### Slovenská technická univerzita v Bratislave

### Fakulta informatiky a informčných technológií

# Zadanie (1. iterácia)

## Databázové systémy

## Peter Kaňuch

## 2015/2016

## 08.04.2016

# Zadanie

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 Vašim 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.

Aplikácia môže mať konzolové alebo grafické rozhranie. Je dôležité aby scenáre boli realizované realisticky - teda aby aplikácia (a teda aj jej používateľské rozhranie) naozaj poskytovala časť funkcionality tak, ako by ju očakával zákazník v danej doméne.

Scenáre, ktoré menia dáta musia byť realizované s použitím transakcií a aspoň jeden z nich musí zahŕňať prácu s viacerými tabuľkami (typicky vytvorenie záznamu a naviazanie cudzieho kľúča).

# Prehľad

Táto aplikácia ponúka jednoduchú evidenciu klientov v kúpeľoch. Najdôležitejšie funkcie sú:

* Pridávanie klientov a ich pobytov v kúpeľoch, taktiež procedúr a doktorov
* Zobrazenie klientov podľa určitého scenára
* Zobrazenie údajov konkrétneho klienta
* Vyhľadávanie klienta podľa priezviska a rodného čísla
* Editovanie údajov klienta a jednotlivých pobytov
* Prepustenie lekára

# Opis scenárov - Kúpele Medzilaborce

Klient sa zaregistruje na pobyt do kúpeľov poslaním e-mailu alebo telefonicky. Do aplikácie ho pridá pracovník, ktorý na karte Clients zvolí tlačidlo Add client, vyplní údaje a pridá klienta (ešte u nás nebol). Ak u nás už klient bol alebo už je pridaný stačí ho vyhľadať podľa priezviska alebo rodného čísla v zozname klientov kliknúť na riadok a pridať pobyt vyplnením dátumov a potvrdením. Následne sa takto pridaný klienti zobrazia na karte Arrivals podľa vybraného dátumu. Tu sa zobrazujú klienti, ktorí majú v daný deň prísť. Pri príchode klienta si ho zamestnanec zobrazí, pridelí mu doktora a izbu zo zoznamov, čím sa stáva riadne ubytovaný. Klient sa už v zozname nebude zobrazovať. Na tejto karte je možné aj aktualizovať alebo doplniť jeho údaje ako meno, priezvisko, dátum narodenia a adresu. Na karte si taktiež môžeme zobraziť všetky aktuálne pobyty.

Na karte Departures vidíme všetkých klientov, ktorí vo vybraný deň majú naplánovaný odchod. Tu ich môžeme odhlásiť z pobytu. Taktiež môžeme odhlásiť neodhlásených klientov do vybraného dátumu, v prípade, že neboli odhlásený.

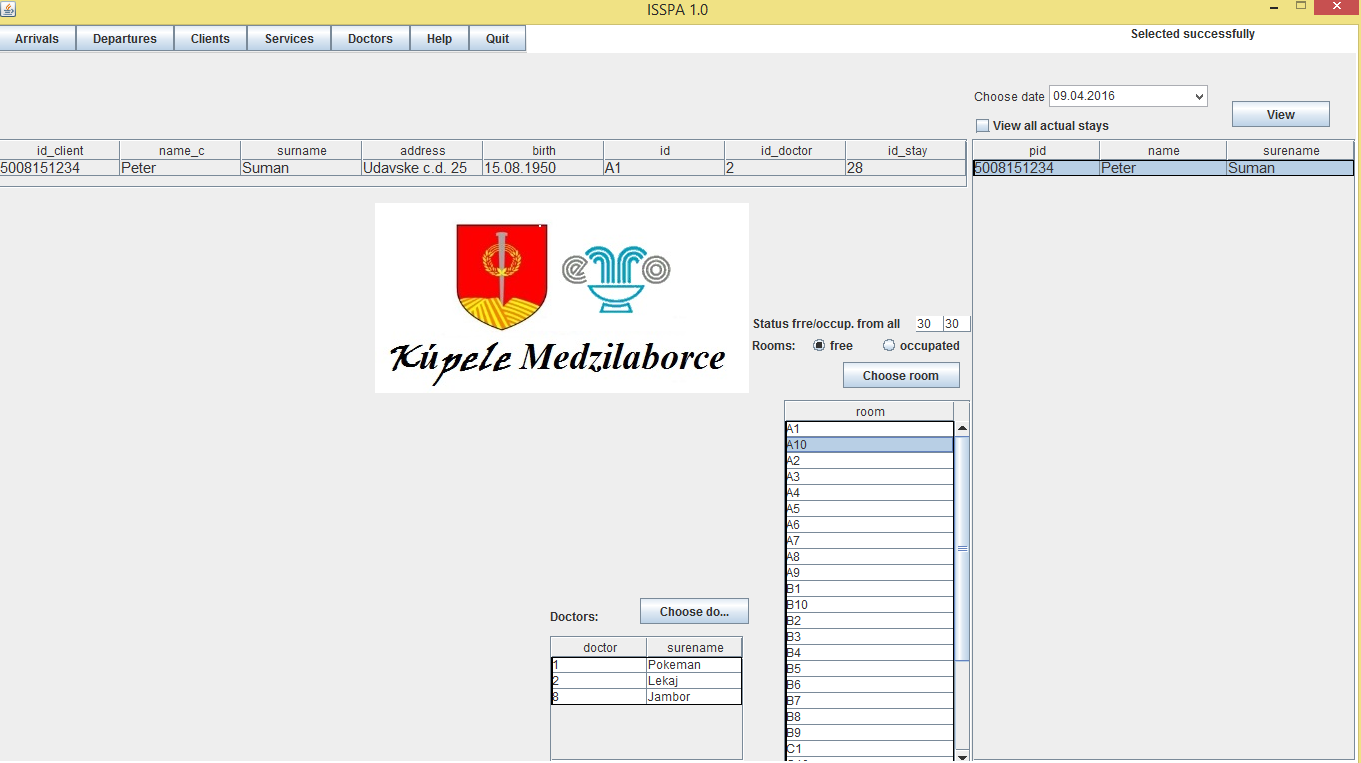
Na karte Clients môžeme vidieť aj všetky doterajšie pobyty jednotlivých klientov.

Karta Services slúži najmä pre manažéra kúpeľov, ktorý môže pridávať alebo prepúšťať doktorov, taktiež pridávať nové procedúry. Tu sa nám zobrazujú aj štatistiky počítané z jednotlivých údajov.

Na karte Doctors sa prihlasujú doktori, ktorí vidia svojich aktuálnych klientov. Tu si vedú o nich záznamy a majú možnosť im prideľovať procedúry.

# Frontend

Používateľské rozhranie je naprogramované v prostredí Java Swing. Obrazovka je statická, ktorá na jednotlivých paneloch zobrazuje komponenty a údaje podľa zadefinovaných scenárov. Celý kód je v Grafika.java. Kód je jednoduchý. S backendom komunikuje prostredníctvom jednotlivých funkcií.



# Backend

Aplikácia je naprogramovaná v jazyku Java s použitím databázy PostgreSQL. Celá funkcionalita sa nachádza v balíku API. Jednotlivé funkcie, ktoré komunikujú s frontendom sú v API.fcie.java.

Pre vyhľadávanie, editovanie a vkladanie dát do databázy sú použité SQL query. Spojenie s databázou je realizované len raz a to pri spustení a uzatvára sa pri ukončení aplikácie.

### Splnenie požiadaviek

* transakcia:

conn.setAutoCommit(**false**);

API.UpdateDb.*Update*(sql2);

API.UpdateDb.*Update*(sql3);

conn.commit();

* použitie JOIN:

"SELECT c.id\_client, c.name\_c, c.surname, c.address, s.id FROM Client c INNER JOIN Stay s ON c.id\_client = s.ID\_Client WHERE s.id\_stay = '"+ table\_click+ "'; "

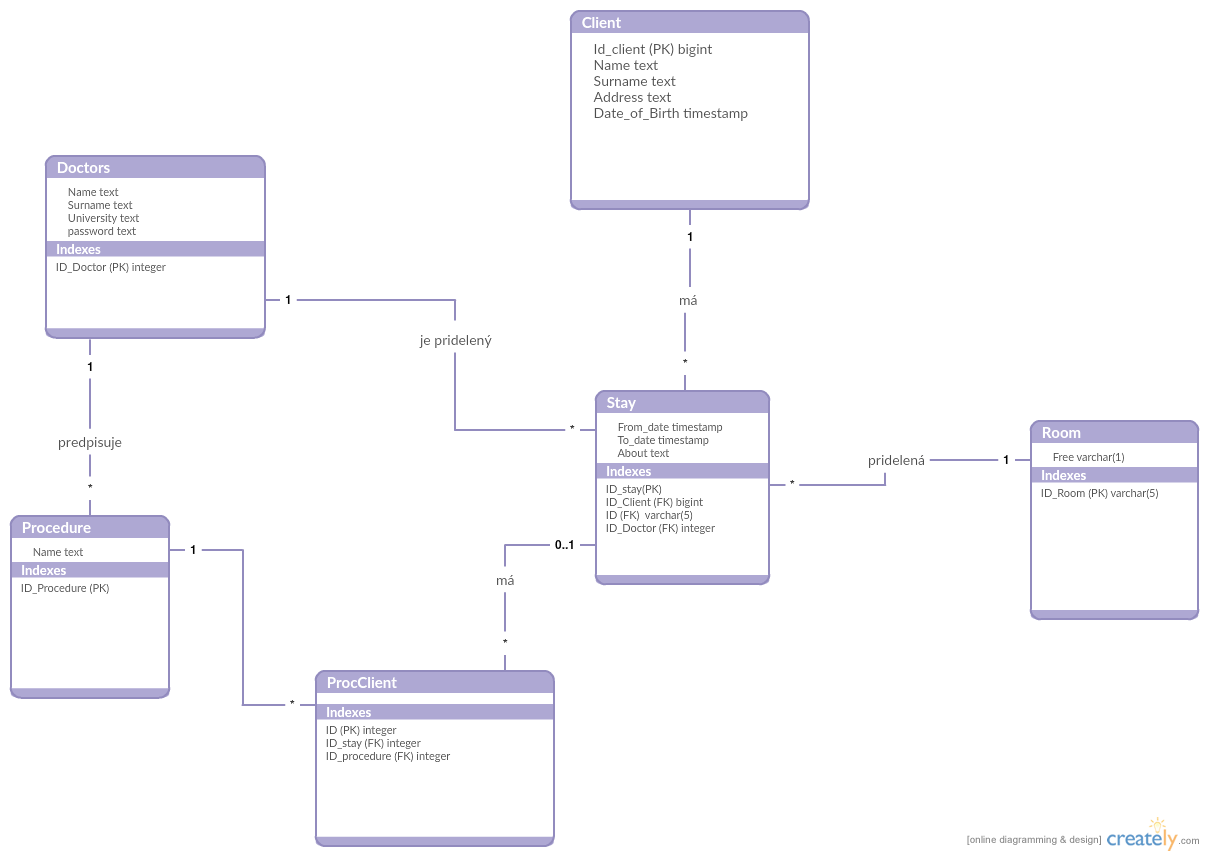
* použitie GROUP BY:

"SELECT name\_c, count(\*) FROM client GROUP BY name\_c ORDER BY count Desc LIMIT 5;"

* agregačná funkcia:

"SELECT round(avg(EXTRACT(YEAR FROM age(date\_birth)))) FROM client;"

# Dátový model



# Zhodnotenie

Aplikáciu som sa snažil navrhnúť a naprogramovať čo najlepšie, s cieľom splniť všetky požiadavky zadania. Je jednoduchá na používanie so základnými poznatkami o nej. Aplikácia nie je ošetrená proti SQL injection, ale prístup k nej má iba zákazník.

Na tomto zadaní som sa naučil pracovať s relačnou databázou SQL.