

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

**Semestrálny projekt - aplikácia na textovú komunikáciu v
štýle IRC (Slack)**

Dávid Bunca, Dominik Daniel

Vývoj progresívnych webových aplikácií - VPWA

Čas cvičení: streda 10:00 - 12:00

Cvičiaci: Ing. Matúš Krajčovič

Obsah

<i>Zadanie</i>	3
<i>Diagram fyzického dátového modelu</i>	7
<i>Diagram architektúry aplikácie</i>	9
<i>Návrhové rozhodnutia</i>	10
socket.io	10
UI Avatars API	10
<i>Snímky obrazoviek</i>	11
Prihlásenie	11
Registrácia.....	11
Domovská obrazovka	12
Vytváranie/joinovanie chatov.....	12
Chat.....	13
List používateľov v chate	13
Zmena stavu používateľa.....	14

Zadanie

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

Tím

Projekt vypracovávate vo dvojici. Každý z dvojice sa musí podieľať na projekte významným dielom (rovnomé rozdelenie práce). Vypracovanie (takmer) celého projektu len jedným z dvojice autorov je neprípustné. Je potrebné, aby bol každý z autorov oboznámený s celým projektom, vrátane častí, na ktorých sám nepracoval. Autori sú hodnotení rovnakým získaným počtom bodov.

Autorstvo

Je zakázané používať programy alebo časti projektov od iných študentov z minulých rokov (automaticky hodnotenie FX). Všetky použité materiály z odbornej literatúry alebo z internetu musia byť citované (použite komentáre v zdrojovom kóde s odkazom na zdroj). Ak použijete cudzí materiál a neuvediete zdroj, práca môže byť považovaná za plagiát. Ak študent použije LLM (GPT-like služby) na generovanie kódu, každú jednu časť študent musí vedieť vysvetliť. Ak študent nebude vedieť kód vysvetliť (riadne mu rozumieť), nebude projekt akceptovaný.

Termíny odovzdania

- **Odovzdanie 1. fázy projektu: koniec 5. týždňa semestra - 20. 10. do 23:59 v AIS, 12 bodov**, vytvorenie responzívneho klikateľného prototypu používateľského rozhrania aplikácie na textovú komunikáciu vo forme Single Page Aplikácie (SPA) pre všetky prípady použitia s použitím rámca Quasar (framework), návrh dátového logického modelu v UML notácii
- **Odovzdanie 2. fázy projektu: koniec 12. týždňa semestra - 8. 12., do 23:59 v AIS, 30 bodov** vytvorenie progresívnej webovej aplikácie (PWA) na textovú komunikáciu v štýle IRC (Slack) podľa požiadaviek v zadaní projektu, dokumentácia

Termíny konzultácií k projektu

- Konzultácie k 1. fáze projektu, na cvičení: 2. - 5. týždeň semestra
- Konzultácie k 2. fáze projektu, na cvičení: 7. - 12. týždeň semestra

Termíny prezentovania

V čase cvičení tím predvedie na svojom počítači svoje riešenie (fázy projektu), a to:

- **Prezentovanie 1. fázy projektu, na cvičení: 6. týždeň semestra - 23.10.**
- **Prezentovanie finálneho projektu: 13. týždeň semestra** (prip. dohodneme termín individuálne pre tím)

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 "vyhodit" iného člena príkazom /kick nickName. ak tak spraví aspoň 3 členovia, používateľ má "trvalý" ban pre daný kanál. správca môže používateľa vyhodit "natrvalo" kedykoľvek príkazom /kick nickName, alebo naopak "obnovit" 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 :-)

Dátový model

V prvej fáze sa odovzdáva JPG (JPEG) obrázok logického dátového modelu (relačnej databázy) reprezentovaného UML class diagramom. V druhej (finálnej) fáze musí byť dátový model vytvorený prostredníctvom migrácií.

Spôsob odovzdávania

Počas semestra musíte mať vytvorený GITHUB repozitár (verejný). Do repozitára budete PRIEBEŽNE odovzdávať svoje výstupy. Budete si navzájom v tíme robiť "code review". V repe musí byť obsiahnutá rovnomerne aktivita každého člena tímu. Aktivita v repe bude slúžiť ako podklad k vášmu hodnoteniu.

Výstupy všetkých kontrolných bodov sa odovzdávajú do AISu. Odovzdáva iba jeden zo študentov v tíme. Dohodnite sa vopred, aby sa nestalo, že neodovzdá ani jeden z tímu. Odovzdávajú sa všetky zdrojové kódy aplikácie, okrem samotných rámcov a knižníc z manažéra balíkov (npm). V prípade, že študent modifikoval používanú knižnicu, je potrebné pribaliť aj zmenenú knižnicu a uviesť zmenu s odôvodnením v dokumentácii. Vytvorte subor "repo.txt", v ktorom bud link na vas verejný GITHUB repo (po skončení kurzu si ho môžete skryť).

Odovzdáva sa ZIP alebo RAR archív, "repo.txt" pripojte do archivu.

Oneskorenie odovzdania

V kontrolnom termíne sa môže odovzdanie oneskoriť maximálne o 3 dni. Za každý deň oneskoreného odovzdania je tímu odobratých 25% bodov z pôvodného maxima (deň po termíne tím získa 3/4 bodov, dva dni po termíne 1/2, atď.) Neskoršie odovzdanie nie je možné. Neodovzdanie niektorej časti projektu znamená nesplnenie podmienok absolvovania predmetu.

Kontrolná fáza progresu implementácie

V kontrolnej fáze - v 9. týždni semestra - sa očakáva implementovaná značná časť aplikácie. Fáza je hodnotená 5 bodmi, a to binárne. Tím letmo predvedie cvičiacemu funkčnosť aplikácie s ohľadom na požadované prípady použitia. Ak aplikácia umožňuje realizovať prvých 6 (z 11) prípadov použitia, každý člen tímu získa 5 bodov. Cvičiaci nebude v tejto fáze podrobne hodnotiť kvalitu kódu a robustnosť riešenia.

Implementačné prostredie

Odporúčané technológie:

- Tučný klient (SPA/PWA) - rámec Quasar
- Služby biznis logiky (backend) - rámec AdonisJS
- relačný databázový systém (napr. PostgreSQL, MySQL)

Použitie iných základných technológií nie je dovolené (netýka sa iných podporných knižníc).

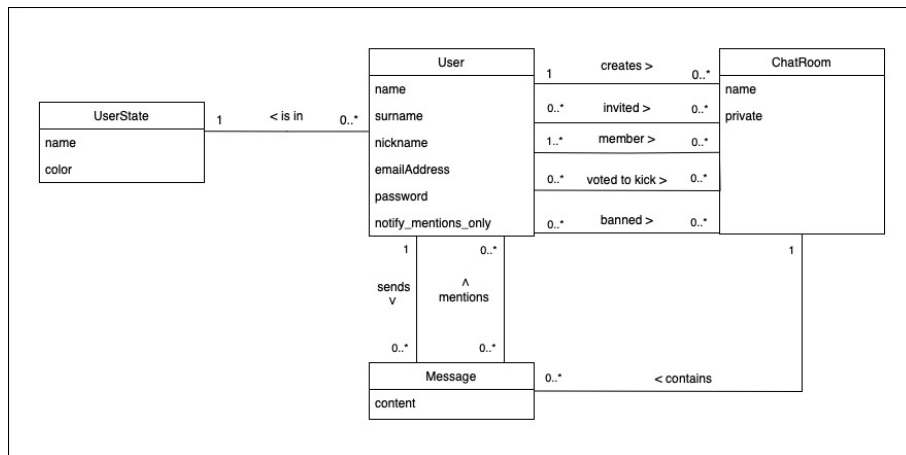
Dokumentácia v 2. (finálnej) fáze

Dokumentácia musí obsahovať minimálne tieto časti:

- zadanie
- diagram fyzického dátového modelu, v prípade zmien z 2. fázy, zdôvodniť zmenu
- diagram/diagramy architektúry aplikácie
- návrhové rozhodnutia (pridanie externej knižnice - zdôvodenie, ...)
- snímky obrazoviek (angl. screenshot, snapshot), aspoň 5 kľúčových obrazoviek (tie, ktoré by ste dali napr. do storu, aby ste zaujali a prezentovali sa)

Diagram fyzického dátového modelu

Pôvodný ER diagram:



Oproti nášmu ER diagramu sme vo fyzickom modeli toho veľa nezmenili. Odstránili sme iba zbytočný vzťah „mentioned“, ktorý sa ukázal ako nepotrebný, a pridali sme tabuľky súvisiace s autentifikáciou používateľov z frameworku AdonisJS. Naše mnohočetné vzťahy sa premietli do vzťahových tabuliek.

Fyzický model:

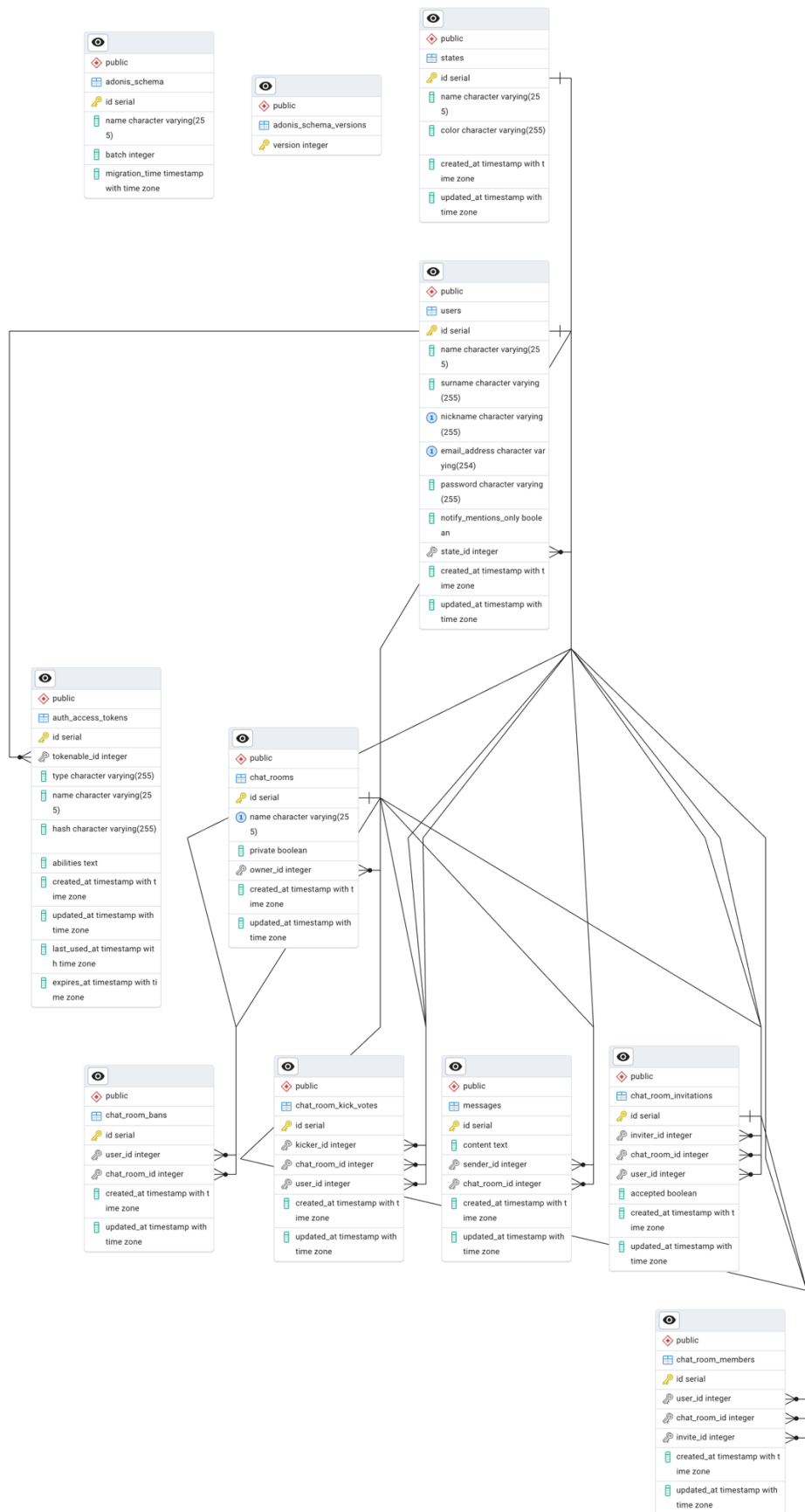
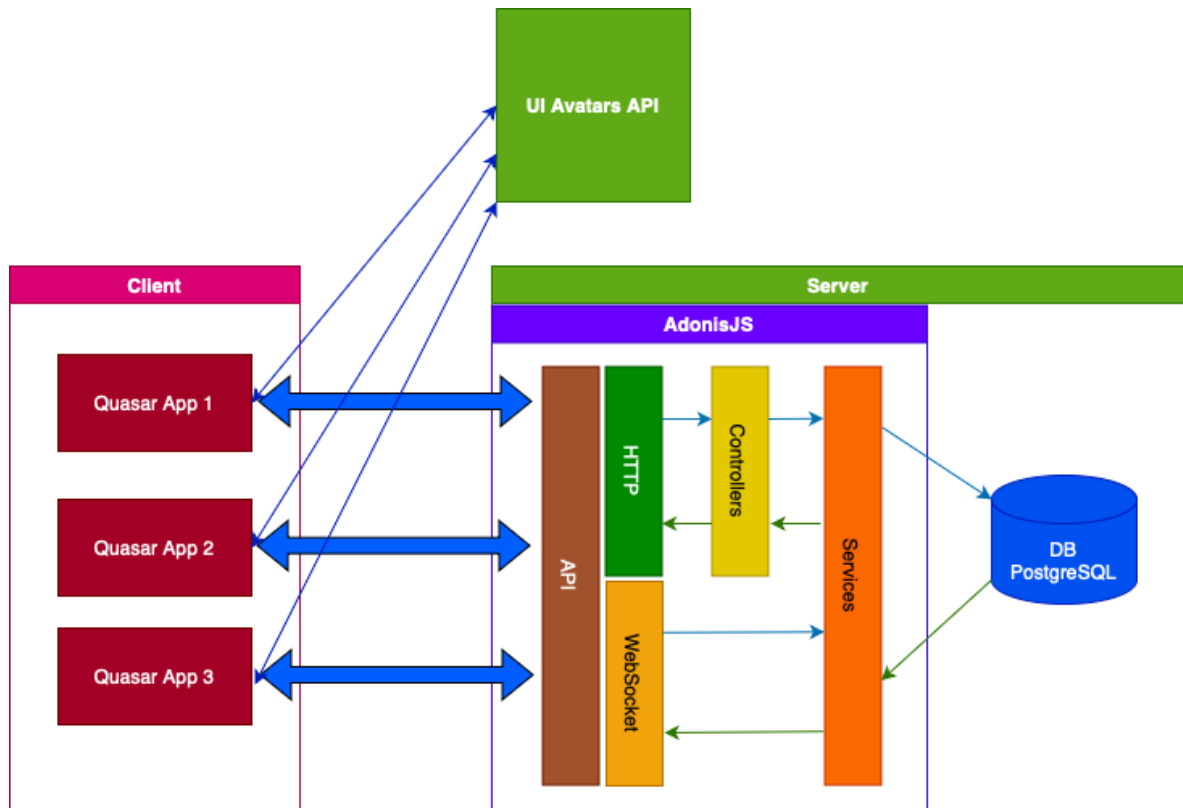


Diagram architektúry aplikácie

Naša architektúra pozostáva z viacerých kľúčových komponentov, ktoré spolupracujú na zabezpečení efektívnej funkcionality aplikácie. Na strane klienta sú Quasar aplikácie, ktoré zabezpečujú interaktívne užívateľské rozhranie. Dané aplikácie ďalej komunikujú so serverom cez HTTP a WebSokety. Serverová časť je postavená na frameworku AdonisJS, kde požiadavky spracovávajú kontroléry využívajúce služby na implementáciu logiky a prístup k databáze PostgreSQL. Architektúra je obohatená o integráciu s externým UI Avatars API, ktoré poskytuje avatarové služby priamo klientskym aplikáciám.



Návrhové rozhodnutia

socket.io

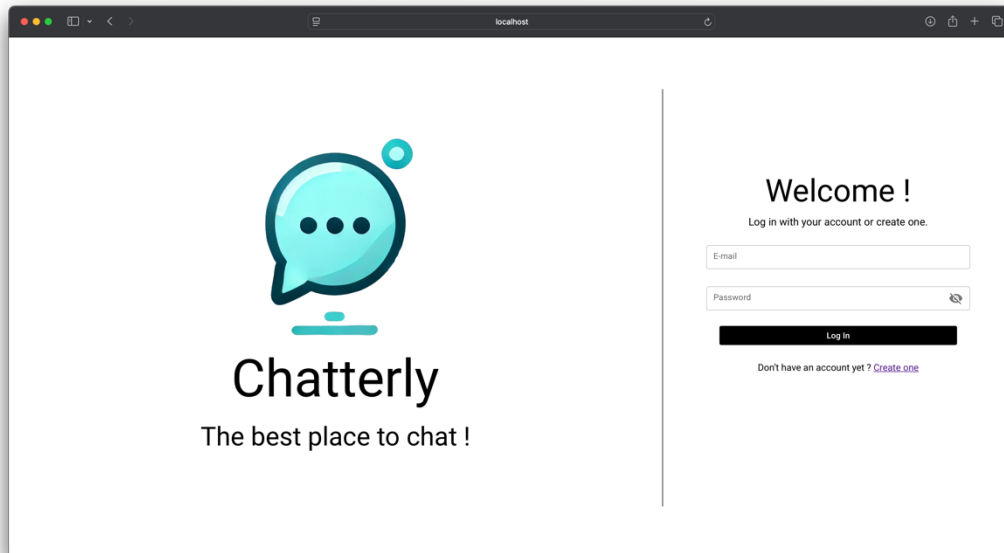
V našom projekte sme implementovali socket.io na komunikáciu v reálnom čase. Inšpirovali sme sa článkom How to configure Socket.IO for AdonisJS v6 (<https://medium.com/@armyabakouan/how-to-configure-socket-io-for-adonisjs-v6-4afbca1a465>), ktorý nám poskytol cenné informácie o konfigurácii. Táto integrácia nám umožnila efektívne zabezpečiť WebSocket spojenia a plynulú výmenu dát.

UI Avatars API

V našom projekte sme použili UI Avatars API (<https://ui-avatars.com>) na generovanie obrázkov pre avatary v chate. Toto API nám umožnilo jednoducho a rýchlo vytvárať personalizované avatary na základe mena používateľa.

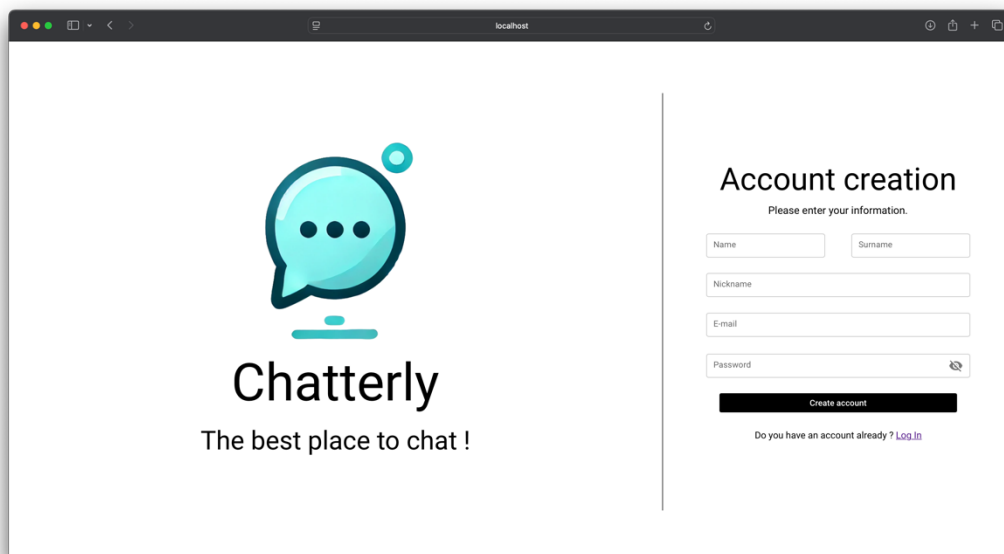
Snímky obrazoviek

Prihlásenie



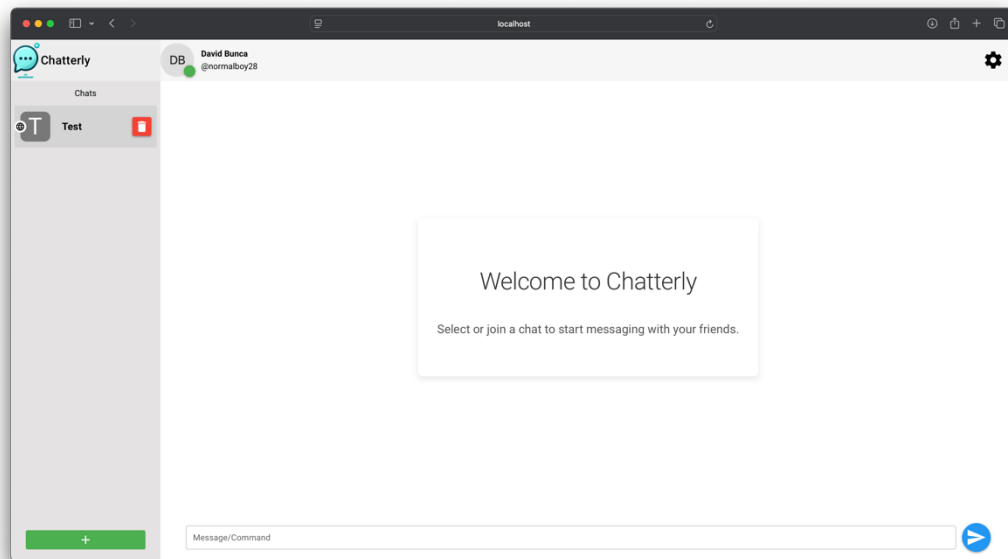
The screenshot shows a web browser window with the URL 'localhost'. The page features the Chatterly logo on the left, which consists of a teal speech bubble with three dots inside, and the text 'Chatterly' and 'The best place to chat !' below it. On the right, there is a 'Welcome !' section with the text 'Log in with your account or create one.' Below this are two input fields: 'E-mail' and 'Password'. The 'Password' field has a small eye icon to its right. A black 'Log in' button is positioned below the input fields. At the bottom of the login section, there is a link that says 'Don't have an account yet ? [Create one](#)'.

Registrácia

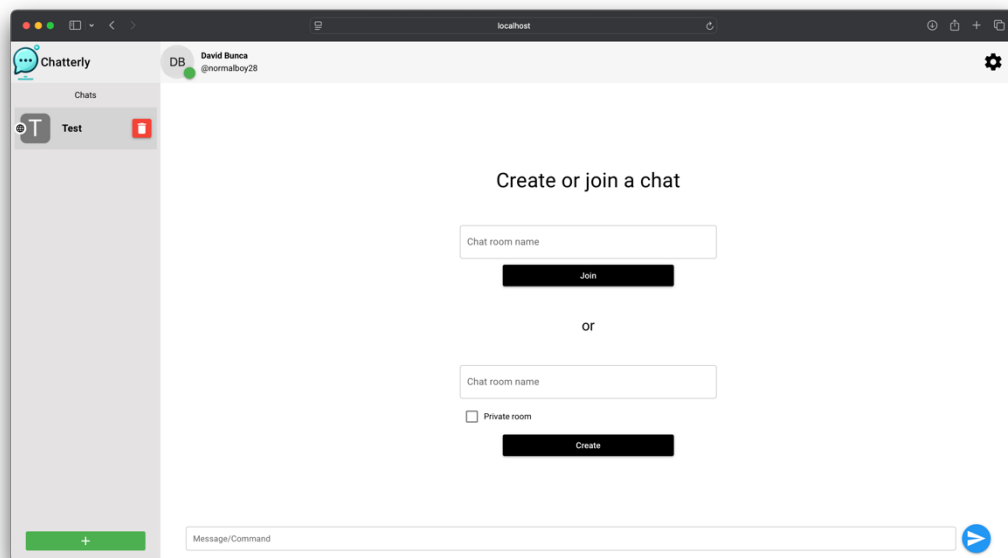


The screenshot shows a web browser window with the URL 'localhost'. The page features the Chatterly logo on the left, which consists of a teal speech bubble with three dots inside, and the text 'Chatterly' and 'The best place to chat !' below it. On the right, there is an 'Account creation' section with the text 'Please enter your information.' Below this are five input fields: 'Name', 'Surname', 'Nickname', 'E-mail', and 'Password'. The 'Password' field has a small eye icon to its right. A black 'Create account' button is positioned below the input fields. At the bottom of the account creation section, there is a link that says 'Do you have an account already ? [Log in](#)'.

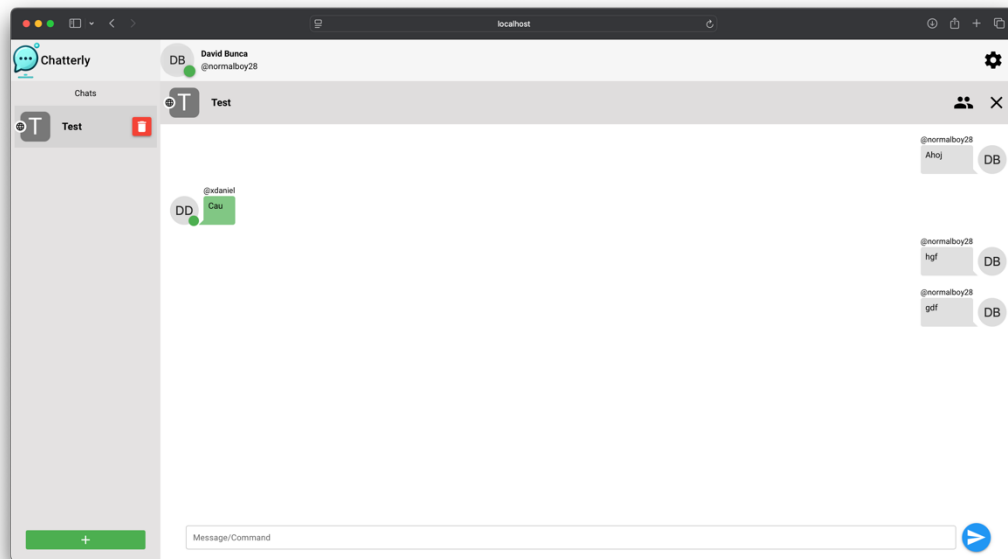
Domovská obrazovka



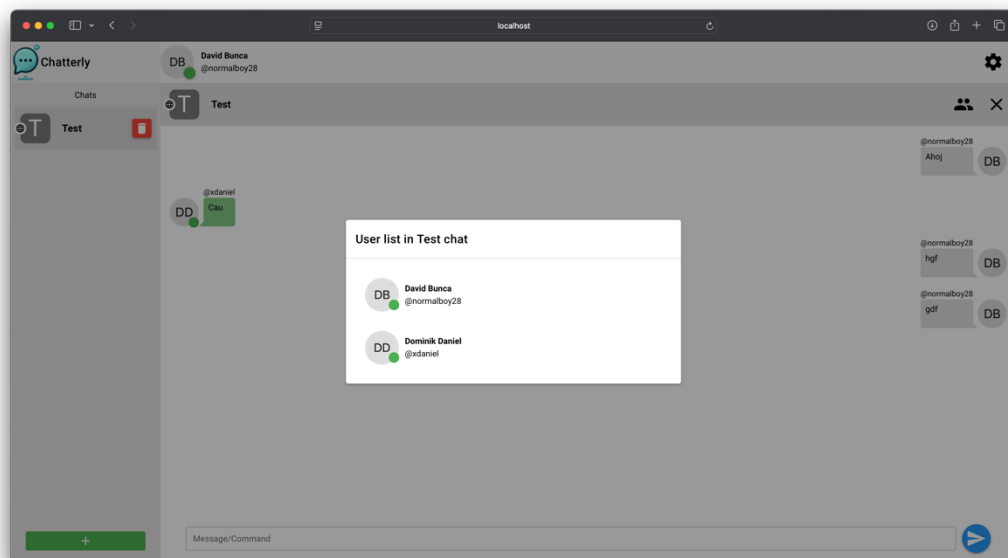
Vytváranie/joinovanie chatov



Chat



List uživatelův v chate



Zmena stavu používateľa

