

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
FIIT-STU

Lukáš Michalčák, Maroš Guráň

Vývoj progresívnych webových aplikácií Skrupulus

Garant: doc. Ing. Peter Trúchly, PhD.
Prednášajúci: Ing. Eduard Kuric, PhD.
Cvičiaci: Ing. Matúš Krajčovič

Akademický rok: 2024/25

Obsah

| | |
|--|----------|
| Obsah..... | 2 |
| Zadanie..... | 3 |
| Aplikácia na textovú komunikáciu v štýle IRC (zjednodušený Slack)..... | 3 |
| Diagramy..... | 5 |
| Verzia z konca 1. fázy..... | 5 |
| Finálna verzia..... | 6 |
| Fyzický diagram..... | 7 |
| Diagram architektúry..... | 8 |
| Návrhové rozhodnutia..... | 9 |

Zadanie

Aplikácia na textovú komunikáciu v štýle IRC (zjednodušený Slack)

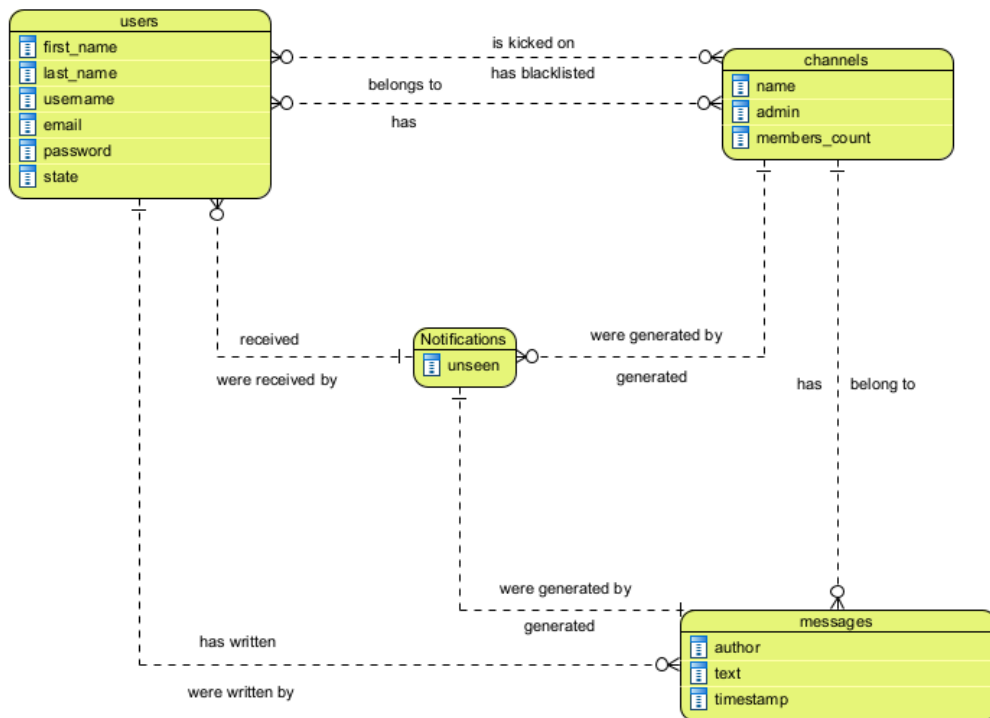
Aplikácia musí realizovať tieto prípady použitia: Akékoľvek iné vylepšenia sú vítané a potešia ma :-)

1. registrácia, prihlásenie a odhlásenie používateľa
 - používateľ má meno a priezvisko, nickName a email
2. používateľ vidí zoznam kanálov, v ktorých je členom
 - pri opustení kanála, alebo trvalom vyhodení z kanála je daný kanál odobratý zo zoznamu
 - pri pozvánke do kanála je daný kanál zvýraznený a topovaný
 - v zozname môže cez používateľské rozhranie kanál vytvoriť, opustiť, a ak je správcom aj zrušiť
 - dva typy kanálov - súkromný (private channel) a verejný kanál (public channel)
 - správcom kanála je používateľ, ktorý kanál vytvoril
 - ak nie je kanál aktívny (nie je pridaná nová správa) viac ako 30 dní, kanál prestáva existovať (následne je možné použiť channelName kanála pre "nový" kanál)
3. používateľ odosiela správy a príkazy cez "príkazový riadok", ktorý je "fixným" prvkom aplikácie. používateľ môže odoslať správu v kanáli, ktorého je členom
4. vytvorenie komunikačného kanála (channel) cez príkazový riadok
 - kanál môže vytvoriť ľubovoľný používateľ cez príkaz /join channelName [private]
 - do súkromného kanála môže pridávať/odoberať používateľov iba správca kanála cez príkazy /invite nickName a /revoke nickName
 - do verejného kanála sa môže pridať ľubovoľný používateľ cez príkaz /join channelName (ak kanál neexistuje, automaticky sa vytvorí)
 - do verejného kanála môže člen kanála pozvať iného používateľa príkazom /invite nickName
 - vo verejnom kanáli môže člen "vyhodiť" iného člena príkazom /kick nickName. ak tak spraví aspoň 3 členovia, používateľ má "trvalý" ban pre daný kanál. správca môže používateľa vyhodiť "natrvalo" kedykoľvek príkazom /kick nickName, alebo naopak "obnoviť" používateľovi prístup do kanála cez príkaz /invite
 - nickName ako aj channelName sú unikátne
 - správca môže kanál zatvoriť/zrušiť príkazom /quit
5. používateľ môže zrušiť svoje členstvo v kanáli príkazom /cancel, ak tak spraví správca kanála, kanál zaniká

6. správu v kanáli je možné adresovať konkrétnemu používateľovi cez príkaz @nickname
 - správa je zvýraznená danému používateľovi v zozname správ
7. používateľ si môže pozrieť kompletnú históriu správ
 - efektívny infinite scroll
8. používateľ je informovaný o každej novej správe prostredníctvom notifikácie
 - notifikácia sa vystavuje iba ak aplikácia nie je v stave "visible" (pozrite quasar docu App Visibility)
 - notifikácia obsahuje časť zo správy a odosielateľa
 - používateľ si môže nastaviť, aby mu chodili notifikácie iba pre správy, ktoré sú mu adresované
9. používateľ si môže nastaviť stav (online, DND, offline)
 - stav sa zobrazuje používateľom
 - ak je nastavený DND stav, neprichádzajú notifikácie
 - ak je nastavený offline stav, neprichádzajú používateľovi správy, po prepnutí do online sú kanály automaticky aktualizované
10. používateľ si môže pozrieť zoznam členov kanála (ak je tiež členom kanála) príkazom /list
11. ak má používateľ aktívny niektorý z kanálov (nachádza sa v okne správ pre daný kanál) vidí v stavovej lište informáciu o tom, kto aktuálne píše správu (napr. Ed is typing)
 - po kliknutí na nickName si môže pozrieť rozpísaný text v reálnom čase, predtým, ako ju odosielateľ odošle (každá zmena je viditeľná) :-)

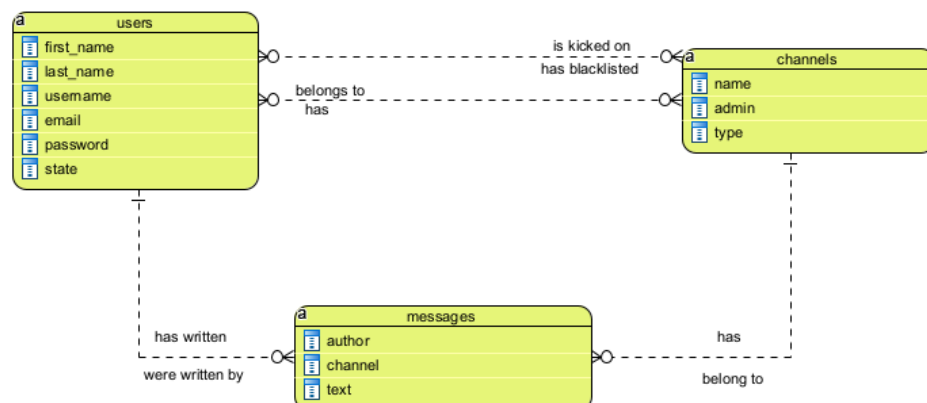
Diagramy

Verzia z konca 1. fázy



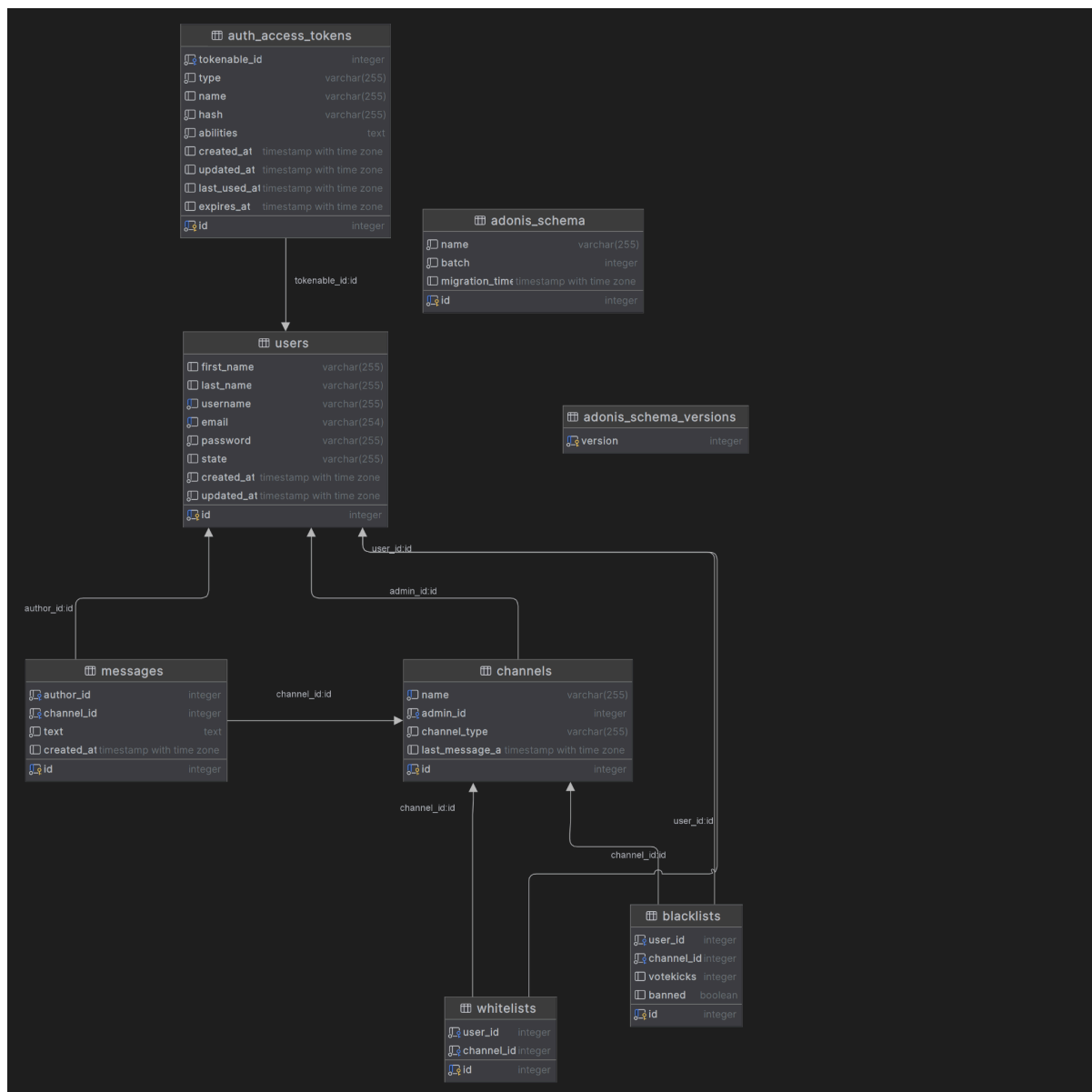
Toto je náš prvotný logický model databázy. Reprezentoval 4 entity, ktoré sme sa (pôvodne) rozhodli reprezentovať v našej aplikácii (priamo konkrétnymi tabuľkami pravdepodobne): users (používatelia), channels (kanály), messages (správy), notifications (notifikácie). V diagrame sú aj vyjadrené vzťahy medzi entitami a many-to-many vzťahy sú vo fyzickom modeli vyjadrené tabuľkami whitelists a blacklists.

Finálna verzia



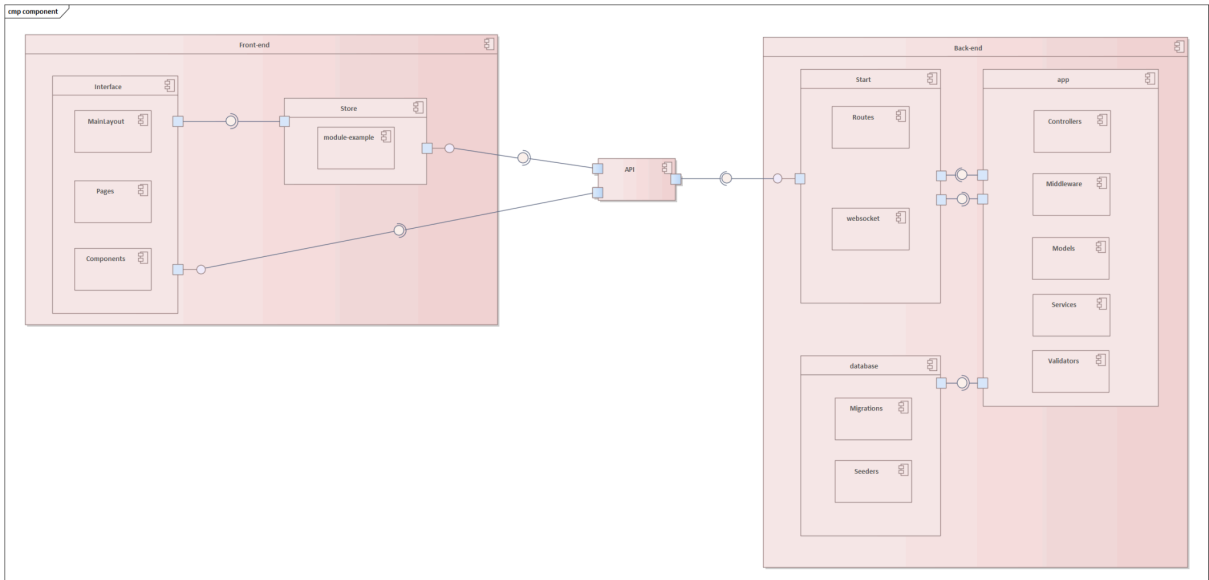
Najväčšia zmena oproti pôvodnému návrhu bolo to, že sme úplne odstránili entitu notifikácie. Nakoniec sme ich totiž neukladali do databázy, lebo nemalo zmysel ich ukladať, a tak by bolo zbytočné ich tu nechať. Takisto sme z messages odstránili timestamp (v databáze je vytvorený created_at, ale na frontende sa s ním nepracuje, používa sa len na triedenie správ) a pridali do nich channel, keďže, samozrejme, si musíme ukladať informáciu o príslušnom kanály. Z channels sme odstránili members_count, taký údaj nikde neukladáme, počet sa dá ľahko získať z whitelists, ale pridali sme tam type, keďže ten predtým nebol nikde vyjadrený.

Fyzický diagram



Toto je fyzický diagram vygenerovaný z našej databázy. Obsahuje aj všetky cudzie kľúče. Približuje fungovanie -lists tabuliek, je tu vidieť, že blacklists nie je len obyčajná many-to-many tabuľka, ale udržiavame si tu aj informáciu, koľko votekickov už dostal používateľ a hlavne, či je na danom kanáli zabanovaný, alebo nie. Obsahuje aj tabuľku na manažment tokenov (tú my na backende neriadime, rieši to automaticky adonis), ako aj pomocné tabuľky pre adonis adonis_schema a adonis_schema_version.

Diagram architektúry



Návrhové rozhodnutia

Vuex store

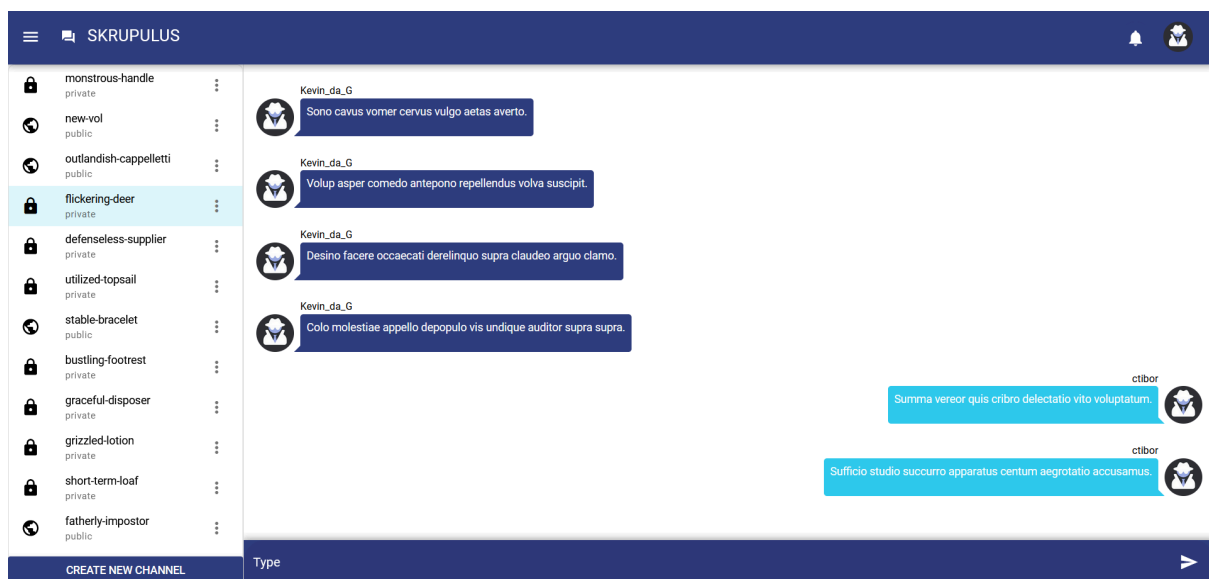
Rozhodli sme sa použiť vuex store, ktorý nám bol odporúčaný prednášajúcim predmetu. A jeho jednoduchosti na implementáciu v našom projekte.

PostgreSQL

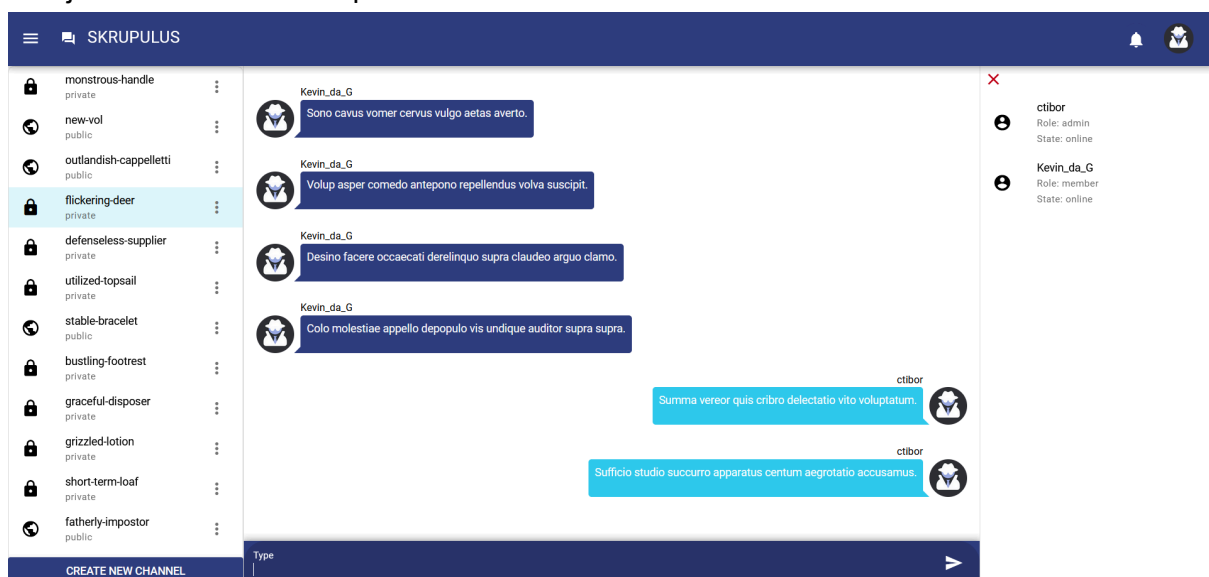
Pre relačný databázový systém sme zvolili postgresql natoľko sme mali v ňom skúsenosti z predošlých predmetov.

AdonisJS v6

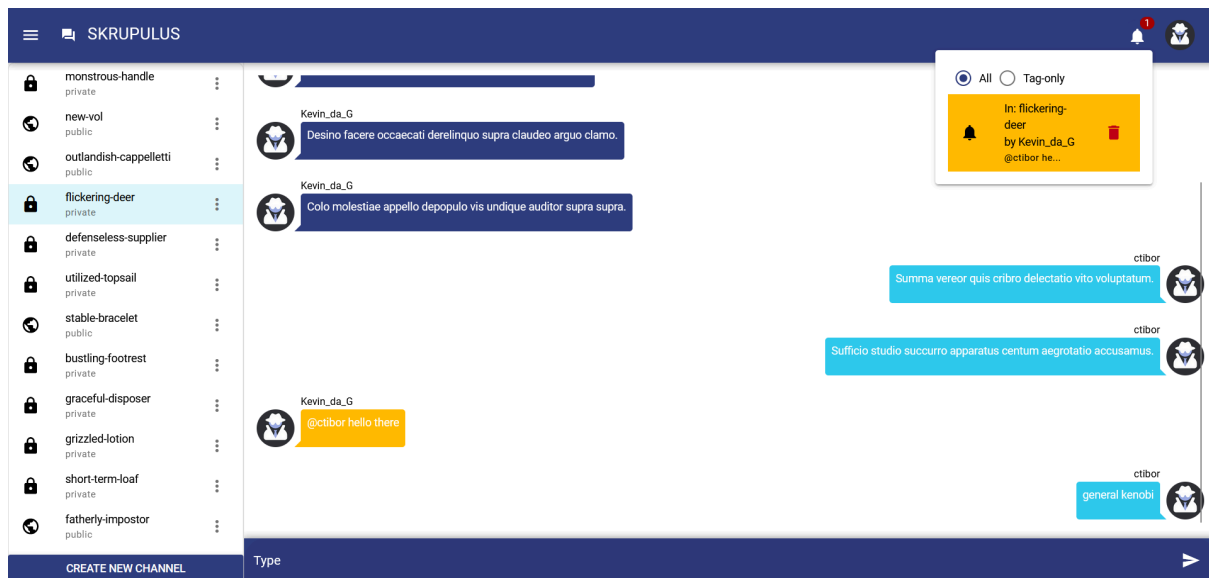
Adonis 6 sme zvolili, lebo sme chceli skúsiť túto najnovšiu verziu back-endového servera.



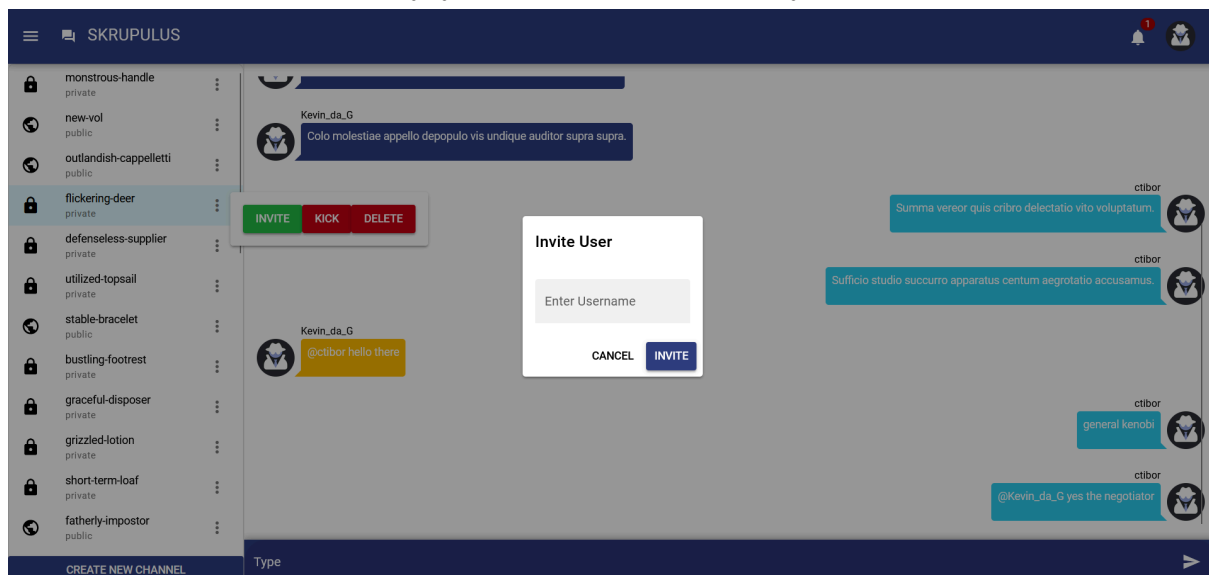
Toto je hlavná obrazovka aplikácie SKRUPULUS



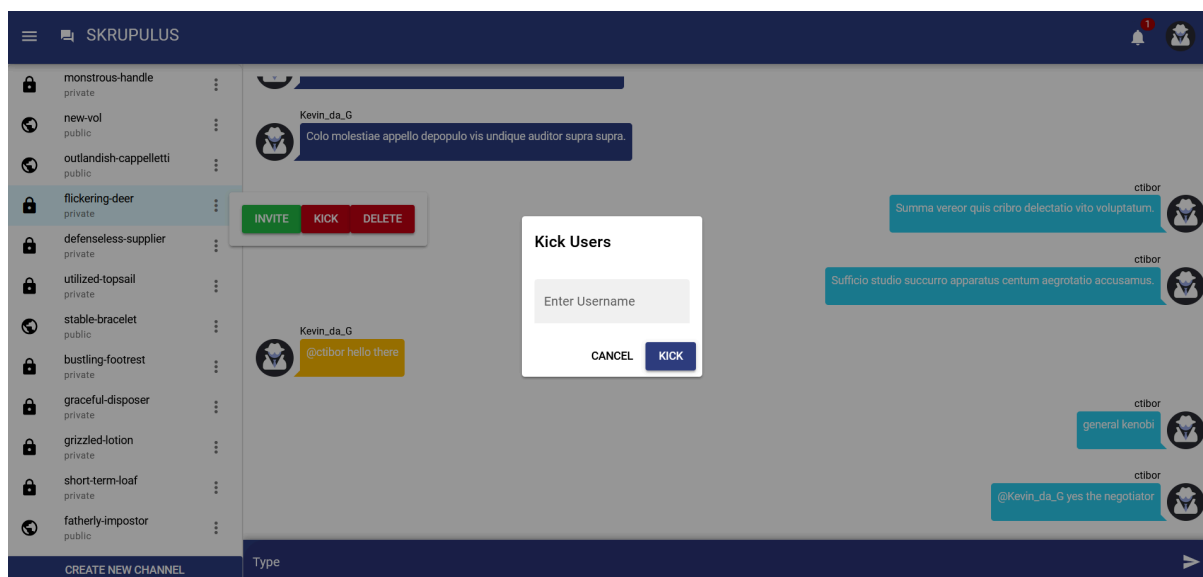
Cez príkaz /list zobrazíme zoznam používateľov z daného kanála



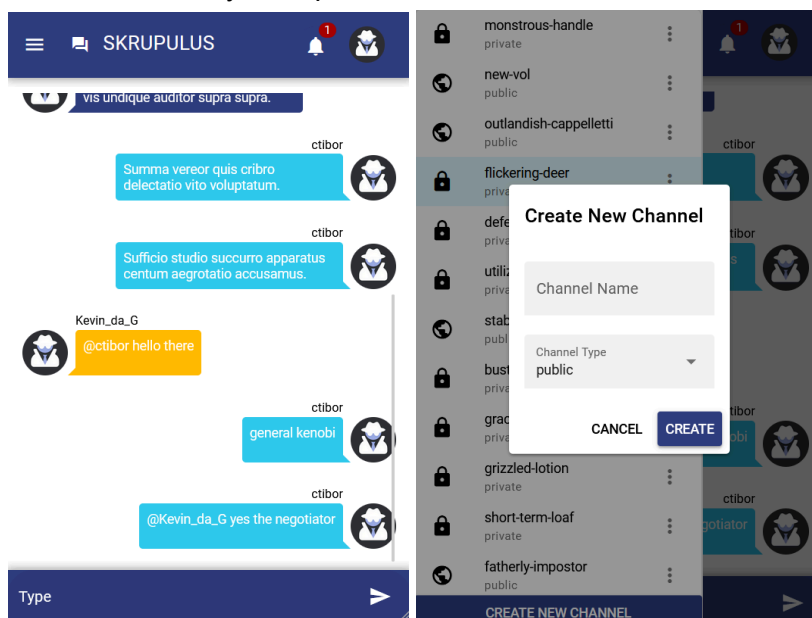
@Username adresované správy týmto spôsobom sa nám zvyraznia.



Cez GUI vieme pridať používateľa do kanála



Cez GUI vieme vyhodit' používateľa z kanála



Cez GUI vieme aj vytvoriť nový kanál všetky tieto príkazy sa dajú spraviť aj z príkazového riadka.