Všeobecný popis aplikácie

1.1 Účel a vlastnosti:

Aplikácia predstavuje zašifrované online úložisko súborov, fungujúce formou Single Page Application (SPA), kde:

- Šifrovanie a dešifrovanie prebiehajú výlučne na strane klienta (End-to-End).

- Privátne kľúče (alebo seed/tajné dáta) nikdy neopúšťajú zariadenie používateľa.

- Na serveri sú uložené len zašifrované súbory a verejné kľúče používateľov.

- Verejné kľúče sú dostupné pre jednoduchú výmenu súborov medzi používateľmi (klient automaticky získava verejný kľúč príjemcu zo servera).

Hlavný cieľ aplikácie je zabezpečiť dôvernosť údajov (server nedokáže dešifrovať súbory), a zároveň zjednodušiť výmenu kľúčov medzi používateľmi (server funguje ako "Key Directory").

1.2 Cieľová skupina:

- Študenti, učitelia, výskumní pracovníci, ktorí potrebujú bezpečne zdieľať dokumenty.

- Firmy a freelanceri, ktorí si cenia dôvernosť údajov (obchodné ponuky, zmluvy).

- Široká verejnosť, ktorej záleží na bezpečnom ukladaní a zdieľaní súborov.

1.3 Použitie na rôznych zariadeniach:

- Desktop (Windows/Linux/macOS): prístup cez webový prehliadač (Chrome, Firefox, Edge).

- Mobilné zariadenia (iOS, Android): responzívny dizajn aplikácie.

2. Používateľské požiadavky

2.1 Používateľské role:

2.1.1) Hosť (neprihlásený):

- Môže prezerať súbory, ku ktorým bol vytvorený verejný odkaz.

- Nemôže nahrávať súbory ani dešifrovať súkromné súbory.

2.1.2) Registrovaný používateľ:

- Pri registrácii generuje pár kľúčov (verejný + privátny) lokálne.

- Server ukladá iba verejný kľúč.

- Môže nahrávať súbory (lokálne šifrovanie pomocou hybridného prístupu).

- Môže zdieľať súbory s jednotlivými používateľmi alebo definovanými skupinami.

- Dešifruje len súbory, ku ktorým má privátny kľúč.

- Spravuje vlastné skupiny používateľov (vytvorenie, úprava, vymazanie).

2.1.3) Administrátor:

- Spravuje používateľské účty (aktivácia účtov, blokovanie používateľov, nastavovanie kvóty).

- Vidí metadáta súborov (veľkosť, hash, vlastník) a verejné kľúče používateľov.

- Nemá prístup k obsahu súborov (nemá privátne kľúče).

2.2 Scenáre použitia (Use Cases):

UC1: Registrácia a generovanie kľúčov:

- Aktér: Nový používateľ

- Scenár:

1. Vyplní registračný formulár (meno, heslo).
2. Aplikácia lokálne vygeneruje pár kľúčov.
3. Verejný kľúč odošle na server, privátny zostáva lokálne.
4. Server vytvorí nový účet s verejným kľúčom.

UC2: Nahranie zašifrovaného súboru:

- Aktér: Registrovaný používateľ

- Scenár:

1. Vyberie súbor.
2. Lokálne zašifruje súbor (symetrický kľúč + verejný kľúč).
3. Odošle súbor na server.
4. Server uloží súbor a vráti ID alebo odkaz.

UC3: Zdieľanie súboru:

- Aktér: Vlastník súboru

- Scenár:

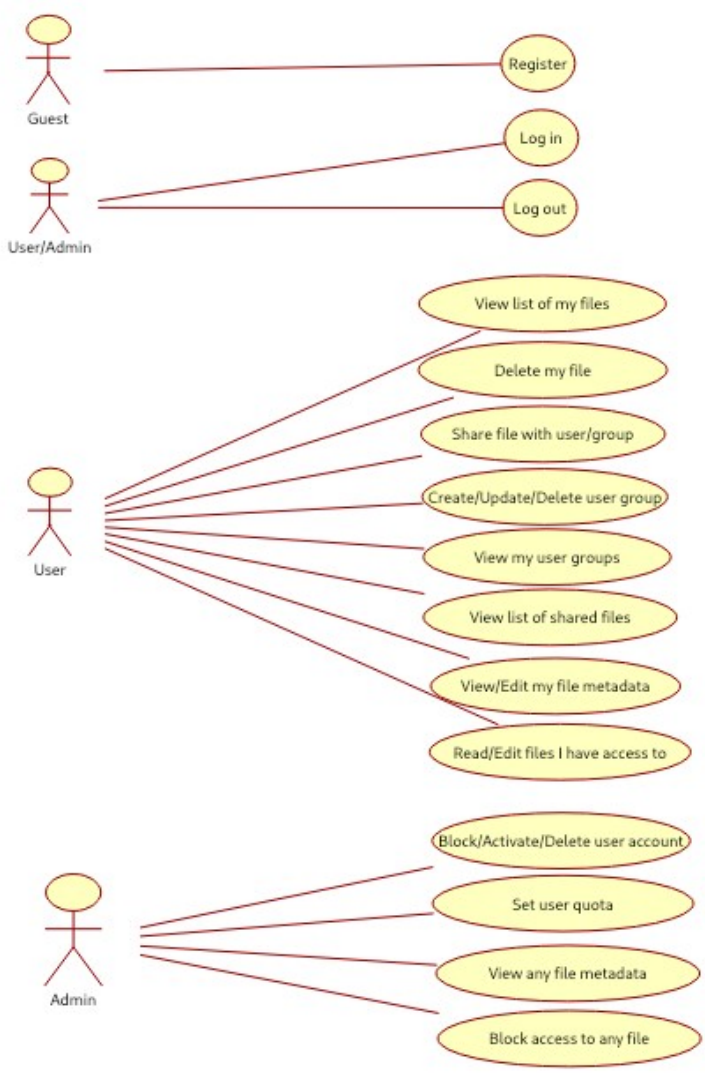
1. Vyberie súbor a príjemcov (jednotlivcov alebo skupinu).
2. Získa verejné kľúče príjemcov zo servera.
3. Lokálne zašifruje symetrický kľúč pod verejné kľúče príjemcov.
4. Príjemcovia dešifrujú súbor svojimi privátnymi kľúčmi.

UC4: Správa skupín používateľov:

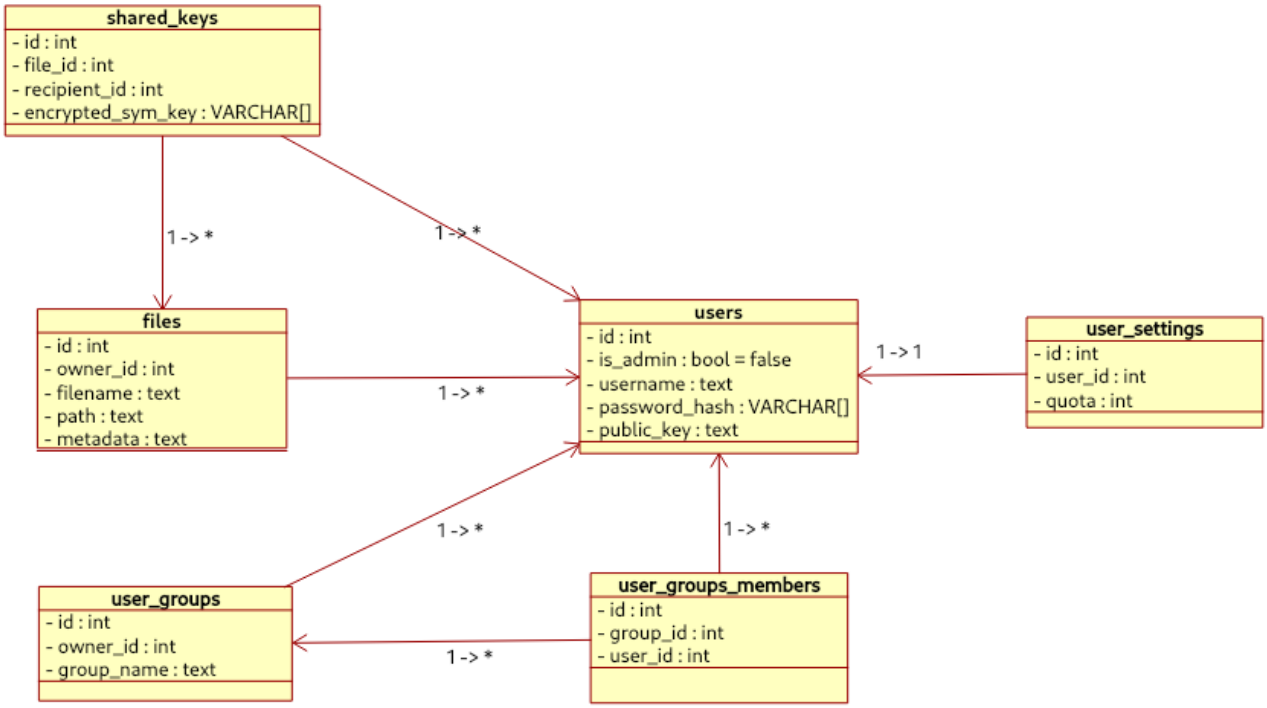
- Aktér: Registrovaný používateľ

- Scenár:

1. Používateľ vytvára, upravuje, zobrazuje alebo vymazáva svoje skupiny používateľov.



3. Dátový model:



users (id, is_admin, username, password_hash, public_key)

files (id, owner_id -> users.id, filename, path, metadata)

shared_keys (id, file_id -> files.id, recipient_id -> users.id, encrypted_sym_key)

user_groups (id, owner_id -> users.id, group_name)

user_groups_members (id, group_id -> user_groups.id, user_id -> users.id)

user_settings (id, user_id -> users.id, quota)

4. Architektúra aplikácie:

Aplikácia funguje ako Single Page Application (SPA):

4.1 Front-end (klientská časť):

- Implementovaná v JavaScript + React.

- Šifrovanie a dešifrovanie prebieha lokálne v prehliadači (Web Crypto API).

- Používateľ komunikuje s jednou HTML stránkou (SPA), ktorá dynamicky mení obsah bez opakovaného načítania.

4.2 Back-end (serverová časť):

- Vytvorený v Djangu (Python 3.12).

- Vystavuje REST API pre registráciu, prihlasovanie, správu súborov, zdieľanie, atď.

- Ukladá zašifrované súbory a verejné kľúče v PostgreSQL (ver. 16).

- Nemá žiadne "server-side" generované HTML - všetko vykresľovanie je len na front-ende.

4.3 Docker infraštruktúra:

- Tri základné kontajnery:

    - Django

    - PostgreSQL

    - Nginx

5. Technologické požiadavky:

5.1 Klientská časť:

- HTML5, CSS3, JavaScript, React, Bootstrap.

- Responzívny dizajn.

5.2 Serverová časť:

- Python 3.12, Django 5.x, PostgreSQL 16.

- HTTPS, hashovanie hesiel, server neukladá privátne kľúče.

5.3 Databáza:

- PostgreSQL, tabuľky: users, files, shared_keys, user_groups, user_groups_members.

5.4 Hosting:

- VPS (netcup.com), HTTPS certifikát.

5.5 Podporované prehliadače:

- Google Chrome, Mozilla Firefox

5.6 Klient-server rozhranie:

- REST API (JSON), HTTPS.

6. Časový plán:

4 týždeň - Návrh DB, ER diagram, výber šifrovacej knižnice, nastavenie docker prostredia.
5 týždeň - Registrácia, generovanie kľúčov
6 týždeň - Nahrávanie súborov (šifrovanie klienta)
7 týždeň - Zdieľanie súborov (shared_keys)
8 týždeň - UI/UX, testovanie
9 týždeň - Optimalizácia, bezpečnosť, testovanie, príprava betaverzie
10 týždeň - Dokumentácia, finalizácia betaverzie
11 týždeň - Zapracovanie spätnej väzby

7. Záver:

- Server: uchováva verejné kľúče a zašifrované súbory bez možnosti dešifrovania.

- Klient: realizuje end-to-end šifrovanie a správu kľúčov. Administrátor: spravuje účty a bezpečnosť bez prístupu k obsahu súborov.