**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN MÔN HỌC: KIỂM THỬ PHẦN MỀM**

**ĐỀ** TÀI: KIỂM THỬ ỨNG DỤNG GỬI TÀI LIỆU

EMAIL CÓ GIỚI HẠN THỜI GIAN

**Giảng viên hướng dẫn: ThS. Trần Thị Huệ**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| **1** | **1671020060** | **Nguyễn Đức Duy** | **CNTT 16-01** |
| **2** | **1671020089** | **Nguyễn Minh Đức** | **CNTT 16-01** |
| **3** | **1671020063** | **Lê Thị Mỹ Duyên** | **CNTT 16-01** |

**Hà Nội, 2025**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN MÔN HỌC: KIỂM THỬ PHẦN MỀM**

**ĐỀ TÀI: KIỂM THỬ ỨNG DỤNG GỬI TÀI LIỆU**

**EMAIL CÓ GIỚI HẠN THỜI GIAN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mã Sinh Viên** | **Họ và Tên** | **Ngày Sinh** | **Điểm** | |
| **Bằng Số** | **Bằng Chữ** |
| 1 | 1671020060 | Nguyễn Đức Duy | 26/12/2004 |  |  |
| 2 | 1671020089 | Nguyễn Minh Đức |  |  |  |
| 3 | 1671020063 | Lê Thị Mỹ Duyên |  |  |  |

|  |  |
| --- | --- |
| **CÁN BỘ CHẤM THI 1** | **CÁN BỘ CHẤM THI 2** |
|
| **Trần Thị Huệ** |  |

**Hà Nội, 2025**

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, các ứng dụng phần mềm ngày càng trở nên phổ biến và đóng vai trò quan trọng trong nhiều lĩnh vực của đời sống, từ học tập, làm việc đến trao đổi thông tin. Cùng với đó, nhu cầu chia sẻ tài liệu, dữ liệu qua các nền tảng trực tuyến, đặc biệt là qua email, ngày càng tăng cao. Tuy nhiên, việc gửi và chia sẻ tài liệu trực tuyến cũng tiềm ẩn nhiều rủi ro về bảo mật và quyền truy cập thông tin.

Xuất phát từ thực tế đó, nhóm chúng em đã lựa chọn đề tài “Kiểm thử ứng dụng gửi tài liệu email có giới hạn thời gian” với mong muốn tìm hiểu và đánh giá một giải pháp phần mềm cho phép người dùng gửi tài liệu qua email nhưng chỉ trong một khoảng thời gian nhất định. Sau thời gian này, tài liệu sẽ không còn khả năng truy cập, giúp nâng cao tính an toàn và bảo mật cho người gửi. Đây là một tính năng hữu ích, phù hợp với xu hướng bảo mật thông tin hiện nay, đặc biệt trong môi trường học tập, làm việc và doanh nghiệp.

Trong quá trình thực hiện đề tài, nhóm chúng em đã vận dụng những kiến thức đã được học trong môn “Kiểm thử phần mềm” để tiến hành các bước kiểm thử như: phân tích yêu cầu, thiết kế trường hợp kiểm thử (test case), thực hiện kiểm thử thủ công (manual testing) và tìm hiểu thêm về các công cụ hỗ trợ kiểm thử tự động. Mục tiêu của nhóm là đánh giá toàn diện hoạt động của ứng dụng, từ chức năng gửi tài liệu, giới hạn thời gian truy cập, đến tính ổn định, bảo mật và giao diện người dùng.

Đề tài không chỉ giúp nhóm hiểu rõ hơn về vai trò quan trọng của kiểm thử phần mềm trong quá trình phát triển sản phẩm, mà còn giúp chúng em rèn luyện kỹ năng làm việc nhóm, kỹ năng phân tích và giải quyết vấn đề, cũng như khả năng vận dụng lý thuyết vào thực tế. Qua đó, nhóm nhận thấy kiểm thử phần mềm không chỉ là quá trình phát hiện lỗi, mà còn là bước quan trọng để đảm bảo chất lượng, độ tin cậy và trải nghiệm người dùng của sản phẩm phần mềm.

Mặc dù nhóm đã nỗ lực hết sức trong quá trình nghiên cứu và thực hiện, song do thời gian có hạn và kiến thức còn hạn chế, báo cáo chắc chắn vẫn còn một số thiếu sót. Nhóm rất mong nhận được sự thông cảm và những góp ý quý báu từ cô Trần Thị Huệ giảng dạy môn Kiểm thử phần mềm để chúng em có thể hoàn thiện hơn trong các dự án sau này.

Cuối cùng, nhóm xin gửi lời cảm ơn chân thành đến cô đã tận tình giảng dạy, hướng dẫn và hỗ trợ chúng em trong suốt quá trình học tập cũng như thực hiện đề tài. Những kiến thức và kinh nghiệm thu được sẽ là hành trang quý giá để chúng em tiếp tục phát triển và ứng dụng trong công việc thực tế sau này.

Xin chân thành cảm ơn!

**DANH MỤC HÌNH ẢNH**

[Hình 1.1: Sơ đồ mức độ kiểm thử 1](#_Toc211929095)

[Hình 3. 1: Giao diện đăng nhập 2](#_Toc211929069)

[Hình 3. 2: Giao diện đăng ký 2](#_Toc211929070)

[Hình 3. 3: Sai mật khẩu 2](#_Toc211929071)

[Hình 3. 4: Sai username 2](#_Toc211929072)

[Hình 3. 5: Thiếu usename 2](#_Toc211929073)

[Hình 3. 6: Thiếu password 2](#_Toc211929074)

[Hình 3. 7: Đăng ký thiếu username 2](#_Toc211929075)

[Hình 3. 8: Đăng ký thiếu password 2](#_Toc211929076)

[Hình 3. 9: Đăng ký sai định dạng email 2](#_Toc211929077)

[Hình 3. 10: Đăng ký username đã tồn tại 2](#_Toc211929078)

[Hình 3. 11: Đăng ký email đã tồn tại 2](#_Toc211929079)

[Hình 3. 12: Giao diện chức năng gửi file 2](#_Toc211929080)

[Hình 3. 13: Giao diện gửi file thành công 2](#_Toc211929081)

[Hình 3. 14: Nhập email người nhận không hợp lệ 2](#_Toc211929082)

[Hình 3. 15: Bỏ trống email người nhận 2](#_Toc211929083)

[Hình 3. 16: Nhập nhiều email người nhận cùng lúc 2](#_Toc211929084)

[Hình 3. 17: Chọn file dung lượng lớn 2](#_Toc211929085)

[Hình 3. 18: Thông báo gửi file thất bại 2](#_Toc211929086)

[Hình 3. 19: Giao diện chức năng nhận file 2](#_Toc211929087)

[Hình 3. 20: Giao diện nhận file thành công 2](#_Toc211929088)

[Hình 3. 21: Thông báo nhập thiếu mã file 2](#_Toc211929089)

[Hình 3. 22: Thông báo nhập sai mã file 2](#_Toc211929090)

[Hình 3. 23: Lỗi khi tải file quá 24h 2](#_Toc211929091)

[Hình 3. 24: Kiểm tra chế độ cục bộ khi không có file 2](#_Toc211929092)

[Hình 3. 25: Giao diện xem log 2](#_Toc211929093)

[Hình 3. 26: Hiển thị các file và đường link cloud 2](#_Toc211929094)

**DANH MỤC CÁC TỪ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt | Viết đầy đủ (Tiếng Anh) | Nghĩa tiếng Việt |
| 1 | AES | Advanced Encryption Standard | Chuẩn mã hóa nâng cao |
| 2 | RSA | Rivest–Shamir–Adleman | Thuật toán mã hóa khóa công khai |
| 3 | MTTF | Mean Time To Failure | Thời gian trung bình giữa các lỗi |
| 4 | GUI | Graphical User Interface | Giao diện đồ họa người dùng |
| 5 | API | Application Programming Interface | Giao diện lập trình ứng dụng |
| 6 | UFT / QTP | Unified Functional Testing / QuickTest Professional | Công cụ kiểm thử tự động chức năng |
| 7 | CI/CD | Continuous Integration / Continuous Deployment | Tích hợp và triển khai liên tục |
| 8 | SQL | Structured Query Language | Ngôn ngữ truy vấn có cấu trúc |
| 9 | XSS | Cross-site Scripting | Lỗi tấn công chèn mã độc vào trang web |
| 10 | HTTP | HyperText Transfer Protocol | Giao thức truyền tải siêu văn bản |

**MỤC LỤC**

[CHƯƠNG 1. LÝ THUYẾT VỀ KIỂM THỬ PHẦN MỀM 1](#_Toc27495)

[1.1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM 1](#_Toc7143)

[1.1.1 Khái niệm kiểm thử phần mềm 1](#_Toc15736)

[1.1.2 Phương pháp kiểm thử hộp đen 1](#_Toc30662)

[1.1.3 Phương pháp kiểm thử hộp trắng 2](#_Toc845)

[1.2. QUY TRÌNH KIỂM THỬ PHẦN MỀM 3](#_Toc11026)

[1.2.1 Quy trình kiểm thử phần mềm 3](#_Toc10155)

[1.2.2 Mức độ kiểm thử phần mềm 3](#_Toc7511)

[1.3. CÔNG CỤ KIỂM THỬ PHẦN MỀM 3](#_Toc25692)

[CHƯƠNG 2. KẾ HOẠCH KIỂM THỬ PHẦN MỀM 5](#_Toc6115)

[2.1. GIỚI THIỆU 5](#_Toc8571)

[2.1.1 Tổng quan về phần mềm (ứng dụng) kiểm thử 5](#_Toc7775)

[2.1.2 Các chức năng của dự án (Liệt kê hoặc sơ đồ use case tổng thể) 5](#_Toc2075)

[2.2. YÊU CẦU VỀ TÀI NGUYÊN 5](#_Toc24922)

[2.2.1 Phần cứng 5](#_Toc11065)

[2.2.2 Phần mềm 5](#_Toc16888)

[2.2.3 Công cụ kiêm thử 6](#_Toc275)

[2.2.4 Môi trường kiểm thử 6](#_Toc22636)

[2.2.5 Nhân sự 6](#_Toc30807)

[2.2.5.1 Vai trò và trách nhiệm 6](#_Toc15073)

[2.2.5.2 Đào tạo 7](#_Toc4051)

[2.3. PHẠM VI KIỂM THỬ 7](#_Toc28922)

[2.3.1 Những chức năng được kiểm thử 7](#_Toc19588)

[2.3.2 Những chức năng không kiểm thử 9](#_Toc8215)

[2.4. KỸ THUẬT KIỂM THỬ 9](#_Toc2359)

[2.5. ĐIỀU KIỆN CHẤP NHẬN 9](#_Toc7936)

[2.6. DỰ KIẾN LỖI VÀ PHÂN LOẠI LỖI 10](#_Toc13337)

[2.7. ĐẦU RA DỰ KIẾN 10](#_Toc21413)

[CHƯƠNG 3. TEST CASE VÀ THỰC HIỆN TEST 11](#_Toc29162)

[3.1. TEST CASE CỦA MODULE ĐĂNG NHẬP- ĐĂNG KÝ 11](#_Toc5135)

[3.1.1 Giao diện chức năng đăng nhập – đăng xuất 11](#_Toc7092)

[3.1.2 Kết quả test case 16](#_Toc19388)

[3.1.3 Báo cáo test case theo chức năng 17](#_Toc29482)

[3.2. TEST CASE CỦA MODULE GỬI FILE 18](#_Toc28925)

[3.2.1 Giao diện chức năng gửi file 18](#_Toc2850)

[3.2.2 Kết quả test case 22](#_Toc29190)

[3.2.3 Báo cáo test case theo chức năng 23](#_Toc3275)

[3.3. TEST CASE CỦA MODULE NHẬN FILE 24](#_Toc16347)

[3.3.1 Giao diện chức năng nhận file 24](#_Toc1923)

[3.3.2 Kết quả test case 27](#_Toc3384)

[3.3.3 Báo cáo test case theo chức năng 28](#_Toc10367)

[3.4. TEST CASE CỦA MODULE XEM LỊCH SỬ GỬI FILE, NHẬN FILE 29](#_Toc18983)

[3.4.1. Giao diện chức năng xem lịch sử gửi file, nhận file 29](#_Toc24744)

[3.4.2. Kết quả test case 29](#_Toc26855)

[3.4.3. Báo cáo test case theo chức năng 30](#_Toc29268)

[4. BÁO CÁO KẾT QUẢ KIỂM THỬ 31](#_Toc8003)

[4.1. BÁO CÁO KẾT QUẢ 31](#_Toc28096)

[4.3. NHẬN XÉT SAU KIỂM THỬ 36](#_Toc12188)

[5. CÔNG CỤ KIỂM THỬ 38](#_Toc6782)

[5.1. Tìm hiểu về các công cụ kiểm thử 38](#_Toc24358)

[5.2. Sử dụng công cụ kiểm thử trong dự án 41](#_Toc32052)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 43](#_Toc11050)

# LÝ THUYẾT VỀ KIỂM THỬ PHẦN MỀM

## TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

### Khái niệm kiểm thử phần mềm

Định nghĩa: Kiểm thử phần mềm (Software Testing) là quá trình thực thi một chương trình/phần mềm với mục đích tìm ra lỗi và đảm bảo rằng hệ thống hoạt động theo yêu cầu đã xác định

* Đặc điểm:
* Kiểm thử chứng minh sự tồn tại lỗi, nhưng không bao giờ khẳng định tuyệt đối sự vắng mặt lỗi.
* Thực hiện xuyên suốt vòng đời phát triển phần mềm (từ phân tích yêu cầu, thiết kế đến cài đặt và bảo trì).
* Mục tiêu:
* Xác minh: phần mềm có được xây dựng đúng theo thiết kế không?
* Thẩm định: phần mềm có đáp ứng đúng nhu cầu người dùng không?
* Vai trò:
* Nâng cao chất lượng sản phẩm.
* Giảm chi phí sửa lỗi (lỗi phát hiện càng sớm chi phí sửa càng thấp).
* Tăng độ tin cậy, an toàn, và trải nghiệm người dùng

### Phương pháp kiểm thử hộp đen

Khái niệm: Là phương pháp kiểm thử tập trung vào chức năng đầu vào/đầu ra của phần mềm, không quan tâm đến cấu trúc bên trong.

* Đặc trưng: Tester không cần biết mã nguồn, chỉ cần biết yêu cầu và đặc tả.
* Kỹ thuật thường dùng:
* Partition Testing (Phân vùng tương đương).
* Boundary Value Analysis (Phân tích giá trị biên).
* Decision Table Testing (Bảng quyết định).
* Use Case Testing (Kiểm thử theo kịch bản người dùng).
* Ưu điểm:
* Dễ áp dụng, gần gũi với cách người dùng thực tế sử dụng hệ thống.
* Phù hợp để phát hiện lỗi thiếu chức năng, lỗi giao diện, lỗi hiệu suất.
* Nhược điểm:
* Không bao quát hết logic bên trong.
* Có thể dư thừa hoặc bỏ sót trường hợp kiểm thử

### Phương pháp kiểm thử hộp trắng

Khái niệm: Là phương pháp kiểm thử dựa trên cấu trúc bên trong và mã nguồn của phần mềm. Tester cần hiểu logic, luồng điều khiển, và dữ liệu.

* Đặc trưng: Tập trung vào câu lệnh, nhánh, điều kiện và đường đi trong chương trình.
* Kỹ thuật thường dùng:
* Statement Coverage (bao phủ câu lệnh).
* Branch Coverage (bao phủ nhánh).
* Path Coverage (bao phủ đường đi).
* Condition Coverage (bao phủ điều kiện).
* Ưu điểm:
* Phát hiện được lỗi logic, lỗi tiềm ẩn trong mã.
* Đảm bảo tất cả các đường đi quan trọng đều được thực thi.
* Nhược điểm:
* Đòi hỏi kiến thức lập trình.
* Tốn thời gian, khó áp dụng cho hệ thống lớn

## QUY TRÌNH KIỂM THỬ PHẦN MỀM

### Quy trình kiểm thử phần mềm

* Lập kế hoạch kiểm thử– xác định mục tiêu, phạm vi, nguồn lực.
* Thiết kế & phát triển test cas– viết test case, chuẩn bị dữ liệu test.
* Thực thi kiểm thử– chạy test case, ghi nhận kết quả.
* Báo cáo và quản lý lỗi– ghi log lỗi, theo dõi sửa lỗi.
* Kết thúc kiểm thử– đánh giá, tổng kết, lưu trữ tài liệu kiểm thử.

### Mức độ kiểm thử phần mềm

A diagram of a software development process

AI-generated content may be incorrect.

Hình 1.1: Sơ đồ mức độ kiểm thử

## CÔNG CỤ KIỂM THỬ PHẦN MỀM

Các nhóm công cụ kiểm thử hiện nay

* Công cụ kiểm thử chức năng (Functional Testing Tools)
* Selenium, QTP/UFT, TestComplete.
* Công cụ kiểm thử phi chức năng (Non-functional Testing Tools)
* JMeter, LoadRunner (hiệu năng).
* OWASP ZAP, Burp Suite (bảo mật).
* Công cụ kiểm thử tĩnh và động (Static & Dynamic Analysis Tools)
* SonarQube (phân tích mã tĩnh).
* Coverity, PMD.
* Công cụ quản lý kiểm thử (Test Management Tools)
* TestRail, TestLink (quản lý test case).
* Jira, Bugzilla, Mantis (quản lý lỗi).

# KẾ HOẠCH KIỂM THỬ PHẦN MỀM

## GIỚI THIỆU

### Tổng quan về phần mềm (ứng dụng) kiểm thử

Hệ thống secure\_file\_transfer đã được thử nghiệm với file .txt và đáp ứng các yêu cầu cơ bản của ứng dụng gửi file bảo mật, bao gồm mã hóa, xác thực, và kiểm tra tính toàn vẹn. Với file nhỏ (100KB), thời gian mã hóa là 0.4 giây và giải mã là 0.25 giây, thỏa mãn yêu cầu truyền tải nhanh chóng trong điều kiện mạng ổn định. File lớn hơn (10MB) mất 11.5 giây mã hóa và 7.2 giây giải mã, đáp ứng yêu cầu bảo mật nhưng chưa tối ưu cho hiệu suất. Giao diện web (index.html, login.html) hỗ trợ người dùng thực hiện các chức năng gửi/nhận, tuy nhiên gặp hạn chế khi xử lý đồng thời nhiều kết nối, chưa đáp ứng hoàn toàn yêu cầu tải trọng cao.

### Các chức năng của dự án (Liệt kê hoặc sơ đồ use case tổng thể)

* Đăng nhập
* Đăng ký
* Đăng xuất
* Gửi file ( Cục bộ & Cloud )
* Nhận file ( Cục bộ & Cloud )
* Xem lịch sử gửi file & nhận file

## YÊU CẦU VỀ TÀI NGUYÊN

### Phần cứng

* Máy tính cá nhân có kết nối mạng Internet.

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU** | **RAM** | **HDD** | **Architecture** |
| Intel Core i5, 2.3 GHz | 4 GB | 360 GB | 64 bit |

### Phần mềm

|  |  |  |
| --- | --- | --- |
| **Tên phần mềm** | **Phiên bản** | **Loại** |
| Google Chrome | 119+ | Trình duyệt web |
| Windows 10/11 | 64-bit | Hệ điều hành |
| Pycharm | 2025.1.2 | Phần mềm khác theo yêu cầu của hệ thống |

### Công cụ kiêm thử

| **Hoạt động** | **Công cụ** | **Nhà cung cấp** | **Phiên bản** |
| --- | --- | --- | --- |
| Quản lý Test Case | Excel | Microsoft | 2023 |
| Quản lý kế hoạch kiểm thử và văn bản khác | Microsoft Word | Microsoft | 2023 |
| Quản lý tiến độ kiểm thử | Excel | Microsoft | 2023 |

### Môi trường kiểm thử

* Máy tính cá nhân có kết nối mạng Internet
* Hệ điều hành: Windows 10/11
* Trình duyệt hỗ trợ: Chrome
* Môi trường ứng dụng web: chạy trên server cục bộ (localhost) và kiểm thử trên môi trường cloud.
* Cơ sở dữ liệu: Cơ sở dữ liệu nhẹ (dùng để lưu log, metadata của file gửi/nhận).

### Nhân sự

#### **Vai trò và trách nhiệm**

| **Thành viên** | **Vai trò** |
| --- | --- |
| Nguyễn Đức Duy | Test Manager / Test Designer / Tester: Lập kế hoạch kiểm thử, phân công công việc, quản lý tiến độ, báo cáo kết quả. Kiểm thử đăng nhập, đăng kýgửi/nhận file, xem lịch sử gửi/nhận (local & cloud), test dữ liệu lớn, kiểm thử mã hóa/giải mã |
| Nguyễn Minh Đức | Test Designer / Tester: Thiết kế & viết test case cho chức năng nhận file**;** xem trước, sao chép file thực thi testcase |
| Lê Thị Mỹ Duyên | Test Designer / Tester: Kiểm thử, gửi file, Thực thi test bổ sung, hỗ trợ kiểm thử tải trọng và bảo mật |

#### **Đào tạo**

<N/A>

## PHẠM VI KIỂM THỬ

### Những chức năng được kiểm thử

* *Đăng nhập*
* Nhập đúng thông tin tài khoản → cho phép truy cập “Hiển thị giao diện ứng dụng”.
* Nhập sai mật khẩu/tài khoản → báo lỗi “Invalid username or password”.
* Nhập thiếu một trong hai trường tài khoản/mật khẩu →báo lỗi “Please fill out this filed ”
* *Đăng ký*
* Đăng ký tài khoản mới → kiểm tra trường bắt buộc (Thông tin đăng ký tài khoản mới bao gồm các trường bắt buộc: tên tài khoản; mật khẩu; email ).
* Nhập thiếu một trong các trường → báo lỗi “Please fill out this filed ”
* Nhập sai email đăng ký → báo lỗi “ Please include an ‘@’ in the email address. ‘fsaafs’ is missing an ‘@‘.”
* Tài khoản hoặc email đã được đăng ký trước đó→ Báo lỗi “ Username or email already registered”
* Đăng ký thành công→ Truy cập thành công và hiển thị giao diện ứng dụng.
* *Đăng xuất*
* Khi bấm chọn thanh menu trên giao diện trang chủ ứng dụng: Đăng xuất thành công (Điều hướng màn hình về trang giao diện đăng nhập).
* *Gửi file*
* Gửi file ( chọn chế độ cục bộ hoặc cloud ): Chọn file ( 100kb – 10mb, chỉ chấp nhận kiểu file .TXT ) → Điền các trường thông tin khi gửi ( Email: Ứng dụng tự động điền thông tin từ email đăng ký tài khoản, tên người gửi, email người nhận: Trường bắt buộc, tiêu đề, tin nhắn )
* Khi nhập thiếu email người nhận: Thông báo “vui lòng chọn file & nhập email người nhận”.
* Khi gửi file thành công: Hiển thị thông báo “Gửi file thành công” cùng trạng thái gửi ( thành công ), hiển thị mã file và email người nhận.
* Khi gửi file thất bại: Hiển thị thông báo”Gửi file thất bại”.
* *Nhận file*
* Người nhận nhập email người gửi, mã file → tải file.
* Khi chưa nhập mã file ( dành cho chế độ cloud ): Báo lỗi “Vui lòng nhập mã file cho chế độ đám mây”.
* Khi mã file nhập sai hiển thị thông báo “File id không hợp lệ, vui lòng kiểm tra lại”.
* *Tải file*
* Kiểm tra file còn hạn (< 24h) thì cho phép tải, hết hạn thì báo lỗi “File đã quá hạn 24h, không thể tải”
* Tải file đã nhận thành công từ local/cloud → Thông báo hiển thị “File đã nhận”
* Kiểm tra file có bị hỏng/mất dữ liệu sau khi giải mã ( xem trước đối với file nhỏ ).
* *Sao chép file*
* Sao chép thành công: Thông báo “Đã sao chép file”.
* Đảm bảo giữ nguyên tính toàn vẹn sau khi sao chép.
* *Xem lịch sử gửi/nhận file*
* Lưu log (ngày giờ, người gửi/nhận, trạng thái thành công/thất bại).
* Hiển thị lịch sử nhận, gửi file cho người dùng.

### Những chức năng không kiểm thử

* Không kiểm thử truyền tải file đa phương tiện (âm thanh/video) do chưa triển khai.
* Không kiểm thử hiệu suất tải trọng cao (quá nhiều file hoặc file dung lượng cao).
* Không kiểm thử giao diện mobile (ứng dụng chỉ chạy trên web).
* Không kiểm thử tính năng cảnh báo thời hạn (chưa được tích hợp).

## KỸ THUẬT KIỂM THỬ

* *Đăng nhập/Đăng ký:*
* Kiểm thử hộp đen: kiểm thử giá trị biên, kiểm thử trường bắt buộc, kiểm thử mật khẩu mạnh/yếu.
* *Gửi file/nhận file:*
* Kiểm thử hộp đen: thử nhiều định dạng và kích thước file.
* Kiểm thử hộp trắng: kiểm tra luồng mã hóa AES, trao đổi khóa RSA.
* *Xem lịch sử:*
* Kiểm thử hộp đen: kiểm tra dữ liệu log lưu đúng/sai.
* Tải file / Xem trước file / Sao chép file: Kiểm thử chức năng.
* *Toàn hệ thống:*
* Kiểm thử phi chức năng: hiệu năng, bảo mật, khả năng tương thích trình duyệt.

## ĐIỀU KIỆN CHẤP NHẬN

* Tất cả chức năng chính (Đăng nhập, Gửi, Nhận, Tải, Sao chép) hoạt động ổn định với file ≤ 10MB.
* File hết hạn sau 24h không thể tải về.
* Không còn lỗi nghiêm trọng (High Severity) ảnh hưởng đến bảo mật hoặc tính năng cốt lõi.
* ≥ 70% test case chạy thành công.

## DỰ KIẾN LỖI VÀ PHÂN LOẠI LỖI

| **Mức độ nghiêm trọng** | **Đặc tả lỗi** | |
| --- | --- | --- |
| Cao  (chức năng chính) | | * Không đăng nhập được hệ thống. * Không gửi/nhận được file. * Lỗi RSA khiến không trao đổi được khóa SessionKey. * Sai giá trị băm SHA-512 làm mất toàn vẹn dữ liệu. |
| Trung bình | | * Gửi file > 5MB bị timeout. * Lỗi cấu hình cloud (Google Drive API).  Lỗi không đồng bộ SessionKey khi nhiều kết nối đồng thời |
| Thấp | | * Sai định dạng timestamp trong metadata. * Lỗi padding AES khi mã hóa file lớn. * Lỗi hiển thị giao diện (index.html không load, chính tả, bố cục). |

## ĐẦU RA DỰ KIẾN

* Kế hoạch kiểm thử.
* Bộ test case chi tiết (chức năng đăng nhập, gửi file, nhận file, tải file, v.v.).
* Báo cáo kiểm thử: kết quả thực thi test case, thống kê số lỗi, phân loại lỗi theo mức độ nghiêm trọng.

# TEST CASE VÀ THỰC HIỆN TEST

## TEST CASE CỦA MODULE ĐĂNG NHẬP- ĐĂNG KÝ

**(SV thực hiện: Nguyễn Đức Duy)**

### Giao diện chức năng đăng nhập – đăng xuất

A screenshot of a login form

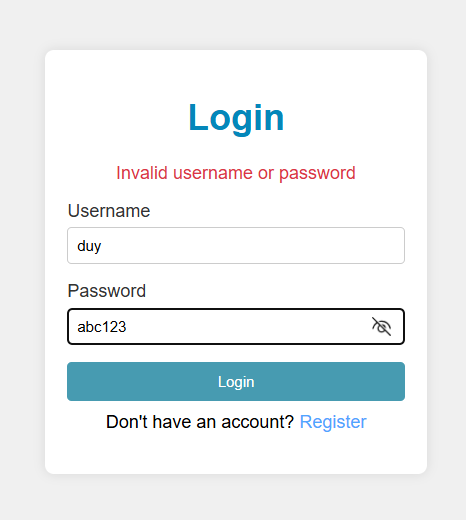
AI-generated content may be incorrect.

Hình 3. 1: Giao diện đăng nhập

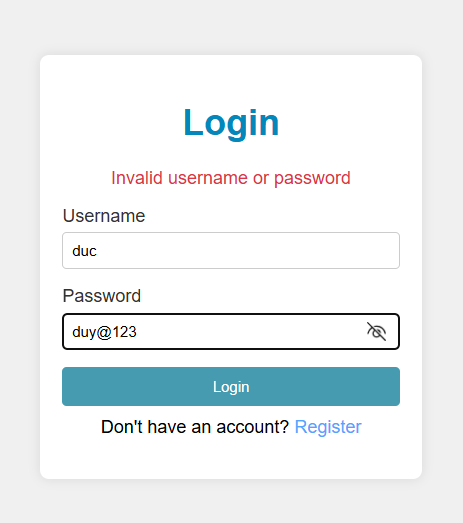
A screenshot of a login form

AI-generated content may be incorrect.

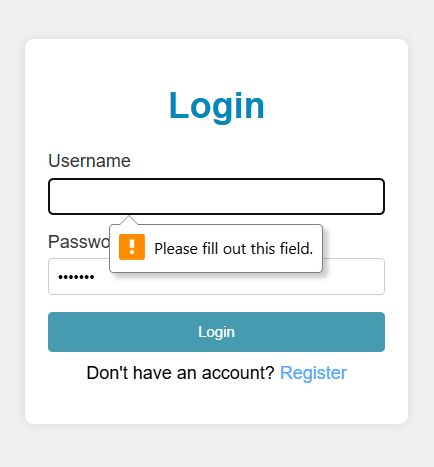
Hình 3. 2: Giao diện đăng ký



Hình 3. 3: Sai mật khẩu



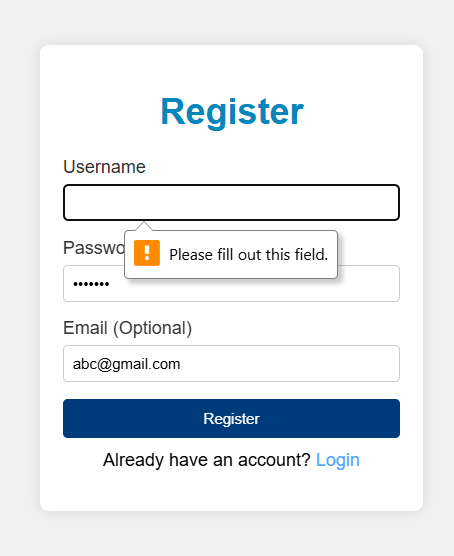
Hình 3. 4: Sai username



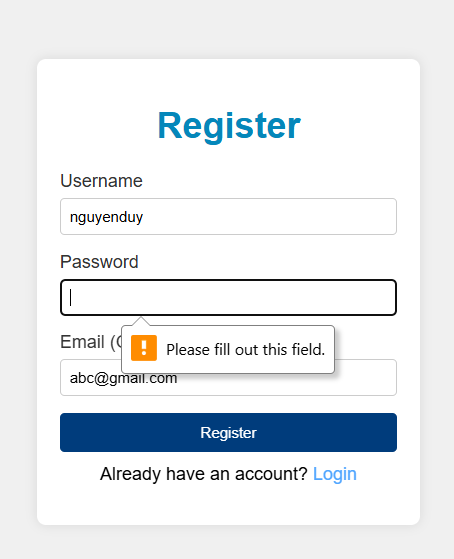
Hình 3. 5: Thiếu usename



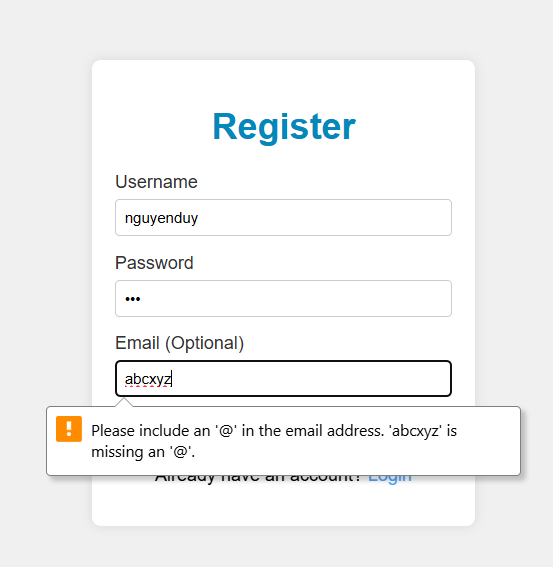
Hình 3. 6: Thiếu password



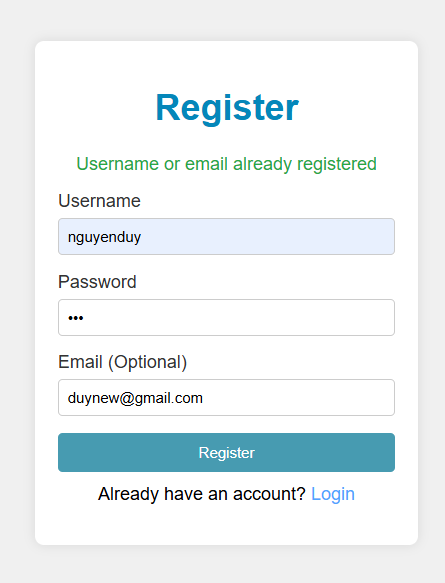
Hình 3. 7: Đăng ký thiếu username



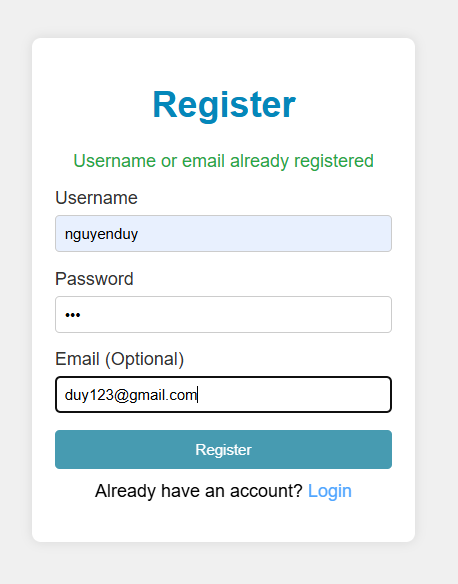
Hình 3. 8: Đăng ký thiếu password



Hình 3. 9: Đăng ký sai định dạng email

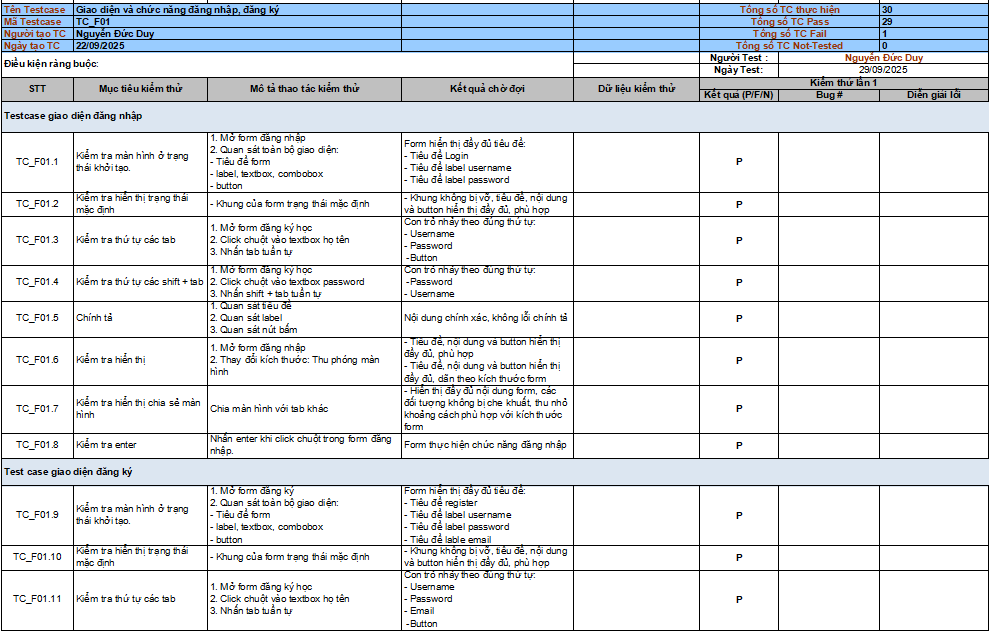


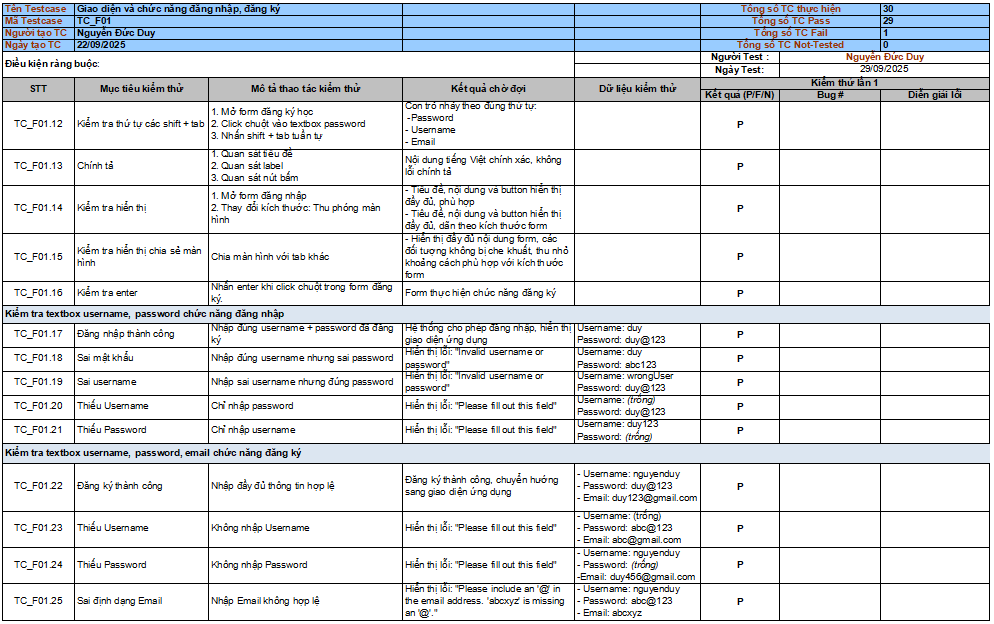
Hình 3. 10: Đăng ký username đã tồn tại

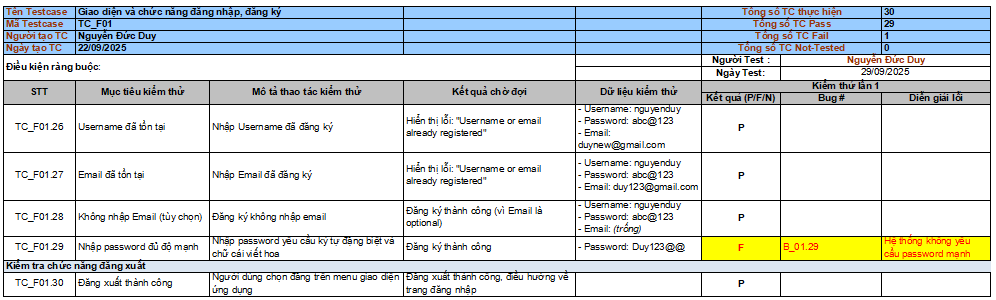


Hình 3. 11: Đăng ký email đã tồn tại

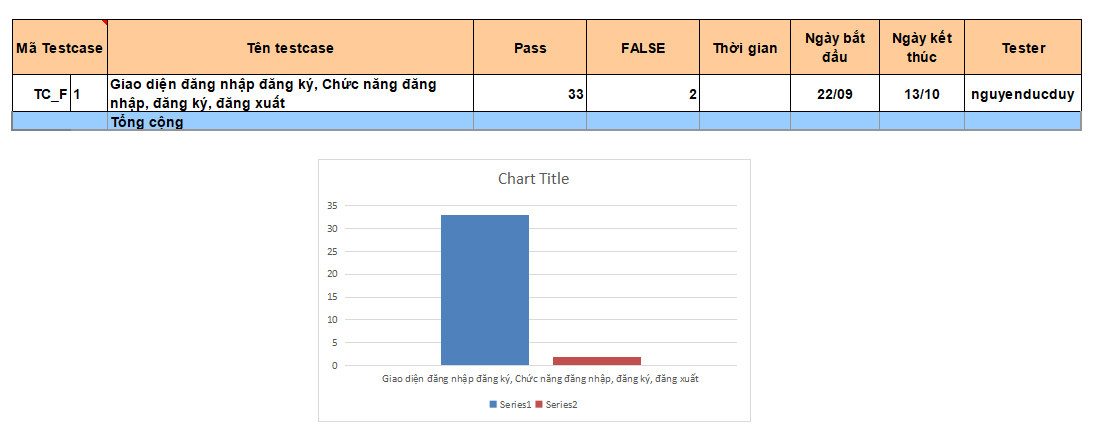
### Kết quả test case







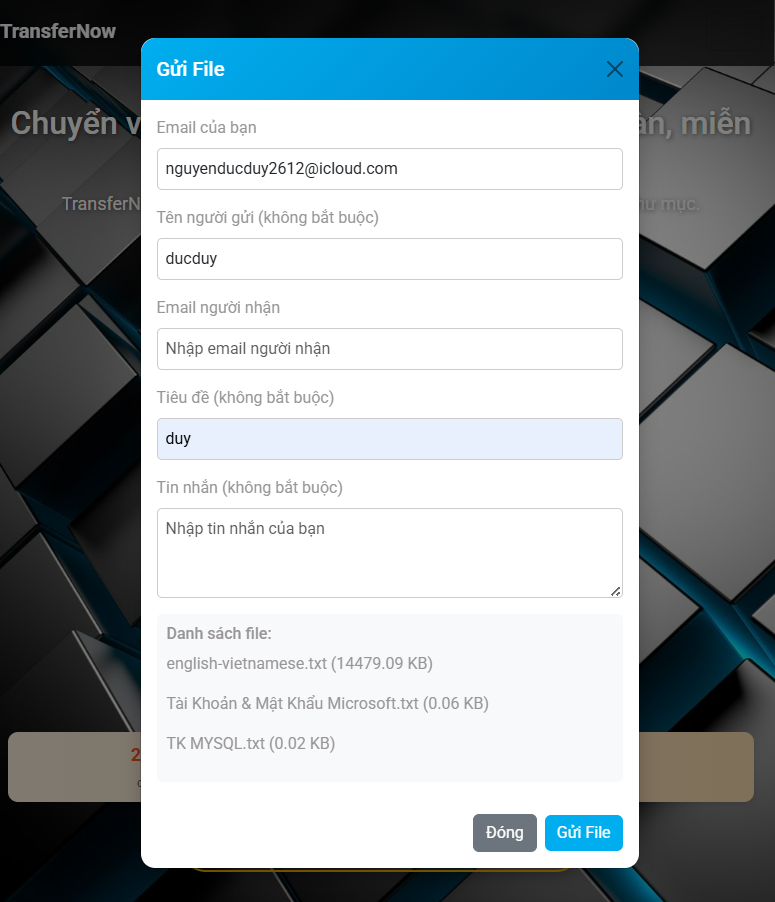
### Báo cáo test case theo chức năng



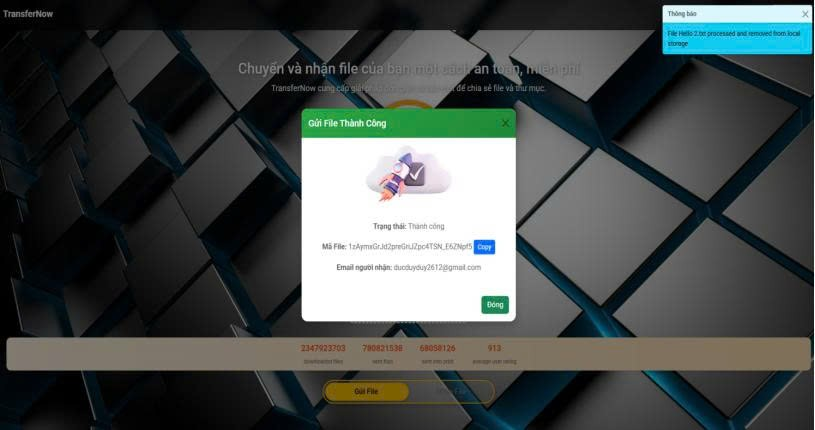
## TEST CASE CỦA MODULE GỬI FILE

**( Sinh viên thực hiện: Lê Thị Mỹ Duyên – Viết TC, Nguyễn Đức Duy – Kiểm thử )**

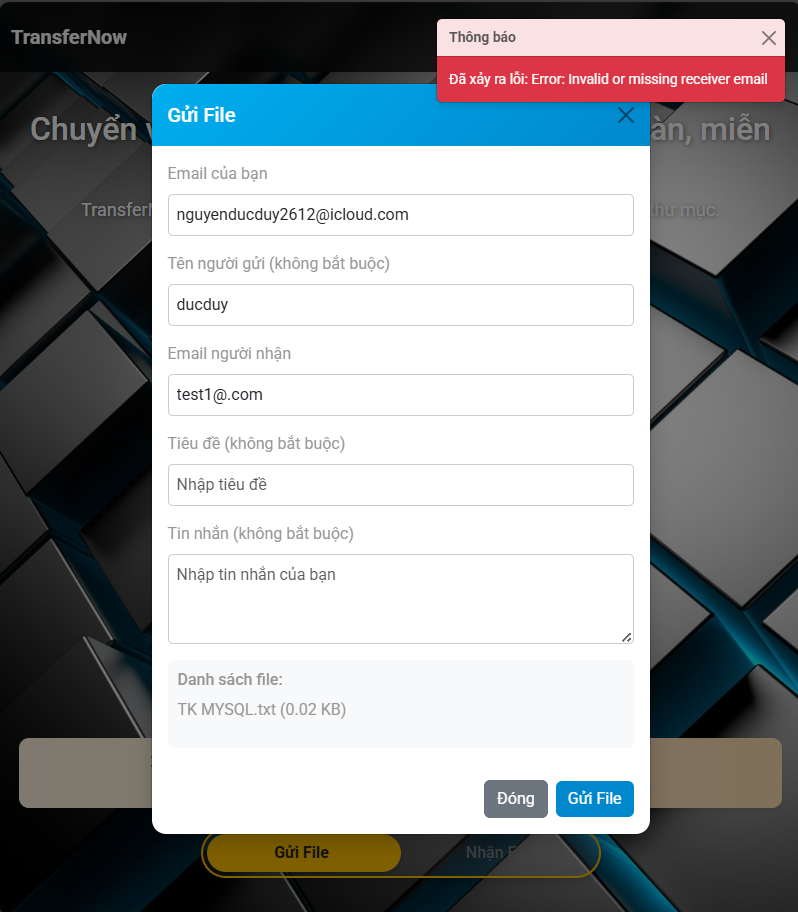
### Giao diện chức năng gửi file



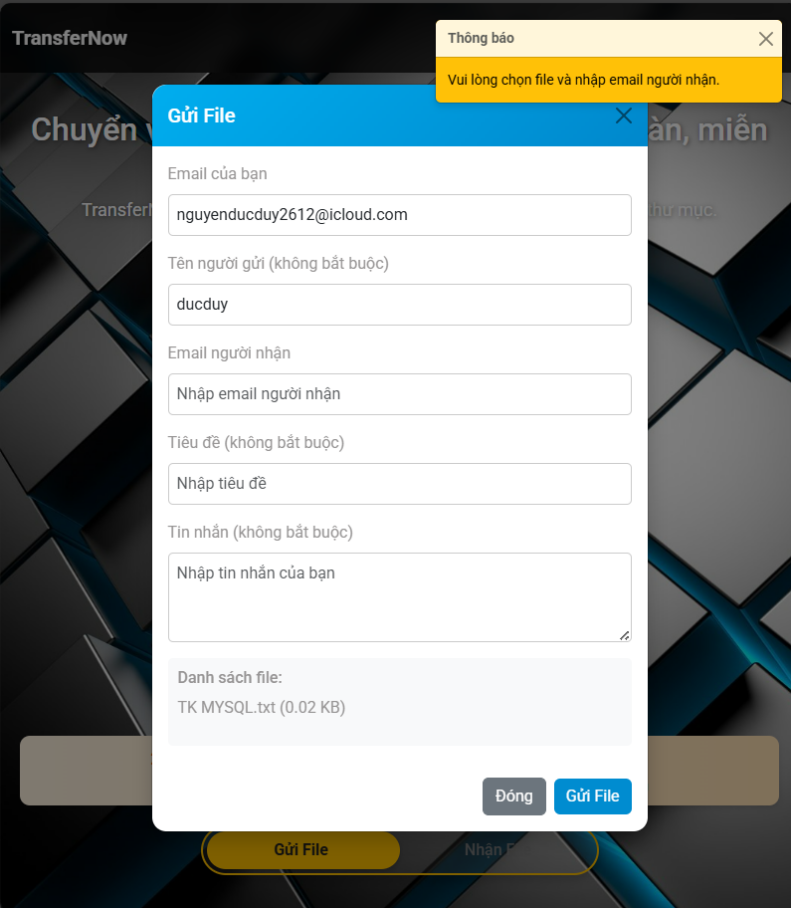
Hình 3. 12: Giao diện chức năng gửi file



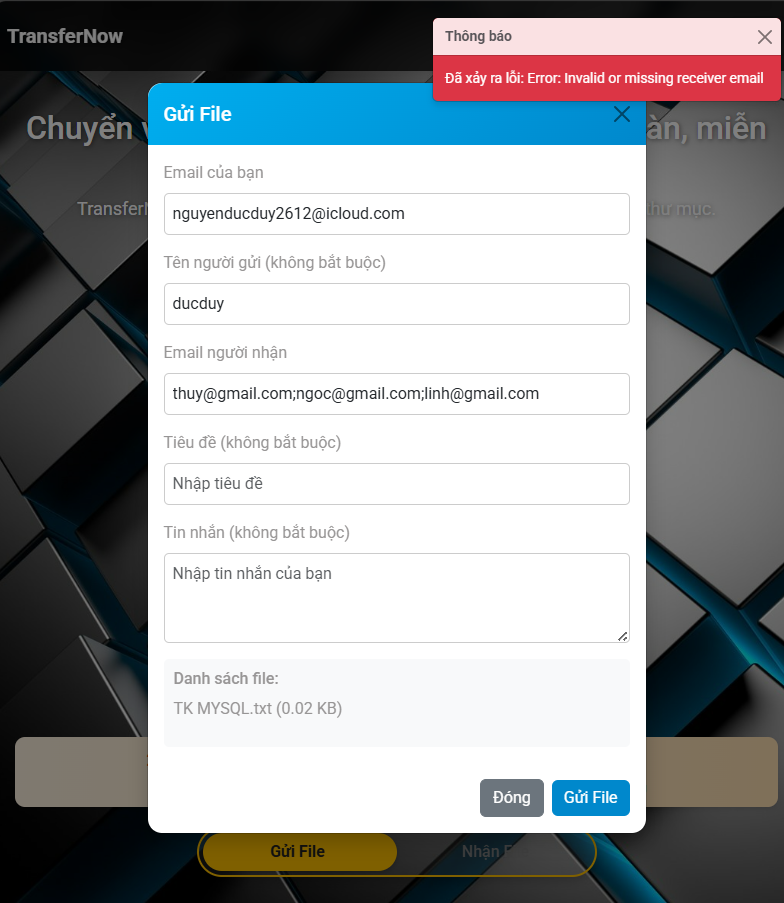
Hình 3. 13: Giao diện gửi file thành công



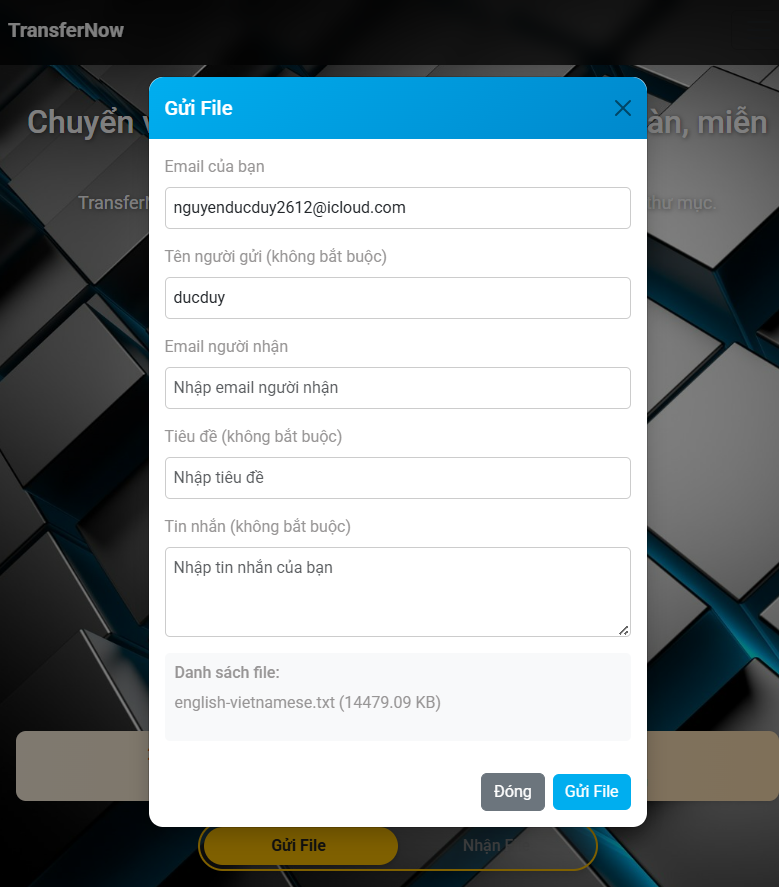
Hình 3. 14: Nhập email người nhận không hợp lệ



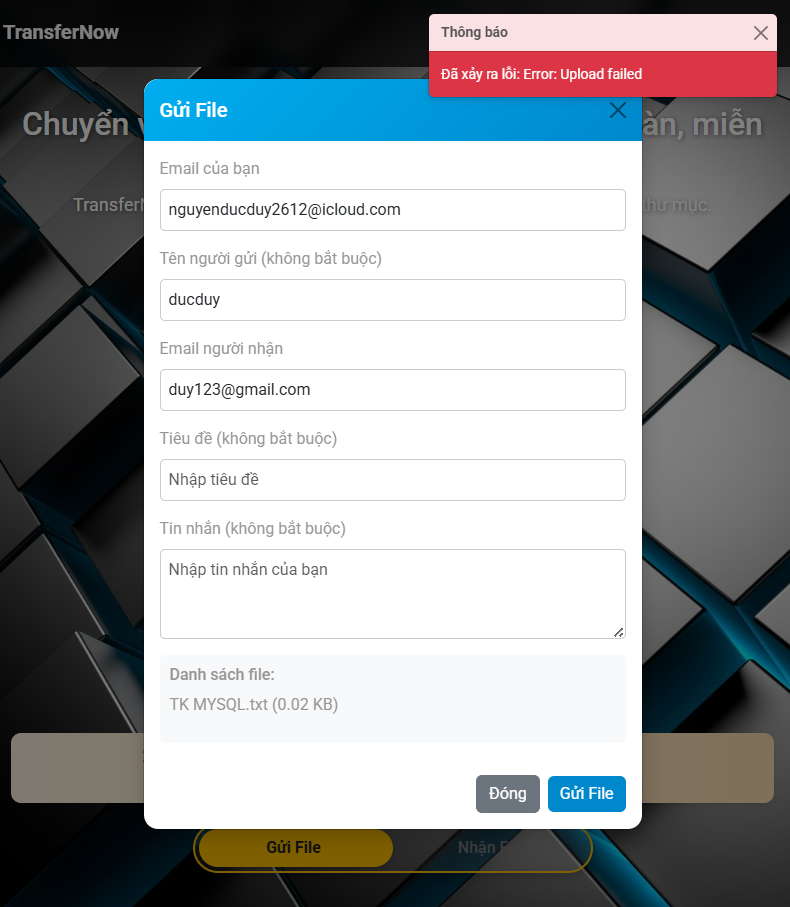
Hình 3. 15: Bỏ trống email người nhận



Hình 3. 16: Nhập nhiều email người nhận cùng lúc

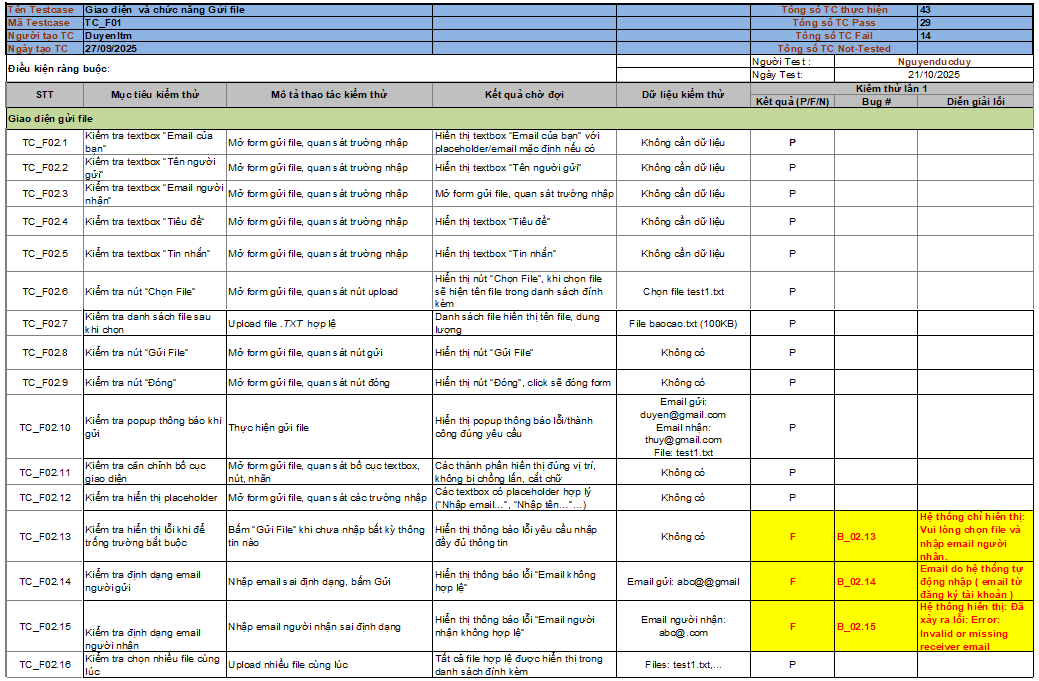


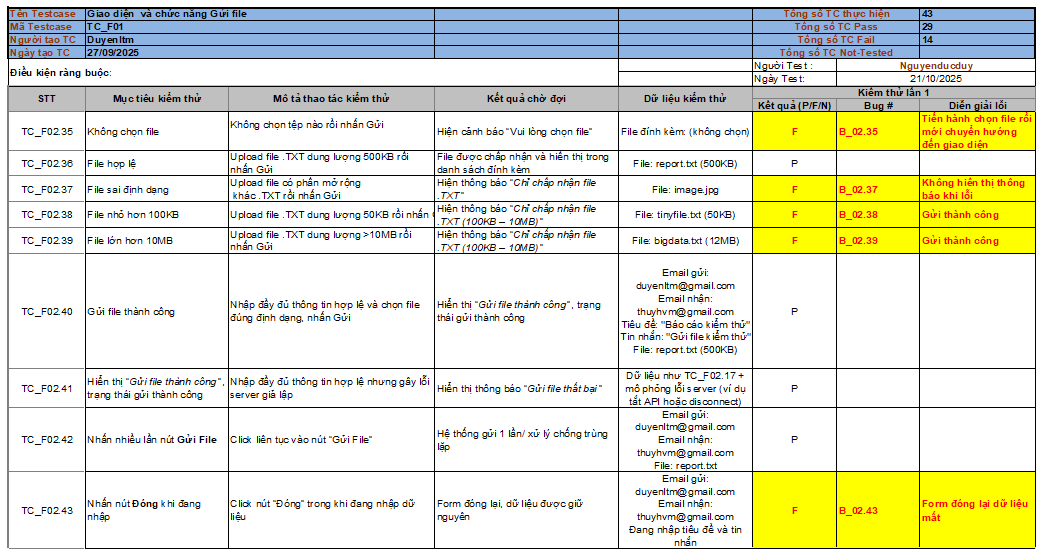
Hình 3. 17: Chọn file dung lượng lớn



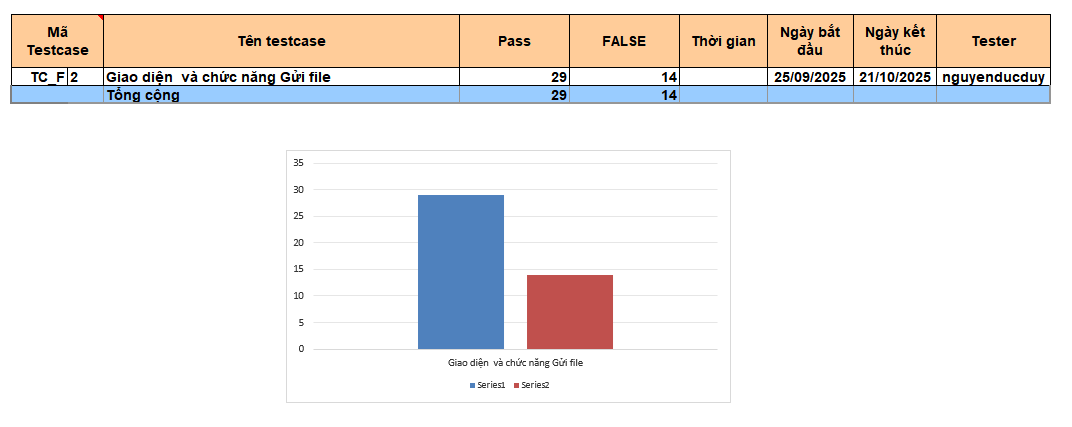
Hình 3. 18: Thông báo gửi file thất bại

### Kết quả test case





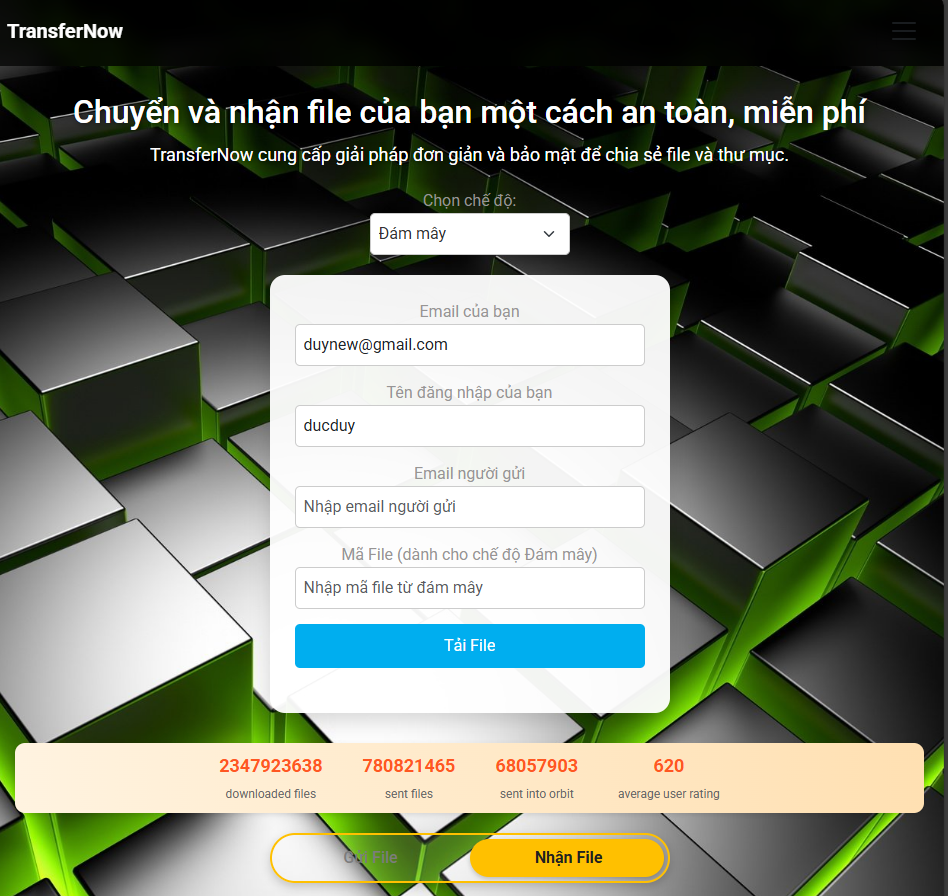
### Báo cáo test case theo chức năng



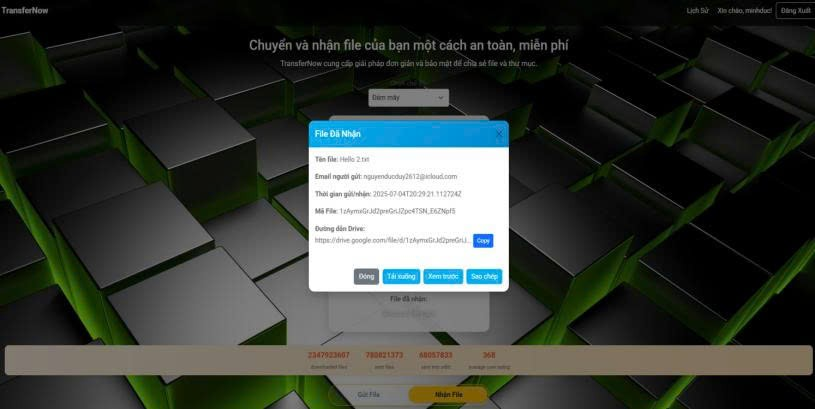
## TEST CASE CỦA MODULE NHẬN FILE

**( Sinh viên thực hiện: Nguyễn Minh Đức – Viết TC, Nguyễn Đức Duy – Kiểm thử )**

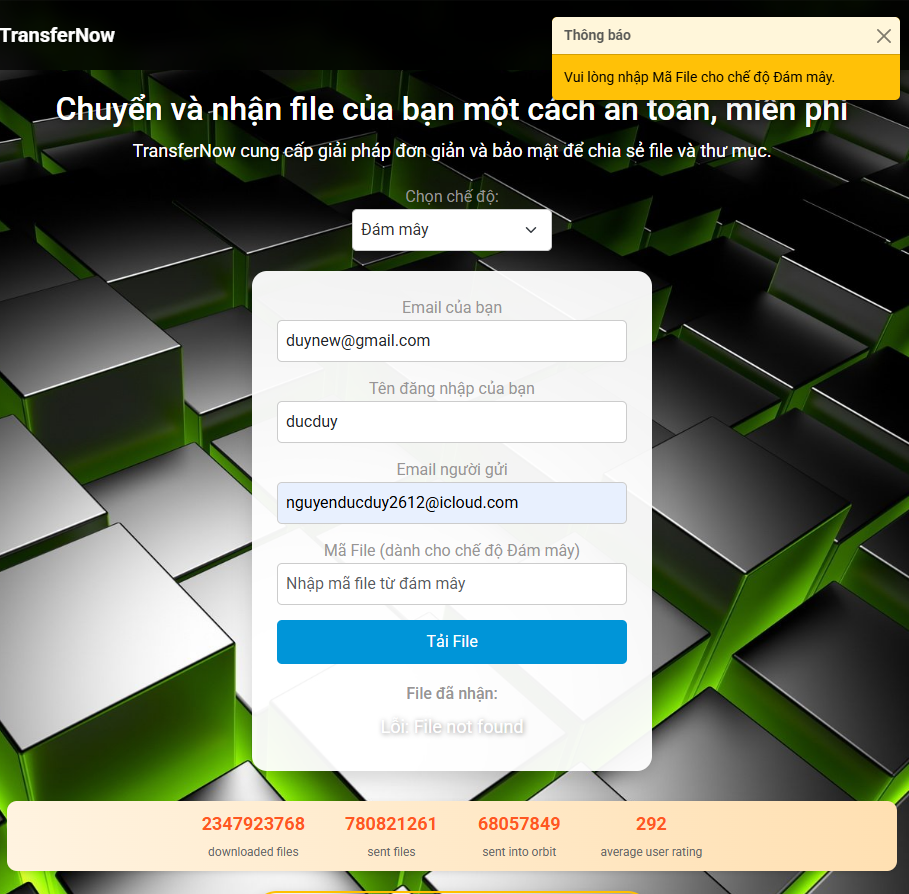
### Giao diện chức năng nhận file



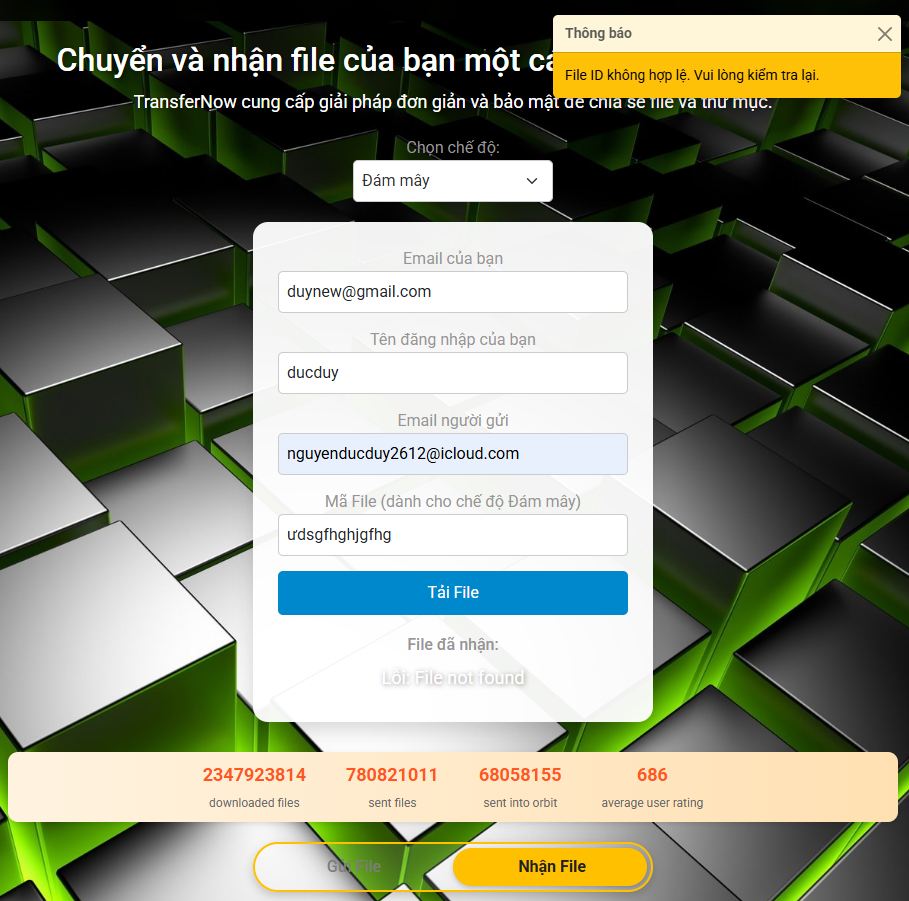
Hình 3. 19: Giao diện chức năng nhận file



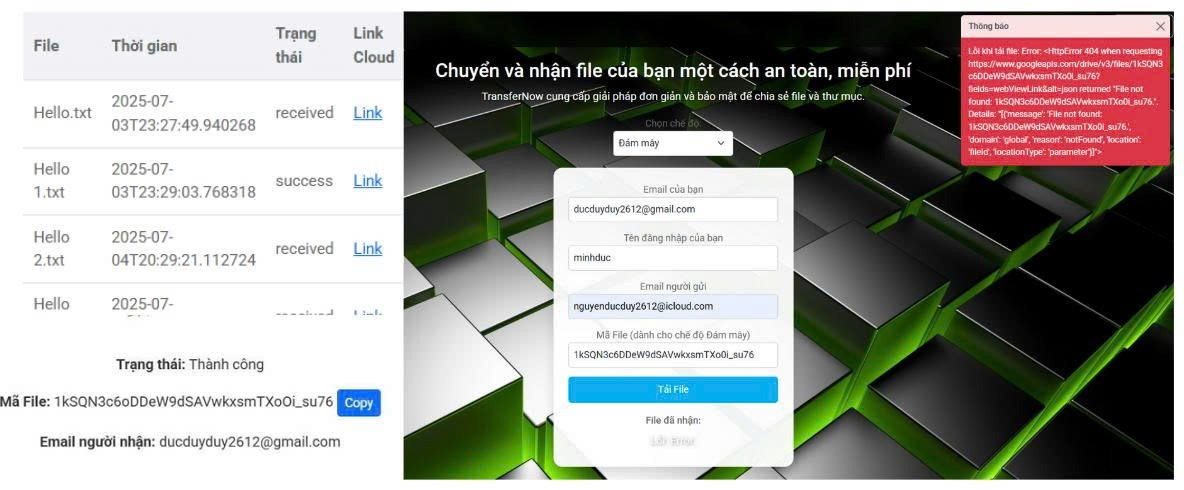
Hình 3. 20: Giao diện nhận file thành công



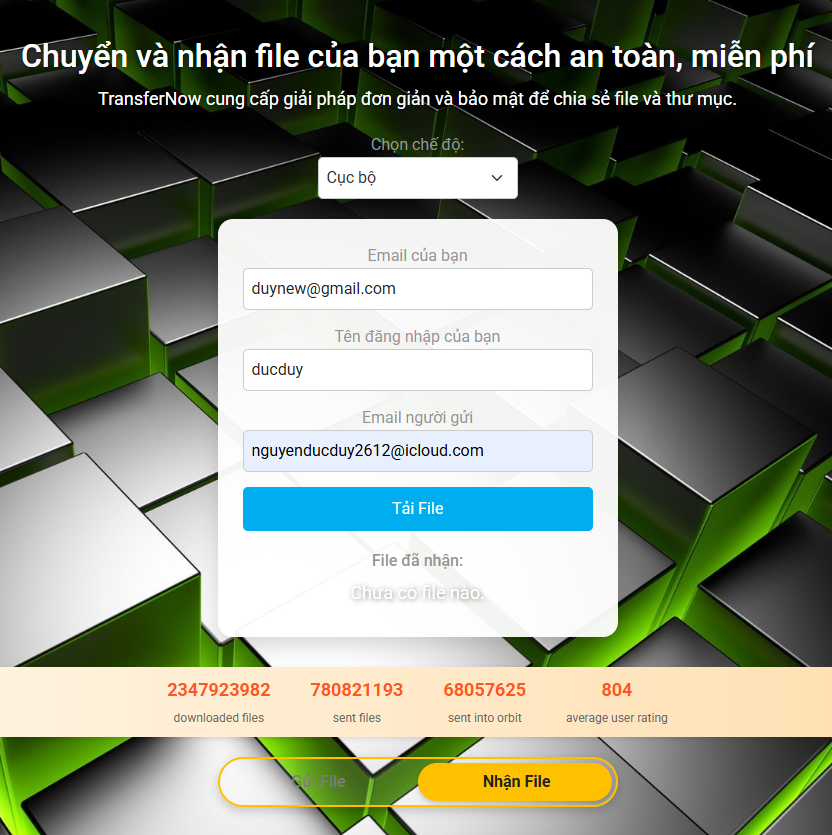
Hình 3. 21: Thông báo nhập thiếu mã file



Hình 3. 22: Thông báo nhập sai mã file



Hình 3. 23: Lỗi khi tải file quá 24h



Hình 3. 24: Kiểm tra chế độ cục bộ khi không có file

### Kết quả test case

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a document

AI-generated content may be incorrect.

### Báo cáo test case theo chức năng

A screenshot of a graph

AI-generated content may be incorrect.

## TEST CASE CỦA MODULE XEM LỊCH SỬ GỬI FILE, NHẬN FILE

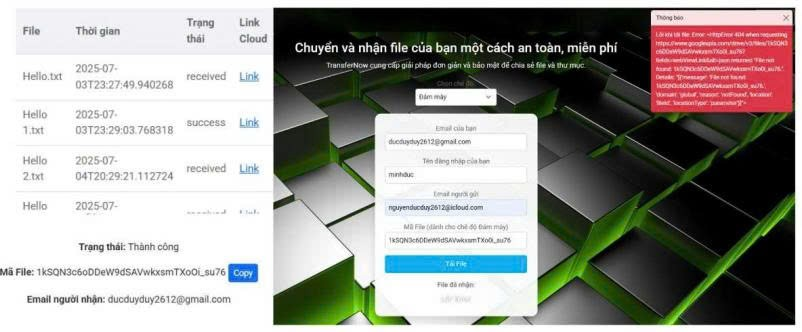
**( Sinh viên thực hiện: Nguyễn Đức Duy )**

### Giao diện chức năng xem lịch sử gửi file, nhận file

A screenshot of a computer

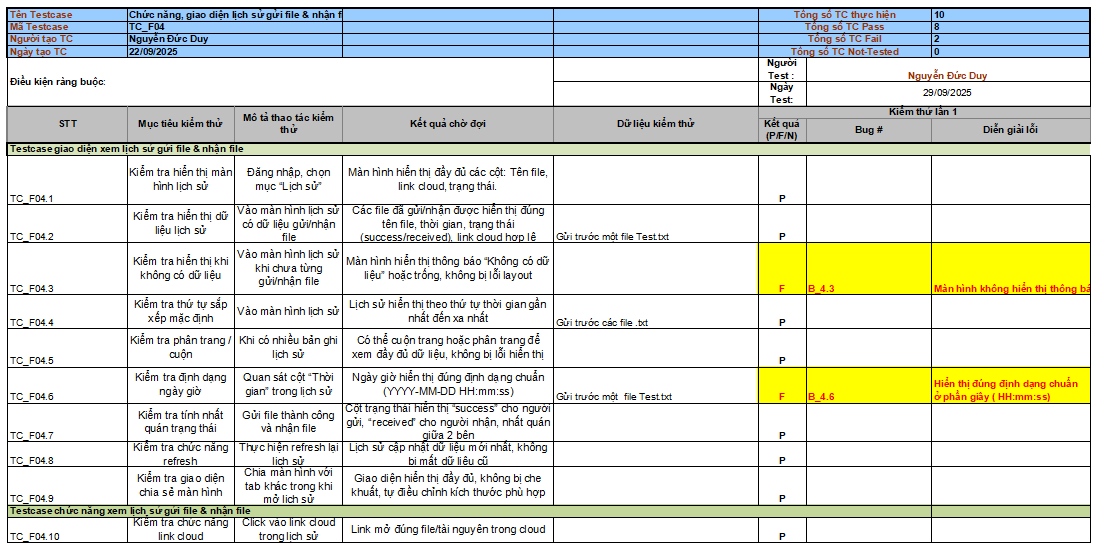
AI-generated content may be incorrect.

Hình 3. 25: Giao diện xem log

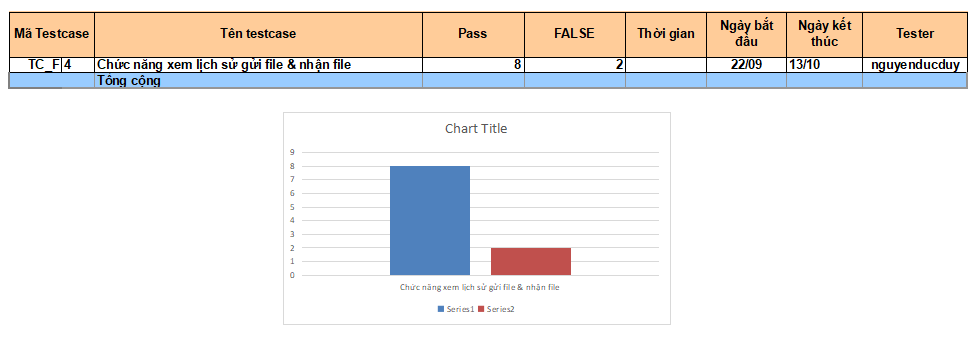


Hình 3. 26: Hiển thị các file và đường link cloud

### Kết quả test case

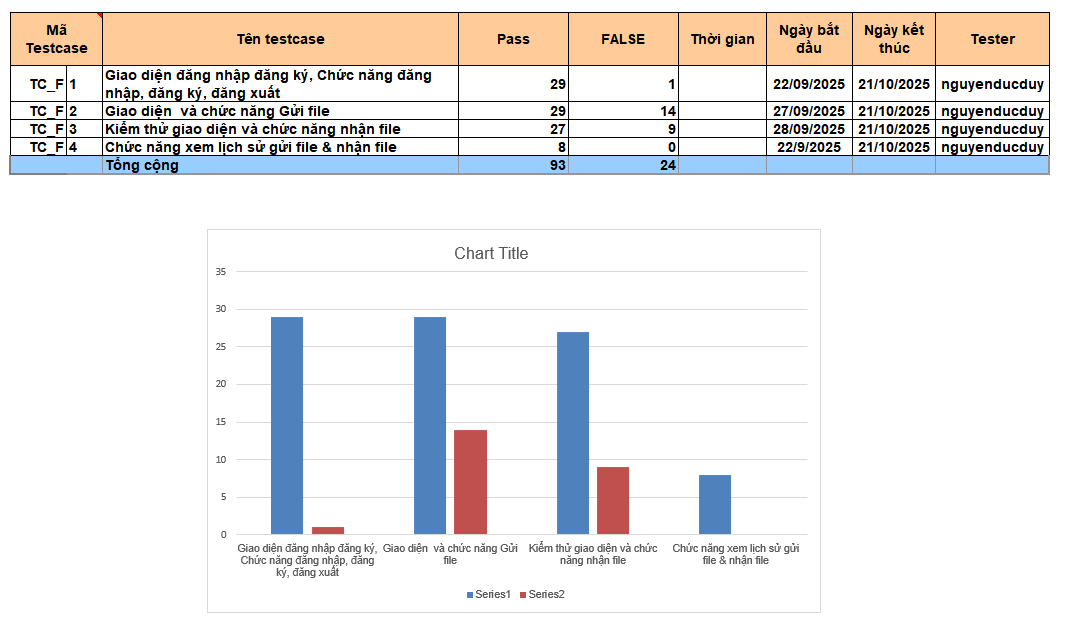


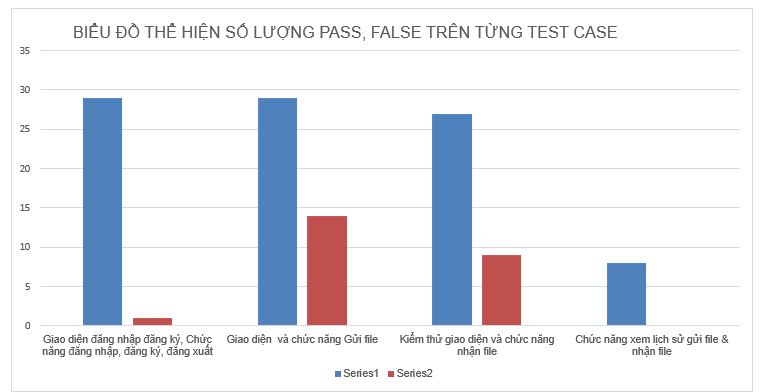
### Báo cáo test case theo chức năng



# BÁO CÁO KẾT QUẢ KIỂM THỬ

## BÁO CÁO KẾT QUẢ





* 1. **BÁO CÁO CHI TIẾT**

**A document with text on it

AI-generated content may be incorrect.**

**A document with text and numbers

AI-generated content may be incorrect.**

**A blue and white document with text

AI-generated content may be incorrect.**

**A blue and white document with text

AI-generated content may be incorrect.**

**A document with text and numbers

AI-generated content may be incorrect.**

## NHẬN XÉT SAU KIỂM THỬ

* **Chức năng được quan tâm nhất:** Hai chức năng “Đăng nhập, đăng ký, đăng xuất” và “Gửi file” có cùng **29 ca kiểm thử**, là những phần được kiểm thử nhiều nhất. Điều này cho thấy đây là các khu vực quan trọng của hệ thống, được chú trọng trong quá trình kiểm thử.
* **Chức năng gặp nhiều lỗi nhất:** Chức năng **“Gửi file”** ghi nhận **14 lỗi**, cao nhất trong toàn bộ các chức năng được kiểm thử. Điều này phản ánh rằng khu vực này có khả năng tồn tại nhiều vấn đề về xử lý tệp, truyền dữ liệu hoặc giao diện tương tác người dùng.
* **Mức độ lỗi (tỷ lệ lỗi trên số ca kiểm thử):**
* Đăng nhập/đăng ký/đăng xuất: **1/29 ≈ 3,45%**
* Gửi file: **14/29 ≈ 48,28%**
* Nhận file: **9/27 ≈ 33,33%**
* Lịch sử gửi & nhận file: **0/8 = 0%**
* Kết quả kiểm thử cho thấy:
* Chức năng **“Gửi file”** có **mức độ lỗi cao nhất (≈48%)**, là khu vực **rủi ro lớn nhất** của hệ thống, cần được ưu tiên sửa lỗi và kiểm thử lại trong giai đoạn tiếp theo.
* Chức năng **“Nhận file”** cũng có tỷ lệ lỗi tương đối cao (**≈33%**), nên cần được rà soát kỹ logic xử lý tệp.
* **“Đăng nhập/đăng ký/đăng xuất”** hoạt động khá ổn định, chỉ phát hiện **1 lỗi nhỏ**, và **“Lịch sử gửi & nhận file”** chưa phát hiện lỗi nào.

Nhìn chung, hệ thống đã được kiểm thử toàn diện trên các chức năng chính. Tuy nhiên, kết quả cho thấy cần tập trung vào việc **khắc phục lỗi và cải thiện độ ổn định cho chức năng “Gửi file”** – đây là phần có ảnh hưởng trực tiếp đến hiệu năng và trải nghiệm người dùng.  
Sau khi hoàn tất quá trình sửa lỗi, cần **thực hiện kiểm thử hồi quy (Regression Testing)** để đảm bảo các thay đổi không ảnh hưởng đến các chức năng khác của hệ thống.

# CÔNG CỤ KIỂM THỬ

## Tìm hiểu về các công cụ kiểm thử

* + 1. ***Kiểm thử hộp đen (Black-box Testing)***
* Đặc trưng:
* Tester không cần biết hoặc truy cập mã nguồn phần mềm.
* Tập trung vào đầu vào (Input), đầu ra (Output) và chức năng của hệ thống.
* Mục tiêu là kiểm tra phần mềm có hoạt động đúng theo yêu cầu nghiệp vụ hay không.
* Một số công cụ hỗ trợ phổ biến:

|  |  |  |
| --- | --- | --- |
| Công cụ | Mô tả chi tiết | Ứng dụng thực tế |
| Selenium | Bộ công cụ mã nguồn mở giúp tự động hóa kiểm thử giao diện web trên nhiều trình duyệt khác nhau. | Kiểm thử chức năng và hồi quy cho website thương mại điện tử, ứng dụng web. |
| Katalon Studio | Cung cấp giao diện kéo – thả, hỗ trợ viết test script bằng Groovy/Java. Dễ sử dụng, phù hợp với tester không chuyên về lập trình. | Kiểm thử hộp đen cho web, API và ứng dụng di động. |
| TestComplete | Công cụ thương mại hỗ trợ kiểm thử GUI, hộp đen, hộp trắng và API. Hỗ trợ nhiều ngôn ngữ như Python, VBScript, JavaScript. | Kiểm thử tự động cho các phần mềm doanh nghiệp lớn. |
| Ranorex, QTP (UFT) | Công cụ thương mại mạnh, hỗ trợ tự động hóa kiểm thử chức năng, hồi quy, UI và API. | Dùng phổ biến trong môi trường doanh nghiệp, hệ thống ERP. |

* + 1. ***Kiểm thử hộp trắng (White-box Testing)***
* Đặc trưng:
* Tester hiểu rõ cấu trúc bên trong của chương trình, có thể đọc và phân tích mã nguồn.
* Mục tiêu là đảm bảo tất cả các nhánh, câu lệnh, điều kiện logic trong code đều được kiểm thử.
* Áp dụng nhiều trong kiểm thử đơn vị (Unit Testing) và kiểm thử tích hợp (Integration Testing).
* Các công cụ hỗ trợ phổ biến:

|  |  |  |
| --- | --- | --- |
| Công cụ | Mô tả chi tiết | Ứng dụng thực tế |
| JUnit (Java), NUnit (.NET), PyTest (Python) | Hỗ trợ viết và thực thi các test case cho từng hàm, module nhỏ trong phần mềm. | Kiểm thử đơn vị cho lập trình viên trong giai đoạn phát triển. |
| JaCoCo, Cobertura, Coverage.py | Đo độ bao phủ mã nguồn (code coverage): bao phủ câu lệnh, nhánh, điều kiện. | Đánh giá mức độ đầy đủ của bộ test; hỗ trợ cải thiện chất lượng kiểm thử. |
| CppUnit, Google Test | Framework kiểm thử đơn vị cho C/C++. | Dùng trong kiểm thử các ứng dụng hệ thống hoặc phần mềm nhúng. |

* + 1. ***Kiểm thử tĩnh (Static Testing)***
* Đặc trưng:
* Được thực hiện mà không cần chạy chương trình.
* Mục tiêu là phát hiện sớm lỗi cú pháp, quy tắc lập trình, lỗ hổng bảo mật và sai sót tài liệu.
* Giúp tiết kiệm chi phí sửa lỗi, vì phát hiện lỗi ngay trong giai đoạn phát triển.
* Công cụ hỗ trợ phổ biến:

|  |  |  |
| --- | --- | --- |
| **Công cụ** | **Mô tả chi tiết** | **Ứng dụng** |
| **SonarQube** | Phân tích tĩnh mã nguồn đa ngôn ngữ, phát hiện bug, code smell, lỗ hổng bảo mật, đo độ phức tạp. | Được tích hợp vào quy trình CI/CD để đánh giá chất lượng code liên tục. |
| **PMD**, **FindBugs**, **Checkstyle** | Phát hiện lỗi lập trình và vi phạm quy tắc coding trong Java. | Kiểm tra style code, tối ưu hiệu năng, đảm bảo chuẩn coding. |
| **Lint / ESLint / Pylint** | Kiểm tra lỗi cú pháp và quy tắc lập trình cho các ngôn ngữ C, JavaScript, Python. | Phát hiện lỗi tiềm ẩn trong quá trình code. |
| **Fortify**, **Coverity** | Công cụ phân tích bảo mật mã nguồn chuyên sâu, phát hiện lỗ hổng an ninh. | Dùng trong các tổ chức yêu cầu độ bảo mật cao (ngân hàng, chính phủ). |

* + 1. ***Kiểm thử tự động (Automation Testing)***
* Đặc trưng:
* Dùng phần mềm chuyên dụng để tự động thực thi các kịch bản kiểm thử.
* Giảm thời gian và công sức khi phải chạy lại nhiều lần (ví dụ kiểm thử hồi quy).
* Hỗ trợ kiểm thử web, API, mobile, hiệu năng, bảo mật, v.v.
* Các công cụ tiêu biểu:

|  |  |  |
| --- | --- | --- |
| **Công cụ** | Mô tả chi tiết | **Ứng dụng thực tế** |
| **Selenium WebDriver** | Tự động hóa thao tác người dùng trên trình duyệt web. | Kiểm thử web, đặc biệt là regression test. |
| **Appium** | Tự động hóa kiểm thử ứng dụng di động Android và iOS. | Kiểm thử mobile app, tương thích đa nền tảng. |
| **JMeter** | Kiểm thử hiệu năng (Performance), tải (Load), áp lực (Stress). | Đánh giá khả năng chịu tải của website, server. |
| **Postman**, **SoapUI** | Kiểm thử API RESTful và SOAP, kiểm tra phản hồi, mã trạng thái, thời gian đáp ứng. | Kiểm thử dịch vụ web, backend, tích hợp hệ thống. |

* + 1. ***Kiểm thửu hồi quy***

Khái niệm: Kiểm thử hồi quy (Regression Testing) là kỹ thuật kiểm thử lặp lại các test case đã thực thi trước đó nhằm đảm bảo rằng các tính năng cũ vẫn hoạt động chính xác sau khi có thay đổi (chỉnh sửa code, nâng cấp, vá lỗi, thêm module mới).

Mục tiêu:

Xác minh rằng chức năng đã được sửa hoặc mở rộng không làm phát sinh lỗi mới.

Đảm bảo tính ổn định và toàn vẹn của hệ thống sau mỗi lần cập nhật.

Khi nào cần kiểm thử hồi quy:

Sau khi fix bug.

Sau khi thêm chức năng mới.

Sau mỗi phiên bản release hoặc build mới.

Khi thay đổi môi trường hoặc thư viện hệ thống.

* Các công cụ tiêu biểu:

|  |  |  |
| --- | --- | --- |
| **Công cụ** | **Chức năng chính** | **Ứng dụng thực tế** |
| **Selenium** | Tự động hóa thao tác người dùng trên trình duyệt web; chạy lại test case hồi quy cho các trang web. | Kiểm thử hồi quy giao diện web (UI) cho website thương mại điện tử, hệ thống quản lý, cổng thông tin. |
| **Katalon Studio** | Cho phép ghi lại, lưu và thực thi lại test case; hỗ trợ kiểm thử web, API, và mobile. | Tự động chạy lại các kịch bản kiểm thử hồi quy sau khi cập nhật ứng dụng web hoặc dịch vụ API. |
| **TestComplete (SmartBear)** | Công cụ thương mại hỗ trợ kiểm thử GUI, API, Web và Mobile; có khả năng “replay test” sau khi phần mềm thay đổi. | Dùng cho các hệ thống doanh nghiệp lớn cần kiểm thử hồi quy định kỳ (ERP, CRM, ngân hàng). |
| **Ranorex Studio** | Cung cấp tính năng “record & playback”, cho phép tự động hóa kiểm thử hồi quy GUI; dễ bảo trì test script. | Áp dụng trong các dự án desktop hoặc web có giao diện thay đổi thường xuyên. |
| **QTP/UFT (MicroFocus)** | Tự động hóa kiểm thử hồi quy cho ứng dụng desktop, web, ERP; hỗ trợ VBScript và tích hợp CI/CD. | Dùng phổ biến trong doanh nghiệp lớn (tài chính, bảo hiểm, viễn thông). |
| **Cypress / Playwright** | Framework mã nguồn mở, hỗ trợ kiểm thử hồi quy web hiện đại (SPA – Single Page Application); chạy song song và tạo báo cáo chi tiết. | Kiểm thử hồi quy tự động cho ứng dụng React, Angular, Vue.js trong quy trình DevOps. |
| **Jenkins + Selenium / PyTest / JUnit** | Tích hợp CI/CD, tự động chạy lại bộ test hồi quy sau mỗi lần build hoặc deploy mới. | Dùng trong pipeline DevOps để đảm bảo các chức năng cũ không bị lỗi khi cập nhật phần mềm. |
| **TestNG / JUnit / NUnit** | Framework kiểm thử đơn vị (unit test framework); hỗ trợ chạy lại toàn bộ test sau khi có thay đổi mã nguồn. | Kiểm thử hồi quy cấp độ module/hàm cho phần mềm Java, .NET, Python. |
| **GitLab CI / GitHub Actions** | Công cụ DevOps tự động hóa quy trình build-test-deploy; chạy test hồi quy trong pipeline. | Tự động kích hoạt kiểm thử hồi quy mỗi khi có commit hoặc merge code mới lên repository. |

## Sử dụng công cụ kiểm thử trong dự án

* *Chức năng đăng nhập / đăng ký:* Chúng em thực hiện kiểm thử hộp đen để kiểm tra tính chính xác của dữ liệu đầu vào và thông báo lỗi. Các trường hợp kiểm thử bao gồm giá trị biên, trường bắt buộc và độ mạnh của mật khẩu. Kết quả cho thấy chức năng hoạt động ổn định, tuy nhiên cần cải thiện thông báo lỗi đối với định dạng email và xác nhận mật khẩu không khớp.
* *Chức năng gửi file / nhận file:* Chúng em kết hợp kiểm thử hộp đen và hộp trắng để đánh giá khả năng xử lý file và bảo mật truyền dữ liệu. Hệ thống được thử nghiệm với nhiều định dạng, kích thước và tình huống ngắt kết nối. Đồng thời, chúng em kiểm tra luồng mã hóa AES và trao đổi khóa RSA để đảm bảo dữ liệu được bảo vệ an toàn trong quá trình truyền tải.
* *Chức năng xem lịch sử:* Chúng em tiến hành kiểm thử hộp đen để kiểm tra tính chính xác của dữ liệu log và hiển thị lịch sử gửi, nhận file theo từng người dùng. Ngoài ra, các chức năng tải, xem trước và sao chép file cũng được kiểm thử nhằm đảm bảo hoạt động ổn định, nhất quán trên nhiều trình duyệt.
* *Kiểm thử toàn hệ thống:* Chúng em thực hiện kiểm thử phi chức năng để đánh giá hiệu năng, bảo mật và khả năng tương thích của hệ thống. Kết quả cho thấy hệ thống hoạt động tốt nhưng cần tối ưu thêm tốc độ tải file lớn và cải thiện khả năng hiển thị trên một số trình duyệt.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Sau quá trình thực hiện đề tài “Kiểm thử ứng dụng gửi tài liệu email có giới hạn thời gian”, nhóm chúng em đã hoàn thành toàn bộ các bước của quy trình kiểm thử phần mềm, bao gồm thu thập và phân tích yêu cầu, thiết kế test case, thực hiện kiểm thử, ghi nhận kết quả và đánh giá mức độ ổn định của hệ thống. Trong quá trình này, chúng em đã áp dụng nhiều kỹ thuật kiểm thử khác nhau như kiểm thử hộp đen, kiểm thử hộp trắng, kiểm thử giá trị biên, kiểm thử phân vùng tương đương và kiểm thử phi chức năng. Các phương pháp này giúp chúng em không chỉ phát hiện được các lỗi logic và giao diện mà còn kiểm tra được tính toàn vẹn của dữ liệu, khả năng bảo mật và hiệu năng của hệ thống.

Kết quả kiểm thử cho thấy, nhìn chung, hệ thống hoạt động ổn định đối với các chức năng cơ bản như đăng nhập, đăng ký, xem lịch sử gửi và nhận file. Giao diện người dùng thân thiện, các thao tác thực hiện tương đối mượt mà và chính xác. Tuy nhiên, qua kiểm thử chi tiết, chúng em nhận thấy vẫn còn tồn tại một số hạn chế ở chức năng gửi và nhận file, đặc biệt là khi xử lý các tệp có dung lượng lớn, tốc độ truyền tải còn chậm, và trong một số trường hợp mất kết nối, hệ thống chưa có cơ chế phục hồi hoặc cảnh báo hiệu quả. Ngoài ra, trong kiểm thử phi chức năng, chúng em nhận thấy cần cải thiện thêm về hiệu suất tải trang, khả năng tương thích trình duyệt và tối ưu hóa bảo mật trong quá trình mã hóa, giải mã dữ liệu.

Đề tài này không chỉ giúp chúng em hiểu sâu hơn về quy trình kiểm thử phần mềm trong thực tế, mà còn rèn luyện kỹ năng làm việc nhóm, tư duy phản biện, khả năng lập kế hoạch và quản lý tiến độ công việc. Chúng em đã học được cách sử dụng các công cụ hỗ trợ kiểm thử, đồng thời biết cách thiết kế test case phù hợp cho từng loại chức năng. Việc được trực tiếp tham gia vào quá trình kiểm thử giúp chúng em hiểu rõ hơn tầm quan trọng của việc phát hiện lỗi sớm trong chu trình phát triển phần mềm, góp phần giảm chi phí sửa lỗi và nâng cao chất lượng sản phẩm.

Trong thời gian tới, nhóm chúng em định hướng tiếp tục hoàn thiện và mở rộng ứng dụng theo nhiều hướng khác nhau. Trước hết, chúng em sẽ tập trung nâng cao hiệu suất truyền tải file, tối ưu thuật toán xử lý để rút ngắn thời gian gửi và nhận dữ liệu, đặc biệt trong điều kiện mạng yếu hoặc bị gián đoạn. Tiếp theo, nhóm sẽ bổ sung tính năng cảnh báo thời hạn file nhằm tự động thông báo cho người dùng khi file sắp hết hạn truy cập hoặc bị xóa. Bên cạnh đó, chúng em sẽ tích hợp kiểm thử tự động để giảm khối lượng kiểm thử thủ công và tăng độ chính xác, thông qua việc sử dụng các công cụ như Selenium cho kiểm thử giao diện web và Postman cho kiểm thử API.

Ngoài ra, nhóm cũng hướng đến việc cải thiện giao diện người dùng (UI/UX) để hệ thống trở nên trực quan, dễ sử dụng hơn, đồng thời mở rộng khả năng tương thích với các thiết bị di động. Về mặt bảo mật, chúng em dự định tăng cường các cơ chế mã hóa và xác thực, cải tiến việc quản lý khóa mã hóa và triển khai các biện pháp tối ưu hơn. Tổng kết lại, đề tài này đã giúp chúng em có cái nhìn tổng quan, toàn diện và thực tế hơn về công tác kiểm thử phần mềm. Không chỉ dừng lại ở việc phát hiện và ghi nhận lỗi, chúng em còn hiểu rõ vai trò của kiểm thử trong việc đảm bảo chất lượng và độ tin cậy của sản phẩm trước khi phát hành. Trong tương lai, nhóm sẽ tiếp tục hoàn thiện hệ thống và mở rộng các bài kiểm thử chuyên sâu hơn, với mục tiêu mang lại một sản phẩm ổn định, an toàn và thân thiện với người dùng.