Békéscsabai SZC Nemes Tihamér Technikum és Kollégium

Szakképesítés megnevezése: Szoftverfejlesztő és -tesztelő  
Azonosító száma: 5 0613 12 03

**VIZSGAREMEK**

Yapper

Készítették:

Baranya Gyula Osztály: 13.b Oktatási azonosító: 7257xxxxxx

Patai Olivér Osztály: 13.b Oktatási azonosító: 72xxxxxxxx

Szabó Márton Osztály: 13.b Oktatási azonosító: 72572159843

Békéscsaba, 2024/2025

Tartalomjegyzék

[Bevezető 3](#_Toc198042911)

[Probléma rövid ismertetése 3](#_Toc198042912)

[A digitális kommunikáció jelentősége 3](#_Toc198042913)

[A jelenlegi helyzet és a felhasználói igények 3](#_Toc198042914)

[A fejlesztendő alkalmazás célja 3](#_Toc198042915)

[A megoldandó problémák részletezése 3](#_Toc198042916)

[Téma indoklása 5](#_Toc198042917)

[A csevegőalkalmazások szerepe a modern társadalomban 5](#_Toc198042918)

[A témaválasztás szakmai és gyakorlati indokai 5](#_Toc198042919)

[A projekt célcsoportja 6](#_Toc198042920)

[A fejlesztés tanulási értéke 6](#_Toc198042921)

[A választott technológiák indoklása 6](#_Toc198042922)

[Összefoglalás 7](#_Toc198042923)

[Téma kifejtése 7](#_Toc198042924)

[Követelmények 7](#_Toc198042925)

[Rendszerterv 8](#_Toc198042926)

[Biztonsági kérdések 9](#_Toc198042927)

[Adatbázis leírása és modell-diagram 9](#_Toc198042928)

[Navigáció és ergonómia 11](#_Toc198042929)

[Főbb funkcionális tesztesetek 13](#_Toc198042930)

[Teszteredmények dokumentációja 13](#_Toc198042931)

[Összefoglalás 14](#_Toc198042932)

# Bevezető

## Probléma rövid ismertetése

### A digitális kommunikáció jelentősége

A 21. században a digitális kommunikáció mindennapjaink szerves részévé vált. Az emberek közötti kapcsolattartás, információcsere, közösségépítés és együttműködés jelentős része már nem személyesen, hanem különböző online platformokon keresztül történik. A gyors, megbízható és biztonságos üzenetküldés iránti igény folyamatosan növekszik, legyen szó magánszemélyekről, baráti társaságokról, családokról vagy akár munkahelyi csapatokról. Az online csevegőalkalmazások (chat appok) ezért napjaink egyik legnépszerűbb és legfontosabb szoftvermegoldásai közé tartoznak.

### A jelenlegi helyzet és a felhasználói igények

Az elmúlt években számos csevegőalkalmazás jelent meg a piacon, amelyek különböző funkciókat és szolgáltatásokat kínálnak. Ezek közül sok azonban vagy túl bonyolult, vagy túlzottan egyszerű, esetleg nem felel meg a modern biztonsági elvárásoknak, vagy nem nyújt megfelelő felhasználói élményt. A felhasználók részéről egyre nagyobb az igény egy olyan platform iránt, amely egyszerre könnyen kezelhető, gyors, megbízható, biztonságos, és lehetőséget ad mind egyéni, mind csoportos kommunikációra.

### A fejlesztendő alkalmazás célja

A Yapper nevű csevegőalkalmazás fejlesztésének célja, hogy egy korszerű, felhasználóbarát, biztonságos és könnyen bővíthető platformot hozzon létre, amely kielégíti a mai felhasználók igényeit. Az alkalmazás lehetőséget biztosít regisztrált felhasználók számára, hogy egymással valós időben üzeneteket váltsanak, képeket osszanak meg, valamint csoportos beszélgetéseket hozzanak létre. A fejlesztés során kiemelt figyelmet fordítottunk a felhasználói élményre, annak intuitív mivoltára.

### A megoldandó problémák részletezése

#### Valós idejű kommunikáció

A felhasználók elvárják, hogy az üzenetek azonnal megérkezzenek a címzetthez, függetlenül attól, hogy az adott pillanatban milyen eszközt használnak. Ezért a fejlesztés során olyan technológiákat kellett alkalmazni, amelyek lehetővé teszik a valós idejű adatátvitelt (pl. WebSocket, Socket.IO).

#### Felhasználói azonosítás és jogosultságkezelés

A biztonságos kommunikáció alapfeltétele, hogy csak hitelesített felhasználók férhessenek hozzá az alkalmazás funkcióihoz. A regisztráció, email-hitelesítés, bejelentkezés, jelszó-visszaállítás, valamint a jogosultságok kezelése mind-mind fontos részei a rendszernek.

#### Adatvédelem és biztonság

A felhasználók által megosztott adatok (üzenetek, képek, profiladatok) védelme kiemelt fontosságú. A fejlesztés során gondoskodni kellett az adatok titkosításáról, a jelszavak biztonságos tárolásáról, a jogosulatlan hozzáférés megakadályozásáról.

#### Felhasználói élmény és ergonómia

Az alkalmazás sikerességének egyik kulcsa a könnyű kezelhetőség és az átlátható, modern felhasználói felület. A navigáció, az üzenetküldés, a profilbeállítások, a csoportos beszélgetések kezelése mind-mind intuitív módon kell, hogy működjön.

#### Platformfüggetlenség és bővíthetőség

A felhasználók különböző eszközökről (asztali gép, laptop, tablet, okostelefon) szeretnék elérni az alkalmazást. Ezért a fejlesztés során törekedtünk a reszponzív, platformfüggetlen megvalósításra, valamint arra, hogy a rendszer könnyen bővíthető legyen új funkciókkal.

#### A fejlesztés során felmerülő kihívások

A valós idejű kommunikáció, a biztonságos adatkezelés, a felhasználói élmény optimalizálása, valamint a rendszer skálázhatósága mind-mind komoly szakmai kihívást jelentettek. A fejlesztőcsapatnak meg kellett találni az egyensúlyt a funkcionalitás, a biztonság, a teljesítmény és a felhasználói élmény között.

#### A dokumentáció célja

Jelen dokumentáció célja, hogy részletesen bemutassa a Yapper csevegőalkalmazás fejlesztésének hátterét, a megoldandó problémákat, a választott technológiákat, a rendszer felépítését, a biztonsági megfontolásokat, valamint a felhasználói és fejlesztői szempontokat. A dokumentáció segítséget nyújt mind a fejlesztőknek, mind a felhasználóknak az alkalmazás megértésében, használatában és továbbfejlesztésében.

#### Összefoglalás

A Yapper csevegőalkalmazás fejlesztése egy valós, napjainkban is aktuális problémára kínál megoldást: a gyors, biztonságos és felhasználóbarát online kommunikációra. A projekt során kiemelt figyelmet fordítottunk a modern technológiai és ergonómiai elvárásokra, a biztonságra, valamint a könnyű használhatóságra és bővíthetőségre. A következő fejezetekben részletesen bemutatjuk a választott témát, a fejlesztési folyamatot, a rendszer felépítését, a biztonsági kérdéseket, az adatbázis felépítését, valamint a tesztelési és felhasználói szempontokat.

## Téma indoklása

### A csevegőalkalmazások szerepe a modern társadalomban

A digitális kommunikáció térnyerése az elmúlt évtizedekben alapvetően átalakította az emberek közötti kapcsolattartás módját. A csevegőalkalmazások, vagyis az online chat platformok, mára a mindennapi élet elengedhetetlen részévé váltak, legyen szó magánéleti, oktatási vagy munkahelyi környezetről. Ezek az alkalmazások lehetővé teszik a gyors, valós idejű információcserét, a csoportos együttműködést, valamint a közösségek építését, fenntartását. A világjárvány időszaka különösen rámutatott arra, mennyire fontosak a megbízható, könnyen használható és biztonságos online kommunikációs eszközök.

### A témaválasztás szakmai és gyakorlati indokai

A Yapper csevegőalkalmazás fejlesztésének ötlete több, egymást erősítő szakmai és gyakorlati megfontolásból született:

**Szakmai kihívás:** Egy valós idejű, többfelhasználós, biztonságos kommunikációs rendszer fejlesztése összetett feladat, amely számos modern technológia (pl. WebSocket, JWT, reszponzív frontend, adatbázis-kezelés, autentikáció, jogosultságkezelés) alkalmazását igényli. Ez kiváló lehetőséget ad a fejlesztőknek arra, hogy elmélyítsék tudásukat a webfejlesztés különböző területein.

**Gyakorlati relevancia:** A csevegőalkalmazások iránti igény folyamatosan nő, mind a magánszemélyek, mind a vállalkozások körében. Egy jól működő, könnyen bővíthető chat platform fejlesztése nemcsak tanulási célból hasznos, hanem a későbbiekben akár valós projektek alapjául is szolgálhat.

**Biztonsági szempontok:** A felhasználói adatok védelme, a hitelesítés, az üzenetek titkosítása, valamint a jogosulatlan hozzáférés megakadályozása mind-mind olyan kihívások, amelyek megoldása elengedhetetlen egy modern alkalmazás esetében. A projekt során lehetőség nyílik ezen biztonsági kérdések gyakorlati megvalósítására.

**Felhasználói élmény:** A mai felhasználók elvárják, hogy egy alkalmazás ne csak funkcionális, hanem esztétikus, gyors és könnyen kezelhető is legyen. A Yapper fejlesztése során kiemelt figyelmet fordítottunk a reszponzív dizájnra, az intuitív navigációra és a modern ergonómiai elvek alkalmazására.

### A projekt célcsoportja

A Yapper alkalmazás elsődleges célcsoportja a fiatal felnőttek, diákok, valamint a kis- és középvállalkozások, akik számára fontos a gyors, megbízható és biztonságos online kommunikáció. Emellett a fejlesztés során figyelembe vettük azokat a felhasználókat is, akik kevésbé jártasak a digitális technológiákban, ezért az alkalmazás kezelőfelülete egyszerű, átlátható és könnyen tanulható.

### A fejlesztés tanulási értéke

A projekt során a fejlesztőcsapat tagjai számos, a munkaerőpiacon is keresett technológiát és módszertant sajátíthattak el, többek között: - REST API tervezés és implementáció - Valós idejű kommunikáció (Socket.IO) - Felhasználókezelés, autentikáció, jogosultságkezelés (JWT, bcrypt) - Adatbázis-tervezés és -kezelés (MongoDB) - Frontend fejlesztés (React, reszponzív dizájn) - Automatizált tesztelés (Selenium) - Biztonsági alapelvek alkalmazása (input validáció, XSS/CSRF védelem, jelszókezelés)

### A választott technológiák indoklása

A projekt során alkalmazott technológiák kiválasztásánál a következő szempontokat vettük figyelembe: - **Skálázhatóság:** A Node.js és a MongoDB lehetővé teszi a rendszer későbbi bővítését, akár nagyobb felhasználószám esetén is. - **Valós idejűség:** A Socket.IO integrációja biztosítja az azonnali üzenetküldést és fogadást. - **Biztonság:** A JWT-alapú autentikáció, a bcrypt-tel történő jelszóhash-elés, valamint a HTTPS támogatás mind hozzájárulnak a felhasználói adatok védelméhez. - **Fejlesztői élmény:** A React és a modern frontend eszközök gyors fejlesztést, könnyű karbantartást és bővíthetőséget tesznek lehetővé. - **Automatizált tesztelés:** A Selenium segítségével a legfontosabb funkciók automatikusan tesztelhetők, így a hibák gyorsan kiszűrhetők.

### Összefoglalás

A Yapper csevegőalkalmazás témájának választása nemcsak a digitális kommunikáció aktuális jelentőségéből fakad, hanem abból is, hogy a projekt komplexitása révén lehetőséget ad a modern webfejlesztési technológiák, módszertanok és biztonsági elvek gyakorlati alkalmazására. A fejlesztés során szerzett tapasztalatok mind a szakmai fejlődést, mind a későbbi munkaerőpiaci érvényesülést támogatják, miközben egy valóban hasznos, a mindennapokban is alkalmazható szoftver született.

## Téma kifejtése

### Követelmények

A Yapper szoftveralkalmazás fejlesztése során az alábbi elvárásoknak kellett megfelelni:

* **Életszerű, valódi problémára nyújt megoldást:**  
  A Yapper egy modern, valós idejű csevegőalkalmazás, amely a mindennapi online kommunikáció igényeire ad választ. Segítségével a felhasználók gyorsan, biztonságosan és kényelmesen tarthatják a kapcsolatot egymással, legyen szó magánéleti, tanulmányi vagy munkahelyi kommunikációról.
* **Adattárolási és -kezelési funkciók:**  
  Az alkalmazás MongoDB adatbázist használ, amelyben a felhasználók, üzenetek, kapcsolatok és egyéb metaadatok strukturáltan, biztonságosan kerülnek tárolásra. Az adatok kezelése RESTful API-n keresztül történik, amely lehetővé teszi a CRUD (Create, Read, Update, Delete) műveleteket.
* **RESTful architektúra:**  
  A backend Node.js és Express alapokon készült, a REST architektúra elveit követve. Az API végpontok logikusan szervezettek, jól dokumentáltak, és minden fő funkció (regisztráció, bejelentkezés, üzenetküldés, profilkezelés stb.) elérhető HTTP kéréseken keresztül.  
  A frontend React-alapú SPA, amely a backenddel HTTP-n és Socket.IO-n keresztül kommunikál.
* **Platformfüggetlen, reszponzív kliens:**  
  A felhasználói felület teljes mértékben reszponzív, így asztali számítógépen, laptopon, tableten és mobiltelefonon is kényelmesen használható. A dizájn mobilbarát, a navigáció és az ergonómia minden eszközön optimális felhasználói élményt biztosít.
* **Tiszta kód elvei:**  
  A forráskód megfelel a tiszta kód (Clean Code) alapelveinek:
  + Átlátható, jól strukturált, olvasható és könnyen karbantartható.
  + A komponensek és függvények egyértelmű, beszédes neveket kaptak.
  + A logika moduláris, a felelősségi körök jól elkülönülnek.
  + A hibakezelés és input validáció mindenhol megvalósul.
* **Dokumentáció:**  
  Jelen dokumentáció részletesen bemutatja a szoftver célját, a komponensek technikai leírását, a működés műszaki feltételeit, valamint a használat rövid bemutatását. A dokumentáció a fejlesztői és felhasználói szempontokat egyaránt lefedi.
* **Önálló szellemi termék:**  
  A Yapper alkalmazás és annak teljes forráskódja, valamint a dokumentáció a fejlesztőcsapat saját, önálló szellemi terméke, külső forrásból származó kódot csak indokolt esetben, megfelelő hivatkozással tartalmaz.

### Rendszerterv

#### Architektúra áttekintése

A Yapper egy klasszikus kliens-szerver architektúrát követ, amely három fő komponensből áll:

* **Frontend (React SPA):**  
  A felhasználói interakciók kezelése, az adatok vizuális megjelenítése, valamint a REST API és a valós idejű kommunikáció (Socket.IO) kezelése.
* **Backend (Node.js + Express):**  
  Az üzleti logika, az adatkezelés, a hitelesítés, a jogosultságkezelés, valamint a valós idejű üzenetküldés megvalósítása.
* **Adatbázis (MongoDB):**  
  A felhasználók, üzenetek, kapcsolatok, csoportok és egyéb adatok tárolása.

##### Fő komponensek és azok feladatai

* **Felhasználókezelés:**  
  Regisztráció, email-verifikáció, bejelentkezés, jelszó-visszaállítás, profilmódosítás, jogosultságkezelés.
* **Chat modul:**  
  Privát és (ha van) csoportos üzenetküldés, üzenetlista, chat előzmények, olvasottsági státuszok.
* **Értesítés modul:**  
  Böngésző értesítések kezelése, push notification támogatás.
* **Teszt modul:**  
  Automatizált végpont- és UI tesztek (Selenium), amelyek a fő funkciók helyes működését ellenőrzik.

### Biztonsági kérdések

#### Jelszókezelés

* A jelszavakat bcrypt-tel hash-eljük, soha nem tároljuk őket tiszta szövegként.
* Jelszó-visszaállítás során időkorlátos, egyszer használatos tokeneket alkalmazunk.

#### Autentikáció és jogosultságkezelés

* A felhasználók JWT tokennel azonosítják magukat minden védett végponton.
* A tokenek érvényességi ideje korlátozott, a backend minden kérésnél ellenőrzi a jogosultságot.

#### Input validáció és védelem

* Minden bemeneti adatot validálunk (pl. email formátum, jelszó hossza).
* Védekezünk XSS, CSRF és egyéb tipikus webes támadások ellen.

### Adatbázis leírása és modell-diagram

A Yapper alkalmazás MongoDB-re épülő adatbázisa több, egymással kapcsolatban álló dokumentumot (modellt) tartalmaz. Az alábbiakban bemutatjuk a főbb adatmodelleket és azok mezőit.

#### Felhasználó (User) modell

* **\_id**: Egyedi azonosító (ObjectId)
* **email:** Email cím (string, kötelező, egyedi)
* **fullName:** Teljes név (string, kötelező)
* **password:** Jelszó hash (string, kötelező, minimum 4 karakter)
* **profilePicture:** Profilkép URL vagy elérési út (string, alapértelmezett: üres)
* **resetPasswordToken:** Jelszó-visszaállítás token (string, opcionális)
* **resetPasswordExpires:** Token lejárati ideje (Date, opcionális)
* **createdAt, updatedAt:** Létrehozás és módosítás dátuma (Date, automatikus)

#### Üzenet (Message) modell

* **\_id:** Egyedi azonosító (ObjectId)
* **senderId:** Feladó felhasználó azonosítója (User.\_id, kötelező)
* **receiverId:** Címzett felhasználó azonosítója (User.\_id, opcionális – privát üzenetnél)
* **groupId:** Csoport azonosítója (Group.\_id, opcionális – csoportos üzenetnél)
* **text:** Üzenet szövege (string, opcionális)
* **image:** Csatolt kép URL vagy elérési út (string, opcionális)
* **createdAt, updatedAt:** Küldés és módosítás időpontja (Date, automatikus)

#### Csoport (Group) modell

* **\_id:** Egyedi azonosító (ObjectId)
* **name:** Csoport neve (string, kötelező)
* **avatar:** Csoport profilképe (string, opcionális)
* **members:** Tagok listája (User.\_id tömb, kötelező)
* **admins:** Adminisztrátorok listája (User.\_id tömb, opcionális)
* **createdBy:** Létrehozó felhasználó (User.\_id, kötelező)
* **createdAt:** Létrehozás dátuma (Date, automatikus)

#### Barátság (Friendship) modell

* **\_id:** Egyedi azonosító (ObjectId)
* **requester:** Barátságot kezdeményező felhasználó (User.\_id, kötelező)
* **recipient:** Barátságot fogadó felhasználó (User.\_id, kötelező)
* **status:** Barátság státusza (string: “pending”, “accepted”, “rejected”; alapértelmezett: “pending”)
* **createdAt, updatedAt:** Létrehozás és módosítás dátuma (Date, automatikus)

#### Push értesítési feliratkozás (PushSubscription) modell

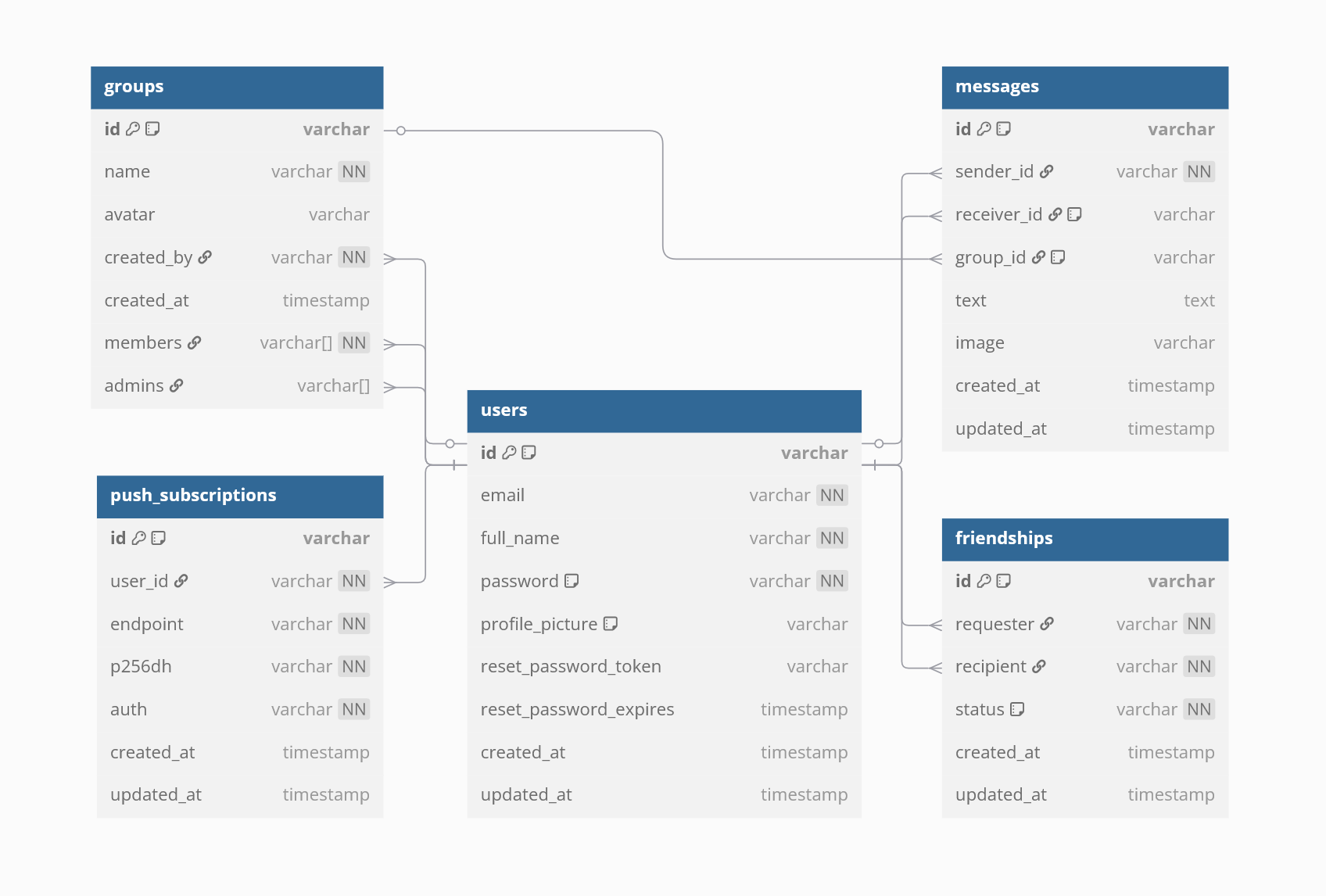
* **\_id:** Egyedi azonosító (ObjectId)
* **userId:** Felhasználó azonosítója (User.\_id, kötelező)
* **subscription:**
  + **endpoint:** Feliratkozás végpontja (string, kötelező)
  + **keys:**
    - **p256dh:** Kulcs (string, kötelező)
    - **auth:** Kulcs (string, kötelező)
* **createdAt, updatedAt:** Létrehozás és módosítás dátuma (Date, automatikus)

#### Adatbázismodell-diagram (szövegesen)

* Egy **User** több **Group**-ban lehet tag és admin.
* Egy **User** több **Friendship**-ben szerepelhet requester vagy recipient szerepben.
* Egy **User** több **Message**-et küldhet (senderId), és kaphat (receiverId).
* Egy **Group**-nak több **User** tagja és adminja lehet.
* Egy **Message** vagy egy **User**-höz (privát), vagy egy **Group**-hoz (csoportos) tartozik.
* Egy **User**-nek lehet több **PushSubscription** rekordja (pl. több eszközről).

#### Adatbázismodell-diagram (vizuálisan)

Az alábbi ábra szemlélteti az adatbázis főbb tábláit (kollekcióit) és azok kapcsolatát:



### Navigáció és ergonómia

A Yapper alkalmazás felhasználói felülete a modern ergonómiai elveknek megfelelően készült, hogy minden eszközön könnyen kezelhető és átlátható legyen.

* **Fő navigációs elemek:**
  + **Főoldal:** Bejelentkezés vagy regisztráció után a felhasználó az üzenetlistát látja.
  + **Csevegések:** Bal oldali sávban a privát és csoportos beszélgetések listája.
  + **Csoportok:** Külön szekcióban jelennek meg a felhasználó által létrehozott vagy adminisztrált csoportok.
  + **Barátok:** Barátlista, barátságkezelés (elfogadás, visszautasítás, törlés).
  + **Profil:** Profilkép, név, email, jelszó módosítása, push értesítések kezelése.
  + **Kijelentkezés:** Egyértelműen elérhető gomb minden nézetből.
* **Ergonómiai szempontok:**
  + **Reszponzív dizájn:** Minden funkció elérhető mobilon, tableten és asztali gépen is.
  + **Átlátható elrendezés:** A legfontosabb funkciók (üzenetküldés, csoportkezelés) gyorsan elérhetők.
  + **Visszajelzések:** Minden művelet után (pl. üzenet elküldése, barátság elfogadása) azonnali vizuális visszajelzés.
  + **Hozzáférhetőség:** Színek, kontrasztok, gombméretek megfelelnek a hozzáférhetőségi ajánlásoknak.
* **Hozzáférhetőség:** Színek, kontrasztok, gombméretek megfelelnek a hozzáférhetőségi ajánlásoknak.
* **Progresszív webalkalmazás (PWA) előnyei:**
  + **Telepíthetőség:** A Yapper PWA-ként működik, így a felhasználók egyszerűen hozzáadhatják az alkalmazást a mobiljuk vagy számítógépük kezdőképernyőjéhez, mintha natív alkalmazás lenne.
  + **Natív élmény:** A PWA offline módban is működik, gyorsan indul, és teljes képernyős, zavaró böngészőelemek nélküli élményt nyújt.
  + **Push értesítések:** A böngésző push notification támogatásának köszönhetően a felhasználók valós időben értesülhetnek az új üzenetekről vagy eseményekről, akkor is, ha az alkalmazás nincs megnyitva.
  + **Automatikus frissítés:** A PWA automatikusan frissül a háttérben, így a felhasználók mindig a legújabb verziót használják, külön telepítés vagy frissítés nélkül.
  + **Platformfüggetlenség:** Ugyanaz a felhasználói élmény biztosított minden modern böngészőn, legyen szó Androidról, iOS-ről vagy asztali rendszerről.

Ezek az ergonómiai és technológiai előnyök hozzájárulnak ahhoz, hogy a Yapper használata kényelmes, gyors és megbízható legyen minden eszközön, a natív alkalmazásokhoz hasonló élményt nyújtva.

### Főbb funkcionális tesztesetek

Az alkalmazás fejlesztése során automatizált és manuális teszteket is végeztünk. Az alábbiakban néhány főbb funkcionális tesztesetet sorolunk fel:

* **Regisztráció és email-verifikáció:**
  + Helyes adatokkal sikeres regisztráció és verifikáció.
  + Hibás email vagy jelszó esetén hibaüzenet jelenik meg.
* **Bejelentkezés:**
  + Helyes adatokkal sikeres bejelentkezés.
  + Hibás adatokkal hibaüzenet jelenik meg.
* **Üzenetküldés:**
  + Privát és csoportos üzenetek küldése, fogadása.
  + Kép csatolása üzenethez.
* **Barátságkezelés:**
  + Barátküldés, elfogadás, visszautasítás, törlés.
* **Csoportkezelés:**
  + Új csoport létrehozása, tagok hozzáadása/eltávolítása, admin jogok kezelése.
* **Push értesítések:**
  + Böngésző értesítések engedélyezése, tesztelése több eszközön.
* **Jelszó-visszaállítás:**
  + Elfelejtett jelszó funkció működése, tokenes visszaállítás.

### Teszteredmények dokumentációja

Az automata tesztek a frontend/tests mappában találhatók, futtatásukról és eredményeikről a README.md is tartalmaz információt.

* **Automata tesztek:**  
  Selenium-alapú végpont- és UI tesztek, amelyek lefedik a fő funkciókat (regisztráció, bejelentkezés, üzenetküldés, barátságkezelés stb.).
* **Teszteredmények:**  
  A tesztek sikeres futtatása igazolja, hogy az alkalmazás fő funkciói helyesen működnek fejlesztői környezetben. A tesztek futtatása során minden fő funkciót ellenőriztünk, hibátlan működés esetén a tesztek zöld státuszt adnak vissza.

### Összefoglalás

A fenti fejezetek részletesen bemutatták a Yapper alkalmazás fejlesztői oldalát: a követelményeket, a rendszertervet, a biztonsági szempontokat, az adatbázis felépítését, a navigációt, ergonómiát, valamint a főbb teszteseteket és azok eredményeit. A következő fejezetben a felhasználói dokumentáció olvasható.