

Aplikácia na textovú komunikáciu v štýle IRC (Slack)

PROJEKT Z PREDMETU VPWA

MAREK PENZEŠ, TOMÁŠ MERAVÝ MURÁRIK

Zadanie

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

Naše riešenie zodpovedá všetkým nižšie stanoveným podmienkam:

1. Používateľ má dostupné všetky základné prihlasovacie funkcionality a má nastaviteľné meno, nickName (username) a email. Screenshoty týchto obrazoviek môžete vidieť nižšie, konkrétne obrázok **login**, **register** a **chatPage** v pravom hornom rohu môžete vidieť ikonu pre odhlásenie.
2. Používateľ vidí zoznam všetkých kanálov, v ktorých je členom. Túto funkcionality môžete vidieť na **chatPage**. Existujú 2 typy skupín, **public** a **private**. Skupiny môže vytvoriť ktokoľvek, a to public aj private, no nemôžu existovať 2 skupiny pod tým istým menom. Používateľ sa môže voľne pripojiť do akejkoľvek public skupiny, no do súkromnej skupiny môže niekoho pozvať iba tvorca tejto skupiny. Pozvať niekoho do skupiny môžete zadaním príkazu **/invite <nickName>** v danej skupine. Pozvánky sa zobrazia v samostatnej sekcii (**Obr invite**) nad všetkými skupinami, ktorých ste členom. Všetky doposiaľ prijaté pozvánky môžete vidieť po rozkliknutí tejto sekcie (**Obr invite**), kde máte následne možnosť pozvánku prijať alebo odmietnuť. Používateľ môže byť tiež vyhodnený, ak tvorca kanálu zadá príkaz **/revoke <nickName>** alebo **/kick <nickName>**, alebo ak za jeho vyhodnenie zahlasujú aspoň 3 členovia kanálu zadaním príkazu **/kick <nickName>**. V oboch prípadoch bude používateľ permanentne vyhodnený. V zozname kanálov sa na pravej strane každej skupiny nachádza dropdown menu s vykonateľnými akciami. Ako bežný používateľ máte možnosť skupinu iba opustiť, no ako tvorca máte možnosť ju vymazať. Ak skupinu opustí tvorca, skupina sa **automaticky vymaže**. Ak do skupiny nie je pridaná nová správa po dobu 30 dní, skupina sa automaticky vymaže.
3. Posielanie správ je realizované pomocou socketov, a teda používateľ má možnosť posilať správy do skupín, ktorých je členom, v reálnom čase pomocou **vstupného (príkazového) poľa**.
4. Tvorba skupiny je možná cez používateľské rozhranie, ale aj cez príkazový riadok. Zadaním príkazu **/join <meno> <popis>** je možné vytvoriť skupinu, ak ešte skupina s takýmto menom neexistuje. Vytvorenie súkromnej skupiny je možné pridaním "[private]" za meno skupiny.
5. Používateľ môže zrušiť svoje členstvo v skupine zadaním príkazu **/cancel**.
6. Správu je možné adresovať konkrétnemu používateľovi začatím správy znakom **"@<nickName>"**. V takomto prípade sa správa danej osobe zvýrazní.
7. Používateľ si môže zobrazíť kompletnú históriu správ pomocou efektívnej implementácie **infinity scroll** komponentu z knižnice Quasar.
8. Používateľ je informovaný o každej novej správe pomocou notifikácie. Pri prvotnom spustení sa spýta browser, či má povolenie posilať notifikácie, ak dá user áno, tak pri poslaní správy od iného user-a v momente, keď nemá appku v stave „visible“, mu príde notifikácia s menom

user-a, ktorý správu poslal a časť jej obsahu. Existuje v našom systéme viacero druhov, ako napr. mention-only, kde príde len notifikácia ktorá obsahuje **@menousera**, pri DnD nechodia žiadne notifikácie ako aj pri offline...

9. Používateľ si môže nastaviť svoj status. Status v našej aplikácii slúži čisto ako manažér notifikácií a pripojenia. Dostupné stavy sú:

online - používateľ prijíma všetky správy aj notifikácie

idle - používateľ prijíma všetky správy, ale notifikácie iba ak sú mu adresované

dnd - používateľ prijíma všetky správy, ale notifikácie má vypnuté

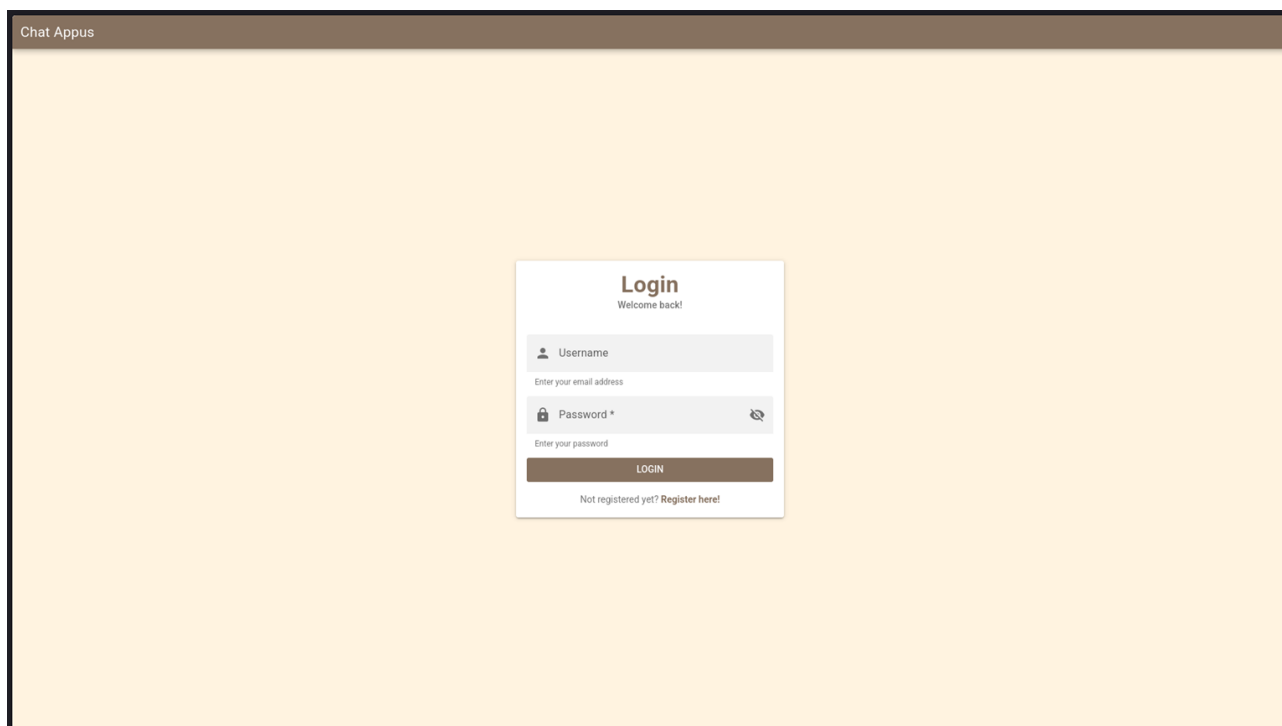
offline – používateľ neprijíma správy ani notifikácie

10. Používateľ si môže zobrazíť zoznam členov pomocou príkazu /list.

11. Používateľ má možnosť „nazrieť“ na práve písané správy inými členmi v kanáli, v ktorom je pripojený. Na správy môže nahliadnuť otvorením dropdown menu a kliknutím na meno príslušnej osoby.

Screenshots:

Obr. login



Obr. register

Chat Appus

Register

Create your account

Username *

Choose your username

First Name*

John

Last Name*

Smith von Gutenberg

Email Address*

example@yahoo.com

Password *

Enter your password (min. 4 characters)

Confirm Password *

Re-enter your password

REGISTER

Already have an account? Login here!

Obr. chatPage

Groups

test

+

☰

☰ Mickeys Clubhouse

☰

👤

test

toto by mali byt lorem ipsum spravy lebo som povedala

third

dalej testujem

third

zaujimalo by ma ci sa bude este nieco diat

third

@test prestan ma pingovat to ci pisem bez alebo s diakikikikou je moja vec

third

Pardon.. Ako ospravednenie poviem vtip. Do you know what happens to a Rock that has overcome it's greatest fear ?

third

it becomes a little boulder .DDDDDDDDDDDDDDDDDDDDDDDD hhhahahahhaahhaahah

test

preco píšeš bez diakritiky?

test

toto je veľmi dôležité

test

@third vysvetli sa

test

no tak taketo pingovanie si teda vyprosim

test

prestán hneď teraz

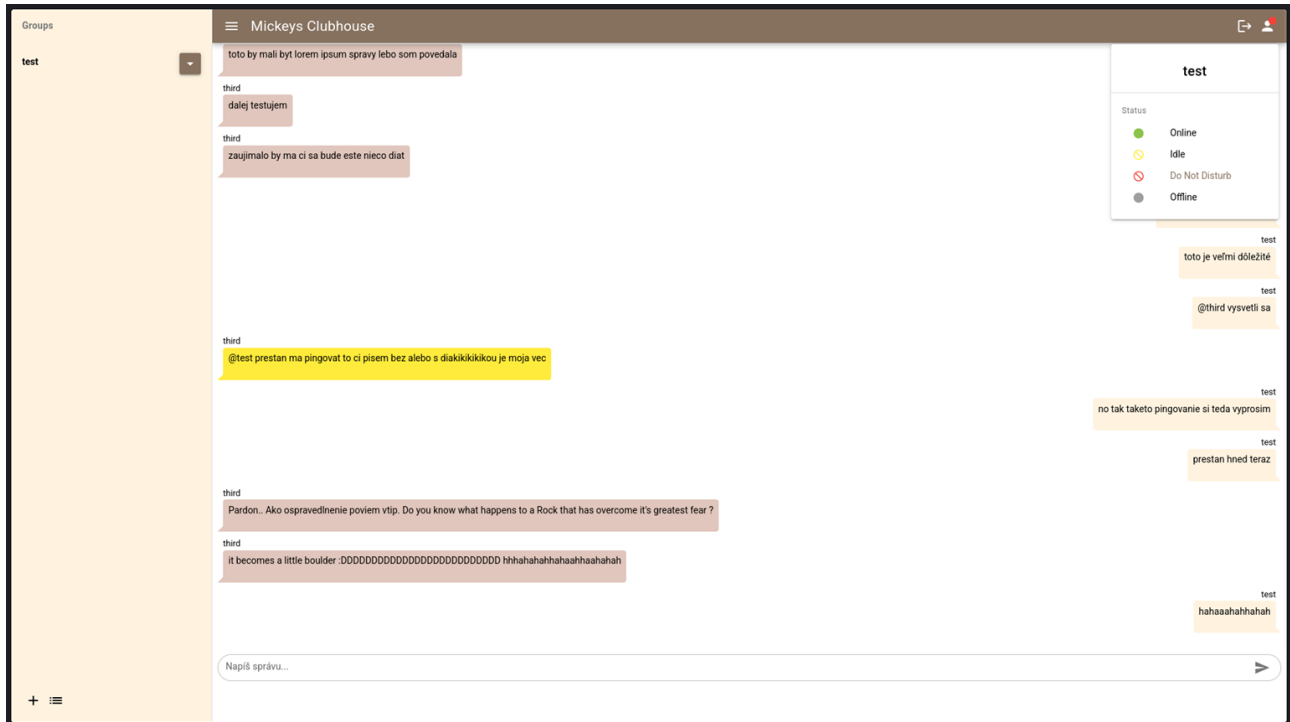
test

hahaaahahaha

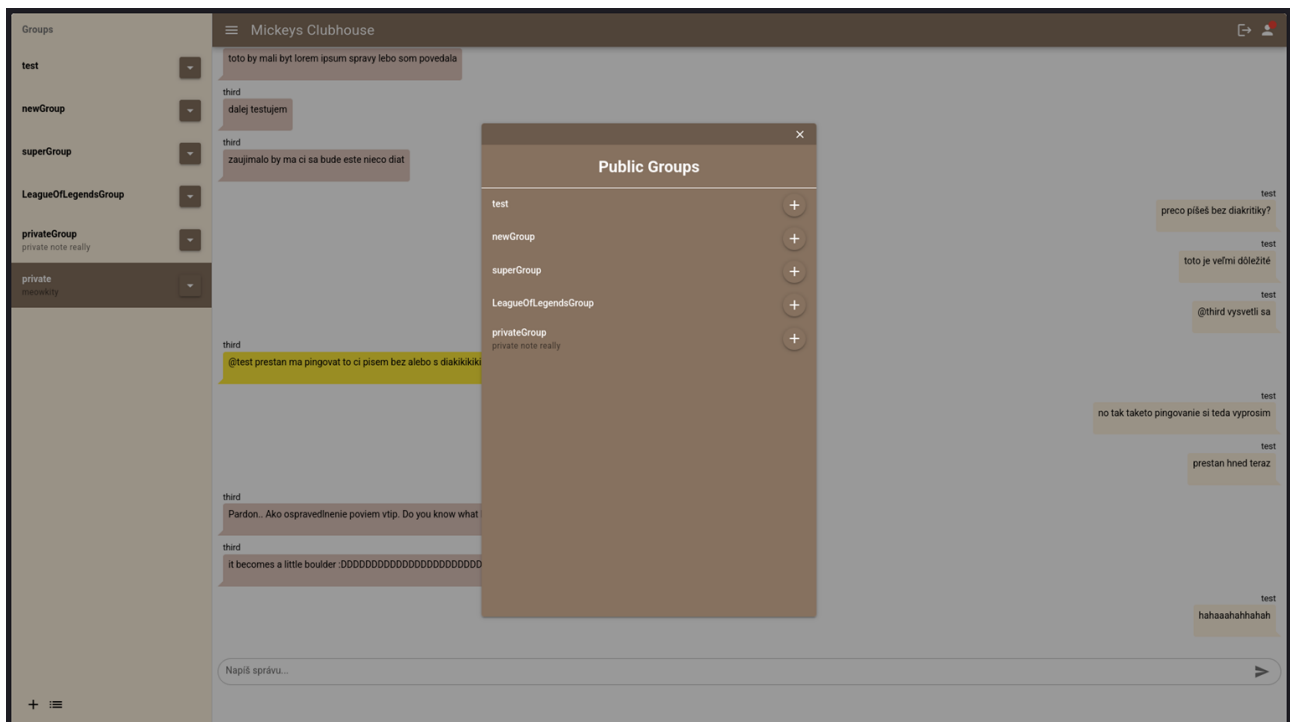
Napiš správu...

➤

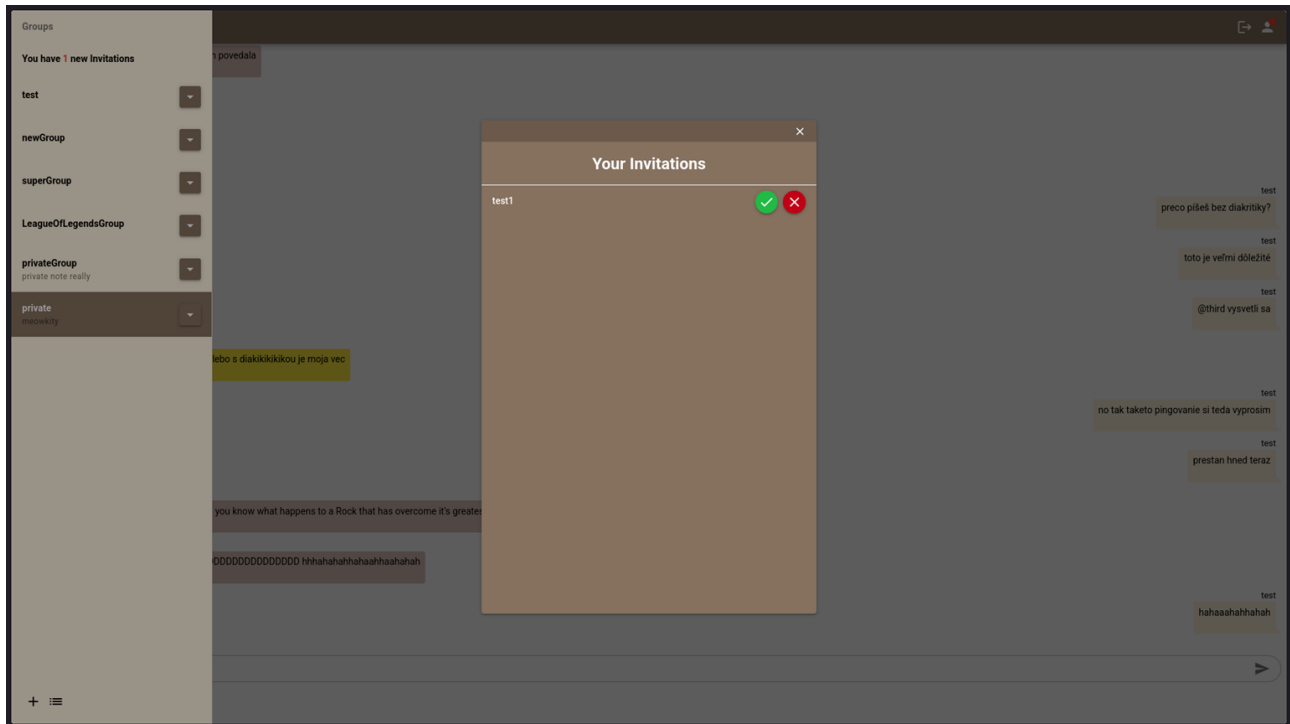
Obr. status



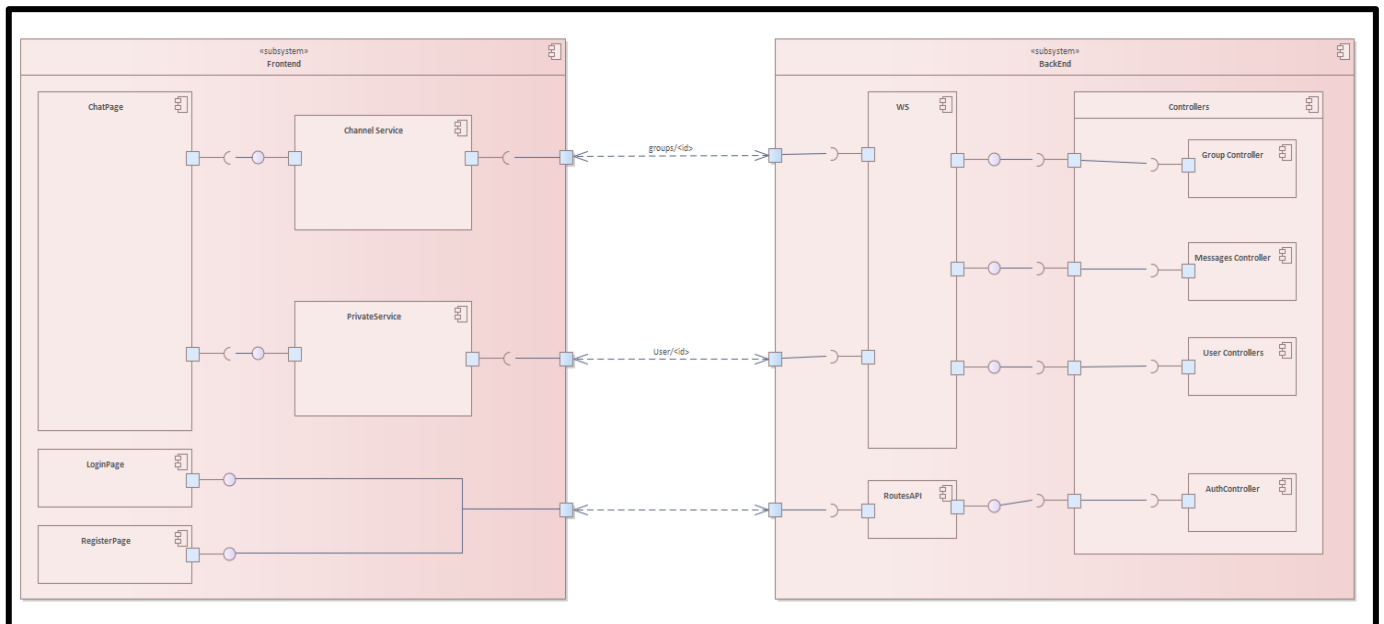
Obr. groupList



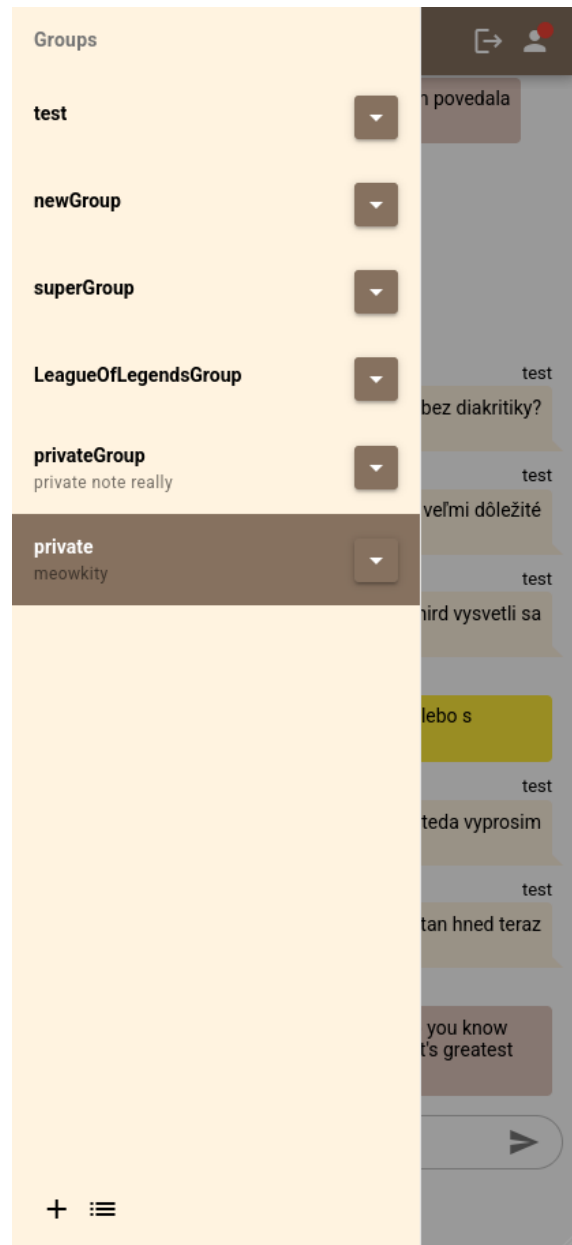
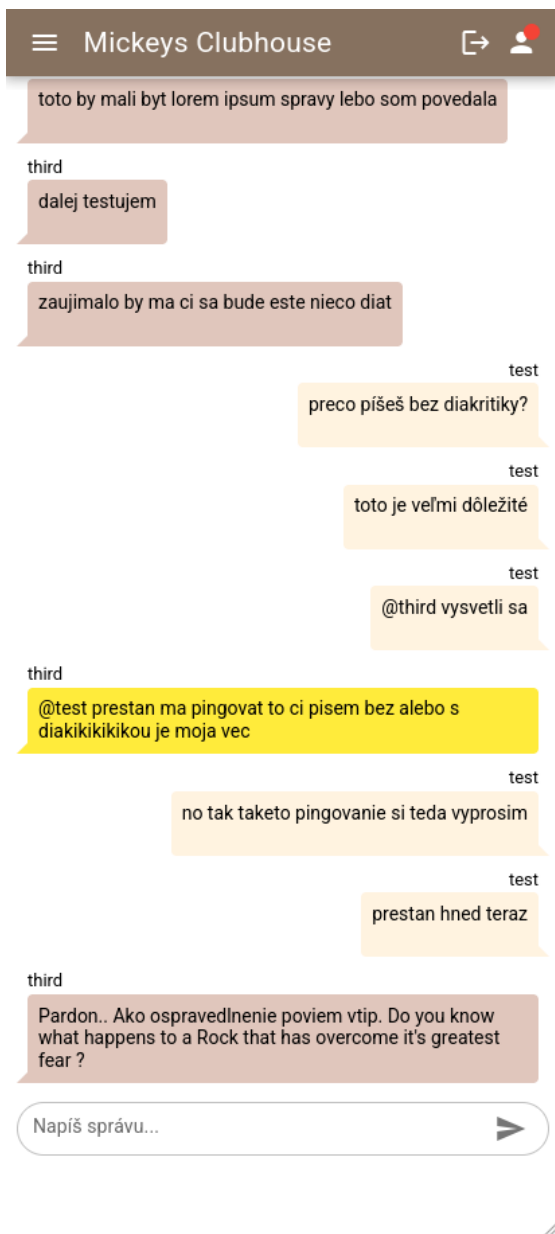
Obr. invite



Obr. Diagram Architektúry aplikácie



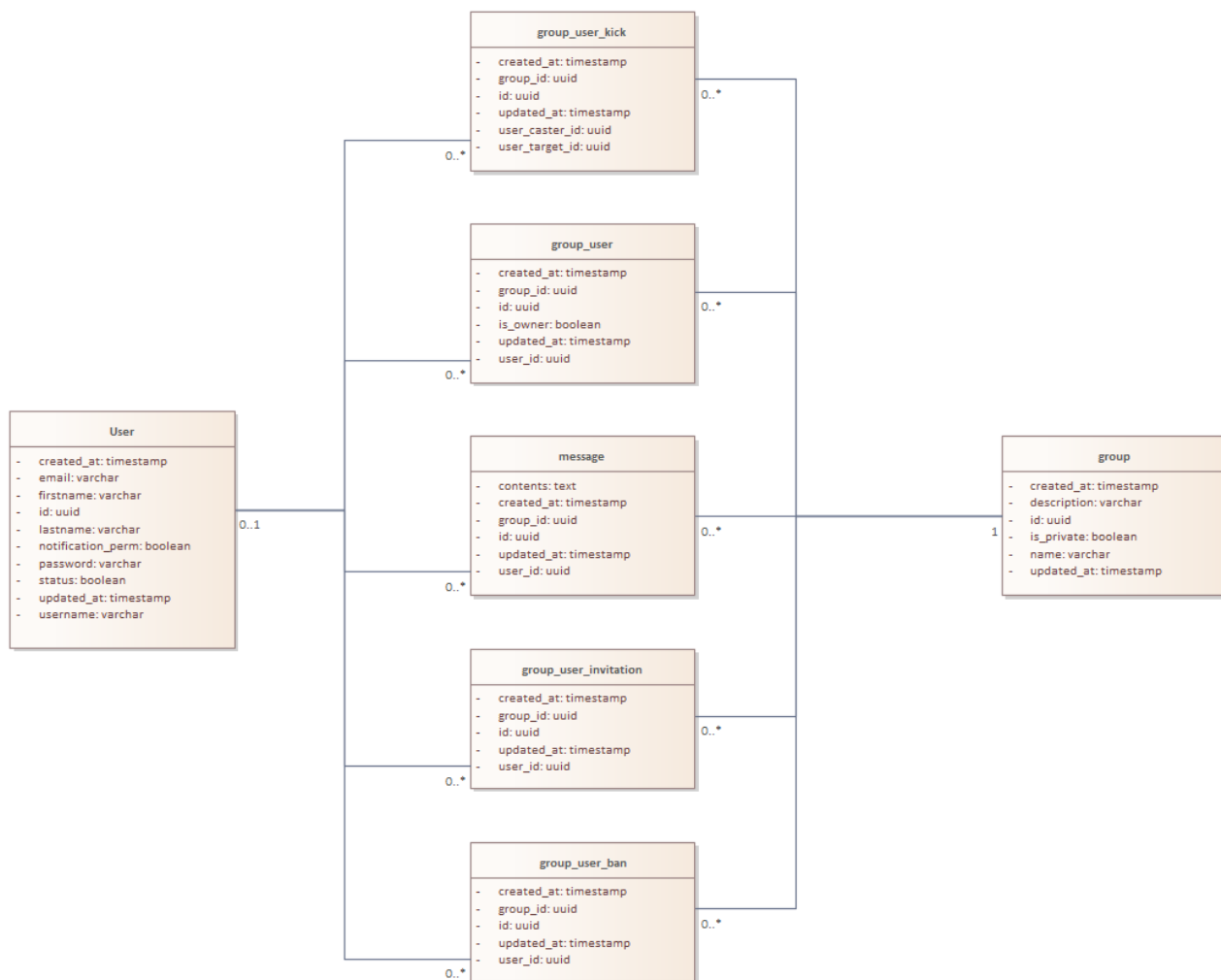
Obr. responsivity



Rozdiely medzi návrhom databázy v prvom odovzdaní projektu a aktuálne používaným databázovým modelom sú nasledovné:

1. tabuľky, ktoré pôvodne nemali id, ho teraz majú. Dôkladnejšie štúdium **LucidORM** nám ukázalo dôležitosť týchto stĺpcov aj pri tabuľkách, kde na prvý pohľad nemusia byť potrebné.
2. pridanie stĺpcov `created_at` a `updated_at`. Pridanie týchto stĺpcov je považované za dobrú prax pri práci s PostgreSQL, ale aj pri nasadzovaní aplikácie na server, keďže tieto údaje môžu byť dôležité pri spätnom dohľadávaní chýb.

Obr. Diagram fyzického dátového modelu



Použité knižnice:

- **Socket.io** - knižnica pre real-time obojsmernú komunikáciu medzi klientom a serverom. Funguje na princípe WebSocketov. Hlavná výhoda Socket.IO je, že umožňuje serveru "pushovať" dáta klientom v reálnom čase bez toho, aby klient musel neustále posielať requesty a pýtať sa na nové dáta. Komunikácia funguje cez eventy - server môže vyslať event (napr. "message") a všetci pripojení klienti, ktorí počúvajú tento event, ho okamžite dostanú. Socket.IO podporuje **namespaces** (oddelené kanály) a rooms (skupiny socketov), čo umožňuje jednoducho posielať správy len určitým používateľom alebo skupinám. Konkrétne v našom prípade sa „namespaces“ hodia práve pre komunikáciu v groups/skupinách, kde správy dostávajú len osoby, ktoré sú v konkrétnej skupine, ostatní nie. Knižnica automaticky rieši reconnect pri výpadku spojenia a má veľmi jednoduchý API na oboch stranách - klient aj server používajú podobné metódy emit() na poslanie a on() na prijímanie eventov.