

Dokumentácia projektu

Databázové systémy

Ak. rok 2016-2017

Meno: **Tomáš Belluš**

E-mail: tomas.bellus@gmail.com

Cvičiaci: **Ing. Ondrej Kachman**

Cvičenie: **štvrtok 13:00**

1. Úvod

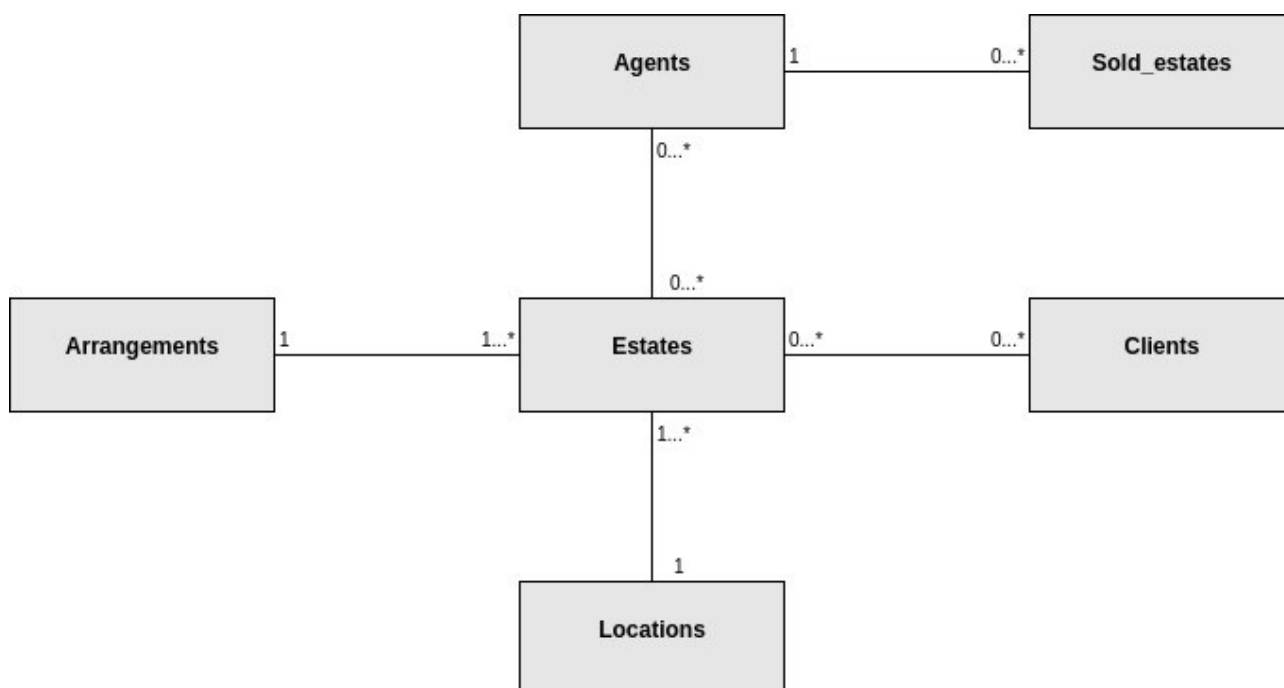
Ide o aplikáciu na spravovanie databázy pre zvolenú doménu. Účel bola práca s databázou cez aplikáciu a sfunčnit' ju s viacerými scenármi, ktoré môže používateľ spúšťať či využívať. Základne naimplementované scenáre sú zobrazenie prvkov z každej tabuľky (ďalej záznam), zobrazenie prvkov na základe nejakého filtra, pridanie nového prvku do hocijakého záznamu ako aj odobranie (ak je to povolené databázou), zobrazenie detailných informácií o každom zázname s prípadným extra scenárom a úprava atribútov vybraného prvku.

1. Zvolená doména

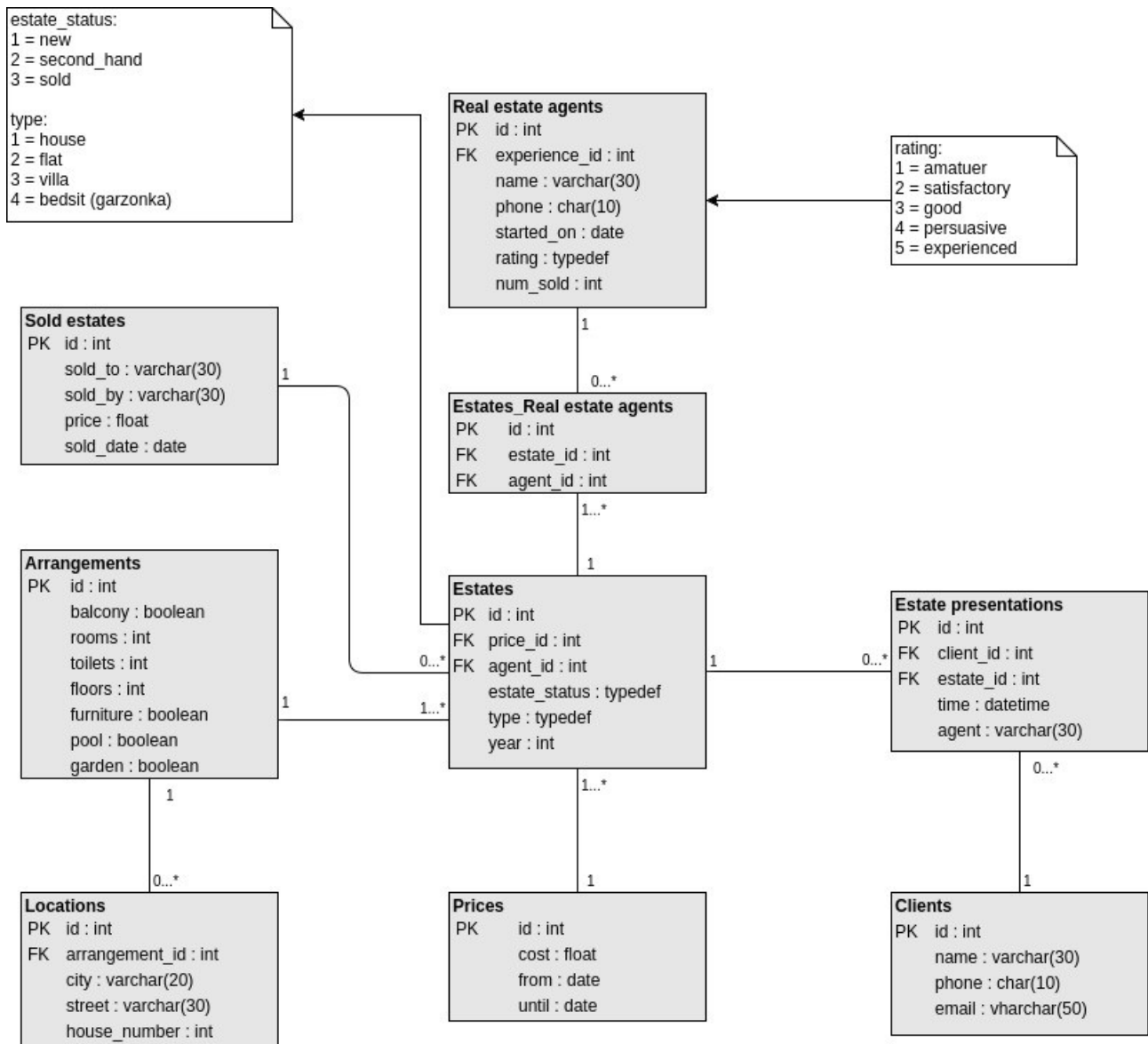
Rozhodol som sa vytvoriť databázovú aplikáciu v doméne poskytovania nehnuteľností realitnými maklérmi. Ide o vhodnú tému, kde sa dá nájsť mnoho scenárov na filtrovanie zobrazenia nehnuteľností a ich vlastností a vzťahy s ostatnými záznamami.

Táto databáza nebude priamym odrazom reálnej realitnej agentúry. Predstava je evidovať nehnuteľnosti, realitných maklérov, lokality nehnuteľností, rôzne zostavy, záujemcov.

Logický model:



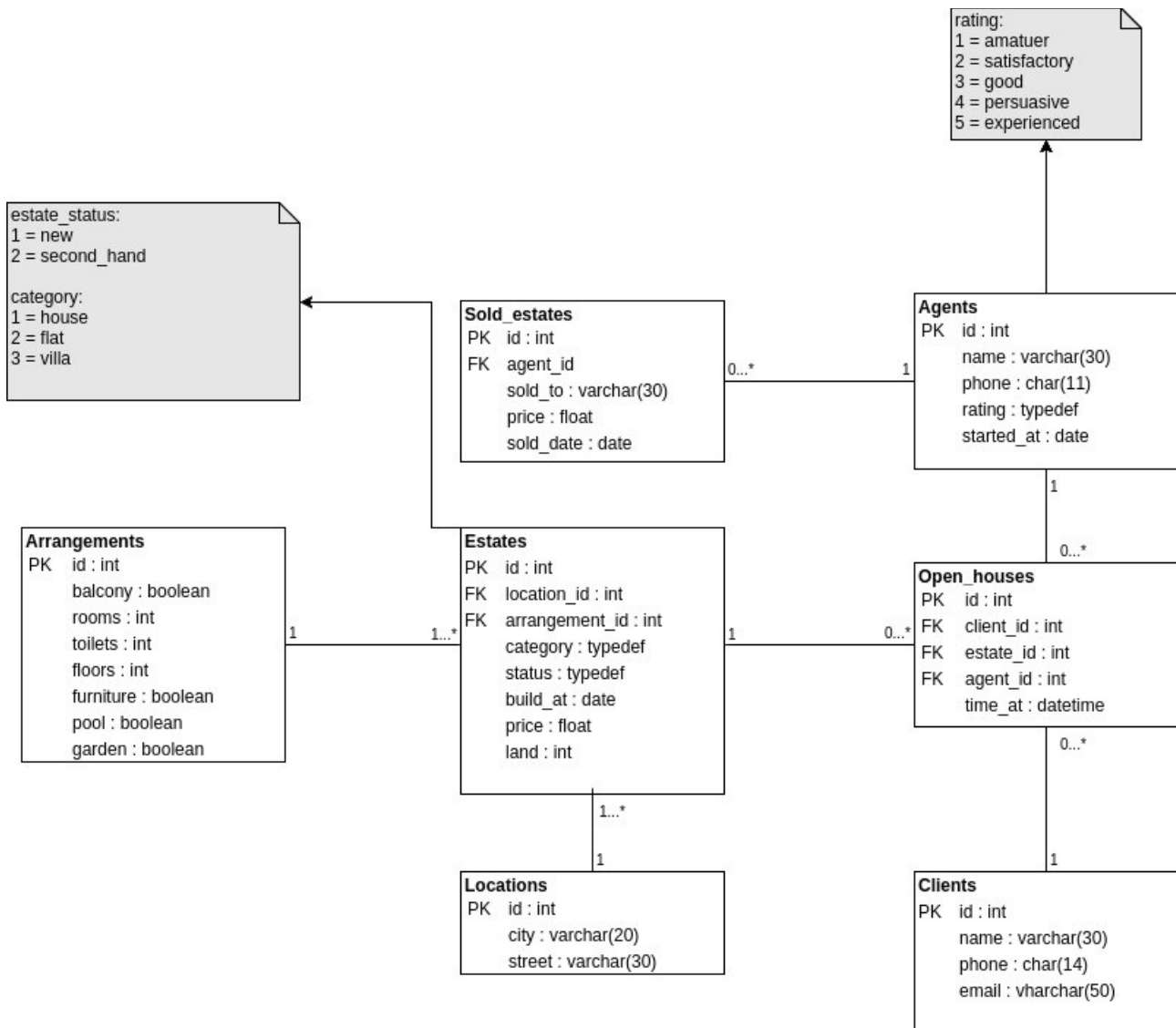
Fyzický model:



V danom fyzickom modeli je otázna evidencia *Sold estate*, ktorá má nesprávne umiestnenie a bude prelinkovaná k realitným maklérom (*agents*). Rovnako aj záznam *prices* sa zaleží do záznamu *estates* kde bude figurovať už iba ako atribút *price*.

2. Finálny datový model

Po prekonzultovaní a pridaní spomínaných zmien a počas generovania dát som prišiel na zopár prekážok, kvôli ktorým som zmenil fyzický model a kardinality. Zmeny je vidieť vo fyzickom modeli nižšie, ktorý je už finálny a podľa neho je zrealizovaná aplikácia.



2. Implementácia databázy a aplikácie

1. Databáza

Databázu mám implementovanú v PostgreSQL v programe na spravovanie PgAdmin3.

2. Aplikácia

Aplikáciu mám implementovanú v Jave s MVC paternom. GUI som implementoval a vytváral použitím plug-inu pre IntelliJ IDEA *JFormDesigner*. Záznamy z databázy mám namapované použitím frameworku Hibernate. Ide o Objektov-Relačné mapovanie, kde hibernate vyžaduje xml

súbor **hibernate.cfg.xml**, ktorý slúži hibernatu ako mapovacia pomôcka pre správne namapovanie do takzvaných persistentných tried v java, kde sú všetky atribúty záznamov namapované.

3. Implementácia a scenáre

Pre namapovanie mám vytcorenú triedu Hibernate, kde sa cez *Configuration* objekt vytvorí *SessionFactory*, ktorý použije hibernate configuračný súbor (hibernate.cfg.xml) pre vytvorenie spojenia. Je práve jeden *SessionFactory* objekt pre jednu databázu, ktorý sa využíva na vytvorenie fyzickej komunikácie s databázou cez *Session* objekt v spojení s transakciou. Pri každom vykonávaní nejakého scenára sa vytvorí nový *Session* a *Transaction* objekt pre vykonanie zmien či spustenie ľubovolnej *query*.

Každý jeden ovládač (Controller) má za úlohu sa starať buď o jednu persistentnú triedu (jeden záznam) alebo o všetky naraz ako napríklad triedy *Manage_public* a *Manage_detail*.

Všetky grafické okná som vytvoril cez *JFormDesigner* a zdrojový kód bol vygenerovaný automaticky.

Každý jeden scenár, či komunikácia s databázou vyzerala ako je znázornené v Kód_1.

Kód_1:

```
public void performAction() {
    Session session = factory.openSession();
    Transaction tx = null;
    try {
        tx = session.beginTransaction();
        // do whatever

        tx.commit();
    } catch (HibernateException e) {
        if (tx != null) tx.rollback();
        e.printStackTrace();
        new ErrorMessage(mainWindow, "error msg");
    } finally {
        session.close();
    }
}
```

1. Zobrazenie záznamu

Zobrazené záznamy sa vypisovali do hlavnej tabuľky v grafickom rozhraní (Príloha **Hlavné GUI**). Pre zobrazenie záznamu som mal funkciu *list<Table>()*, kde som použil *query*

```
SELECT *
FROM <Table>
ORDER BY id
LIMIT <limit>
```

ktorej výsledok som vrátil ako zoznam. V aplikácii je toto zobrazenie obmedzené maximálnym počtom zobrazených prvkou konštantou, respektíve ako limit v *query*.

Zobrazené atribúty neboli len vlastné ale aj zo súvisiacich záznamov alebo cez cudzí kľúč. V takom prípade som neupoužil JOIN ale vďaka ORM som to získal cez getter danej triedy.

Zoznam klientov

Id	Name	Phone	E-mail
----	------	-------	--------

Zoznam zostáv

Id	Floors	Rooms	Toilets	Balcony	Furniture	Pool	Garden
----	--------	-------	---------	---------	-----------	------	--------

Zoznam nehnuteľností

Id	Build_at	Category	Status	Land	Price	City	Rooms
----	----------	----------	--------	------	-------	------	-------

Zoznam lokalít

Id	City	Street
----	------	--------

Zoznam prezentácií

Id	Date and time	Location	Agent	Client
----	---------------	----------	-------	--------

Zoznam predaných nehnuteľností

Id	Price	Sold_to	Sold_at	Sold by agent
----	-------	---------	---------	---------------

Zoznam realitných maklérov

Id	Name	Phone	Rating	Started_at
----	------	-------	--------	------------

2. Zobrazenie s filtrom

Zobrazenie prvkov s použitím filtra je vypísané rovnako do hlavnej tabuľky s použitím *query*

```
SELECT *  
FROM <Table>  
WHERE <attribute> = <filter>
```

alebo pre string

```
SELECT *  
FROM <Table>  
WHERE <attribute> like <filter>
```

ktorá vráti všetky nájdené prvky do zoznamu. Atribúty sú rovnaké ako pri bežnom zobrazovaní.

3. Pridanie do záznamu

Pridávanie sa vykonáva na základe vybraného záznamu v hlavnom výbere (comboBox záznamov) a stlačením tlačidla *add new*. Pre pridanie treba vytvoriť nový objekt daného záznamu a ponastavovať atribúty a prípadne cudzie kľúče. *Query*, ktorý sa vykoná je INSERT.

```
INSERT INTO locations(city, street) VALUES (Nitra, Trnavská cesta);
```

Príklad okna pridania novej nehnuteľnosti je uvedený v prílohe **Pridanie nehnuteľnosti**.

4. Vymazanie zo záznamu

Pre vymazanie sa označí daný prvok záznamu a stlačením tlačidla *delete* sa vymaže zo záznamu v prípade, že boli splnené podmienky (constraints) v databáze. Pre vymazanie sú pridané do databázy podmienky a to pre cudzie kľúče, kedy je vymazanie povolené a teda kaskádne alebo je zamietnuté. *Query* pre vymazanie je DELETE.

```
DELETE FROM <Table> WHERE id = <target_id>
```

V prípade vymazania lokality alebo zostavy, ktorá je odvolávaná v zázname nehnuteľností sa nevymažú kvôli podmienke v databáze. V prípade vymazanie ostatných prvkov nejakého záznamu sa podľa podmienok kaskádovito vymažú príslušné prvky v odvolávaných záznamoch. Čiže napríklad pri vymazaní klienta, agenta alebo nehnuteľnosti sa vymaže aj prezentácia prepojená s týmto prvkom.

5. Detail a úprava

Zobrazenie detailu je dvoj-klik na hociktorý prvok vypísaného zoznamu v tabuľke v hlavnom okne. Detailné okno prvkov obsahuje rovnaké atribúty a extra k tomu dodatočné informácie. Zároveň je v tomto okne možné upraviť ktorýkoľvek poskytnutý atribút záznamu na zvolenú hodnotu. *Query* pre uprávu atribútu.

```
UPDATE <Table> SET <attribute> = <new_value> WHERE id = <target_id>
```

Detailné okno je uvedené v prílohe **Detailné okno**.

Detail klienta

V detaile klienta je extra zobrazený počet prezentácií na ktoré má klient prísť. Použitý *query* s GROUP BY:

```
SELECT count(o.id)
FROM clients c
JOIN open_houses o ON o.client_id = c.id
WHERE c.id = <target_id>
GROUP BY c.id
```

Detail zostavy

V detaile zostavy je extra pridaný počet nehnuteľností s danou zostavou, podľa *query*:

```
SELECT count(e.id)
FROM arrangements a
JOIN estates e ON e.arrangement_id = a.id
WHERE a.id = <target_id>
GROUP BY a.id
```

Detail nehnuteľnosti

V detaile nehnuteľnosti je extra pridaný počet prezentácií pre danú nehnuteľnosť, podľa *query*:

```
SELECT count(o.id)
FROM estates e
JOIN open_houses o ON o.estate_id = e.id
WHERE e.id = <target_id>
GROUP BY e.id
```

Detail lokality

V detaile lokality je extra pridaný počet nehnuteľností s danou lokalitou, podľa *query*:

```
SELECT count(e.id)
FROM locations l
JOIN estates e ON e.location_id = l.id
WHERE l.id = <target_id>
GROUP BY l.id
```

Detail prezentácie nehnuteľnosti

V detaile nehnuteľnosti nie je nič extra, keďže ani GROUP BY sa tam nehodí vzhľadom na kardinality.

Detail predanej nehnuteľnosti

Detail predanej nehnuteľnosti neobsahuje žiaden extra scenár.

Detail realitného makléra

Detail realitného makléra obsahuje extra počet predaných nehnuteľností a zárobok spojený s týmto počtom, podľa *query*:

```
SELECT sum(s.price) as earned, count(s.id) as count
FROM agents a
JOIN sold_estates s ON s.agent_id = a.id
WHERE a.id = <target_id>
GROUP BY a.id
```

4. Zhodnotenie

Celkovo sa dá aplikácia zhodnotiť ako funkčná s implementovanými scenármi podľa zadania a to zobrazenie záznamov, zobrazenie s filtrom, pridanie, zmazanie, detail prvku záznamu a úprava atribútov v detailnom zobrazení. Chýba stránkovanie zoznamu, ale to sa dá predísť vyhľadávaním s filtrom. Vo fyzickom modeli dát, záznam predaných nehnuteľností by mal byť prepojený so záznamom nahnuteľností, čo by znamenalo že predané nehnuteľnosti sú tiež evidované, ale v aktuálnom prípade je predaná nehnuteľnosť nevidovaná. Čo sa atribútov týka ako je *rating*, *status* a *category*, mali by byť použité číselníky pre väčšiu konzistenciu a lepšiu udržateľnosť aplikácie.

Implementačne bolo použitie ORM veľmi nápomocné, a to najmä pri práci so súviciacimi atribútmi so spojeného záznamu kedy netreba stále používať JOIN na získanie potrebnej hodnoty. Nevýhodou je čas trvania mapovania záznamou do príslušných perzistentných tried. Vyhľadávanie s filtrom je obmedzené poskytnutými atribútmi alebo dátovým typom, pričom sa nedá vyhľadať s filtrom typu dátum s iba zadaným rokom, mesiacom, alebo dňom.

5. Prílohy

1. Hlavné GUI

Table: Clients display

search: id search by

id	name	phone	e-mail
1	Carl Welch	(727) 444-5045	vaos@uze.com
2	Lola Montgomery	(584) 201-6761	jkwot@ew.gov
3	Myrtle McKinney	(676) 405-3563	zupo@ra.co.uk
4	Hallie Wade	(383) 398-5793	ejape@omhon.net
5	Maude Jacobs	(254) 985-5500	pa@opcorcoz.net
6	Frances Steele	(719) 979-2478	lezivtom@ina.net
7	Gavin Jones	(451) 748-5543	nevsu@jenela.com
8	Vera Reid	(282) 996-2418	doh@ug.gov
9	Ann Peters	(542) 392-9395	cifuj@es.edu
10	Gary White	(271) 900-7755	wok@caj.edu
11	Lester McCarthy	(836) 396-5894	bipewara@wituwi.edu
12	Hunter Stanley	(540) 550-4063	uvideohi@zipuksub.net
13	Teresa Harris	(461) 330-9025	jhcun@vepfagna.co.uk
14	Scott Schmidt	(489) 897-9711	mel@kazew.edu
15	Vera Holloway	(721) 384-7046	et@rukhonfe.gov
16	Jose Matthews	(956) 828-2863	nota@fec.io
17	Alberta McBride	(405) 825-1328	toduek@sabe.gov
18	Hilda Barnes	(537) 889-2178	nekde@tuper.net
19	Hunter Hale	(327) 697-3132	bafo@succok.com
20	Stella Watts	(865) 922-3895	buwug@pas.net
21	Estella Francis	(288) 201-1141	acjigbam@wuna.org

number of records: 200001 add new delete

2. Pridanie nehnuteľnosti

New estate item

name:

status: new

category: villa

land:

build_at:

price:

new location: ☒ or

new arrangement: ☐ or

cancel add

Set location

city:

street:

Set arrangement

rooms: 1

floors: 1

toilets: 1

furniture: with

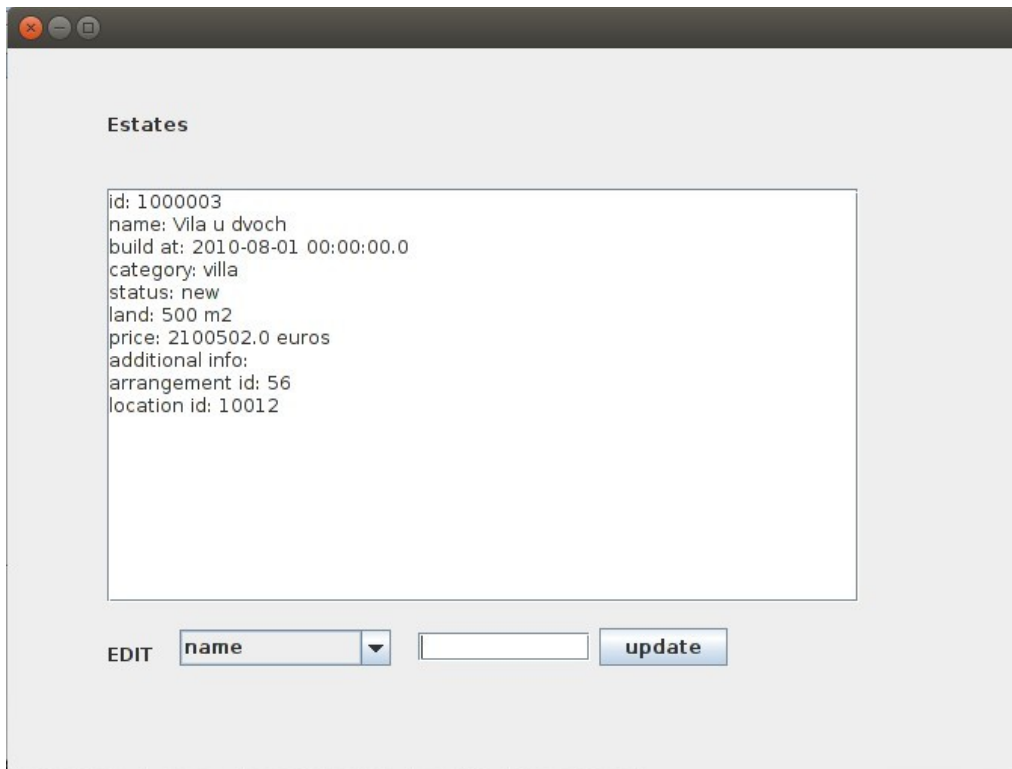
balcony: with

pool: with

garden: with

Pridá novú nehnuteľnosť s novou lokalitou a so zostavou s id = 56.

3. Detailné okno



Estates

id: 1000003
name: Vila u dvoch
build at: 2010-08-01 00:00:00.0
category: villa
status: new
land: 500 m2
price: 2100502.0 euros
additional info:
arrangement id: 56
location id: 10012

EDIT name