

MTAA

Zadanie - Zoznamka Digeridoo

Samuel Bernát
AIS ID : 110764
LS 2022

Rest API volania

- GET - aplikácia zavolá volanie HTTP GET napríklad v prípade, keď bude chcieť používateľ rozkliknúť profil eventu a získať informácie o ňom. Volanie bude HTTP GET /getEventData[id], kde id bude platné id eventu, ktorý chce používateľ načítať
- PUT - aplikácia zavolá volanie HTTP PUT napríklad v prípade, kedy bude chcieť používateľ upraviť informácie o ním vytvorenom evente. Po upravení želaných dát bude volanie HTTP PUT /saveChangedEvent[id], kde id bude platné id eventu
- POST - aplikácia zavolá volanie HTTP POST v prípade že používateľ chce zmeniť informácie pre vlastnosti eventu. Volanie bude HTTP POST /events[id]/eventDescription/description , kde id je platné id eventu a description je názov vlastnosti ktorú chceme zmeniť
- DELETE - aplikácia zavolá volanie HTTP DELETE v prípade , keď bude chcieť používateľ zrušiť svoju účasť na evente. Volanie bude HTTP DELETE /Events[id]/attendants/name, kde id bude platné id eventu a name meno používateľa

Účel aplikácie

Naša aplikácia slúži ako záujmová zoznamka, pre ľudí všetkých vekových kategórií, pokrývajúc všetky sféry záujmov. Vychádzame zo situácií, kedy by sa stalo, že vaši kamaráti boli zaneprázdnení, keď ste potrebovali pomoc so školou, nemali ste parťáka na pivo, alebo vám chýbal jeden hráč do párneho počtu na futbal. V tejto online ére tinderu a podobných aplikácií sa javí jednoduchšie dostať nejakú slečnu na rande, ako si zohnať kamaráta so spoločnými záujmami a vidinou hlbšieho priateľstva. Preto sme sa rozhodli vytvoriť záujmovú zoznamku, ktorá pomôže spoznať nových ľudí, vytvoriť priateľstvá, či vzťahy na základe spoločných záujmov, aktivít a hlbších konexií, nie iba výzoru. Používatelia budú mať možnosť vytvorenia si profilu na základe záujmov a následne majú možnosť vytvárať eventy, na ktoré sa ďalší používatelia môžu pripojiť v prípade záujmu a splnenia kritérií eventu.

Akceptačné testy

Front-end

Test 1: Vytvorenie nového eventu (positive test)	
Vstupné podmienky	Používateľ je zaregistrovaný do aplikácie a aplikácia má pripojenie na internet
Výstupné podmienky	V aplikácii pribudne event, ktorý sa uloží do databázy a ľudia ho budú môcť nájsť a interagovať s ním
Postup:	<ol style="list-style-type: none">1. Používateľ stlačí tlačidlo Vytvoriť Event2. Na ďalšej stránke vyplní kolónky s názvom, kategóriou, dátumom, kapacitou, miestom, popisom a nahrá obrázok3. Používateľ klikne na tlačidlo Publikovať Event4. Zobrazí sa potvrdenie o vytvorení a publikovaní článku a presmeruje používateľa do sekcie Moje Eventy
Výsledok: PASS	

Test 2: Nájdenie a pripojenie sa na event (positive test)	
Vstupné podmienky	Používateľ je zaregistrovaný do aplikácie a aplikácia má pripojenie na internet
Výstupné podmienky	Používateľ nájde event ktorý mu vyhovuje, prihlási sa naň a event bude dostupný v Moje Eventy
Postup:	<ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo Prehľadávať Eventy 2. Na ďalšej stránke klikne na tlačidlo Náhodné prehľadávanie 3. Otvorí sa mu obrazovka s prehľadávaním eventov, kde klikne na tlačidlo "X" pod prvým eventom, pretože sa mu nepáči 4. Objaví sa mu druhý event, ktorý sa mu páči, preto klikne na tlačidlo so srdiečkom pod eventom 5. Event je dostupný, preto sa na neho používateľ prihlási a môže k nemu pristupovať v kolónke Moje Eventy
Výsledok: PASS	

Test 3: Odhlásenie sa z prihláseného eventu (positive test)	
Vstupné podmienky	Používateľ je zaregistrovaný do aplikácie, aplikácia má pripojenie na internet a používateľ je prihlásený na event
Výstupné podmienky	Používateľ sa odhlási z eventu, nebude k nemu môcť pristupovať v kolónke Moje Eventy a z databázy eventu sa uvoľní jedno miesto
Postup:	<ol style="list-style-type: none"> 1. Používateľ klikne na tlačidlo Moje Eventy 2. Zobrazí sa mu list eventov, na ktoré je prihlásený, vyberie event z ktorého sa chce odhlásiť a klikne naň 3. Zobrazí sa mu profil eventu, používateľ scrollne nižšie a klikne na tlačidlo odhlásiť z eventu 4. Zobrazí sa mu potvrdzujúca obrazovka o odhlásení z eventu
Výsledok: PASS	

Test 4: Vytvorenie eventu s rovnakými parametrami	
Vstupné podmienky	Používateľ je zaregistrovaný do aplikácie a aplikácia má pripojenie na internet
Výstupné podmienky	Event nebude vytvorení používateľ bude upozornení aby zmenil parametre pre event , ktorý sa snaží vytvoriť.
Postup:	<ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo vytvoriť event. 2. Zobrazí sa mu screen na vytvorenie nového eventu 3. Používateľ vyplní údaje kde názov čas a miesto je totožné s eventom nachádzajúcim sa v databáze. 4. Používateľ klikne tlačidlo vytvoriť event 5. Rozpozna sa zhoda a vyskočí varovné okno ktoré hlási že takýto event už existuje.
Výsledok: FAIL	

Test 5: Vytvorenie eventu bez miesta konania	
Vstupné podmienky	Používateľ je zaregistrovaný do aplikácie a aplikácia má pripojenie na internet
Výstupné podmienky	Event nebude vytvorení používateľ bude upozornení na chýbajúci parameter zo skupiny povinných .
Postup:	<ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo vytvoriť event. 2. Zobrazí sa mu screen na vytvorenie nového eventu 3. Používateľ vyplní všetky povinne a nepovinné údaje až na kolónku meno eventu 4. Používateľ klikne tlačidlo vytvoriť event 5. Zobrazí sa obrazovka s informáciou o chýbajúcom povinnom parametre eventu
Výsledok: FAIL	

Back-end

Test 1: Úprava existujúceho eventu (negative test)	
Vstupné podmienky	V každom volaní musí byť zakomponovaný autorizačný token
Výstupné podmienky	Vytvorený event v databáze sa nijako nezmení
Postup:	<ol style="list-style-type: none">1. Aplikácia zavolá HTTP GET /Moje Eventy/getEventData[id] - kde v id bude platné id existujúceho eventu2. Volanie vráti 200 OK, v body HTTP sa budú nachádzať polia status_type, name,image, category, date, capacity, city a additional info3. Aplikácia zavolá HTTP PUT /saveChangedEvent/[id] a v tele požiadavky sa bude nachádzať zmenená kolónka date a zvyšné kolónky budú nezmenené okrem name, ktoré bude prázdne4. Volanie vráti 400 BAD REQUEST a informácie o evente ostanú bez zmeny
Výsledok: FAIL	

Test 2: Odhlásenie sa z prihláseného eventu (negative test)	
Vstupné podmienky	V každom volaní musí byť zakomponovaný autorizačný token
Výstupné podmienky	Používateľovi sa nepodarí odhlásiť z eventu
Postup:	<ol style="list-style-type: none"> 1. Aplikácia zavolá HTTP GET User[id]/Prihlásené Eventy/getEventData[id] - kde v id bude platné id používateľa a v druhom id existujúceho eventu 2. Volanie vráti 200 OK, v body HTTP sa budú nachádzať polia attendance, chat 3. Aplikácia zavolá HTTP PUT User[id]/saveEventChange/[id] a v tele požiadavky sa bude nachádzať kolónka attendance, ktorá bude stále nastavená na True, zvyšná kolónka bude nezmenená 4. Volanie vráti 400 BAD REQUEST a používateľ ostane prihlásený na event
Výsledok: FAIL	

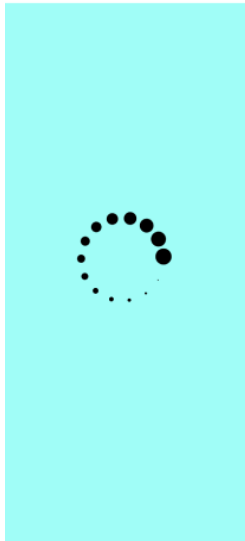
Test 3: Úprava existujúceho eventu	
Vstupné podmienky	V každom volaní musí byť zakomponovaný autorizačný token
Výstupné podmienky	Vytvorený event v databáze sa aktualizuje
Postup:	<ol style="list-style-type: none"> 1. Aplikácia zavolá HTTP GET /Moje Eventy/getEventData[id] - kde v id bude platné id existujúceho eventu 2. Volanie vráti 200 OK, v body HTTP sa budú nachádzať polia status_type, name,image, category, date, capacity, city a additional info 3. Aplikácia zavolá HTTP PUT /saveChangedEvent/[id] a v tele požiadavky sa bude nachádzať zmenená kolónka date a zvyšné kolónky budú nezmenené 4. Volanie vráti 200 OK a informácie o evente sa zmenia
Výsledok: PASS	

Test 4: Úprava profilu	
Vstupné podmienky	V každom volaní musí byť zakomponovaný autorizačný token
Výstupné podmienky	Zmenené meno používateľa
Postup:	<ol style="list-style-type: none"> 1. Aplikácia zavolá HTTP GET/MojProfil/getProfilData[id]- kde id bude platné id používateľa 2. Volanie vráti 200 OK , v body HTTP sa budú nachádzať polia meno, priezvisko, vek, pohlavie, dátum, image a záujmy 3. Aplikácia zavolá HTTP PUT /editProfile/[id] v tele požiadavky sa bude nachádzať zmenené meno používateľa 4. Volanie vráti 200 OK a meno používateľa sa aktualizuje
Výsledok: PASS	

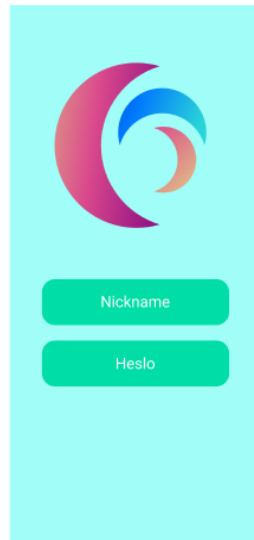
Test 5: Zobrazenie eventu	
Vstupné podmienky	V každom volaní musí byť zakomponovaný autorizačný token
Výstupné podmienky	Zobrazí profil eventu
Postup:	<ol style="list-style-type: none"> 1. Aplikácia zavolá HTTP GET/MyEvents[id] - kde id je platné id používateľa 2. Volanie vráti 200 OK v body sa budú nachádzať názvy všetkých eventov v ktorých je používateľ prihlásený 3. Aplikácia zavolá HTTP GET/Event_info[id] - kde id bude id eventu, v body HTTP sa budú nachádzať polia s dátami o evente 4. Volanie vráti 200 OK a zobrazí sa profil eventu
Výsledok: PASS	

Návrhy obrazoviek

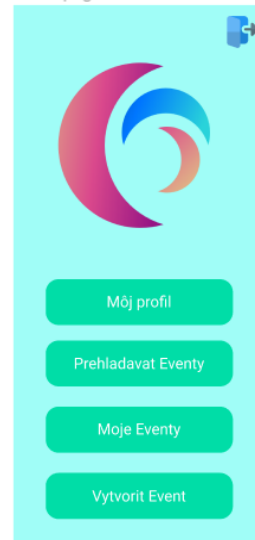
loading



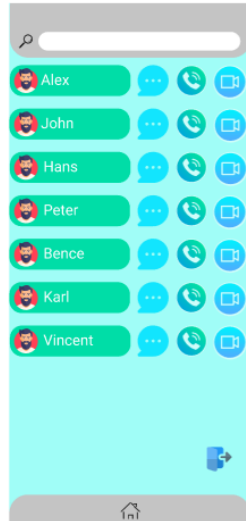
LOGIN



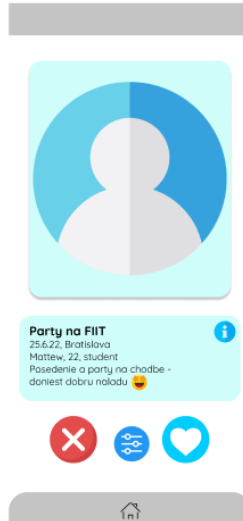
Home page 2.0



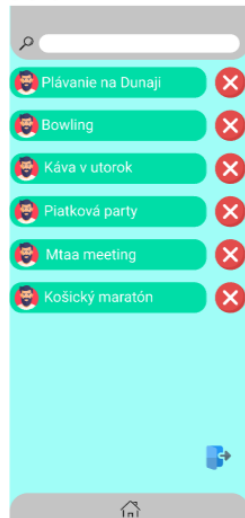
attendant list event



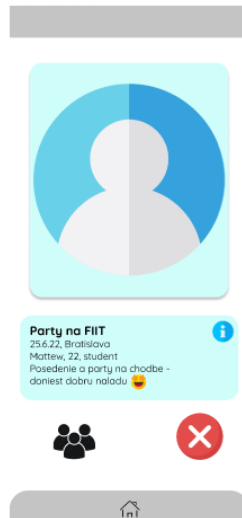
Swipe Eventov



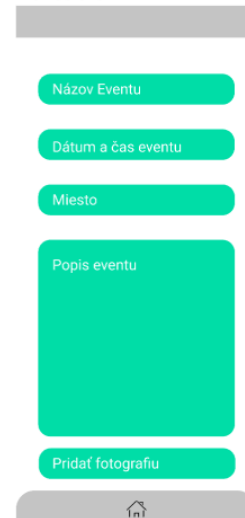
events



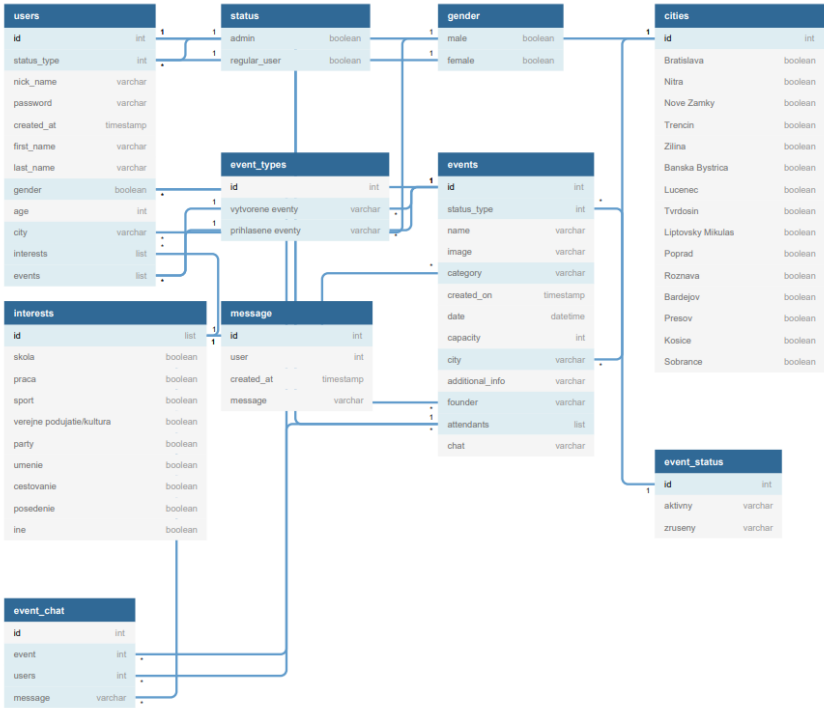
Prehľad eventu



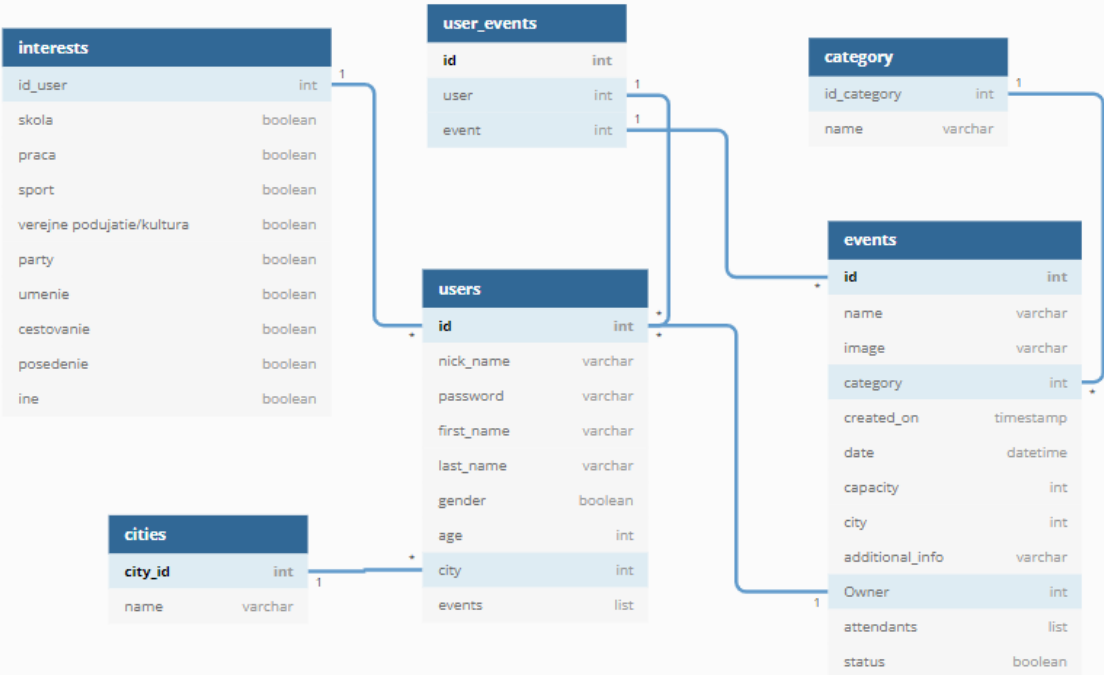
Prehľad eventu



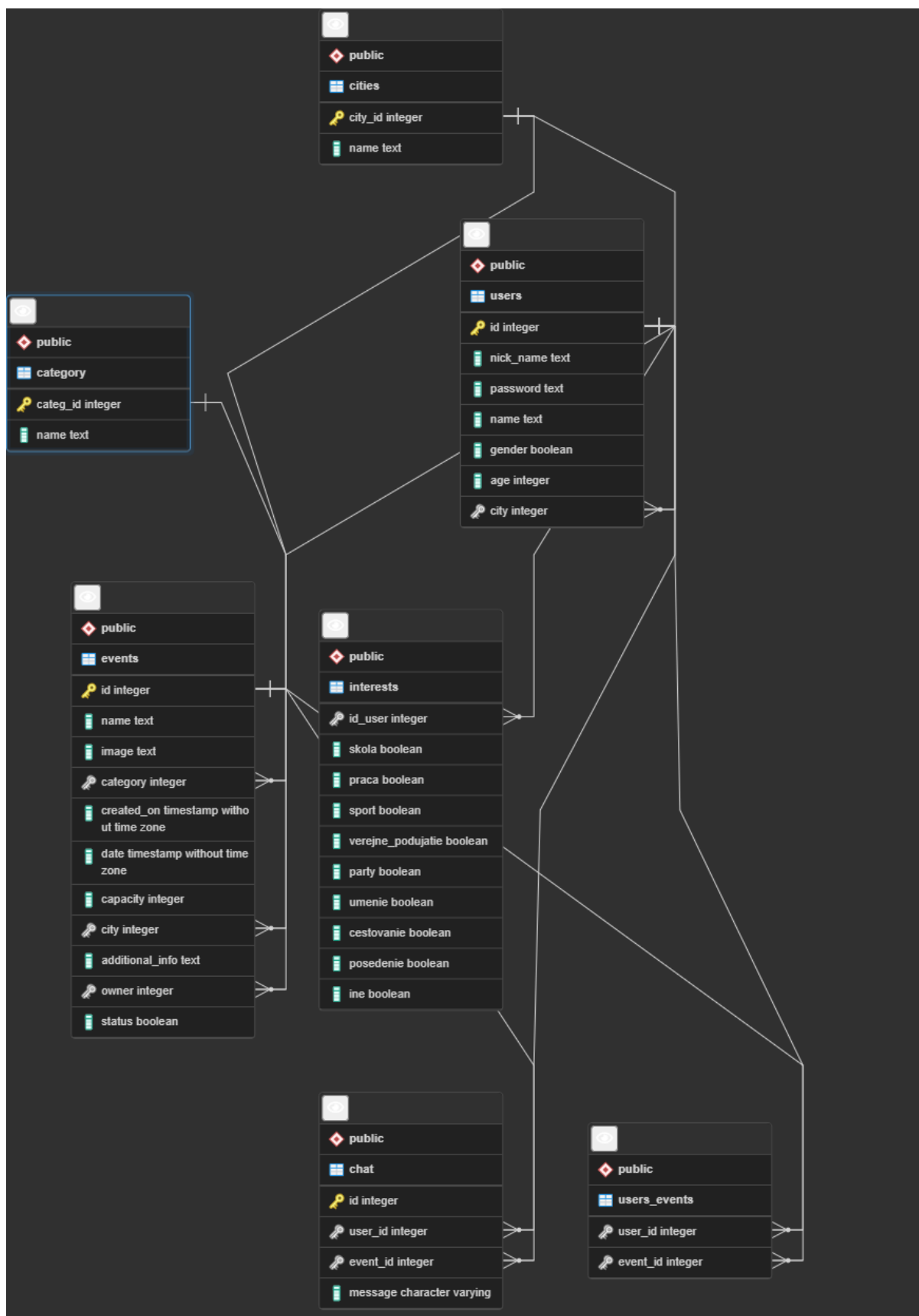
Dátový model



Dátový model úprava



Dátový model finálna úprava



Úpravy a pridanie rest API volaní

POST /api/signEvent - pridanie noveho eventu do tabuľky users_events

POST /api/login/:nick_name - po úspešnom zalogovaní zistím id

používateľa z jeho nicku

POST /api/login - overenie loginu pouzivatela pri zadani nicku, hesla do
gui

PUT /api/updateUser/:id - updatenem info o userovi

GET /api/swipeEvents/:id - vypíšem si eventy, ktore mozem pozerat a
prihlasiť sa na ne

GET api/Event/Attendants/:id - vypíšem si userov, ktorí su prihlaseni na
event

GET /api/myEvents/:id - vypíšem si eventy, na ktore je user prihlásený

GET /api/Event/:id - vypíšem si informacie o evente

GET /api/user/:id - vypíšem si informacie o userovi

POST /api/createEvent - vytvorím nový event

POST /api/register - vytvorím noveho pouzivatela

POST /api/register/check - skontrolujem či už používateľský nick existuje
v databáze

DELETE /api/deleteUser/:id - vymažem usera

DELETE /api/deleteEvent/:id - vymažem event