Dokumentácia VPWA

Filip Hromada

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

xhromadaf@stuba.sk

December 2025

Abstrakt

Projekt implementuje jednoduchú real-time chatovaciu aplikáciu s podporou kanálov, posielania správ, správy členov a základných moderátorských akcií. Používatelia môžu vytvárať a spravovať kanály, komunikovať v nich, sledovať aktivitu ostatných členov, zobrazovať stav písania správ a využívať systém upozornení. Súčasťou je aj evidencia neprečítaných správ, pozvánok, rolí používateľov a hlasovanie o vyhodení člena z kanála. Aplikácia poskytuje prehľadné rozhranie pre každodennú komunikáciu v menších skupinách.

1 Dátový model

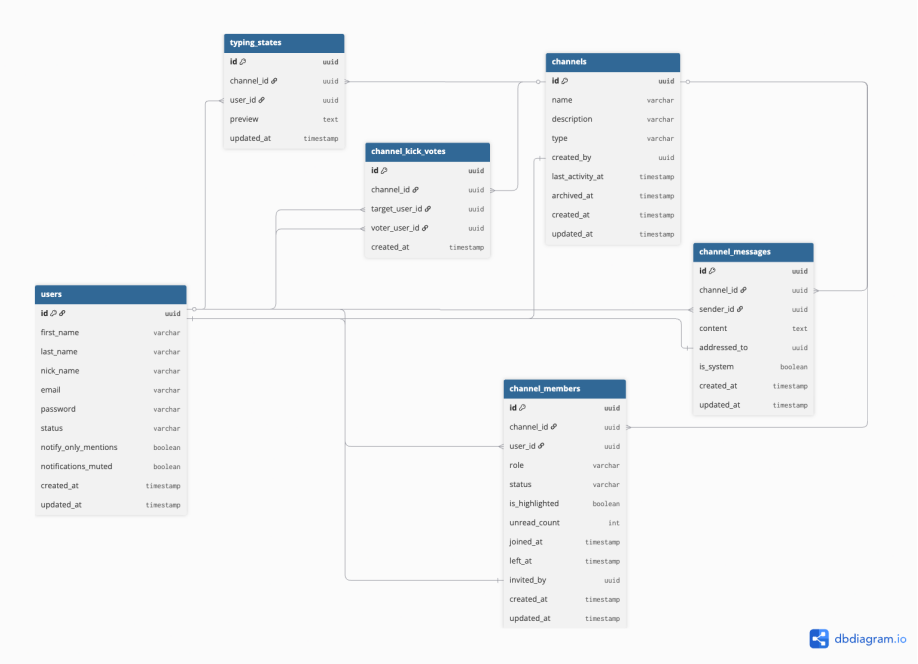
1.1 Pôvodný návrh

Pôvodný logický dátový model, ktorý vytvoril kolega v skoršej fáze projektu, zachytával základné entity systému: používateľov, kanály, správy a používateľské preferencie. Model obsahoval korektné základné vzťahy, napríklad vzťah M:N medzi používateľmi a kanálmi a väzbu „používateľ posiela správu“. Išlo však len o konceptuálny návrh, ktorý nebol ďalej rozpracovaný do detailov potrebných na implementáciu.

Počas vývoja backendu sa ukázalo, že pôvodný model nezohľadňuje všetky požiadavky aplikácie a niektoré entity chýbajú alebo sú príliš zjednodušené. Preto bolo potrebné model rozšíriť a upraviť tak, aby verne reflektoval implementované funkcionality.

1.2 Finálny fyzický model

Finálny dátový model vychádza priamo z implementovaných migrácií a presne zodpovedá aktuálnej štruktúre databázy. Obsahuje šesť tabuliek, ktoré pokrývajú všetky funkcie real-time chatovacej aplikácie.



Obr. 1: Finálny dátový model aplikácie

users

Tabuľka obsahuje údaje o používateľoch vrátane mena, e-mailu, hesla, prezývky a notifikačných preferencií. Preferencie boli presunuté priamo sem, aby sa odstránila potreba samostatnej tabuľky.

channels

Reprezentuje jednotlivé komunikačné kanály. Obsahuje informácie o názve, type, archivačnom stave a používateľovi, ktorý kanál vytvoril.

channel_members

Pivot tabuľka prepájajúca používateľov a kanály. Okrem základného členstva obsahuje: roly (admin/member), stav (active, invited, banned, left), počet neprečítaných správ, informáciu o tom, kto používateľa pozval, a čas vstupu alebo odchodu z kanála. Oproti pôvodnému modelu ide o výrazné rozšírenie.

channel_messages

Obsahuje správy v kanáloch vrátane systémových správ a voliteľného adresovania konkrétnemu používateľovi. Oproti pôvodnému návrhu má viacero doplňujúcich atribútov.

typing_states

Sleduje stav „používateľ píše správu“. Táto entita v pôvodnom modeli vôbec nebola.

channel_kick_votes

Podporuje komunitné hlasovanie o vyhodení člena z kanála. Táto funkcionalita tiež chýbala v pôvodnom návrhu.

1.3 Zdôvodnenie zmien oproti pôvodnému modelu

Nižšie sú uvedené všetky zmeny, ktoré bolo potrebné uskutočniť pri prechode z pôvodného logického modelu k finálnemu fyzickému návrhu.

1. Odstránenie tabuľky „User preferences“

V pôvodnom modeli bola navrhnutá samostatná tabuľka pre používateľské preferencie, avšak obsahovala iba dve logické hodnoty. Keďže aplikácia k preferenciám pristupuje pri každom načítaní používateľa, oddelená tabuľka by spôsobovala zbytočné JOIN operácie. Preferencie boli preto presunuté do tabuľky `users`.

2. Rozšírenie pivot tabuľky `channel_members`

Pôvodný model rátal len s obyčajným M:N vzťahom, ktorý nepostačoval pre požiadavky na moderovanie kanálov, pozývanie členov či počítanie neprečítaných správ. Pivot tabuľka bola rozšírená o roly, stavy, pozývajúceho používateľa a ďalšie atribúty.

3. Prepracovanie tabuľky správ

Kým pôvodný model rátal len s textom a časom vytvorenia, reálne správy musia podporovať reply, mention, systémové udalosti a rýchle načítavanie podľa času. Preto bola vytvorená tabuľka `channel_messages` s doplnenými stĺpcami a indexami.

4. Pridanie tabuľky `typing_states`

Táto tabuľka sleduje informáciu o tom, kto práve píše správu v danom kanáli. Ide o čisto real-time funkcionalitu, ktorú pôvodný model neobsahoval.

5. Pridanie tabuľky `channel_kick_votes`

Na podporu komunitného hlasovania o vyhodení člena bolo potrebné vytvoriť novú tabuľku, keďže takáto entita v pôvodnom návrhu neexistovala a nedala sa realizovať cez pivot tabuľku ani cez správový systém.

1.4 Zhrnutie

Pôvodný dátový model vytvorený kolegom bol vhodný ako konceptuálny základ, no nepostačoval pre implementáciu všetkých funkcionalít aplikácie. Finálny model databázy odráža skutočné potreby real-time chatovacej aplikácie a umožňuje správu správ, členstiev, notifikácií, moderátorských akcií aj real-time stavov.

2 Knižnice a návrhové rozhodnutia

Táto kapitola opisuje všetky externé knižnice, ktoré boli použité pri implementácii frontendu aj backendu aplikácie. Zameriava sa najmä na zdôvodnenie ich výberu, porovnanie s alternatívami a prínos pre celkovú architektúru systému.

2.1 Frontend

Frontend aplikácie bol implementovaný pomocou **Vue 3** a frameworku **Quasar**. Cieľom bolo vytvoriť moderné, responzívne a komponentovo orientované používateľské rozhranie, ktoré funguje rovnako dobre na mobilných zariadeniach aj na desktopoch.

2.1.1 Vue 3

Vue 3 bol zvolený ako hlavný frontendový framework z nasledovných dôvodov:

- poskytuje reaktívny dátový model a jednoduchý komponentový systém,
- umožňuje rýchlu implementáciu real-time časti aplikácie,
- má nízku vstupnú zložitosť a prehľadnú syntax,
- ponúka lepšiu integráciu s TypeScriptom v porovnaní s predchádzajúcou verziou.

Alternatívy ako React alebo Angular boli zvážené, avšak Vue poskytol najlepší pomer jednoduchosti a flexibility pre potreby projektu.

2.1.2 Quasar Framework

Quasar bol použitý ako UI framework, ktorý ponúka rozsiahlu sadu komponentov pre mobilné aj desktopové rozhranie.

Dôvody výberu:

- rýchla tvorba responzívneho dizajnu bez nutnosti rozsiahleho CSS,
- jednotný vizuálny štýl naprieč celou aplikáciou,
- hotové komponenty ako karty, prepínače, panely, zoznamy a navigácia,
- výborná integrácia s Vue 3 a Vite.

Ako alternatíva bol zvažovaný Vuetify či Tailwind CSS, no Quasar poskytol lepší mobilný dizajn a kratší čas vývoja.

2.1.3 Axios

Knižnica **Axios** bola zvolená pre realizáciu HTTP komunikácie s backendom.

Výhody:

- jednoduchšia práca s interceptormi a error handlingom,
- pohodlné nastavovanie hlavičiek pri autentifikácii,
- konzistentné spracovanie JSON odpovedí.

Hoci moderné prehliadače obsahujú `fetch` API, Axios poskytol vyšší komfort programovania.

2.1.4 Vue Router

Vue Router zabezpečuje navigáciu medzi stránkami aplikácie, ako napríklad Login, Register, Home či Chat. Jeho integrácia je priamočiara a bez konfliktných nastavení.

2.1.5 `@vpwa/shared`

Táto interná monorepo knižnica obsahuje zdieľané typy a dátové modely pre frontend aj backend. Výhodou je odstránenie duplicity dátových štruktúr a zníženie rizika vzniku nekompatibilných modelov.

2.2 Backend

Backend aplikácie bol implementovaný na frameworku **AdonisJS**, čo poskytlo silný základ pre spracovanie požiadaviek, prácu s databázou a validáciu vstupov.

2.2.1 AdonisJS Core

Framework **AdonisJS 5** bol zvolený pre svoju jasnú projektovú štruktúru a komplexný ekosystém. V porovnaní s Express.js ponúka oveľa viac funkcionality out-of-the-box.

Výhody výberu:

- zabudovaný DI kontajner,
- podpora TypeScriptu,
- integrovaný životný cyklus aplikácie,
- jednoduché definovanie route, middleware a kontrolérov.

Oproti NestJS je AdonisJS menej robustný, ale výrazne jednoduchší pre menší tím a rýchly vývoj chatovacej aplikácie.

2.2.2 Lucid ORM

Knižnica **Lucid ORM** slúži na prácu s databázou a definovanie modelov. Umožňuje jednoduché mapovanie vzťahov (*1:N*, *M:N*) a automatické generovanie migrácií.

Lucid bol zvolený ako najprirodzenejšia voľba, keďže je natívnou súčasťou AdonisJS.

2.2.3 PostgreSQL

Aplikácia používa databázový ovládač **pg** pre komunikáciu s PostgreSQL. Táto databáza bola vybraná pre:

- spoľahlivosť a stabilitu,
- podporu silných konzistentných transakcií,
- schopnosť efektívne spracovávať správy, členstvá a pivot tabuľky.

2.2.4 Adonis Validator

Slúži na validáciu všetkých vstupov, najmä pri registrácii, prihlásení, tvorbe kanálov a odosielaní správ. Zabraňuje nekorektným alebo škodlivým vstupom ešte pred ich spracovaním backendom.

2.2.5 Luxon

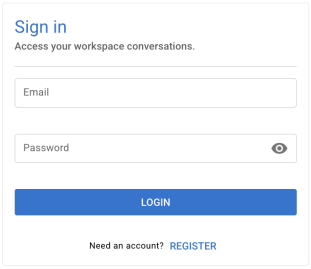
Knižnica **Luxon** bola použitá na prácu s dátumami a časom – najmä pri ukladaní časov odoslania správ a sledovaní aktivity používateľov. Oproti moment.js má nižšiu režiu a lepšiu podporu pre časové zóny.

2.3 Zhrnutie

Použité knižnice významne urýchlili vývoj aplikácie a poskytli stabilný základ pre real-time komunikáciu, validáciu dát a responzívne používateľské rozhranie. Výber knižníc bol motivovaný jednoduchou integráciou, dlhou podporou komunít a jasnou architektúrou vhodnou pre tímový vývoj.

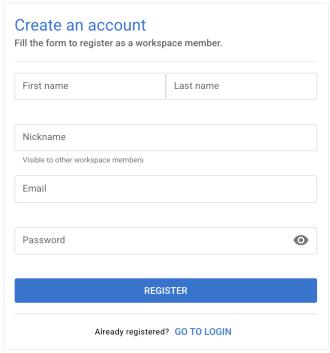
3 Obrazovky

Aplikácia podporuje nobilné a PC rozhranie. A na základe toho sa prispôsobujú jednotlivé komponenty a obrazovky.



Obr. 2: Login obrazovka

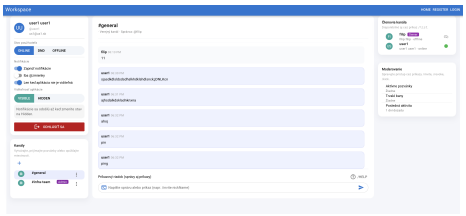
Login screen slúži na prihlásenie existujúceho používateľa. Používateľ zadá e-mail a heslo, pričom formulár validuje vstupy ešte pred odoslaním. Rozhranie je jednoduché, prehľadné a optimalizované pre rýchly vstup do aplikácie. V prípade nesprávnych údajov sa zobrazí chybové hlásenie.



Obr. 3: Registračná obrazovka

Registračný formulár umožňuje vytvoriť nový účet. Používateľ zadáva meno, e-mail a heslo. Proces registrácie zahŕňa kontrolu správnosti údajov (napr. formát e-mailu, silu hesla). Po úspešnej registrácii je používateľ automaticky

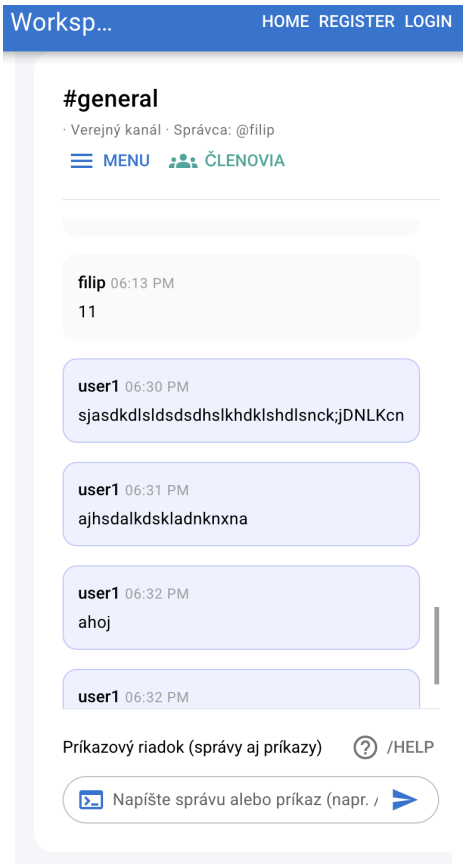
presmerovaný na login alebo priamo do aplikácie.



Obr. 4: Hlavná obrazovka

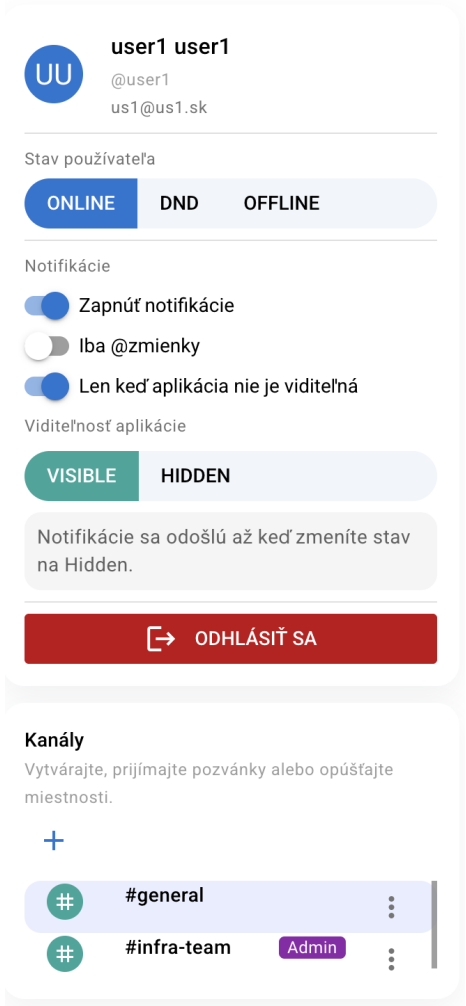
Hlavná obrazovka zobrazuje zoznam všetkých kanálov, v ktorých je používateľ členom. Každý kanál obsahuje názov, počet neprečítaných správ a poslednú aktivitu. Slúži ako centrálny navigačný bod aplikácie. V hornej časti sa nachádza možnosť vytvárať nové kanály alebo prejsť do nastavení.

Teraz si ukážeme rozhranie pre telefón.



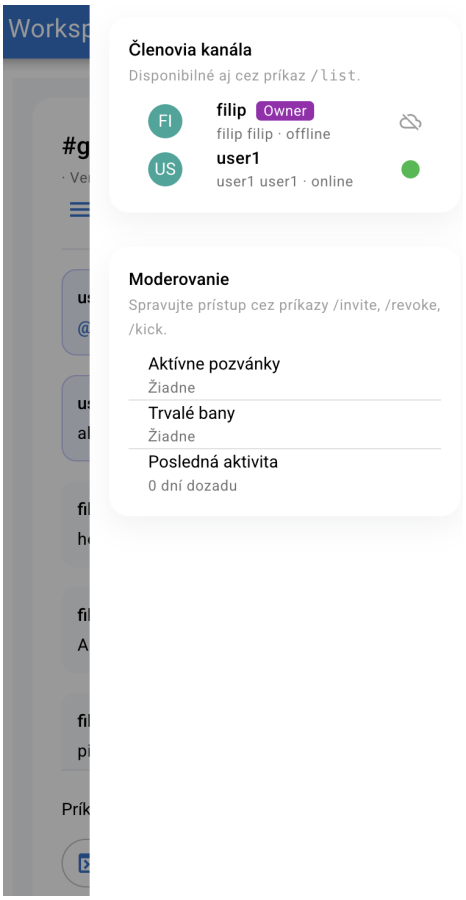
Obr. 5: Caption

Detailná obrazovka kanála zobrazuje zoznam správ v real-time režime. Používateľ vidí komunikáciu, môže reagovať, spomínať členov, prípadne sledovať systémové hlásenia. Súčasťou je aj vstupné pole pre odosielanie novej správy. Rozhranie sa prispôsobuje mobilným aj desktopovým zariadeniam.



Obr. 6: Caption

Detailná obrazovka kanála zobrazuje zoznam správ v real-time režime. Používateľ vidí komunikáciu, môže reagovať, spomínať členov, prípadne sledovať systémové hlásenia. Súčasťou je aj vstupné pole pre odosielanie novej správy. Rozhranie sa prispôsobuje mobilným aj desktopovým zariadeniam.



Obr. 7: Caption

Detail kanála poskytuje prehľad členov, ich rolí, stavov (active, invited, banned, left) a moderátorských možností. Používateľ môže vidieť, kto kanál vytvoril, kedy sa pripojil, prípadne môže odhlasovať vyhodenie člena. Obrazovka slúži ako hlavný manažmentový panel pre administráciu kanála.