

Objektové programování II

Semestrální práce

Contact book

Dominik Heller

Aplikovaná informatika

Jihočeská univerzita v Českých Budějovicích

Zimní semestr 2019/20

Prohlašuji, že jsem tuto semestrální práci zpracoval samostatně a použil jen uvedených pramenů a literatury.

České Budějovice, leden 2019

Obsah

1.	Schválená formulace zadání projektu.	3
2.	Analýza a popis funkcionality cílové aplikace	5
3.	Diagram tříd projektu a jejich popis.....	7
3.1.	Serverová část.....	7
3.2.	Klientská část.....	8
4.	Popis návrhu a podoba grafického rozhraní	9
5.	Návod na zprovoznění aplikace	14
6.	Zhodnocení výsledného projektu v porovnání s návrhem	16
7.	Diskuze – klady a zápory	17
8.	Literatura a prameny.....	18

1. Schválená formulace zadání projektu.

Téma projektu semestrální práce – OOP II

Jméno autora:

Dominik Heller

Název projektu:

ContactBook (alternativně: Contact Management Tool, Kniha kontaktů)

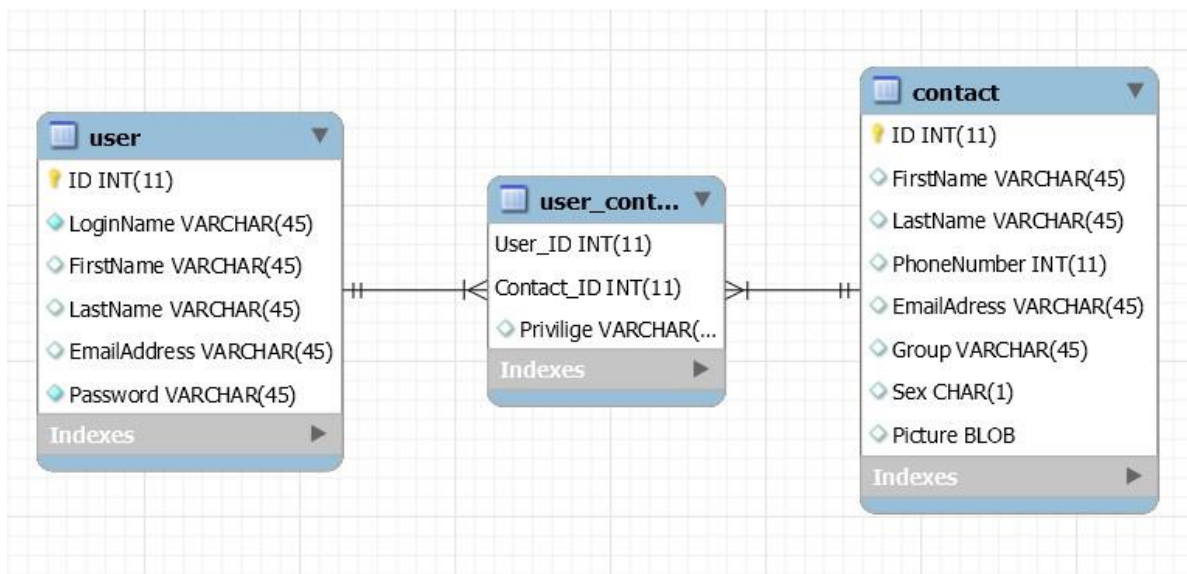
Formulace řešeného problému, nastínění cílové funkcionality aplikace:

Předmětem mé semestrální práce bude tvorba mnohauživatelské aplikace pro správu kontaktů nad databází MySQL. Aplikace bude poskytovat možnost přihlášení existujícího uživatele, respektive registraci uživatele nového. Pro každého registrovaného uživatele bude poté k dispozici vlastní prostor pro správu svých kontaktů, tedy možnosti jako přidat nový kontakt, upravit stávající, příp. jej úplně smazat. Významnou funkcionalitou bude také možnost vyhledávání v již sestaveném seznamu kontaktů (dle zvolených kritérií) a také možnost sdílení kontaktů s vícero uživateli na základě předdefinovaných přístupových práv.

Aplikace vyučovaných témat v projektu:

Jak řečeno výše aplikace by měla primárně pracovat s databázovým systémem. Za tímto účelem bude proto využito frameworku JDBC a s ním spojených tříd a interfejsů jako *DriverManager* a *Connection* pro navázání spojení s databází, *Statement* a *ResultSet* pro formulaci příkazů a následné shromáždění výsledků, či v neposlední řadě ošetřování výjimek typu *SQLException* atp.

Zevrubné schéma databázového prostoru popisuje následný obrázek:



Zde navržená struktura databáze se však v průběhu zpracování projektu samozřejmě může změnit.

Dále bude do vypracování projektu zahrnuto téma síťového programování a téma vláken. Komunikace s databází bude totiž řešena přes serverové připojení, čímž bude zajištěn nepřímý přístup uživatelů k uloženým datům a bude tak mj. posílena i bezpečnostní stránka celé aplikace. Server bude tedy naslouchat na přiděleném portu a v případě žádosti o připojení od klienta vytvoří nové vlákno, které zajistí obsluhu klientských požadavků, zatímco hlavní serverové vlákno bude pokračovat v naslouchání. S ohledem na to, bude využito balíčku *java.net* a tříd jako *ServerSocket*, *Socket*, *InputStreamReader*, *OutputStreamWriter* atp.

Kromě zmíněného bude poté součástí tvorby projektu také práce s vlákny v rámci grafického rozhraní, a to za využití knihovny *javax.swing*. Předpokládáné je tedy zpracování synchronizace mezi počátečním vláknem, event-dispatch vláknem a vlákny pracovními. Mj. bude tedy pracováno např. s třídou *Swingworker*.

Závěrem přikládám ryze zevrubný náčrt vizuálního zpracování své aplikace:

The image displays three screenshots of a Java Swing application interface, all featuring a dark gray background with light gray text and form elements.

Přihlašovací stránka (Login Page): This window contains two text input fields labeled "Přihl. jméno:" and "Heslo:". Below these fields are two buttons: "Přihlásit" and "Registrovat".

Registrační stránka (Registration Page): This window contains four text input fields labeled "Přihl. jméno:", "Heslo:", "Potvrdit heslo:", and "Emailová adresa:". Below these fields is a single button labeled "Registrovat".

ContactBook uživatele xy (Contact Book User Interface): This window is divided into two main sections. On the left, there are seven text input fields for contact details: "ID:", "Jméno:", "Příjmení:", "Telefonní číslo:", "Emailová adresa:", "Skupina:", and "Pohlaví:". On the right, there is a section titled "Seznam kontaktů:" containing a list box with five items labeled "Item 1", "Item 2", "Item 3", "Item 4", and "Item 5". At the bottom of the window, there are five buttons: "Přidat kontakt", "Odebrat kontakt", "Upravit kontakt", "Vyhledat kontakt", and "Sdílet kontakt(y)".

2. Analýza a popis funkcionality cílové aplikace

Námi předkládaná aplikace, jak název napovídá, představuje platformu pro online správu uživatelských kontaktů. Celkem je přitom tvořena dvěma samostatnými projekty, a to serverovou aplikací na jedné straně a klientskou na straně druhé. Obě poté samozřejmě vzájemně nezbytně spolupracují a pro naplnění plné funkcionality je v tomto smyslu jedna neoddělitelná od druhé.

Z klientského hlediska se poté konkrétně jedná o „víceuživatelskou“ aplikaci nabízející mimo běžného přidávání, úprav či odebírání kontaktů s okamžitým uložením všech změn v databázovém prostoru, také možnost sdílení svých kontaktů (či skupiny kontaktů) s ostatními vybranými uživateli aplikace. To vše přitom na základě jasně definovaných přístupových práv, které může majitel kontaktu (*owner*) blíže specifikovat a nastavit, případně odebrat a změnit (znovunastavit). Konkrétně lze odlišit 3 kategorie uživatelských pravomocí:

- a) pouze číst
- b) upravit
- c) odstranit
- d) kombinace b) a c)

Jako doplňková funkcionality pro pestřejší úpravu jednotlivých kontaktů, stejně jako samotného uživatelského účtu byla poté oproti návrhu přidána možnost nahrávání profilových obrázků. Každý kontakt tak může být opatřen konkrétním obrázkem, který lze společně ostatními údaji sdílet mezi uživateli. Ti jej poté mohou, v případě příslušných práv (*update*), kdykoli změnit.

Mimo zmíněného, se s ohledem na svůj víceuživatelský charakter mezi základní funkcionality aplikace řadí také registrace nových uživatelů, stejně jako přihlašování uživatelů stávajících. S tím jsou samozřejmě spojeny nezbytné nároky na bezpečnost

přenosu přístupových údajů¹, obdobně jako jejich uchovávání v databázi a zpětné získávání pro autentizaci uživatele. Tím se tedy dostáváme k serverové části aplikace.

Veškerá komunikace mezi uživatelskými aktivitami a databází probíhá přes server, respektive přes serverové vlákno, které je každému nově připojenému uživateli přiděleno. Konkrétně poté vzájemná komunikace probíhá víceméně na základě posílání heslovitých signálů, které přivádějí serverové vlákno do požadovaného stavu, tak aby bylo synchronizováno s klientskými podněty. Tento signálový protokol přitom stojí v jádru celé síťové komunikace².

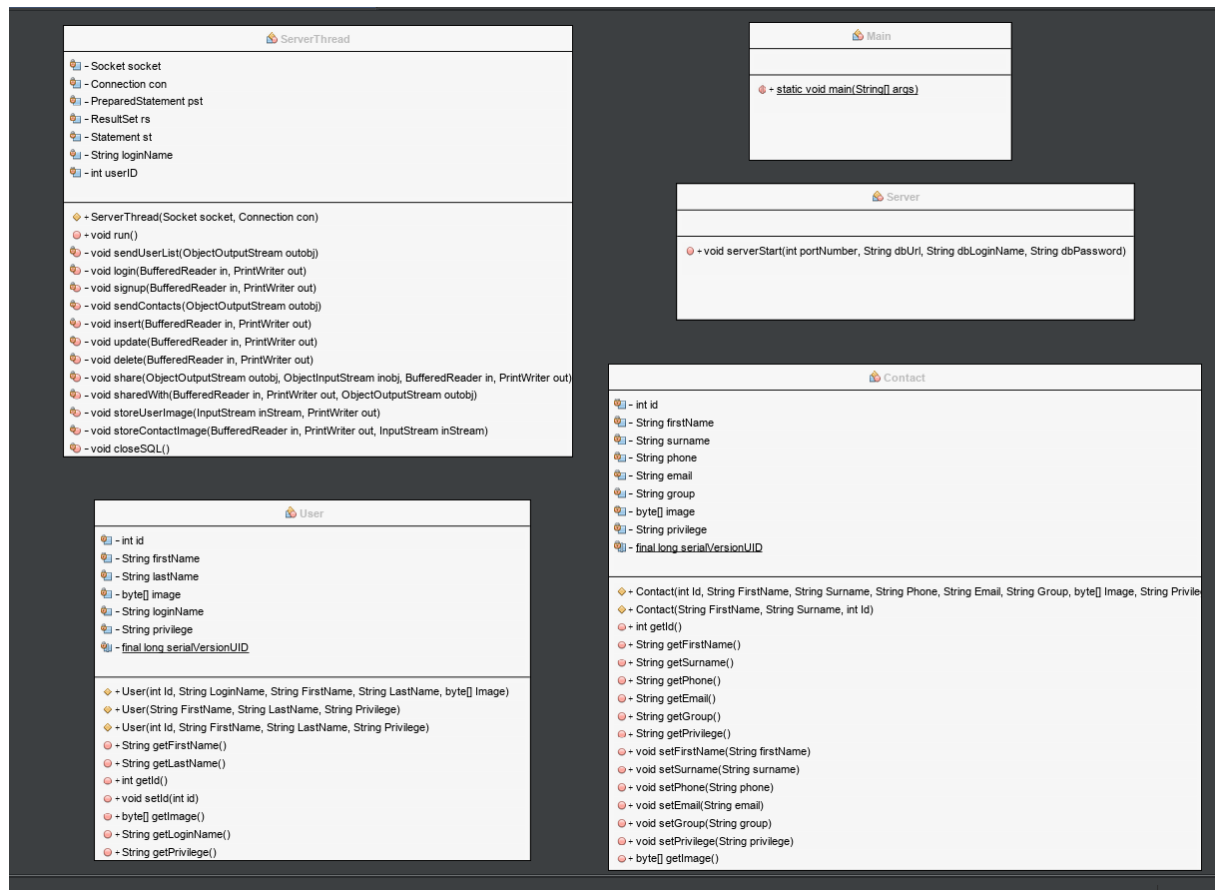
Kromě ryze síťové komunikace mezi serverem a klientem je zde však ještě jedna vrstva, a to komunikace mezi serverem a databázovým systémem. Pro náš účel byl vybrán relační databázový model implementovaný pomocí systému řízení báze dat MySQL na straně jedné a rozhraním JDBC (Java Database Connectivity) pro javovské aplikace na straně druhé.

¹ Využívána je definovaná hašovací funkce pro kódování přístupového hesla.

² Např. posílání signálu „GETCONTACT“ z klientského soketu zajistí, že serverové vlákno vybere předdefinovaným SQL příkazem z databáze příslušné kontakty, zabalí do objektu typu ArrayList<Contact> a pošle zpět na klientskou stranu, která již na objekt daného typu čeká a zobrazí jej uživateli.

3. Diagram tříd projektu a jejich popis

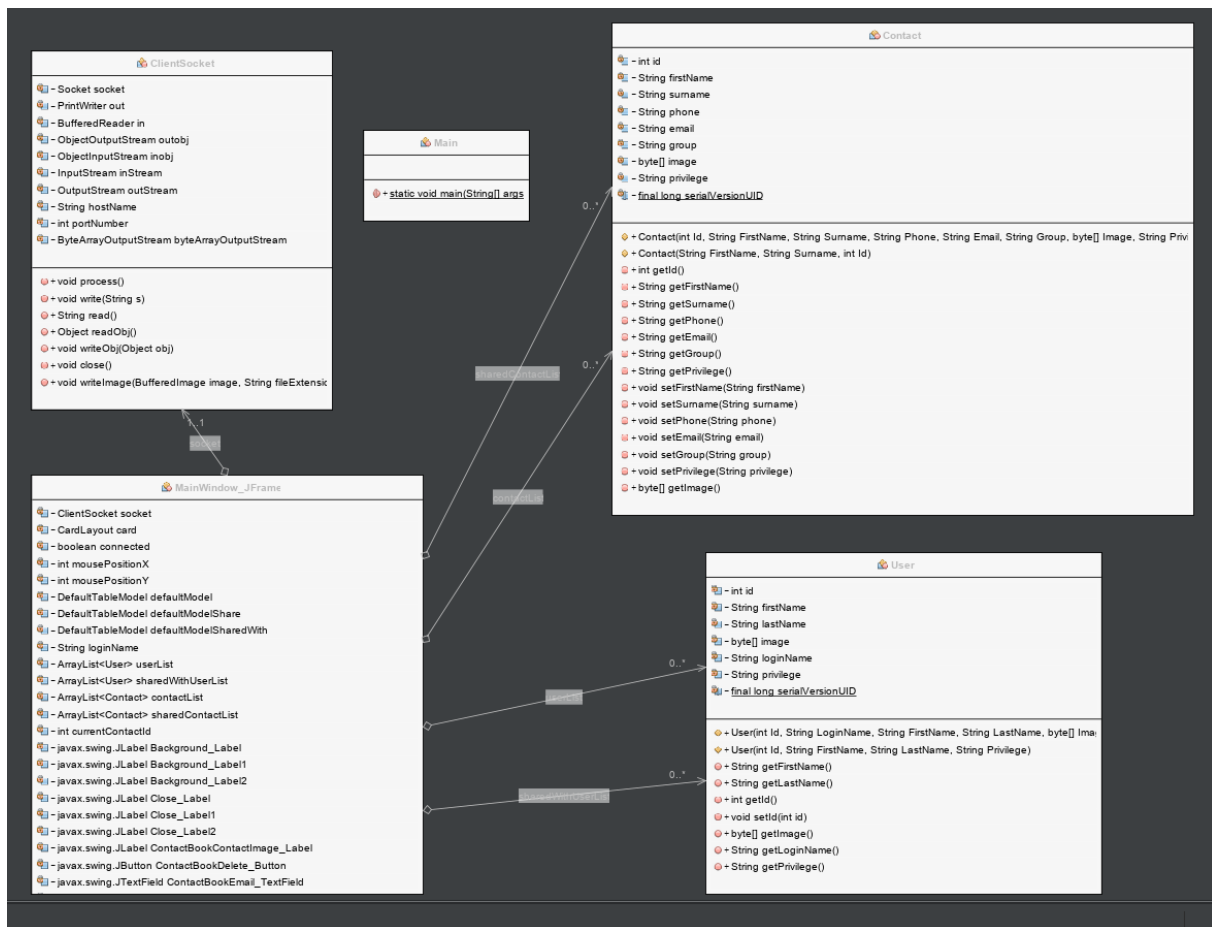
3.1. Serverová část



Projekt na straně serveru tvoří celkem pět samostatných tříd, které jsou rozděleny do dvou balíčků. V prvním balíčku jsou specifikované třídy, pro tvorbu datových objektů typu *User* a *Contact*. Ty jsou využívány k ukládání dat získaných z databáze a následné distribuci po síti na stranu klienta. Obsahují tak jednotlivé instanční metody a proměnné, které by se daly očekávat, stejně jako příslušné typy konstruktorů pro konkrétní případy.

Druhý balíček již obsahuje aplikační logiku a konkrétně třídy *Server* a *ServerThread*. Jak řečeno, tak klíčovou funkcí instance třídy *Server* je naslouchání na příslušném portu a následné distribuci obsluhy nových spojení na nově vytvořené vlákno (instanci třídy *ServerThread*). Třída *ServerThread* poté obsahuje portfolio metod nezbytných pro komunikaci jak s databází, tak s klientem, tedy pracuje s obsahem balíčků jako *java.net* či *java.sql*.

3.2. Klientská část



Projekt na straně klienta lze taktéž rozdělit do dvou balíčků. První znovu obsahuje třídy *Contact* a *User*, které jsou nezbytně identické se serverovou částí pro zajištění vzájemné kompatibility. Druhým balíček poté tvoří třída *ClientSocket* a *MainWindow*. První zmíněná poskytuje klientské „síťové metody“, tedy zajišťuje navázání spojení a přenos/příjem dat. Třída *MainWindow* poté tvoří jádro klientské části projektu a řeší aplikační i prezentační logiku, včetně celé vizuální specifikace grafického uživatelského rozhraní. Definuje tudíž s pomocí integrovaného grafického editoru veškeré vizuální komponenty, stejně jako na ně navázané odpovídající události atp. Zajišťuje přitom i přepínání jednotlivých oken, stejně jako zpracování a prezentaci příchozích dat ze strany serveru.

4. Popis návrhu a podoba grafického rozhraní

Pro vytvoření naší aplikace bylo využito tzv. kartového layoutu (*card layout*) poskytujícího možnost obměňovat jednotlivé pohledy v rámci jediného okna (*fram*). To zajišťuje plynulé přepínání pohledů (panelů) a činí tak aplikaci mnohem příjemnější k užívání (i samotné tvorbě) než při přepínání mezi samostatně definovanými okny. Pouze drobnou výjimku v tomto smyslu tvoří užití internal frames pro komplexnější operace týkající se sdílení kontaktů. Vizuální prezentace aplikace poté využívá tzv. nedekorovaného okna (*undecorated*), tudíž zde nenalezneme klasické orámování, což dále napomáhá vizuální atraktivitě celé aplikace.

Podobu grafického rozhraní stejně jako některé funkcionality si nyní prezentujeme na několika obrázcích.

1. Přihlášení uživatele



Přihlašovací stránka

Login:

Heslo: ☐ Zobrazit heslo

Potvrdit

Registrovat

2. Registrace uživatele

– X

Registrační stránka

Login:

Jméno:

Příjmení:

Heslo:

Heslo znovu:

Emailová adresa:

Provést registraci

Zpět

3. Správce kontaktů

Odehlásit

– X

ContactBook uživatele TEST

Přidejte fotografii

Jméno:

Příjmení:

Telefonní číslo:

Emailová adresa:

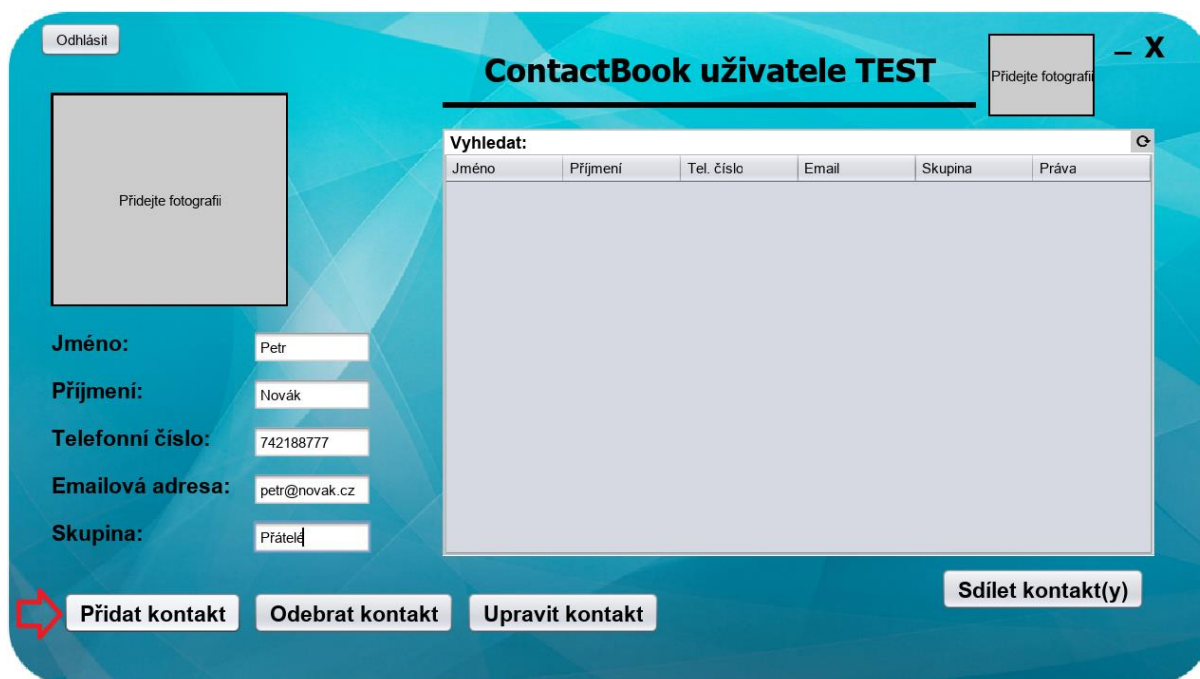
Skupina:

Vyhledat:

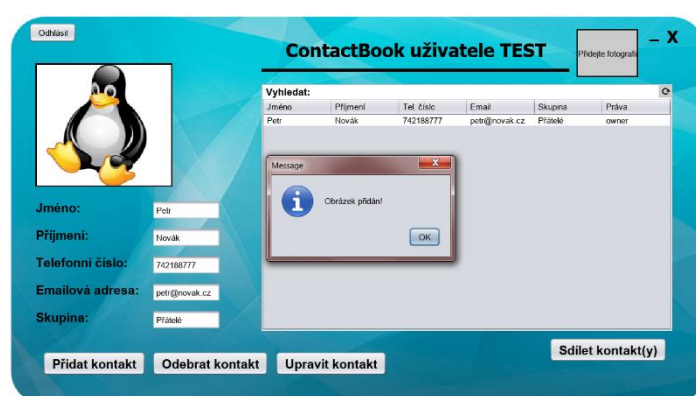
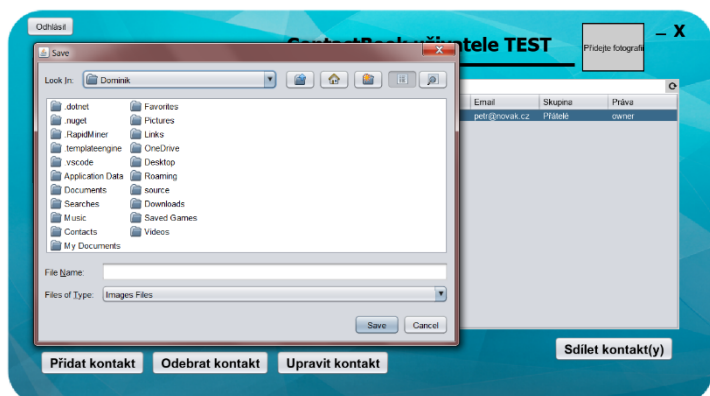
Jméno	Příjmení	Tel. číslo	Email	Skupina	Práva
-------	----------	------------	-------	---------	-------

Přidat kontakt **Odebrat kontakt** **Upravit kontakt** **Sdílet kontakt(y)**

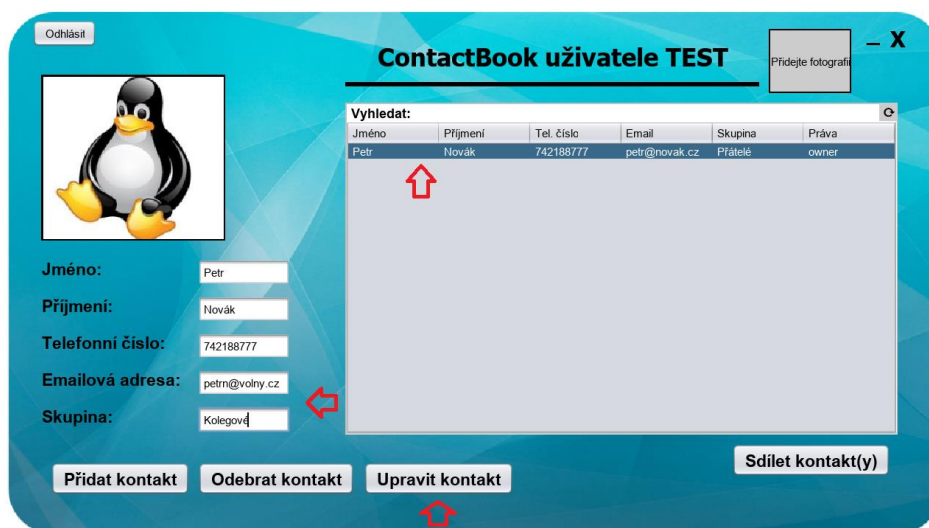
a) Přidání uživatele



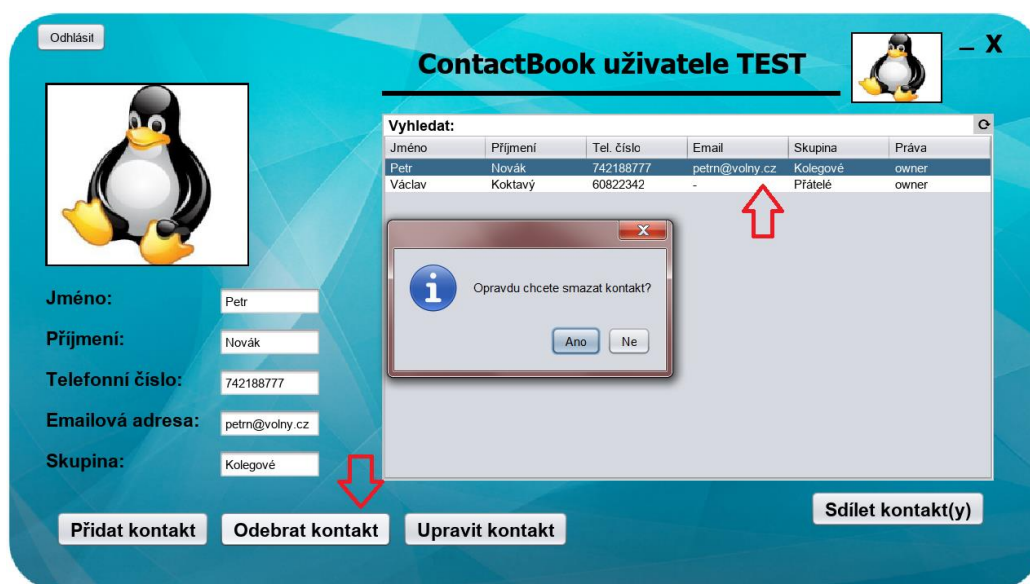
b) Přidání fotografie



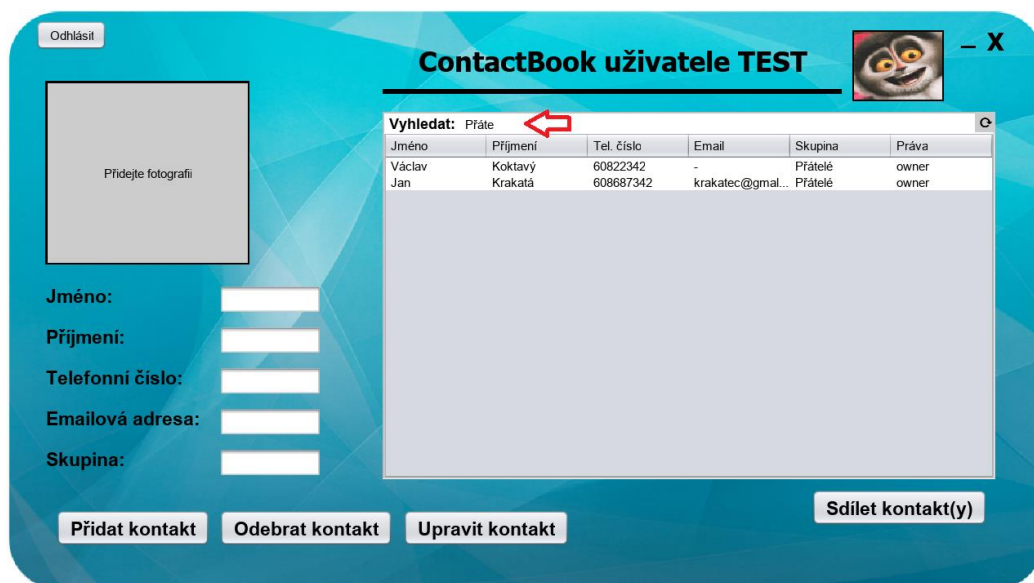
c) Úprava kontaktu



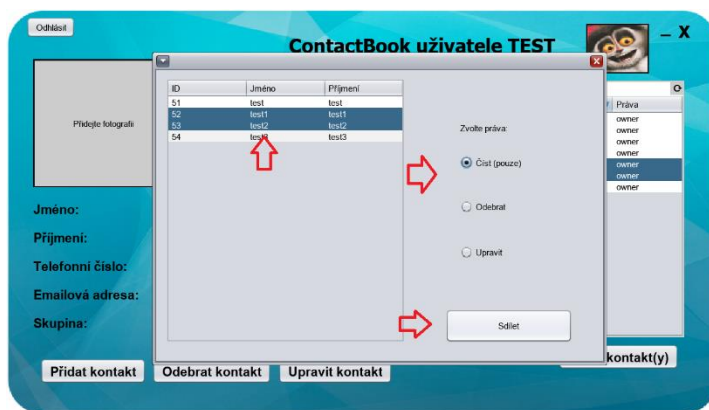
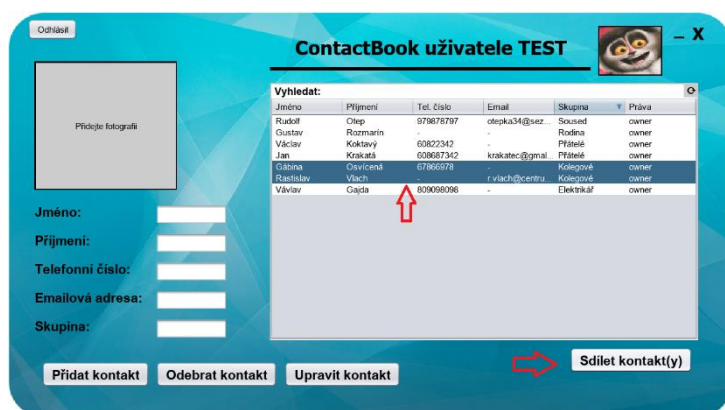
d) Odstranění kontaktu



e) Vyhledávání kontaktů



f) Sdílení kontaktů



Odhlásit

ContactBook uživatele TEST

Přidejte fotografii

Jméno:

Rastislav

Příjmení:

Vlach

Telefonní číslo:

-

Emailová adresa:

n@centrum.com

Skupina:

Kolegové

Přidat kontakt

Odebrat kontakt

Upravit kontakt

Sdílet kontakt(y)

Sdíleno s

Vyhledat:

Jméno	Příjmení	Tel. číslo	Email	Skupina	Práva
Rudolf	Otep	979878797	otepka34@sez...	Soused	owner
Gustav	Rozmarin	-	-	Rodina	owner
Václav	Koktavý	60822342	-	Přítel	owner
Jan	Krakatá	698987342	krakatec@gmail	Přítel	owner
Gábina	Osvícená	67866978	-	Kolegové	owner (shared)
Rastislav	Vlach	-	r.vlach@centru...	Kolegové	owner (shared)
Václav	Gajda	800098038	-	Elektrikář	owner

Odhlásit

ContactBook uživatele TEST

Přidejte fotografii

Jméno:

Rastislav

Příjmení:

Vlach

Telefonní číslo:

-

Emailová adresa:

n@centrum.com

Skupina:

Kolegové

Přidat kontakt

Odebrat kontakt

Upravit kontakt

Sdílet kontakt(y)

Sdíleno s

Vyhledat:

Jméno	Příjmení	Tel. číslo	Email	Skupina	Práva
Id	Jméno	Příjmení	Práva		
52	test1	test1	R		
53	test2	test2	R		

Odebrat

Změna práv:
1 odeberte uživatele
2 nastavte sdílení znovu

Odhlásit

ContactBook uživatele TEST1

Přidejte fotografii

Jméno:

Rastislav

Příjmení:

Vlach

Telefonní číslo:

-

Emailová adresa:

n@centrum.com

Skupina:

Kolegové

Přidat kontakt

Odebrat kontakt

Upravit kontakt

Sdílet kontakt(y)

Vyhledat:

Jméno	Příjmení	Tel. číslo	Email	Skupina	Práva
Gábina	Osvícená	67866978	-	Kolegové	R
Rastislav	Vlach	-	r.vlach@centru...	Kolegové	R

Message

i

Nemáte oprávnění

OK

5. Návod na zprovoznění aplikace

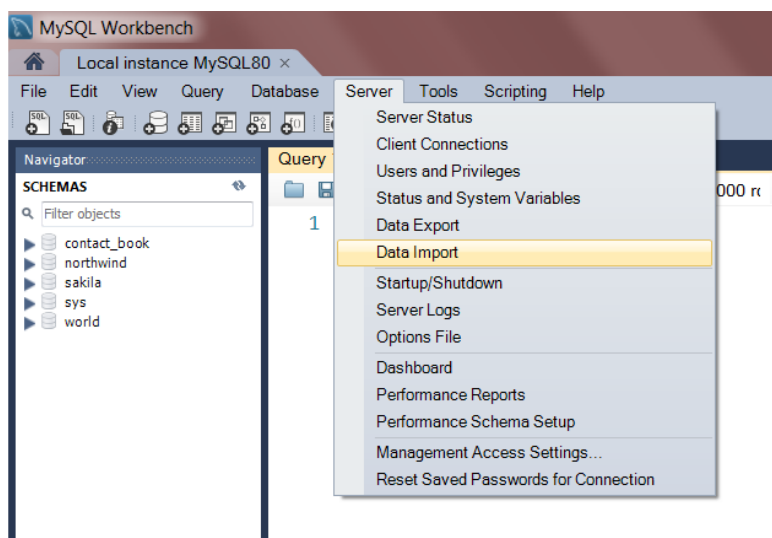
Aplikace ke svému spuštění vyžaduje splnění dvou základních požadavků. Za prvé implementaci driveru MySQL connector³, který je nutno přiložit jako externí knihovnu do serverového projektu. A poté nadefinovat přihlašovací údaje do databáze, a to konkrétně na straně serveru v třídě *Main*, metodě *public static void main()*.

```
package semestralnipraceoopiiserver;

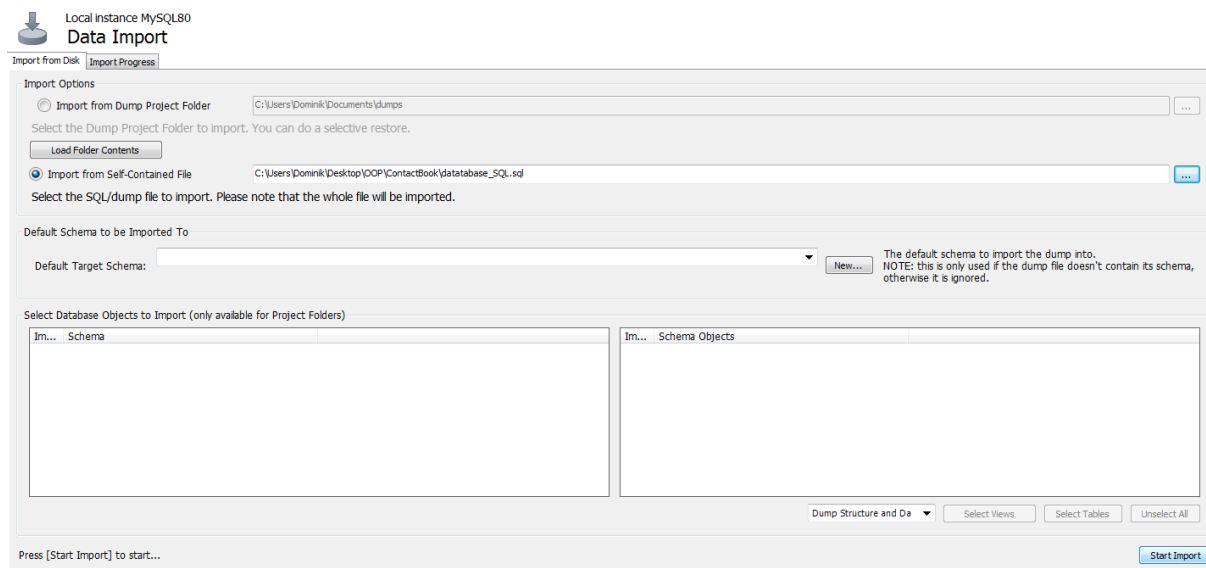
/**
 * Main class to initialize Server side application.
 *
 * @author Dominik
 */
public class Main {

    /**
     * Creates new Sever instance on which the serverStart method is initialized.
     * Please specify the method parameters based on your database login
     * credentials.
     *
     * @param args
     */
    public static void main(String[] args) {
        Server server = new Server();
        server.serverStart(4337, "jdbc:mysql://localhost:3306/contact_book", "root", "root");
    }
}
```

Druhou nezbytnou věcí je poté implementace samotné databáze z přiloženého *database_SQL.sql* souboru. V MySQL Workbench můžeme tento krok provést cestou Server→Data Import→Import from Self-Contained File→zvolit soubor→ Start Import.



³ Přiloženo jako součást zip souboru. Případně dostupné zde <https://www.mysql.com/products/connector/>.



Následně již pak stačí pouze spustit serverový a následně klientský projekt v prostředí *Netbeans*.

6. Zhodnocení výsledného projektu v porovnání s návrhem

Dle mého názoru se prakticky v celé míře podařilo naplnit cíle stanovené v semestrálním návrhu. Aplikace protíná všechny v návrhu zmíněné sféry vyučované v průběhu semestru, tzn. síťové programování, práce s databází i vláknové programování. Z hlediska funkcionality poté umožňuje příslibené operace nad kontakty včetně sdílení a přidává i možnost ukládání obrázků. Z hlediska vizuálního poté aplikace dostala oproti návrhu značného vylepšení a prezentuje se o poznání přívětivějším grafickým rozhraním.

7. Diskuze – klady a zápory

Začneme-li zápory předložené aplikace, tak těmi z mého pohledu největšími jsou určitě samotná „architektura“ primárně klientské části aplikace a za druhé v řadě pasáží komplikovaná forma komunikace mezi klientem, serverem a databází. To v obou případech přispívá k horší přehlednosti projektu a samozřejmě též k problematičtějšímu odhalování a nápravě chyb. V některých případech se tedy může objevit problém s přenosem datových souborů případně s korupcí celého streamu mezi serverem a klientem. Problematické je také přenášení/načítání obrázků, které značně zatěžuje datový tok a u velkých obrázků je náchylné na chybová hlášení. Aplikace poté také neposkytuje některé funkcionality, které by u ní asi byly očekávány. Např. možnost správy uživatelského účtu (změny hesla atp.), či customizace vizuálního prostředí atp.

Všechny tyto nedostatky jsou však v mnohém zastíněny nejenom výše zmíněnými pozitivy, ale také přínosem z hlediska osobního rozvoje, který mi práce na tomto projektu přinesla. Na začátku semestru jsem totiž o daných konceptech měl jen velmi omezené znalosti. Práce na daném projektu mi tudíž přinesla možnost alespoň částečně si osvojit některé klíčové kapitoly objektového programování, a to si myslím byl hlavní přínos celého předmětu.

8. Literatura a prameny

Baeldung (2020). *A Guide to Java Sockets* (<https://www.baeldung.com/a-guide-to-java-sockets>), 10. 1. 2020)

Calvert, K.L. - Donahoo, M.J. (2001). *TCP/IP Sockets in Java: Practical Guide for Programmers* (Morgan Kaufmann Publishers: San Francisco).

GeeksForGeeks (2019). *SwingWorker in Java* (<https://www.geeksforgeeks.org/swingworker-in-java/>, 25.12.2019).

JDK 11 Documentation (2020). *JDK 11 Documentation* (<https://docs.oracle.com/en/java/javase/11/>, 9. 1. 2020).

Jenkov (2019). *Tutorials*. (<http://tutorials.jenkov.com/java-concurrency/index.html>, 10. 1. 2020).

Materiály na stránkách předmětu Objektové programování II 2019. Dostupné na: <https://elearning.jcu.cz/course/view.php?id=3213>

MySQL (2020). *MySQL 8.0 Reference Manual* (<https://dev.mysql.com/doc/refman/8.0/en/>, 4. 1. 2020).

TutorialPoint (2020). *Java Tutorial* (<https://www.tutorialspoint.com/java/index.htm>, 8. 1. 2020).