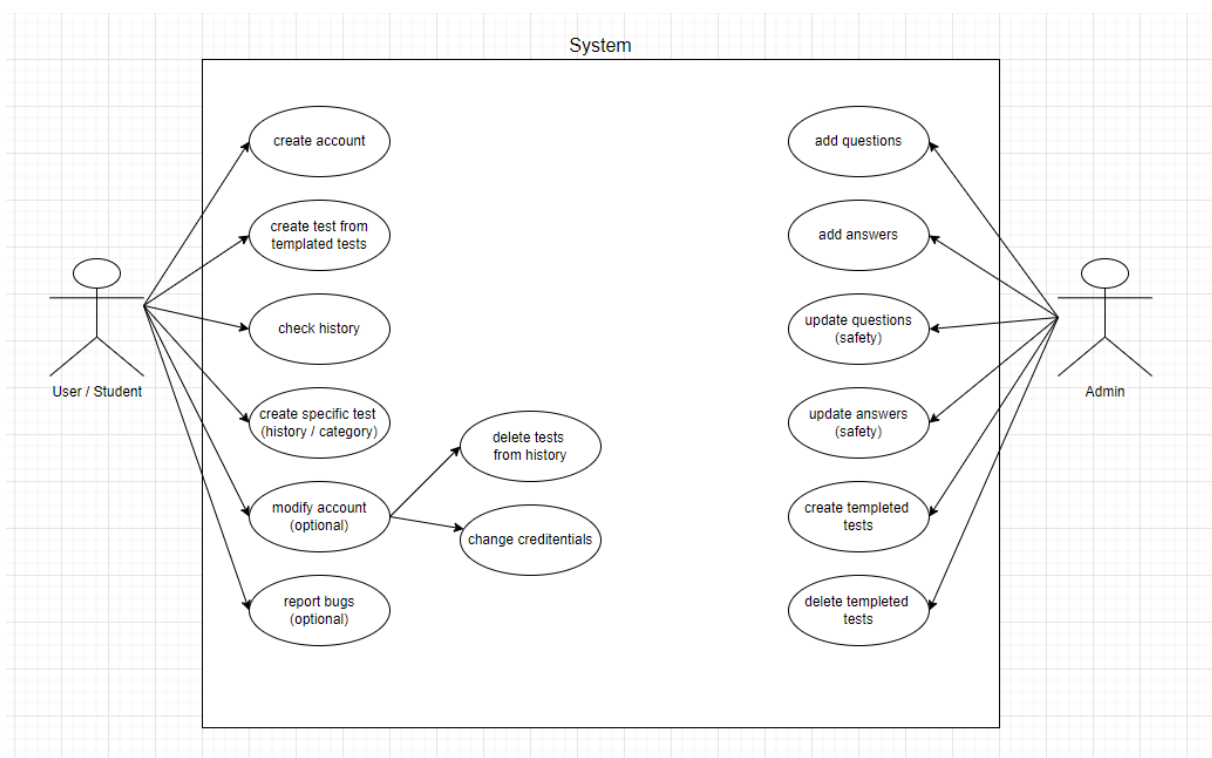
O akú aplikáciu sa jedná: Aplikácia na prípravu na prijímacie skúšky na medicínu.

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

2. Používateľské požiadavky

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

Admin: Správa databázu otázok a kategórií, môže pridávať alebo updatovať otázky. Zatiaľ nepožadujem, za potrebné mazať otázky zo strany admina, z dôvodu štruktúry príjmačov. (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

MedicineStudents: 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

question_category: Kategória, do ktorej otázka patrí.

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).

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

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

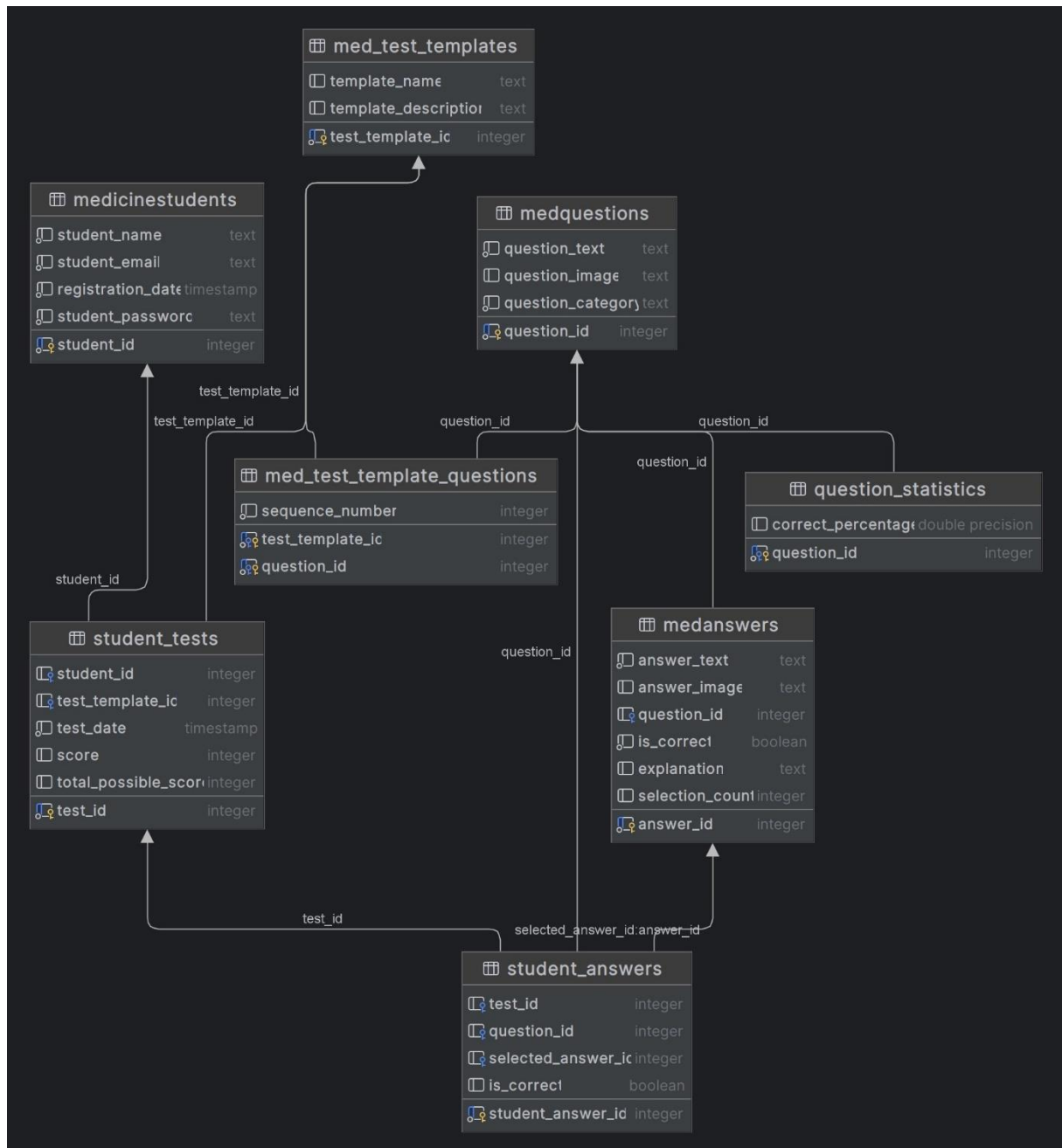
student_answer_id: Unikátne identifikačné číslo pre každú vybranú odpoveď, primárny kľúč.

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.



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 rozmýšľam 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 „testovnic“ vybavenie autorského povolenia na prácu a manipuláciu s daným materiálom ...