

# Dokumentácia

Dominik Mojto xmojtod  
Alexander Ovšonka xovsonka

December 8, 2024

## Contents

<b>1</b>	<b>Zadanie</b>	<b>3</b>
1.1	Aplikácia na textovú komunikáciu v štýle IRC (zjednodušený Slack)	3
<b>2</b>	<b>Diagram fyzického dátového modelu</b>	<b>5</b>
2.1	Zmeny v tabuľkách . . . . .	5
<b>3</b>	<b>Návrhové rozhodnutia (pridanie externej knižnice - zdôvodnenie)</b>	<b>6</b>
<b>4</b>	<b>Diagram architektúry aplikácie</b>	<b>7</b>
<b>5</b>	<b>Snímky obrazoviek</b>	<b>7</b>

# 1 Zadanie

Vytvorte progresívnu webovú aplikáciu na textovú komunikáciu v štýle IRC (Slack), ktorá komplexne rieši nižšie definované prípady použitia.

## 1.1 Aplikácia na textovú komunikáciu v štýle IRC (zjednodušený Slack)

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

- registrácia, prihlásenie a odhlásenie používateľa
  - používateľ má meno a priezvisko, nickName a email
- 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)
- používateľ odosiela správy a príkazy cez "príkazový riadok", ktorý je "fixným" prvkom aplikácie. používateľ môže odoslať správu v kanáli, ktorého je členom
- 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
- používateľ môže zrušiť svoje členstvo v kanáli príkazom /cancel, ak tak spraví správca kanála, kanál zaniká
- správu v kanáli je možné adresovať konkrétnemu používateľovi cez príkaz @nickname
  - správa je zvýraznená danému používateľovi v zozname správ
- používateľ si môže pozrieť kompletnú históriu správ
  - efektívny infinite scroll
- 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
- 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
- používateľ si môže pozrieť zoznam členov kanála (ak je tiež členom kanála) príkazom /list
- 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 :-)

## 2 Diagram fyzického dátového modelu

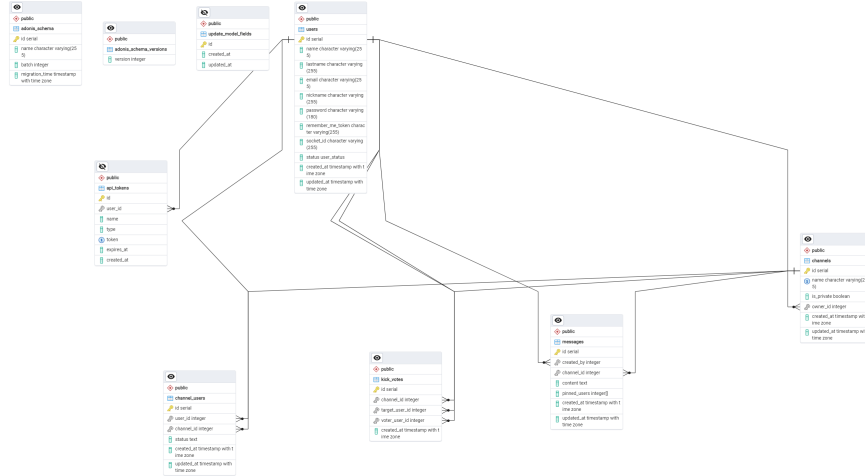


Figure 1: ERD model databázy

### 2.1 Zmeny v tabuľkách

Pri návrhu databázového modelu chatovej aplikácie sme vykonali niekoľko zásadných zmien a optimalizácií, ktoré reflektujú potrebu zvýšenia efektivity, rozšírenia funkcionality a zabezpečenia konzistencie dát. Tieto úpravy sú popísané nižšie spolu so zdôvodnením.

### Úpravy tabuliek

#### 1. Tabuľka users:

- Pridali sme stĺpec **remember\_me\_token**, ktorý umožňuje implementáciu funkcie „zapamätania používateľa“ pri prihlasovaní. Táto funkcionality zjednodušuje opätovné prihlásenie pre používateľov, ktorí sa často vracajú do aplikácie.
- Pridali sme stĺpec **socket\_id**, ktorý umožňuje jednoznačnú identifikáciu WebSocket spojenia používateľa. Táto úprava je kľúčová pre správne smerovanie správ v reálnom čase.
- Pridali sme stĺpec **updated\_at**, ktorý umožňuje sledovať čas poslednej aktivity používateľa. Tento údaj bude využitý na identifikáciu neaktívnych používateľov alebo neaktívnych kanálov (neaktívnych viac ako 30 dní).

#### 2. Tabuľka channel\_users:

- Tabuľka `channel_membership` bola premenovaná na `channel_users`, aby názov presnejšie odrážal jej obsah a vzťah medzi kanálmi a používateľmi.
- Pridali sme stĺpec `updated_at` na sledovanie času poslednej aktivity v rámci kanála.

### 3. Tabuľka `channels`:

- Stĺpec `created_by` bol premenovaný na `owner_id`, aby sa jednoznačne označilo, že tento atribút predstavuje vlastníka kanála.

### 4. Tabuľka `messages`:

- Pridali sme stĺpec `updated_at` na zaznamenanie poslednej úpravy správy.
- Pridali sme stĺpec `pinned`, ktorý umožňuje identifikovať pripnuté správy. Táto funkcionality je dôležitá na zvýraznenie dôležitých správ v kanáli.

### 5. Tabuľka `kick_votes`:

- Tabuľka `bans` bola premenovaná na `kick_votes`, aby lepšie reprezentovala mechanizmus hlasovania za vykázanie používateľa z kanála. Tento nový prístup je transparentnejší a demokratickejší.

### 6. Nová tabuľka `api_tokens`:

- Vytvorili sme tabuľku `api_tokens`, ktorá spravuje prácu so socketmi a zabezpečuje autentifikáciu pri komunikácii prostredníctvom API.

## Zrušené tabuľky

Tabuľky, ktoré sa ukázali ako nadbytočné, boli z návrhu odstránené. Táto optimalizácia umožňuje udržať databázový model jednoduchý a prehľadný.

## 3 Návrhové rozhodnutia (pridanie externej knižnice - zdôvodnenie)

Pri návrhovom rozhodnutí sme využili iba Quasar na frontend a AdonisJS na realizáciu backendu. Na ukladanie údajov bola zvolená databáza PostgreSQL z dôvodu predošlej práce s touto databázou.

## 4 Diagram architektúry aplikácie

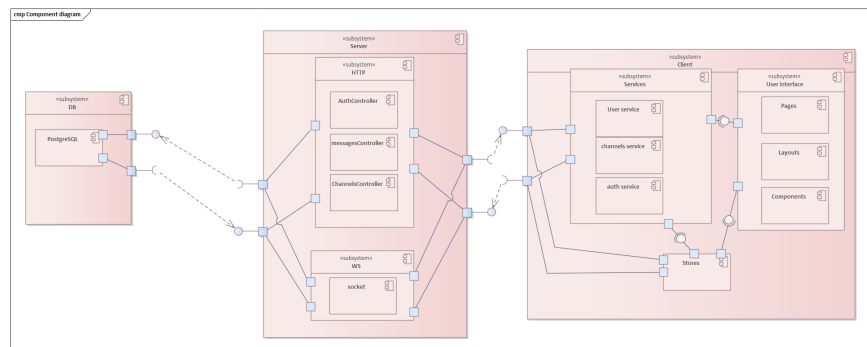


Figure 2: Diagram fungovania aplikácie

## 5 Snímky obrazoviek

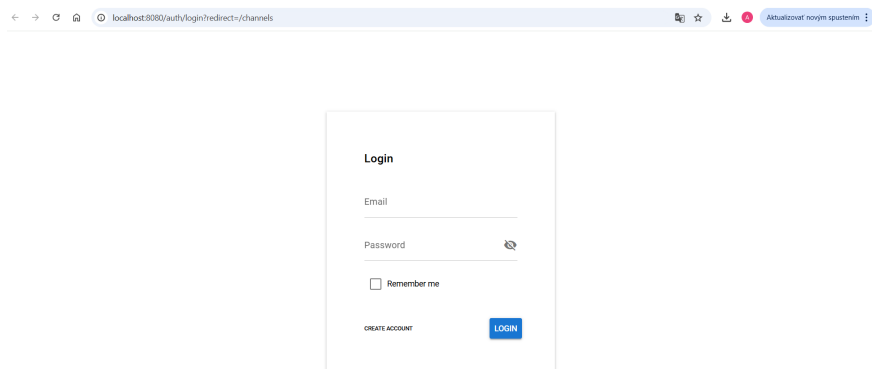


Figure 3: Prihlásenie sa

Jednoduchý prihlasovací formulár s možnosťou prihlásenia, vytvorenia nového účtu a poskytovanou možnosťou "Zapamätaj si ma".

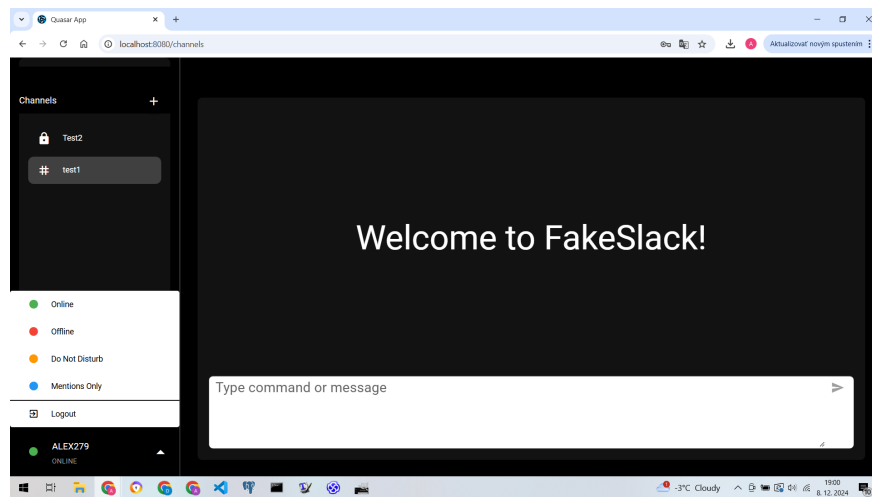


Figure 4: Homepage

Jednoduchá domovská obrazovka, na ktorej si môže používateľ vybrať stav svojho účtu (DND, offline a pod.), zvoliť jednotlivé kanály, ktorých je členom, alebo prípadne vytvoriť nový kanál.

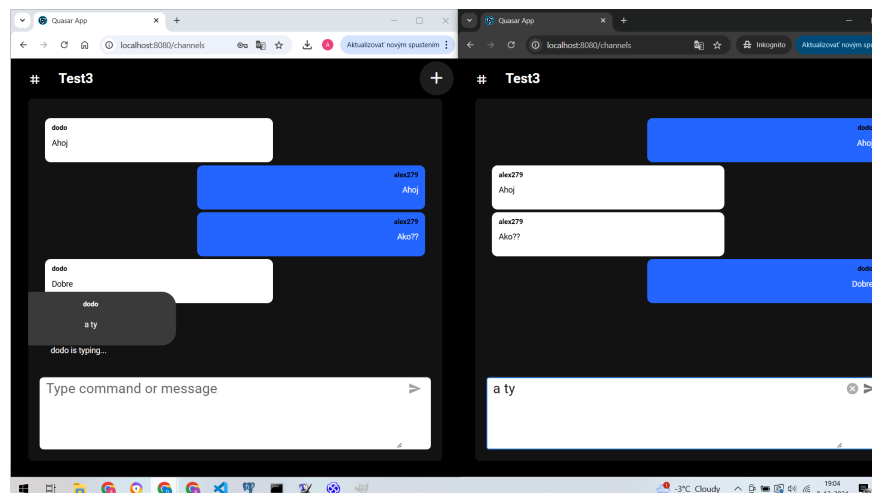


Figure 5: Ukážka komunikácie

Aplikácia poskytuje jednoduché posielanie správ s prehľadom o tom, čo píše používateľ.



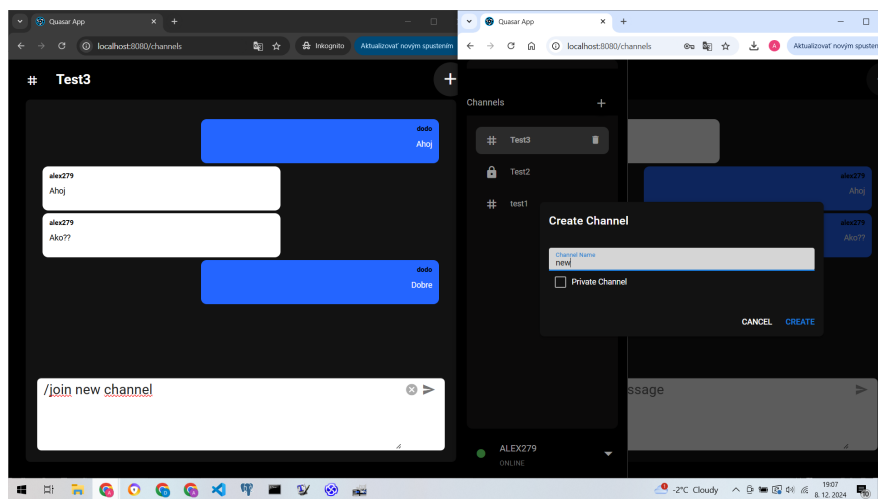


Figure 6: Vytváranie nového kanálu

Ľahké a jednoduché vytváranie nového kanála prostredníctvom nami poskytnutého formulára alebo príkazu "join".

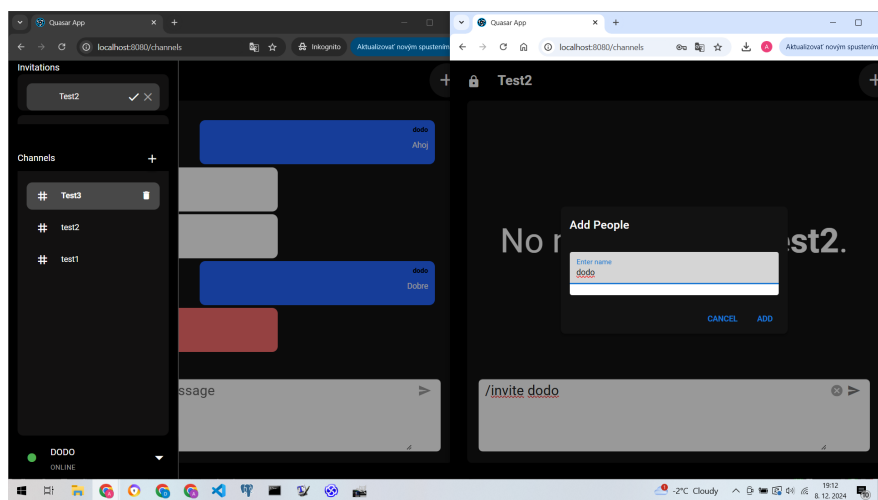


Figure 7: Pozývanie ľudí do kanálu

Pozvanie ľudí do kanála je tiež intuitívne, buď na základe nášho formulára, alebo prostredníctvom príkazu. Okrem toho si môžete pozrieť, ako vyzerá dané oznámenie o pozvánke.

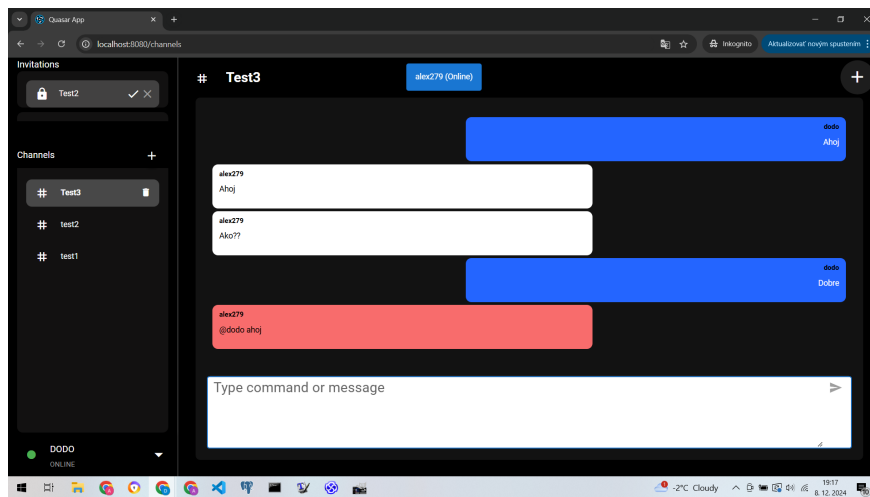


Figure 8: Príkaz list a topovaná správa

V aplikácii je množstvo príkazov, z ktorých jeden je aj príkaz "ist", ktorý poskytne oznámenie o výpise členov kanála s ich aktuálnym stavom. Taktiež aplikácia umožňuje zasielanie adresovaných správ, ktoré sú zvýraznené červenou farbou.