ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA

Khoa Khoa học và Kỹ thuật Máy tính

ĐỒ ÁN TỔNG HỢP
CÔNG NGHỆ PHẦN MỀM

HỆ THỐNG FILE SHARING

Nhóm: Web
GVHD: Thầy Lê Đình Thuận

Thành phố Hồ Chí Minh, tháng 11 năm 2025

CHƯƠNG 1

TỔNG QUAN DỰ ÁN

1.1 Mô tả dự án

Hệ thống **File Sharing and Management System** là một nền tảng web hỗ trợ người dùng tải lên, quản lý và chia sẻ tệp tin một cách an toàn, tiện lợi và linh hoạt. Hệ thống được xây dựng theo kiến trúc dịch vụ REST với backend phát triển bằng **Golang**, giao tiếp thông qua API chuẩn hoá, và frontend dưới dạng ứng dụng web (SPA) do nhóm frontend triển khai.

Hệ thống cho phép người dùng tải tệp dưới hai hình thức: *ẩn danh* hoặc *đăng nhập*. Mỗi tệp được gán một *share token* duy nhất giúp người nhận có thể truy cập nhanh, đồng thời hỗ trợ các cơ chế bảo vệ như mật khẩu, TOTP, whitelist theo email và giới hạn thời gian hiệu lực.

Thông tin tệp (metadata), lịch sử truy cập và các chính sách hệ thống được quản lý tập trung trong cơ sở dữ liệu. Bên cạnh đó, hệ thống hỗ trợ cơ chế dọn dẹp tệp tự động (cleanup) thông qua tác vụ cron với mã xác thực `X-Cron-Secret`. Tính năng thống kê lượt tải và xem lịch sử download hỗ trợ quá trình theo dõi và kiểm toán.

Mục tiêu tổng thể của dự án là xây dựng một nền tảng lưu trữ tệp hiệu quả, mở rộng tốt, đảm bảo bảo mật và dễ tích hợp với các dịch vụ phụ trợ như email service hoặc cloud storage trong tương lai.

1.2 Yêu cầu chức năng và phi chức năng

1.2.1 Yêu cầu chức năng

FR1 – Quản lý người dùng

- FR1.1: Hệ thống cho phép người dùng đăng ký và đăng nhập thông qua email và mật khẩu; xác thực bằng JWT.
- FR1.2: Người dùng có thể kích hoạt và xác minh TOTP để tăng cường bảo mật.
- FR1.3: Người dùng ẩn danh được phép tải lên tệp nhưng không thể chỉnh sửa hoặc quản lý tệp sau đó.
- FR1.4: Người dùng đăng nhập có thể truy cập danh sách tệp cá nhân, chỉnh sửa metadata, thay đổi bảo mật, hoặc xóa tệp.

- FR1.5: Quản trị viên có thể truy cập các API quản lý hệ thống, bao gồm cập nhật chính sách và thực thi tác vụ cleanup.

FR2 – Upload và chia sẻ tệp

- FR2.1: Hệ thống hỗ trợ tải tệp bằng phương thức multipart thông qua endpoint `/files/upload`.
- FR2.2: Sau khi tải thành công, hệ thống sinh ra *share token* và liên kết duy nhất cho phép tải xuống.
- FR2.3: Người dùng có thể thiết lập các thuộc tính:
  - Khoảng thời gian hiệu lực `availableFrom` – `availableTo`.
  - Danh sách email được phép tải (*whitelist*).
  - Mật khẩu bảo vệ.
  - Trạng thái công khai hoặc riêng tư.
- FR2.4: Các tệp hết hạn sẽ không còn khả dụng và trả về mã lỗi HTTP 410.

FR3 – Bảo mật và kiểm soát truy cập

- FR3.1: Mật khẩu tệp được băm bằng **bcrypt**; secret TOTP lưu trữ an toàn trong cơ sở dữ liệu.
- FR3.2: Quy trình kiểm tra truy cập bao gồm: trạng thái tệp → kiểm tra whitelist → yêu cầu mật khẩu (nếu có).
- FR3.3: Người dùng ẩn danh không thể chỉnh sửa thông tin tệp đã tải lên.
- FR3.4: Endpoint `/admin/cleanup` yêu cầu token admin hoặc header `X-Cron-Secret`.

FR4 – Quản lý vòng đời tệp

- FR4.1: Metadata tệp được lưu trong cơ sở dữ liệu bao gồm thuộc tính, thời hạn, thông tin bảo mật và trạng thái.
- FR4.2: Người dùng có thể xem thống kê tệp, bao gồm tổng lượt tải và thông tin truy cập gần đây.
- FR4.3: Hệ thống lưu lịch sử tải xuống để phục vụ mục đích kiểm toán.
- FR4.4: Tệp có thể bị xóa bởi chủ sở hữu hoặc bị dọn dẹp tự động khi hết hạn.

1.2.2 Yêu cầu phi chức năng

Hiệu năng

- NFR1: Thời gian phản hồi đăng nhập không vượt quá 5 giây.
- NFR2: Mã TOTP hoặc OTP phải được sinh và gửi trong vòng 30 giây (tối đa 60 giây khi tải cao).
- NFR3: Thời gian tải lên tệp dưới 100 MB không vượt quá 30 giây.
- NFR4: Việc tải xuống hỗ trợ streaming đảm bảo ổn định với tệp dung lượng lớn.

Bảo mật

- NFR5: Toàn bộ giao tiếp giữa FE và BE sử dụng HTTPS (TLS 1.2 trở lên).
- NFR6: JWT có thời hạn và hỗ trợ refresh token.
- NFR7: Share token được sinh ngẫu nhiên, chống dò quét.
- NFR8: Mật khẩu, TOTP và thông tin nhạy cảm được băm và xử lý đúng chuẩn bảo mật.

Khả năng mở rộng và bảo trì

- NFR9: Backend được thiết kế theo mô-đun, dễ mở rộng theo chức năng.
- NFR10: CSDL hỗ trợ đánh index và phân trang cho dữ liệu lớn.
- NFR11: Sẵn sàng tích hợp cloud storage và email service trong tương lai.

Chịu lỗi

- NFR12: Metadata và dữ liệu tệp được tách biệt, tránh mất dữ liệu khi lỗi lưu trữ.
- NFR13: Các thông báo lỗi tuân theo chuẩn JSON trong tài liệu OpenAPI.

Giao diện và trải nghiệm người dùng

- NFR14: Dashboard hiển thị danh sách tệp, trạng thái, thời hạn và thông tin bảo mật.

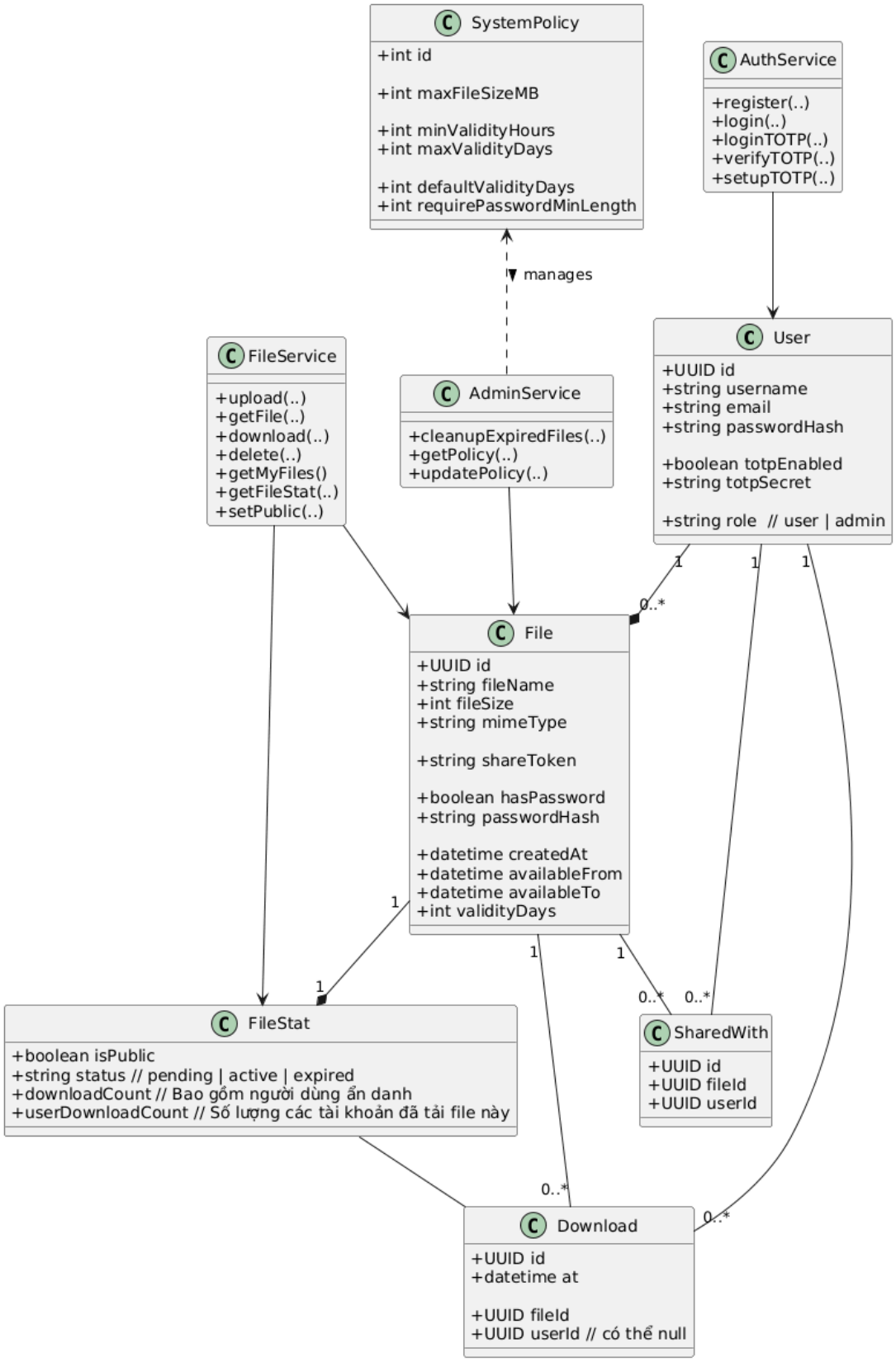
1.3 Use-case diagram cho toàn bộ hệ thống

Hình 1.1: Sơ đồ Use Case của hệ thống chia sẻ file

No.	Use-case	Job Description
1	Create Account	Dùng để người dùng tạo tài khoản mới trong hệ thống.
2	Login	Cho phép người dùng đăng nhập để truy cập các chức năng của hệ thống.
3	Logout	Đăng xuất khỏi hệ thống, kết thúc phiên làm việc.
4	Enable TOTP	Bật tính năng xác thực hai lớp (2FA) cho tài khoản người dùng.
5	Setup TOTP	Sinh mã QR và secret key để người dùng đăng ký Google Authenticator.
6	Verify TOTP	Kiểm tra mã xác thực sáu chữ số được tạo từ ứng dụng xác thực.
7	Upload files	Cho phép người dùng hoặc khách ẩn danh tải file lên hệ thống và tạo link chia sẻ.
8	Set password	Thiết lập mật khẩu bảo vệ file để chỉ người có mật khẩu mới tải được.
9	Share with	Chia sẻ file cho danh sách người dùng cụ thể qua email.
10	Download files	Tải file về nếu thỏa mãn điều kiện truy cập.
11	Download History	Ghi lại lịch sử tải file.
12	Delete files	Cho phép chủ sở hữu file xóa file khỏi hệ thống.
13	Update files	Cập nhật thông tin hoặc thuộc tính của file.
14	List my files	Hiển thị danh sách file của người dùng.
15	Validate Period	Kiểm tra thời gian hiệu lực file.
16	Set attribute file input	Thiết lập thuộc tính file.
17	System Policy	Xem và quản lý cấu hình hệ thống.
18	Cleanup Expired Files	Xóa các file đã hết hạn.

Bảng 1.1: Bảng mô tả Use Case của hệ thống chia sẻ file

1.4 Class Diagram



Hình 1.2: Sơ đồ Class của hệ thống chia sẻ file

No.	Class	Giải thích
1	User	Lưu trữ thông tin về tài khoản người dùng.
2	File	Lưu các metadata về các file được tải lên.
3	FileStat	Chứa các thông tin về tình trạng và thống về của file.
4	SharedWidth	Cho biết file này được chia sẻ với các người dùng nào.
5	Download	Ghi nhận những lượt tải về (bao gồm tải về ẩn danh).
6	FileService, AdminService và AuthService	Các interface để quản lý, tương tác với File, User, FileStat.
7	SystemPolicy	Các ràng buộc về tải lên và chia sẻ file.

Bảng 1.2: Bảng mô tả về Class diagram

CHƯƠNG 2

API SPEC

2.1 Giới thiệu chung

Hệ thống chia sẻ tệp tin được xây dựng với kiến trúc dịch vụ RESTful và được mô tả chính thức bằng chuẩn OpenAPI 3.0.4. Toàn bộ các endpoint, kiểu dữ liệu (schema), tham số và mã phản hồi được định nghĩa tập trung trong tệp `openapi.yaml`. Từ đặc tả này, nhóm có thể:

- Sinh tài liệu API tương tác cho backend thông qua Swagger UI.
- Import vào Postman để sinh collection phục vụ test và debug.
- Tự động kiểm tra tính hợp lệ (validate) của các request/response trong quá trình phát triển.

2.1.1 Công cụ xem và kiểm thử API

Để làm việc với tệp `openapi.yaml`, nhóm sử dụng các công cụ sau:

- **Swagger Editor**: copy nội dung `openapi.yaml` để xem live preview và kiểm tra cú pháp.
- **Swagger UI**: deploy cùng backend tại `http://localhost:8080/swagger/` để gửi thử request trực tiếp.
- **Postman**: import `openapi.yaml`/`swagger.json` để tạo collection kiểm thử tự động và thủ công.

2.2 Tổng quan API

2.2.1 Base URL

- **Development**: `http://localhost:8080/api`
- **Production**: `https://api.filesharing.com/api`

Mọi đường dẫn sau đây đều là tương đối so với Base URL.

2.2.2 Cơ chế xác thực

- Kiểu: Bearer Token (JWT)
- Header: `Authorization: Bearer <token>`
- Token cấp sau khi đăng nhập (có thể kèm bước xác thực TOTP cho tài khoản)

Các endpoint yêu cầu xác thực đều khai báo security schema trong OpenAPI.

2.3 Tóm tắt các nhóm Endpoint

2.3.1 Nhóm Authentication

- `POST /auth/register` – Đăng ký tài khoản mới.
- `POST /auth/login` – Đăng nhập bằng email/password; nếu tài khoản bật TOTP sẽ trả `requireTOTP = true`.
- `POST /auth/login/totp` – Xác thực mã TOTP để hoàn tất đăng nhập và nhận JWT.
- `POST /auth/totp/setup` – Thiết lập TOTP (yêu cầu Bearer token), trả về secret và QR code.
- `POST /auth/totp/verify` – Xác minh mã TOTP để kích hoạt 2FA (yêu cầu Bearer token).
- `POST /auth/logout` – Đăng xuất (client tự xóa token).
- `GET /user` – Lấy thông tin tài khoản cùng danh sách file (cần Bearer token).

2.3.2 Nhóm Files

- `POST /files/upload` – Upload tệp mới (có thể bảo vệ bằng mật khẩu, whitelist, cấu hình hiệu lực).
- `GET /files/{id}` – Lấy chi tiết tệp theo UUID (chỉ owner hoặc admin).
- `GET /files/{id}/stats` – Thống kê lượt tải (owner/admin).
- `GET /files/{id}/download-history` – Lịch sử tải chi tiết (owner/admin).
- `GET /files/{shareToken}` – Lấy metadata qua share token.
- `GET /files/{shareToken}/download` – Tải tệp (kiểm tra trạng thái, whitelist, mật khẩu).
- `DELETE /files/{id}` – Xóa tệp (chỉ owner; anonymous upload không được xóa).

2.3.3 Nhóm Admin

- `POST /admin/cleanup` – Xóa tệp hết hạn (yêu cầu Bearer token admin hoặc header `X-Cron-Secret`).
- `GET /admin/policy` – Lấy cấu hình hệ thống.
- `PATCH /admin/policy` – Cập nhật cấu hình hệ thống (chỉ admin).

2.4 Mã phản hồi HTTP

Bảng 2.1: Các mã phản hồi HTTP chính

Mã	Ý nghĩa	Mô tả
200	OK	Request thành công.
201	Created	Tạo mới resource thành công (ví dụ upload file).
400	Bad Request	Lỗi validate dữ liệu hoặc payload sai.
401	Unauthorized	Thiếu/không hợp lệ token.
403	Forbidden	Không có quyền, sai mật khẩu.
404	Not Found	Không tìm thấy resource.
409	Conflict	Dữ liệu xung đột (email/username tồn tại).
410	Gone	File đã hết hạn.
413	Payload Too Large	File vượt quá giới hạn dung lượng.
423	Locked	File chưa đến thời gian hiệu lực.

2.5 Các bảng cơ sở dữ liệu liên quan

Bảng 2.2: Các bảng chính

Bảng	Mô tả	Đặc điểm chính
`users`	Tài khoản người dùng	TOTP cho tài khoản, vai trò (user/admin).
`files`	Metadata file	Share token, password, hiệu lực thời gian.
`shared_with`	Danh sách chia sẻ	Quan hệ many-to-many giữa file và user (whitelist).
`file_statistics`	Thống kê lượt tải	Tổng lượt tải, người tải duy nhất.
`download_history`	Lịch sử tải	Log từng lượt tải (có thể anonymous).
`system_policy`	Policy hệ thống	Giới hạn dung lượng, thời hạn mặc định.

Schema chi tiết: `pkg/database/schema.sql`; migrations: thư mục `migrations/`; demo queries: `pkg/database/demo_queries.sql`.

2.6 Luồng TOTP/2FA cho tài khoản

Bật TOTP

1. `POST /auth/register` để tạo tài khoản.
2. `POST /auth/login` để nhận `accessToken`.
3. `POST /auth/totp/setup` (kèm Bearer token) để lấy secret + QR code.
4. Quét QR bằng Google Authenticator/Authy.
5. `POST /auth/totp/verify` với mã 6 số, tài khoản được đánh dấu `totpEnabled = true`.

Đăng nhập với TOTP

1. `POST /auth/login` với email/password.
2. Nếu bật 2FA, backend trả `requireTOTP = true`.
3. `POST /auth/login/totp` với mã 6 số hiện tại.
4. Nhận `accessToken` để gọi các endpoint bảo vệ.

2.7 Thống kê và phân tích lượt tải

2.7.1 Thống kê tổng quan

`GET /files/{id}/stats` (owner/admin) trả về:

- `downloadCount` – Tổng lượt tải.
- `uniqueDownloaders` – Số user đã xác thực khác nhau.
- `lastDownloadedAt` – Lần tải gần nhất.

Nguồn dữ liệu: bảng `file_statistics`. Anonymous upload có thể không đủ dữ liệu.

2.7.2 Lịch sử tải chi tiết

`GET /files/{id}/download-history` (owner/admin) cung cấp:

- Thông tin downloader (username/email hoặc null nếu anonymous).
- Thời điểm tải.
- Trạng thái hoàn thành hay bị gián đoạn.

Hỗ trợ phân trang với `page` và `limit`, dữ liệu từ `download_history`.

2.8 Trạng thái file và thời hạn hiệu lực

2.8.1 Trạng thái

- `pending`: chưa đến `availableFrom`.
- `active`: đang trong khoảng `availableFrom`–`availableTo`.
- `expired`: đã qua `availableTo`.

2.8.2 Logic mặc định

- Có cả FROM/TO: hiệu lực FROM → TO.
- Chỉ TO: hiện tại → TO.
- Chỉ FROM: FROM → FROM + 7 ngày.
- Không cung cấp: hiện tại → +7 ngày.

2.9 Bảo mật và kiểm soát tải xuống

2.9.1 Token và secret

- **Bearer Token (JWT)**: cấp từ `/auth/login` hoặc `/auth/login/totp`.
- **X-Cron-Secret**: secret cho cron job, gửi qua header `X-Cron-Secret` khi gọi `/admin/cleanup`.

2.9.2 Thứ tự kiểm tra khi tải file

1. **Trạng thái file**: hết hạn → 410; chưa đến giờ → 423.
2. **Whitelist**: nếu file private (có `sharedWith`), yêu cầu Bearer token; thiếu token → 401, không thuộc whitelist → 403.
3. **Mật khẩu**: nếu có password, phải cung cấp đúng; thiếu/sai → 403.

2.9.3 Mã phản hồi cho `/files/{shareToken}/download`

- 200 – Thành công.
- 401 – Thiếu Bearer token khi file yêu cầu whitelist.
- 403 – Sai/thiếu mật khẩu hoặc không thuộc whitelist.
- 404 – Share token không tồn tại.
- 410 – File hết hạn.
- 423 – File chưa đến thời gian hiệu lực.

2.10 Các kịch bản sử dụng điển hình

2.10.1 Upload ẩn danh và chia sẻ công khai

```
POST /files/upload
→ Nhận shareToken
→ Chia sẻ link: https://domain.com/f/{shareToken}
```

2.10.2 Upload với mật khẩu bảo vệ

```
POST /files/upload
Body: { file, password: "secret123" }
→ Người download cần nhập đúng password
```

2.10.3 Chia sẻ với whitelist

```
POST /files/upload
Body: {
  file,
  isPublic: false,
  sharedWith: ["user1@gmail.com", "user2@gmail.com"]
}
→ Chỉ user1 và user2 (đăng nhập) mới tải được
```

2.10.4 Xem ai đã tải file

```
1. GET /files/{id}/stats            → Thống kê tổng quan
2. GET /files/{id}/download-history → Lịch sử chi tiết
```

2.10.5 Tải file với nhiều lớp bảo mật (password + whitelist)

```
1. Đăng nhập để pass whitelist
2. GET /files/{shareToken}/download?password=secret123
```

2.11 Migration và dữ liệu demo

```
# Áp dụng migration
make migrate-up

# Kiểm tra version hiện tại
make migrate-version

# Rollback
make migrate-down

# Tạo migration mới
make migrate-create NAME=my_migration
```

Dữ liệu demo:

```
psql -h localhost -U postgres -d file_sharing_db \
  -f pkg/database/demo_queries.sql

# Hoặc vào container
make db-shell
# Sau đó chạy các truy vấn demo
```

Chương API spec trên phản ánh đầy đủ đặc tả hiện tại: cấu trúc endpoint, cơ chế bảo mật (Bearer token, whitelist, password), thống kê, migration và các luồng nghiệp vụ chính, làm cơ sở cho phát triển, kiểm thử và vận hành hệ thống chia sẻ tệp an toàn.