

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

SKIALPI

DOKUMENTÁCIA
JURAJ ŠIMEK

ZADANIE

Vo vami zvolenom prostredí vytvorte databázovú aplikáciu, **ktorá komplexne rieši minimálne 5 scenárov** vo vami zvolenej doméne. Presný rozsah a konkretizáciu scenárov si dohodnete s Vaším cvičiacim na cvičení. Aplikácia musí realizovať tieto všeobecné scenáre:

- Vytvorenie nového záznamu,
- Aktualizácia existujúceho záznamu,
- Vymazanie záznamu,
- Zobrazenie prehľadu viacerých záznamov (spolu vybranou základnou štatistikou),
- Vyhľadanie konkrétneho záznamu.

Aplikácia môže mať konzolové alebo grafické rozhranie (nie je kladený dôraz na GUI a tento aspekt sa nebude hodnotiť).

SPRESENIE ZADANIA

Na predmet databázové systémy som sa rozhodol riešiť systém na ohodnocovanie úspechov v skialpinizme. Podobných systémov je, čo sa týka horolezectva, viacero. Najznámejší je snád climb.sk. Pre potreby skialpinizmu však podobný systém neexistuje. Tento systém som sa čiastočne rozhodol riešiť i na základe žiadosti o vytvorenie takéhoto systému zo strany jedného nemenovaného horolezeckého klubu, ktorý sa venuje i skialpinizmu.

Systém bude vedieť realizovať nasledovnú funkcionlitu:

Ľubovoľný človek sa do systému môže zaregistrovať, pričom po registrácii môže do systému pridávať zjazdy. Každý zjazd patrí do určitej oblasti, pričom oblasť je súčasťou väčšej lokality. Zjazd má rôzne parametre ako výška (nadmorská výška daného bodu, kde sa nachádza), obtiažnosť, čo predstavuje to, ako je zjazd náročný a expozícia, ktorá predstavuje psychickú náročnosť zjazdu. Podrobnosti o hodnotení obtiažnosti zjazdov nájdete na stránke <http://www.miropeto.sk/sprievodca/vseobecne-informacie-o-sprievodcovi/>.

Každý registrovaný používateľ má denníček, do ktorého si môže pridávať svoje úspechy. Tieto úspechy predstavujú zjazdy, ktoré sa mu podarilo zlyžovať. Zaznamenáva si i dátum, kedy daný zjazd zlyžoval. Používateľovi je potom sú potom na základe istého algoritmu pridelené body za tento úspech. Tak sa tvoria rebríčky najlepších lyžiarov, ktoré možno sledovať.

Neregistrovaný používateľ si môže prezerať zjazdy evidované v systéme, takisto i rebríčky, no nemôže zjazdy pridávať ani nijako meniť obsah systému.

Ako implementačné prostredie som si zvolil jazyk Ruby na frameworku Ruby on Rails.

Systém umožňuje lyžiariom evidovať to, aké zjazdy zdolali a následne ich ohodnotí podľa náročnosti zjazdu. Takto užívateľ vidí to, aké trasy prekonal v minulosti, jeho úspechy, vidí to, ako postupom času zdolával väčšie výzvy a vidí to, ako je na tom v porovnaní s ostatnými lyžiarmi registrovanými v systéme.

Systém robí štatistiky pre každého lyžiara osobitne, ale aj rebríčky v rámci všetkých registrovaných používateľov. Keďže sú kluby, ktoré by si radi viedli štatistiky len medzi ich členmi, v budúcnosti by bolo možné túto vlastnosť sprístupniť za nejaký finančný poplatok.

Rozhodol som sa teda riešiť nasledovné scenáre:

REGISTRÁCIA POUŽÍVATEĽA + INFORMÁCIE O ŇOM

- používateľ sa zaregistruje v systéme
- vyplní svoje kontaktné informácie
- prezeranie profilov používateľov

PRIDANIE ZJAZDU DO DATABÁZY A NÁHĽAD ZJAZDU

- užívateľ pridá zjazd
- ohodnotí jeho náročnosť (i napriek tomu, že užívatelia náročnosť ohodnotia takmer vždy správna, bolo by dobré implementovať mechanizmy, ktoré by riešili omyly)
- zjazdy si možno prezerať v tabuľke
- pre každý zjazd možno pozrieť jeho detail

ADMINISTRÁCIA DENNÍČKA

- každý užívateľ má denníček, kde si eviduje svoje zjazdy
- pridanie zjazdu (aký zjazd, kedy, komentár)
- systém zjazd bodovo ohodnotí podľa jeho náročnosti
- užívateľ vidí všetky svoje zjazdy (možno spraviť i filtrovanie: v tomto roku, v roku XXXX, celkovo)
- rebríček zjazdov (top 10 najlepších)

ŠTATISTIKY

- užívatelia môžu sledovať rebríčky
- môžu sa dostávať na lepšie pozície v rámci nich
- zobrazovanie najlepších užívateľov

NAHLÁSENIE CHYBY

- používateľ môže nahlásiť nezrovnalosť v ohodnotení trasy
- používateľ môže nahlásiť chybu

Alternatívne scenáre, ktoré ale nebudú v súčasnej verzii implementované:

VYTVORENIE SKUPINY

- po zaplattení umožní vytvoriť skupinu používateľov

KOMENTOVANIE ZJAZDOV A PRIDÁVANIE FOTOGRAFIÍ

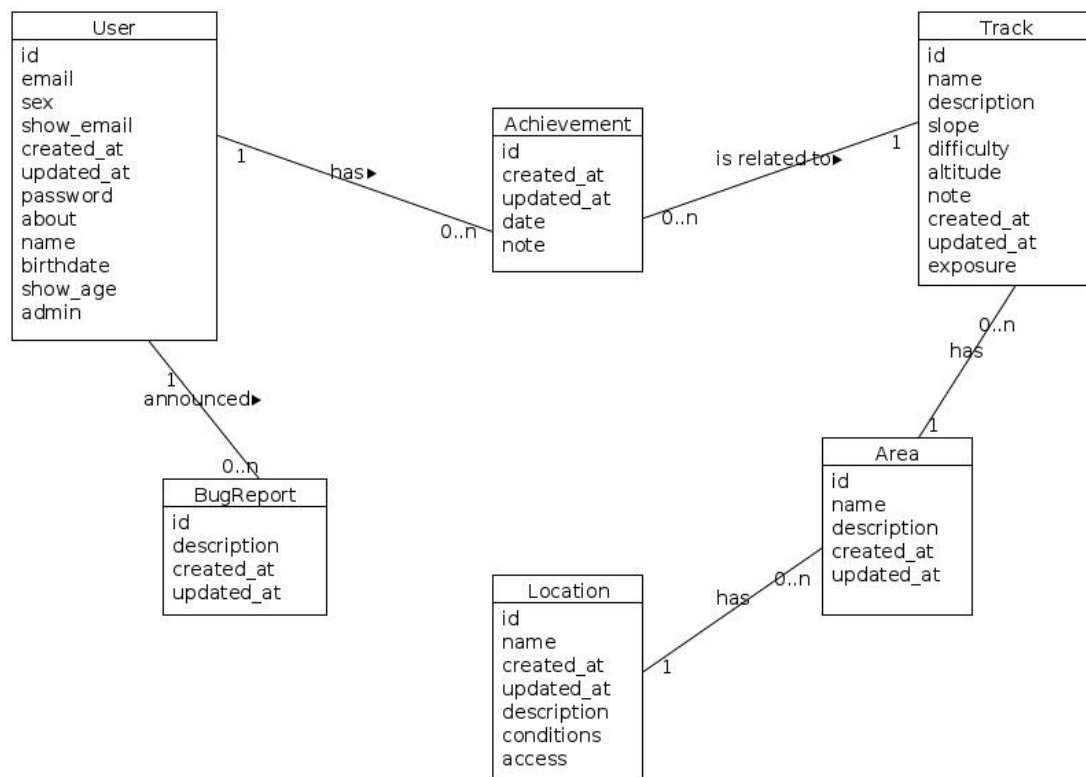
- používateľ môže pridať komentár ku zjazdu a pridať fotografie zjazdu

KOMENTOVANIE ÚSPECHOV POUŽÍVATEĽOV

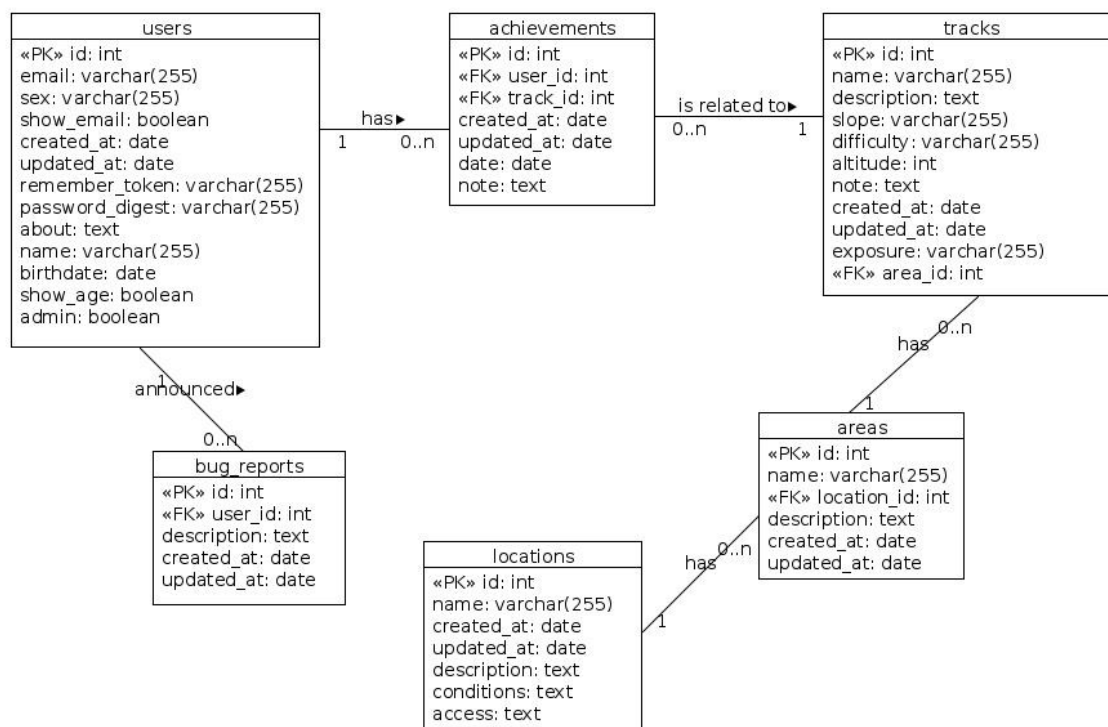
- používateľ môže komentovať úspechy iného používateľa, ktoré sú evidované v jeho denníčku

MODELÝ

LOGICKÝ MODEL



FYZICKÝ MODEL



OPIS

Používateľa (User) charakterizujú atribúty uvedené v tabuľke users. Každý používateľ sa môže rozhodnúť, či chce, aby bol jeho email viditeľný ostatným používateľom. Na to slúži atribút show_email. Atribút password_digest obsahuje hashovanú verziu hesla použitého na prístup používateľa do systému. Ak používateľ nechce zverejniť v systéme svoj vek, nastaví atribút show_age na hodnotu false. Atribút remember_token obsahuje hodnotu určenú na autentifikáciu používateľa pomocou cookies.

Každá trať (Track) uvedená v tabuľke tracks má meno, popis, sklon a ostatné v diagrame uvedené atribúty. Trať patrí do istej oblasti, ktorá je súčasťou nejakej lokality.

Používateľ má možnosť evidovať svoje úspechy pomocou tabuľky achievements. Na základe evidovania toho, aký zjazd zlyžoval systém bude vedieť následne bodovo ohodnotiť jeho výkon a z týchto záznamov bude vedieť počítať i rebríčky. Pri implementácii tohto scenára možno bude potrebné dodať i ďalšie atribúty, nakoľko táto časť systému ešte nie je plne špecifikovaná. Na zrýchlenie výpočtu celkových bodov používateľa bude možno lepšie denormalizovať databázu tým, že pri každom používateľovi si budeme pamätať počet jeho celkových bodov (prípadne i počet bodov dosiahnutých v danej sezóne prostredníctvom ďalšej tabuľky).

Používateľ môže oznámiť chyby v systéme. Toto oznámenie bude uložené v tabuľke bug_reports.

OPIS NÁVRHU A IMPLEMENTÁCIE

PROGRAMOVACIE PROSTREDIE

Za programovacie prostredie som si zvolil programovací jazyk Ruby vo verzii 1.9.3 spolu s webovým frameworkom Ruby on Rails vo verzii 3.2.12. Tento framework umožňuje O/R mapovanie pomocou ActiveRecord-u, ktoré som v projekte používal.

SYSTÉM

Systém beží pod frameworkom ruby on rails. Používa tému bootstrap od spoločnosti Twitter. Grafický dizajn však prejde v budúcnosti rozsiahlymi zmenami. Na simulovanie dát bol použitý gem faker, ktorý umožňuje vytvárať falošné dáta do databázy na testovacie účely. Systém beží nad databázou postgres zatiaľ len v development móde. Pri implementovaní systému som postupoval podľa návodov z rôznych internetových zdrojov, ale najmä <http://ruby.railstutorial.org/ruby-on-rails-tutorial-book>.

REGISTRÁCIA POUŽÍVATEĽA

Scenár bol implementovaný v súlade s požiadavkami na tento scenár. Používateľ sa môže do systému zaregistrovať, pri registrácii vyplní meno, pohlavie, email, dátum narodenia, heslo, potvrdenie hesla, informácie o sebe a vyberie si možnosti ohľadom zverejňovania niektorých informácií. O kryptovanie hesla sa stará gem bcrypt-ruby.

Po registrácii sa používateľ môže prihlásiť do systému a vykonávať akcie neprístupné

neregistrovanému návštevníkovi stránky.

Administrátor má práva na mazanie používateľov a nimi pridanych dát.

Dáta používateľov sú kontrolované na ich platnosť.

Používateľ je autentifikovaný na základe hodnoty remember_token-u v jeho cookies.

PRIDANIE ZJAZDU

V rámci tohto scenára je umožnené používateľovi pridať lokalitu a oblasti, pričom lokalitu a oblasti možno následne upraviť, alebo administrátorom stránky zmazať. Pri zmazaní oblasti sa oblasť zjazdu nastaví na hodnotu NULL, čiže oblasť nie je evidovaná. Administrátor môže takéto zjazdy následne presunúť do inej oblasti.

Používateľovi je umožnené pridať a ohodnotiť náročnosť zjazdu. Zjazd môže zmazať len administrátor. Tabuľka náročnosti zjazdu je reprezentovaná pomocou hashu v súbore tracks_helper.rb. Nakoľko sa bodové ohodnotenie časom meniť nebude (a ak sa bude, je nutné zmeniť aj bodové ohodnotenie zjazdov vykonaných do daného okamihu) nie je potrebné tieto informácie uchovávať v databáze. Aplikácia sa tým zrýchli.

NAHLÁSENIE CHYBY

Používateľ môže oznámiť chybu, ktorú v systéme našiel. Administrátor môže túto chybu potom po jej opravení odstrániť. Taktiež vidí, kto koľko chýb nahlásil. V prípade falošných správ môže zmazať záškodníckeho používateľa a spolu s ním aj všetky falošne nahlásené chyby.