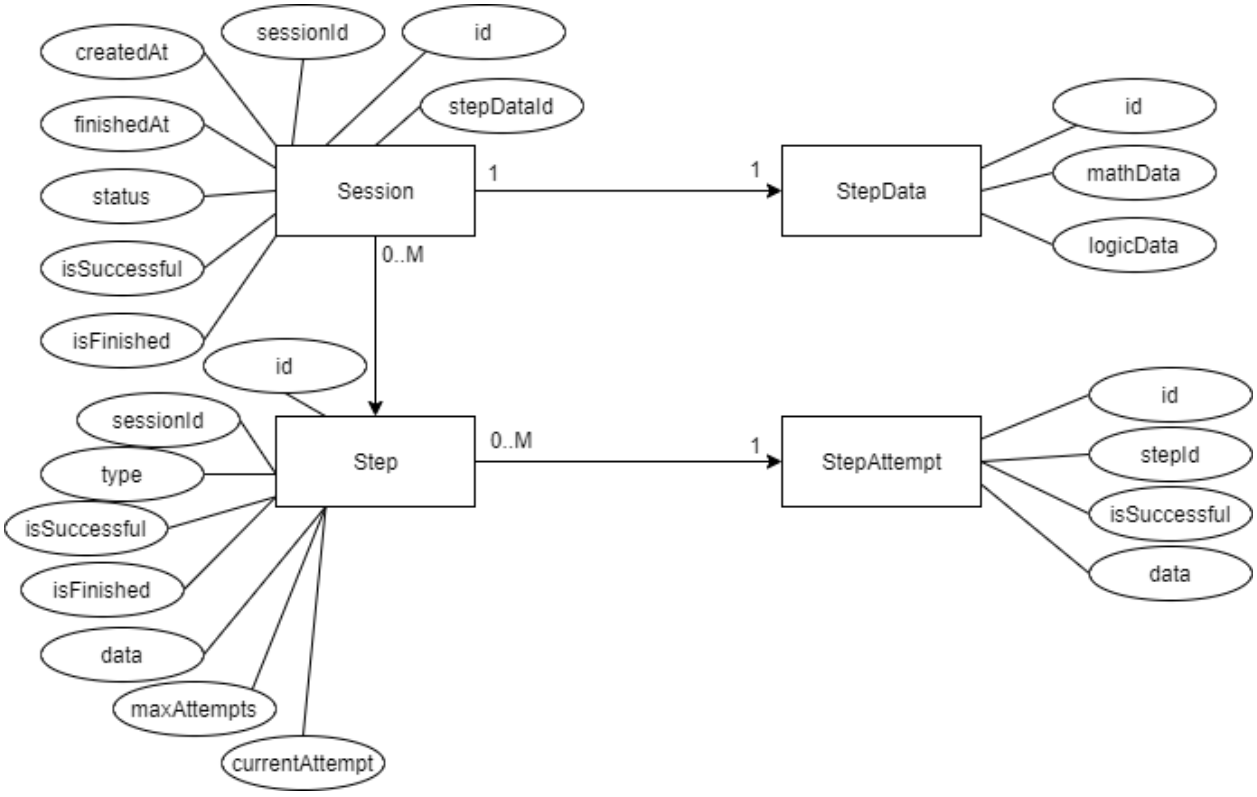
Backend task

Za sledeći projekat potrebno je napraviti aplikaciju koja rukovodi upravljanjem korisničkih sesija. Poželjno je koristiti kombinaciju NodeJS + TypeScript.

Sesija (Session) se sastoji od više koraka (Step). Sesija sadrži informacije kao što su: kada je kreirana, kada je završena, koji je status sesije, da li je sesija uspešno kompletirana i slično.
Svaki korak u sesiji može imati određene podatke (data), kao i stanje da li je uspešan ili ne. Postoje dva tipa koraka, Math i Logic.
Svaki korak se može pokušati više puta (StepAttempt). Krajnji rezultat koraka se smešta u tabelu StepData.

1. Napraviti PostgreSQL bazu u skladu sa priloženom slikom relacionog modela



Session

Naziv	Tip	Opis
id	number	Primarni ključ tabele
sessionId	string	guuid koji jedinstveno identifikuje sesiju
createdAt	date	U timestamp formatu
finishedAt	date	U timestamp formatu
status	Enum: [‘Created’, ‘In progress’, ‘Completed’]	Vrednost koja pokazuje u kom se statusu nalazi sesija
isSuccessful	boolean	Vrednost koja naznačava da li je sesija uspešno završena
isFinished	boolean	Vrednost koja naznačava da li je sesija završena
stepDataId	number	Spoljni ključ ka tabeli StepData

Step

Naziv	Tip	Opis
id	number	Primarni ključ tabele
sessionId	number	Spoljni ključ ka tabeli Session
type	Enum: [Math, ‘Logic’]	Vrednost koja pokazuje kog je tipa korak
isSuccessful	boolean	Vrednost koja naznačava da li je korak završen uspešno
isFinished	boolean	Vrednost koja naznačava da li je korak završen

data	{
	“mathData”: null,
	“logicData”:null
	}
		Objekat u kome se nalazi najažurnija vrednost za prikupljene podatke za ovaj korak.

maxAttempts	number	Broj pokušaja koliko puta može da se pokuša korak
currentAttempt	number	Trenutni pokušaj na kome se nalazi korsinik

StepAttempt

Naziv	Tip	Opis
id	number	Primarni ključ tabele
stepId	number	Spoljni ključ ka tabeli Step
isSuccessful	boolean	Vrednost koja naznačava da li je pokušaj koraka završen uspešno

data	{
	“mathData”: null,
	“logicData”:null
	}
		Objekat u kome se nalazi vrednost za prikupljene podatke za ovaj pokušaj.

StepData

Naziv	Tip	Opis
id	number	Primarni ključ tabele
mathData	string	Finalna vrednost koja je prikupljena u Math koraku.
logicData	string	Finalna vrednost koja je prikupljena u Logic koraku.

2. Napraviti Node.js servis koji će imati tri rute. Metode je moguće testirati iz nekog REST klijenta poput Postmana ili Arc-a. Nije potrebno razvijati front end aplikaciju za ove potrebe.
    a. /createSession
    b. /finishStep
    c. /getSessions
        i. By Status
        ii. Between certain dates
        iii. By isSuccessful
        iv. By isFInished

Metoda **createSession** kao ulazni parametar prima ***niz koraka*** koje ta sesija treba da sadrži. Prilikom kreiranja sesije, generiše se guuid sesije i zajedno sa ostalim parametrima se čuva u

tabeli. Njen status je Created. Svaki korak u nizu mora definisati kojeg je tipa korak. Staviti maksimalan broj pokušaja na vrednost 2. Svaka sesija mora sadržati oba tipa koraka.

Metoda **finishStep** služi za pokušaj prolaska datog koraka na kome je sesija. Prima sessionId, id koraka i payload. Payload zavisi od tipa koraka. Ukoliko je korak tipa Math, metoda treba da proveri da li je ulazni parametar jednak zbiru 2+2. Ukoliko jeste (ulazni parametar je 4), pokušaj datog koraka je uspešan, samim tim i korak postaje uspešan i sistem prebacuje sesiju na sledeći korak. Ukoliko unos nije 4, pokušaj za korak je neuspešan. Korisnik može ponovo pokušati unos, dok ne potroši sve pokušaje ili ne bude uspešan. Ukoliko se iskoristi maksimalan broj pokušaja, korak postaje neuspešan i sistem prebacuje sesiju na sledeći korak. Pri svakom pokušaju, u StepAttempt data kolonu potrebno je sačuvati payload (4 u uspešnom slučaju, drugi broj u neuspešnom). Kada se korak završi, uspešno ili neuspešno, potrebno je ažurirati vrednost kolone data na tabeli Step. U datu kolonu je potrebno upisati vrednost iz data kolene iz pokušaja koji je bio uspešan, ili vrednost data kolone poslednjeg neuspešnog pokušaja. Na isti način je potrebno popuniti i data kolonu u tabeli StepData.

Ukoliko je korak tipa Logic, potrebno je proveriti da li je uneta vrednost ispravna email adresa. Potrebno je koristiti regex za proveru unosa. Logika za popunjavanje je ista kao i za Math tip koraka.

Čim nastane prvi pokušaj prvog koraka, status sesije se ažurira na In progress.

Ukoliko je poslednji korak sesije završen, status sesije se postavlja na Completed i zapisuje se vreme završetka. Status sesije je uspešan ako su svi koraci uspešni. U suprotnom je neuspešna.

Metoda **getSessions** služi za dovlačenje svih sesija u json formatu.

Bonus poeni:

- Ukoliko se koristi Typescript:
    - Preporučuje se da se koristi ORM pri pravljenju ovog servisa i komunikacije sa bazom. Npr: Typeorm
- Odraditi neki vid autorizacije pri komunikaciji sa ovim servisom, jwt ili slično
- Uvesti transakcije kroz sve rute gde dolazi do bilo kakve izmene stanja na nekim od tabela
- Uvesti neki vid error handling-a
- Standardizovati responsove
- Uvesti paginaciju na /getSessions rutu