Objektové programování v Javě 2

Zimní semestr 2016/17

Chat server - klient

Pavel Máca

Projekt by vypracován samostatně za pomoci zdrojů popsaných níže.
8. ledna 2017

1 Zadání projektu

Komunikační klient připojitelný na specifické rozhraní, které umožňuje komunikaci mezi ostatními klienty. Server a klienti budou poskytovat možno vytváření tzv. chatovacích místností.

Forma komunikace bude textová.

Základní rozhraní bude tvořit nabídka dostupných chatovacích místností a okno s aktuálně otevřenou místností a zprávami.

Pokud některý uživatel odešle zprávu do stejné místnosti, zobrazí se ostatním uživatelům, kteří jsou připojeni v dané místnosti. Pokud uživatel obdrží zprávu v jiné místnosti, zobrazí se počet nepřečtených zpráv k dané místnosti v seznamu všech místností.

Uživatel bude vstupovat do místnosti, tím že začne naslouchat její komunikaci.

Online i offline komunikace. Pokud klient nebude online, zprávy pro něj se mu zobrazí po přihlášení.

Systém ukáže klientovi, kdo je aktuálně online, pokud se někdo přihlásí/odhlásí, všichni klienti na to budou ihned reagovat

Skupiny budou definovány obecně, uživatel bude moci byt ve vice skupinách, GUI bude mít funkcionalitu, pomoci které se uživatel bude moci sám přihlašovat a odhlasovat ze skupin

Klient i server v javě.

Projekt bude využívat síťové programování a vláknové programování při naslouchání na více chatovacích místnostech a databázi.

2 Analýza funkcionality a řešení

2.1 Funkcionalita

Projekt obsahuje jak serverovou část, tak klienta. Komunikace probíhá na úrovni TCP socketů. Každý klient se připojuje na server, pomocí nastavené IP adresy a portu.

2.2 Server

Aplikace bez grafického rozhraní, pouze konzolový vstup a výstup. Po spuštění lze nastavit číslo portu pro naslouchání nových klientů. Při prvním spuštění se vyplní přístupové údaje k databázi.

Každý připojený klient vytvoří samostatné vlákno, které má za úkol příjem veškeré komunikace od daného klienta. Pokud klient vytvoří novou místnost, tato místnost bude reprezentována vláknem, rozesílá veškerou komunikaci všem klientům, kteří se do místnosti připojí. Ve chvíli kdy je místnost prázdná, se vlákno ukončí. Místnost je však nadále uchována v databázi až do jejího smazání, které může provést pouze tvůrce místnosti. Po připojení do místnosti, které je prázdná, se opět spustí vlákno této místnosti.

Uživatel, který byl off-line a neopustil místnost, obdrží veškeré zprávy z této doby, jakmile bude opět online.

Zpráva od klienta je po obdržení zapsána do databáze a následně odeslána vláknem místnosti všem připojeným klientům. Ostatní události, jako odchod z místnosti, přejmenování místnosti apod. jsou také rozesílány všem aktuálně připojeným klientům.

2.3 Databáze

Schéma databáze se nachází v souboru '/sql/install.sql'

Každý uživatel je reprezentován záznamem v tabulce 'user'. Záznam obsahuje id, která je jedinečné a automaticky generované. Dále pak jméno a heslo.

Místnost je identifikovatelná pomocí číselného identifikátoru, který je jednoznačný. Každá místnost má také svého autora, který je automaticky administrátorem této místnosti. Po smazání místnosti je tato místnost označena jako smazaná, ale data zůstávají nadále zachována.

Seznam uživatelů v místnosti je reprezentován záznamy v tabulce 'user_room'. Seznam blokovaných uživatelů určité místnosti se nachází v tabulce 'user_block'.

Zpráva obsahuje čas odeslání, odesílatele a místnost, ve které byla odeslána.

2.4 Klient

Grafické rozhraní umožňuje nastavit IP adresu a port serveru, ke kterému se klient připojí. Tuto volbu lze uložit. Po připojení na server se uživatel přihlásí svou přezdívkou a heslem. Pokud uživatel neexistuje, automaticky se vytvoří záznam na serveru.

Přihlášenému uživateli se zobrazí seznam všech místností, ve kterých je připojen. Dále má možnost připojit se do již existujících místností, nebo vytvořit novou. Místnost lze chránit heslem a vytvořit ji tak soukromou pro specifické uživatele.

U každé místnosti je zobrazen seznam online / připojených uživatelů. Tvůrce místnosti má možnost změnit název, nastavit heslo a vyloučit (zablokovat) ostatní uživatele z místnosti.

Nastavení klienta (server IP a port) se ukládají do lokálního souboru.

Uživatel má možnost se odhlásit, či odpojit od serveru.

2.5 Komunikace klient-server

Komunikaci na straně klienta obstarává jedno vlákno.

Mezi server a klientem se posílají objekty typu 'Request' a 'Response', který se podle parametru dále zpracovávají. Data formát dat pro komunikaci je specifikován modely v 'pavelmaca.chat.share.model'.

Klient po vytvoření TCP spojení odešle 'Hand-shake' request, na který server odpovídá 'Response.OK'. Následně klient odesílá autorizační požadavek, na který server odpovídá Response, která obsahuje informace o přihlášeném uživateli.

Po obdržení identity klient posílá požadavek celkový status uživatele. Server odpovídá seznamem místností, stavy jednotlivých uživatelů a historií zpráv. Tím se současně klient připojí do všech místností pro odběr dalších zpráv.

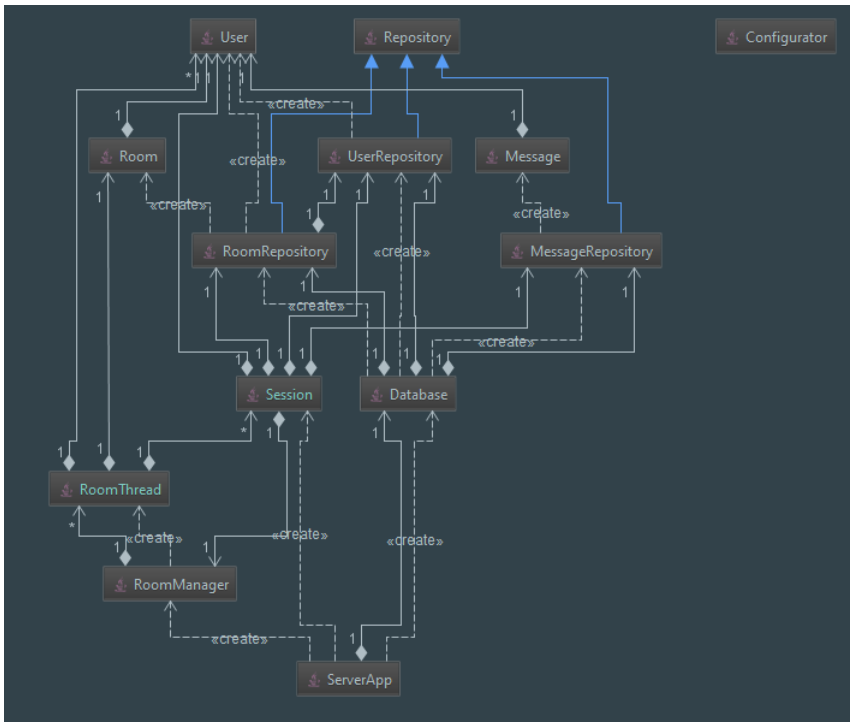
Stavové informace odesílá server z RoomThread vlákna pomocí objektu Request s vyplněným typem a parametry. Klient těmto informacím naslouchá pomocí vlákna Session, které je řadí do blokující fronty, od kud si je přebírá vlákno GUIRequestListener, které běží pod EDT a má možnost aktualizovat GUI.

Po připojení do existující místnosti uživatel obdrží historii pro tuto místnost.

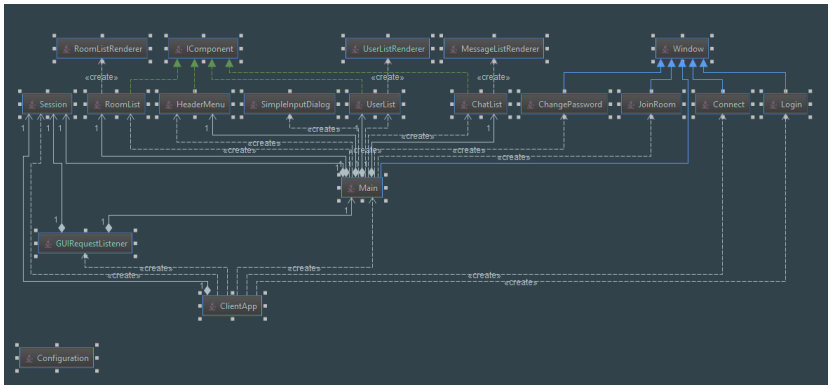
Každý připojený klient má nastavený aktuální stav připojení. Ten určuje, které požadavky může server od klienta přijímat. Například nepřihlášený klient nemůže odesílat nové zprávy.

3 Diagramy tříd

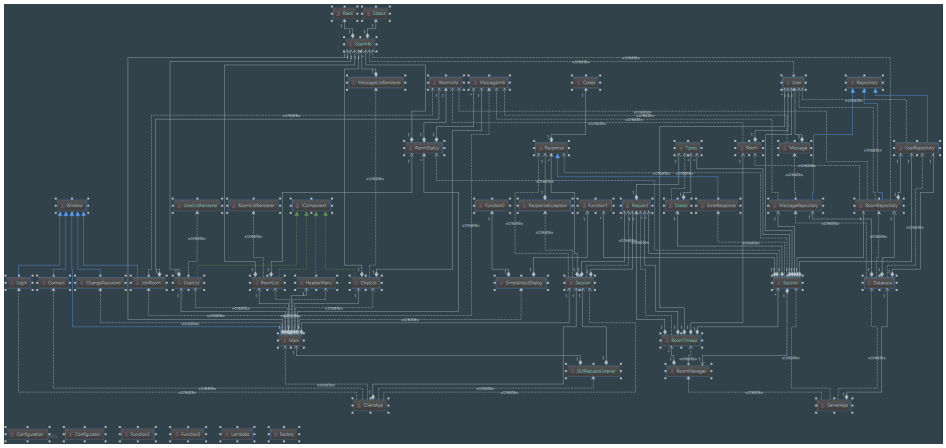
3.1 Server



3.2 Klient



3.3 Server a klient společně



4 Zhodnocení výsledku

Výsledný projekt splňuje zadání a výrazně se nijak neliší od původní analýzy. Aplikace umožňuje rozšiřitelnost o další funkcionalitu díky univerzální implementaci požadavků a odpovědí. Napojení na databázi by mohlo být řešeno pomocí ORM (např. Hibernate), nicméně to nebylo součástí předmětu a proto bylo zvoleno hybridní řešení s použitím JDBC API.

Díky použití blokující fronty na obou stranách komunikace jsou zprávy přijímány a odesílány synchronně.

Celý projekt je publikován na adrese github.com/pavelmaca/jcu-java-chat

5 Použité zdroje

- Prezentace a výukové materiály k předmětu "Objektové programování II - UAI/685"
- Dokumentace Javy 8 od Oracle - docs.oracle.com/javase/8
- Diskuzní fórum stackoverflow.com