

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 812 19 Bratislava

Databázové systémy

Zadanie č. 1 – Vytvorenie aplikácie komunitného fóra na relačnou databázou

Marek Kováč

Cvičiaci: Ing. Róbert Móro

Študijný odbor: Informatika

Ročník: 2. Bc

Akademický rok: 2015/2016

1. Znenie zadania

Oficiálne znenie zadania

Vo vami zvolenom prostredí vytvorte databázovú aplikáciu, **ktorá komplexne rieši minimálne 6 scenárov** vo vami zvolenej doméne. Presný rozsah a konkretizáciu scenárov si dohodnete s Vaším cvičiacim na cvičení. Aplikáciu vytvoríte v dvoch iteráciach. V prvej iterácii, postavenej nad relačnou databázou, musí aplikácia realizovať tieto všeobecné scenáre:

- Vytvorenie nového záznamu,
- Aktualizácia existujúceho záznamu,
- Vymazanie záznamu,
- Zobrazenie prehľadu viacerých záznamov (spolu vybranou základnou štatistikou),
- Zobrazenie konkrétneho záznamu,
- Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom.

Spresnenie

Mojou úlohou bolo vytvoriť komunitné fórum, v ktorom budú môcť používatelia vytvárať, editovať a rušiť svoje kontá ako aj príspevky.

2. Špecifikácia realizovaných scenárov

• *Pridanie fóra (Scenár: Vytvorenie nového záznamu)*

Fórum ako také je delené na niekoľko hlavných podfór v kt. sa nachádzajú jednotlivé témy a v týchto témach sa ďalej nachádzajú príspevky, ktoré sú usporiadané do jednotlivých skupín na základe toho či sa jedná o reakcie alebo originálne príspevky na kt. užívateľ reaguje.

Pridanie fóra má za úlohu pridať podfórum do celkového fóra. Takéto podfórum môže byť pridávané len administrátorom, čo je pri pridávaní samotného fóra overené aplikáciou náhľadom do databázy. Pri pridaní podfóra je nevyhnutné vložiť aspoň názov a popis fóra, pričom užívateľ je automaticky zistený pomocou dát o prihlásení. Dátum vytvorenia fóra je automaticky generovaný vložením záznamu do databázy. Fórum bude ďalej možné administrátorom zmazať, prípadne zamknúť nastavením príslušného príznaku.

• *Upravenie konta (Scenár: Aktualizácia existujúceho záznamu)*

Po prihlásení sa do používateľského účtu prostredníctvom príkazu „prihlásenie“ je možné si pomocou príkazu „upravenie konta“ upraviť používateľské konto, a to napríklad zmeniť heslo alebo kontaktnú emailovú adresu.

- **Zmazanie fóra (Scenár: Vymazanie záznamu)**

Pomocou príkazu „zmazanie fóra“ je možné fórum zmazať. Pri mazávaní fóra je potrebné mať admin. oprávnenia inak sa procedúra mazania skončí neúspechom.

- **Zobrazenie prehľadu kont (Scenár: Zobrazenie prehľadu viacerých záznamov (+štatistika))**

Zobrazenie prehľadu kont pomocou príkazu „zobrazenie kont“ zobrazí vš. užívateľov v databáze spoločne s ich email adresami, id a počtom príspevkov. Tiež zobrazí celkový počet registrovaných používateľov.

- **Vypísanie info o akt. prihlas používateľovi (Scenár: Zobrazenie konkrétneho záznamu)**

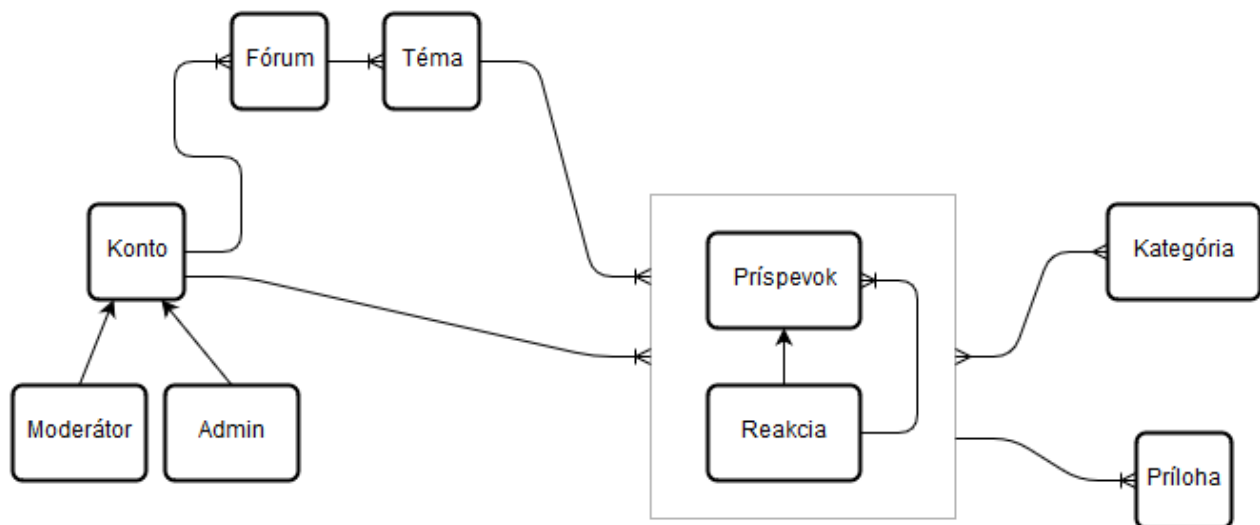
Pomocou príkazu „ktosom“ vypíšeme informácie o akt. Prihlásenom používateľovi. Aby sme sa mohli prihlásiť použije sa príkaz „prihlásenie“, kt. nás prihlási do systému.

- **Zobrazenie kont na základe prednastavených špecifikácií (Scenár: Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom)**

Pomocou príkazu „filtrovanie kont“ môžeme filtrovať konta na základe časového rozsahu ich vytvorenia, ako aj iných parametrov ako napr. min. počtu príspevkov.

3. Diagram logického a dátového modelu

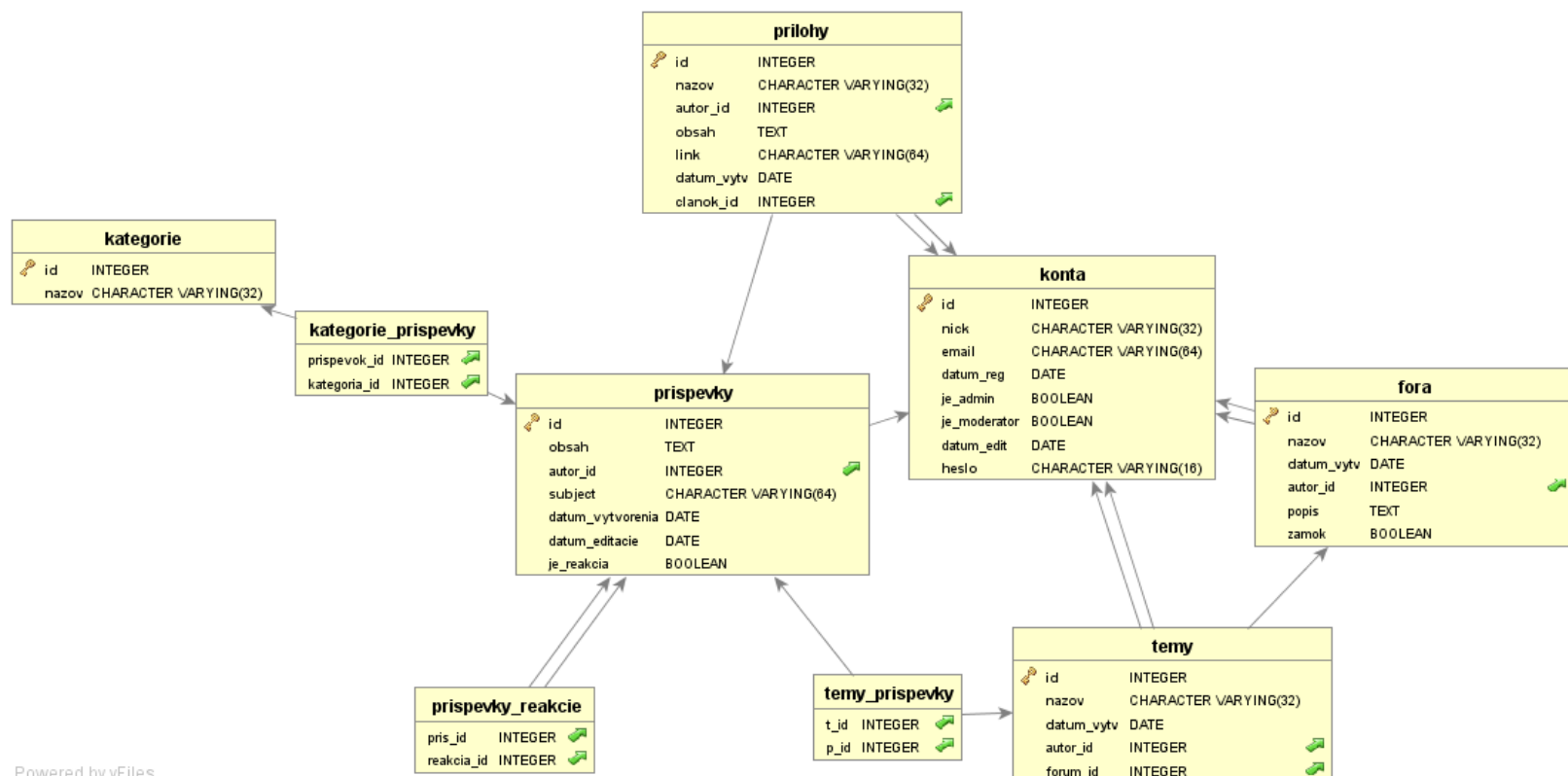
3.1 Logický model



Obr 1. Logický model

- Entita **Konto** – databáza obsahuje viacero kônt, pričom na každé jedno konto môže ale nemusí obsahovať aspoň jeden príspevok. Konto reprezentuje užívateľa, inak povedané jedná sa o užívateľský účet. Tento užívateľský účet sa dá na základe príznaku generalizovať na konto moderátora, konto administrátora alebo obyčajné používateľské konto. Tieto tri typy kont sa neskôr v aplikácii rozlišujú na základe nastavených príznakov.
- Entita **Fórum** – databáza, resp. celé fórum obsahuje niekoľko podfór (tu označené ako fórum) v kt. sa ďalej nachádzajú témy a v nich jednotlivé príspevky utriedené do skupín na základe toho či sa jedná o pôvodný príspevok alebo reakciu na pôvodný príspevok. Jedným užívateľom – administrátorom môže byť vytvorených niekoľko fór, pričom jedno fórum môže obsahovať niekoľko rozličných tém. Jedna konkrétna téma môže sa môže nachádzať iba v jednom podfóre.
- Entita **Téma** – hierarchický prvok na nižšej úrovni, ktorý obsahuje príspevky. Jedna téma sa môže nachádzať súčasne iba v jednom konkrétnom fóre, ale samotná téma môže obsahovať niekoľko príspevkov resp. žiadny.
- Entity **Príspevok a Reakcia** – Samotné príspevky sú obsiahnuté v témach, pričom jeden príspevok môže byť buď originálnym príspevkom alebo reakciou na príspevok. Príspevky ako také sú logicky združené v podobe jedného hlavného príspevku a následne výlučne reakciami na tento príspevok. Jeden príspevok môže byť vytvorený a editovaný iba jedným kontom, pričom to isté konto môže vytvoriť niekoľko príspevkov. Takisto jeden príspevok sa môže nachádzať iba v jednej téme.
- Entita **Kategória** – Príspevky je možné kategorizovať, pričom jeden príspevok môže mať niekoľko dopredu vytvorených kategórií a jedna kategória môže byť použitá u viacerých príspevkov.
- Entita **Príloha** – K jednotlivým príspevkom je možné pridávať prílohy. Na jeden príspevok je možné mať niekoľko príloh, pričom na jedna konkrétna príloha sa môže nachádzať len v jednom príspevku.

3.2 Fyzický model



Obr 2. Fyzický model

- Tabuľky **kategorie_prispevky**, **prispevky_reakcie** a **temy_prispevky** su väzobné tabuľky.
- Tabuľka **kategorie** – obsahuje primárny kľúč a názov kategórie, kt. je potrebný pre identifikáciu kategórie. Každá použitá kategória v príspevku by sa mala vložiť do väzobnej tabuľky **kategorie_prispevky** s potrebným id príspevku.
- Tabuľka **prilohy** – obsahuje PK, cudzí kľúč na ID autora, kt. je tento identifikovaný. Obsah definuje obsah prílohy pokiaľ sa jedná o dlhý text, resp. sa dá použiť link, kde sa uvedie URL zdroja. **clanok_id** sa odkazuje do tabuľky príspevkov. **datum_vytv** je generovaný automaticky pri vytvorení nového záznamu.
- Tabuľka **prispevky** – obsahuje text príspevku(**obsah**), predmet(**subject**), odkaz na autora(**autor_id**) a **datum_vytv** – generovaný automaticky pri vytvorení záznamu, **datum_editacie** – dátum posl. editácie jeho spravovanie má na starosti aplikácia. Príznak **je_reakcia** definuje či sa jedná o reakciu na už existujúci príspevok, čo sa využíva pri výpise príspevkov.

- Tabuľka **konta** – obsahuje informácie o regist. užívateľoch. Nick definuje meno autora, email je kontakt, heslo je automaticky generované pri registrácii (vytv. Nového záznamu) a je potrebné ho meniť. Datum_reg je vytvorený pri registrácii a nastavený na súčasný dátum. Príznamy je_moderator a je_admin definujú či sa jedná o moderátora a/alebo administrátora.
- Tabuľka **temy** – obsahuje informácie o témach. tj. názov, autora a dátum vytvorenia, kt. je generovaný pri vložení záznamu.
- Tabuľka **fora** – podobne ako temy. Navyše obsahuje príznak zámku, kt. je štandardne odomknutý a následne môže byť administrátorom uzamknutý.

4. Návrh a implementácia

Projekt bol vytvorený v programovacom jazyku JAVA s použitým IDE Eclipse Mars 2 na platforme Microsoft Windows 8. V projekte som na pripojenie sa k databáze využil ovládač JDBC. Pri tvorbe aplikácie som používal databázu PostgreSQL.

Implementácia jednotlivých scenárov

Všeobecný opis

Jednotlivé scenáre sú implementované s použitým transakcií, ktoré sú nastavené pomocou príznaku `spojenie.setAutoCommit(false)`; , ktorý zabezpečí zapnutie transakcií. Po každom vykonaní transakcie sa požadovaný príkaz, ktorým sa vykonal dotaz do databázy ako aj samotné spojenie ukončia.

Aplikácia obsahuje niekoľko pomocných tried v balíku (databaza) obsahujúcich potrebné statické metódy poskytujúce článkovú funkcionálnosť pre jednotlivé scenáre.

Napríklad scenár „registrácia používateľa“ najprv overí či dané konto pre kt. boli vložené parametre už neexistuje a to pomocou príslušnej metódy `existujeKonto` v triede `KontaManager`, ktorá vráti objekt používateľa v prípade existencie konta, následne sa vykoná metóda `vytvorKonto` tej istej triedy, kt. nadviaže opätovne kontakt z databázou a vložením dopredu užívateľom zadaných údajov zaregistruje užívateľa do databázy.

Popis implementácie vybraných scenárov

- **Pridanie fóra (Scenár: Vytvorenie nového záznamu)**

Scenár pridania fóra sa inicializuje vložení príkazu „`pridanie fora`“ na príkazovom riadku. Tento príkaz je následne spracovaný v triede `Spracovanie` z balíka `aplikacia`. Najskôr dôjde k overeniu toho či je užívateľ, ktorý vložil tento príkaz riadne zaregistrovaný pomocou pomocného objektu `Prihlásenie` uchovávajúceho akt. stav o prihlásení. Pokiaľ sa meno prihláseného konta rovná reťazcovej konštante „guest“ jedná sa o neprihláseného používateľa a celý scenár končí. V opačnom prípade sa prostredníctvom volania databázy cez metódu `existujeKonto` zistia informácie

o práve prihlásenom používateľovi ako je napr. príznak admin, keďže len adminy majú povolenie vytvárať nové fóra. Po úspešnom overení systém vyzve používateľa aby zadal informácie obsahujúce popis nového fóra, následne aplikácia pokračuje vložением týchto údajov do databázy volaním metódy vytvorFórum.

- **Upravenie konta (Scenár: Aktualizácia existujúceho záznamu)**

Scenár upravenia konta sa začína príkazom „upravenie konta“ na príkazovom riadku. Tento príkaz je následne spracovaný v triede Spracovanie z balíka aplikacia. Následne dôjde pomocou pomocného objektu triedy Prihlásenie k overeniu či je užívateľ vôbec prihlásený a to pomocou porovnania mena akt. používateľa s reťazcom „guest“. V prípade ak sa rovnajú užívateľ nie je prihlásený a teda nie je čo upravovať. V inom prípade je používateľ vyzvaný vložiť typ editácie a táto je následne vykonaná prostredníctvom metódy upravKonto, ktorá pozmení danú položku v databáze.

- **Zmazanie fóra (Scenár: Vymazanie záznamu)**

Zmazanie fóra sa inicializuje príkazom „zmazanie fora“ na príkazovom riadku. Najskôr užívateľ zadá id fóra, ktoré chce zmazať – toto môže získať pomocou príkazu „zobrazenie fora“. Zmazanie ďalej prebieha overením či je konto administrátorske s pomocou objektu Prihlásenie. Ak áno, tak sa pomocou metódy zmaž záznam fórum zmaže volaním do databázy. V prípade, že sa snažíme fórum zmazať z neadministrátorského konta, sme o tom upozornení a scenár končí.

- **Zobrazenie prehľadu kont (Scenár: Zobrazenie prehľadu viacerých záznamov (+štatistika))**

Zobrazenie prehľadu kont sa inicializuje príkazom „zobrazenie kont“. Tento príkaz najprv volaním do databázy cez metódu zistiInfoKonta získa celkový počet registrovaných užívateľov v databáze a následne pomocou zistiPocty získa prehľadnú tabuľku o celkovom počte registrovaných používateľov v databáze spoločne s prehľadom ich mien, emailových kontaktov ako aj počtov príspevkov na daného používateľa. Týmito údajmi vyššie menovaná metóda naplní zoznam objektov typu UserLoaded, kt. aj vypíše.

- **Vypísanie info o akt. prihlas používateľovi (Scenár: Zobrazenie konkrétneho záznamu)**

Príkazom „ktosom“ sú zobrazené informácie o akt. Používateľovi, tento scenár sám osobe nenadväzuje žiadne spojenie z databázou ale pomocou prihlásenie naplní objekt Prihlásenie z databázy údajmi o aktuálne prihlásenom používateľovi.

- **Zobrazenie kont na základe prednastavených špecifikácií (Scenár: Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom)**

Príkazom „filtrovanie kont“ sa vykonáva filtrovanie záznamov kont pomocou parametrov dátumu a min. počtu príspevkov na základe autora. Scenár je riešený dopytom do databázy, kde ako parametre sú použité informácie vložené používateľom do databázy.