

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



AN TOÀN VÀ BẢO MẬT DỮ LIỆU
TRONG HỆ THỐNG THÔNG TIN

TÊN ĐỀ TÀI : SECURE FILE STORAGE APP

Nhóm 19:

Lưu Thành Đạt - 3118410073

Nguyễn Tân Đạt - 3118410076

Vũ Đình Cao - 3118410037

TP. HCM tháng 4/2020

Mục lục

Chương 1: Giới thiệu chung	5
I. Lý do chọn đề tài	5
II. Mục tiêu	5
III. Phạm vi	5
Chương 2: Tìm hiểu về AES	6
I. AES là gì ?	6
1. Giải thích	6
2. Lịch sử AES	6
II. Chế độ thao tác mã hóa ECB và CBC	7
1. ECB	7
2. CBC	8
III. Cơ chế hoạt động	9
1. Encryption	9
2. Decryption	13
Chương 3 : GIỚI THIỆU SECURE FILE STORAGE APP	17
I. Giới thiệu	17
II. Sơ đồ chức năng	17
III. Mô tả hệ thống	21
IV. Cách thức hoạt động của KDC	22
V. Usecase	24
VI. Giao diện phần mềm	28
Chương 4 : KẾT LUẬN	40
I - Các vấn đề đạt được	40
II- Hạn chế	40

Chương 1: Giới thiệu chung

Cùng với sự phát triển của công nghệ thông tin, công nghệ mạng máy tính và sự phát triển của mạng internet ngày càng phát triển đa dạng và phong phú. Các dịch vụ trên mạng đã thâm nhập vào hầu hết các lĩnh vực trong đời sống xã hội. Các thông tin trên Internet cũng đa dạng về nội dung và hình thức, trong đó có rất nhiều thông tin cần được bảo mật cao hơn bởi tính kinh tế, tính chính xác và tính tin cậy của nó.

Bên cạnh đó, các hình thức phá hoại mạng cũng trở nên tinh vi và phức tạp hơn. Do đó đối với mỗi hệ thống, nhiệm vụ bảo mật được đặt ra là hết sức quan trọng và cần thiết. Xuất phát từ những thực tế đó, chúng ta sẽ tìm hiểu về thuật toán mã hóa AES.

I. Lý do chọn đề tài

Trong những năm gần đây, Việt Nam ngày càng phát triển và nhất là về mặt công nghệ thông tin. Đặc biệt là về bảo mật thông tin, hầu như mọi người ai cũng từng sử dụng các bộ lưu trữ dữ liệu để lưu trữ những dữ liệu công việc, thông tin cá nhân,... Việc lưu trữ dữ liệu chưa được mã hóa sẽ rất nguy hiểm khi bị hacker tấn công đánh cắp thông tin. Như thế, chúng ta cần có một ứng dụng để mã hóa dữ liệu giúp cho việc bảo mật dữ liệu được tốt hơn. Chính vì thế nhóm tôi đã lựa chọn đề tài ứng dụng bảo mật dữ liệu áp dụng thuật toán mã hóa AES nhằm giúp cho việc bảo mật dữ liệu được tốt hơn.

II. Mục tiêu

Giúp chúng ta hiểu hơn về thuật toán mã hóa AES và cách thức bảo mật thông tin người dùng qua ứng dụng lưu trữ bảo mật file (Secure file storage app).

III. Phạm vi

Tìm hiểu về thuật toán mã hóa AES.

Chương 2: Tìm hiểu về AES

I. AES là gì ?

1. Giải thích

AES là viết tắt của Advanced Encryption Standard, chuẩn mã hóa dữ liệu rất phổ biến, dùng cho nhiều mục đích và được cả chính phủ Mỹ sử dụng để bảo vệ các dữ liệu tuyệt mật.

AES là kiểu mã hóa đối xứng dạng khối, nghĩa là mỗi khối văn bản có một kích thước nhất định (128 bit) được mã hóa, khác với mã hóa dạng chuỗi khi từng kí tự được mã hóa. Đối xứng nghĩa là khóa để mã hóa và giải mã đều là một.

2. Lịch sử AES

AES được phát triển từ cuối những năm 90s để thay thế chuẩn mã hóa trước đó là Data Encryption Standard (DES) do IBM tạo ra đầu những năm 70s. Nó được chính phủ Mỹ dùng trong năm 1977 nhưng sau đó có nhiều lỗ hổng dễ bị tấn công (brute force, phân tích mật mã khác biệt/tuyến tính) do dựa trên thuật toán 56 bit, nên không còn hữu ích nữa khi vi xử lý máy tính ngày càng mạnh hơn.

Vào năm 1998, DES trở thành 3DES hay còn gọi là Triple DES, dùng thuật toán DES để truyền thông điệp 3 lần liên tiếp với 3 khóa mã hóa khác nhau. 3DES khiến dữ liệu an toàn hơn trước kiểu tấn công brute force thời đó.

15 thuật toán được đề xuất thay thế DES, bắt đầu quy trình 5 năm của chính phủ Mỹ. AES được hai nhà mật mã học là Vincent Rijmen và Joan Daemen đề xuất, sau được gọi là “đơn Rijndael”.

AES là chuẩn mở vì khi đó chuẩn thực sự cũng chưa được xác định. Trong quá trình thiết kế, nó cũng nhận bình luận, góp ý. Nó được Viện tiêu chuẩn và kỹ thuật quốc gia Hoa Kỳ phát triển với mục tiêu dễ dùng cho cả phần cứng và phần mềm. Một số thay đổi về khóa và khối được thực hiện để tăng tính an toàn.

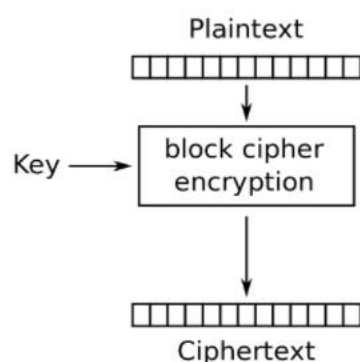
NSA cũng tham gia xem xét 15 bản đề xuất. Tới tháng 8/1999 chỉ còn 5 thuật toán (Rijndael, Serpent, RC6, Twofish và MARS). Các “ứng viên” được phân tích thêm về độ bảo mật, tính dễ sử dụng, bản quyền, tốc độ, độ chính xác khi mã hóa và giải mã.

Người chiến thắng sau cùng là Rijndael, sau đó được đưa lên cho chính phủ Mỹ vào năm 2002 và cả NSA cùng các tổ chức khác. Đến giờ, AES vẫn được dùng cho các tài liệu tuyệt mật, được cho là FIPS (Federal Information Processing Standard - tiêu chuẩn xử lý thông tin liên bang).

Sau đó nó được dùng trong khối tự nhân, là chuẩn mã hóa phổ biến nhất với mã hóa khóa đối xứng.

II. Chế độ thao tác mã hóa ECB và CBC

AES là một loại mã hóa khối: nó sẽ nhận 128 bits của văn bản sẽ được chuyển đổi để tạo thành 128 bits dữ liệu được mã hóa khác. Tuy nhiên 128 bits hay 16 ký tự sẽ không đủ để đáp ứng toàn bộ dữ liệu mà chúng ta muốn mã hóa, vậy làm thế nào để AES mã hóa toàn bộ các tài liệu chứa văn bản?

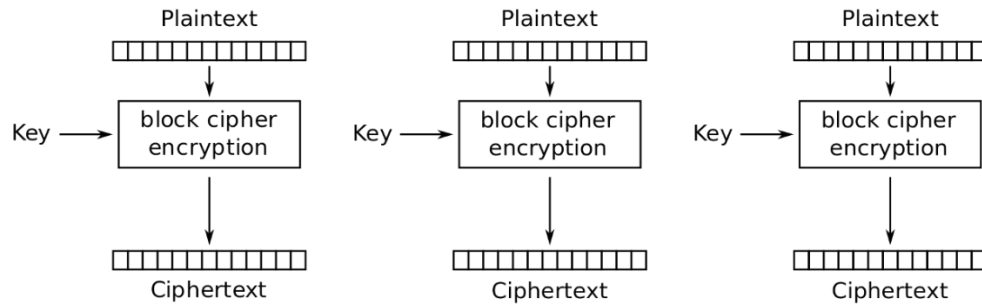


Bằng cách chia để trị, AES tách dữ liệu bạn muốn mã hóa thành các khối có kích thước 128 bit và thực hiện mã hóa chúng. Nếu dữ liệu đầu vào không phải là bội của 128, chúng ta phải mở rộng độ dài của tin nhắn để nó thỏa mãn điều kiện. Quá trình này được gọi là **padding**, và ở hình thức đơn giản nhất, nó chỉ cần thêm các bit 0 và cuối tin nhắn cho đến khi là bội của 128 bits.

Làm thế nào và theo thứ tự nào để mã hóa và hiển thị các khối dữ liệu dưới tên gọi **Các chế độ thao tác mã hóa**. Các chế độ này không phải là duy nhất cho AES, trên thực tế, các chế độ được đề cập trong bài viết này có thể được áp dụng cho hầu hết mọi loại mã hóa khối khác.

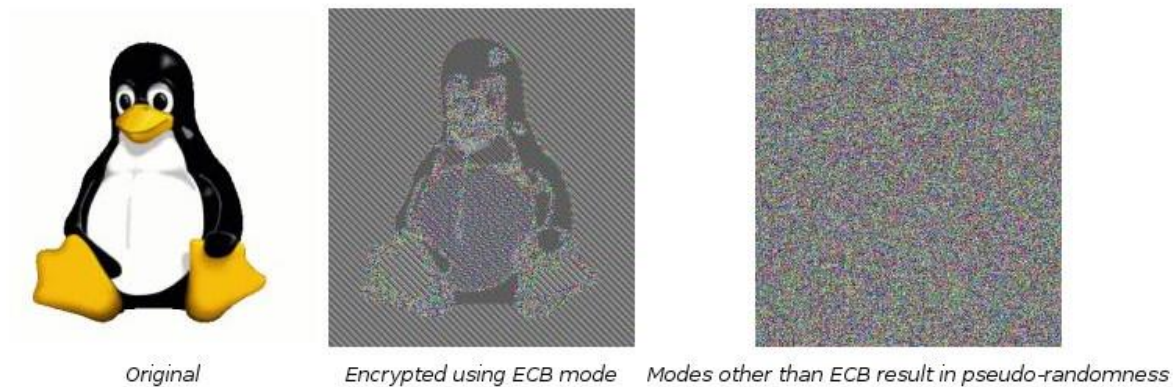
1. ECB

Loại đơn giản nhất, ECB, tự mã hóa từng khối và hiển thị các khối được mã hóa nối tiếp nhau.



Electronic Codebook (ECB) mode encryption

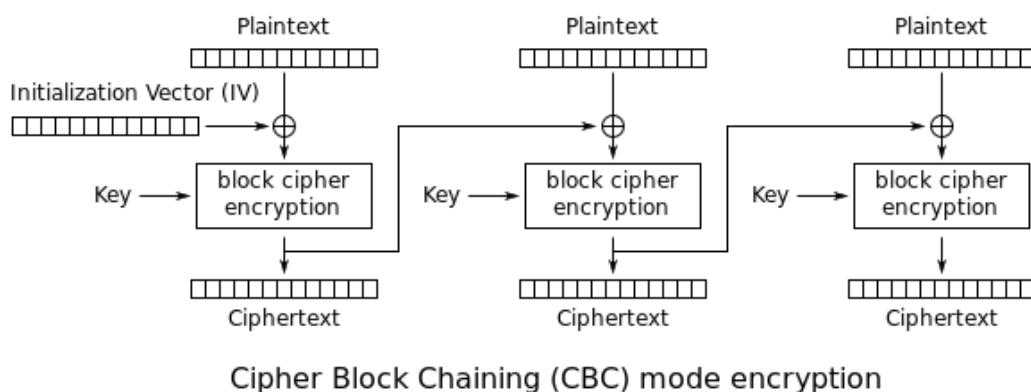
Tuy nhiên có một điểm yếu liên quan đến ECB, đó là nó thất bại trong việc che giấu các thông tin bị trùng lặp. Để minh họa cho điều này, hãy nhìn vào kết quả mã hóa ở chế độ ECB đối với chim cánh cụt Linux:



Mặc dù bị mất đi một số thông tin, hình ảnh sau khi được mã hóa vẫn có thể nhận ra và khác xa so với xuất hiện ngẫu nhiên. Đối với điều này, bạn nên tránh sử dụng chế độ **ECB** và thay vào đó sử dụng các chế độ phức tạp hơn, chẳng hạn như **CBC**.

2. CBC

Mã hóa AES theo **CBC** là kết quả của phép toán **xor** giữa khối văn bản thuần hiện tại với khối đã được mã hóa trước đó. Đối với khối văn bản thuần đầu tiên, vì không có khối đã được mã hóa trước đó, nên cần phải có một **vector khởi tạo (IV)** để thực hiện phép **xor**. **IV** này sẽ có cùng kích thước với các khối, 128 bits hoặc 16 ký tự.



Tạo một **IV** mới mỗi lần bạn mã hóa được xem như là một cách thực hành tốt để tạo ra sự ngẫu nhiên cho dữ liệu đầu ra. Tuy nhiên để giải mã ở chế độ **CBC**, chúng ta phải biết IV đã được sử dụng để mã hóa, vì vậy nó nên được đặt trước các khối được mã hóa và là một phần của dữ liệu mã hóa đầu ra. Với cách tiếp cận này, khi muốn giải mã, chúng ta biết rằng khối đầu tiên của tin nhắn được mã hóa chính là **IV** của nó.

Mặc dù có nhiều chế độ mã hóa khác nữa, **ECB** và **CBC** vẫn là những loại mô phạm nhất vì chúng cho thấy rằng việc chọn một chế độ hoạt động cũng quan trọng như việc chọn thuật toán mã hóa. Các chế độ hoạt động phổ biến khác có thể kể đến như **OFB**, **CTR**, hay **CFB**. Chúng sẽ không được đề cập trong bài viết này nhưng hãy tìm hiểu thêm về chúng nếu bạn cảm thấy tò mò.

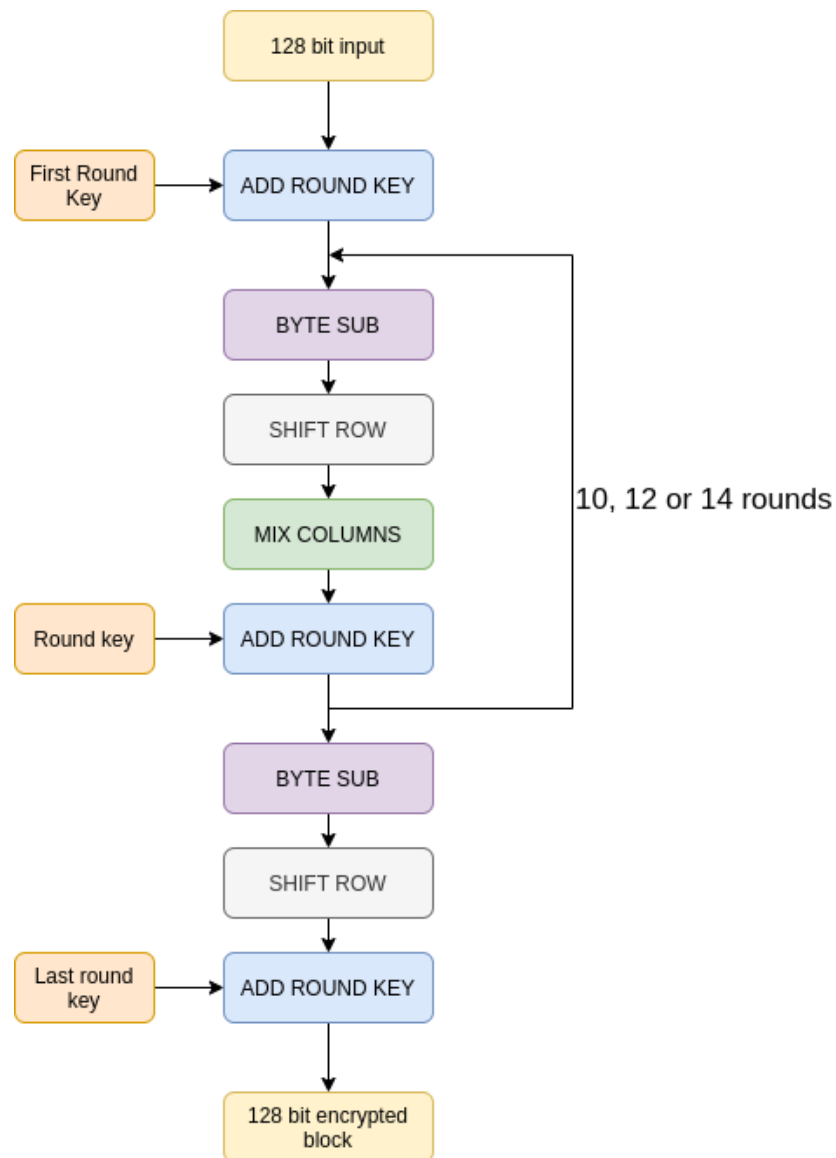
III. Cơ chế hoạt động

1. Encryption

Cho đến lúc này, chúng tôi đã đề cập đến những cách AES có thể mã hóa các dữ liệu đầu vào lớn bằng cách chia chúng thành các khối 128 bits, nhưng chúng tôi vẫn chưa đề cập đến việc những dữ liệu này thực sự được mã hóa bằng cách nào: những gì diễn ra bên trong AES khi nó mã hóa dữ liệu.

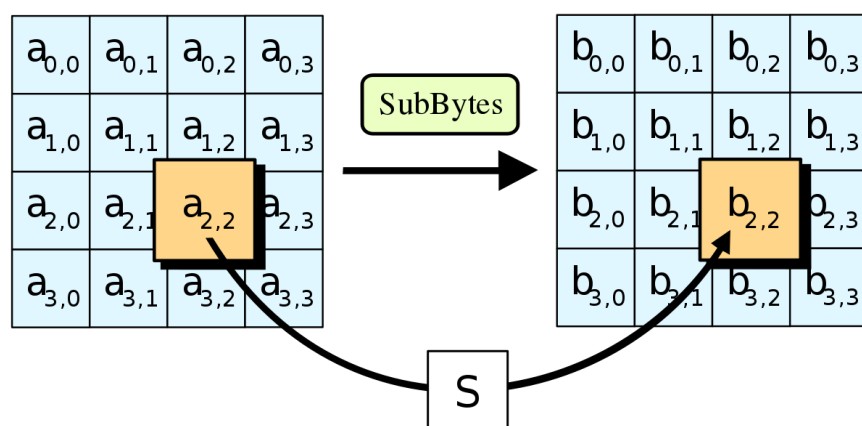
Bạn có thể nhớ lại rằng trước đó chúng tôi đã đề cập rằng AES có thể có khóa thuộc ba loại kích thước: 128, 192, 256 bits và khóa càng dài thì mã hóa càng mạnh. Để hiểu được điều này, chúng ta phải biết rằng khi AES mã hóa, nếu áp dụng cùng một thuật toán mã hóa giống nhau cho một số lượng các vòng nhất định, mỗi vòng sử dụng một khóa phụ khác nhau được tạo ra từ khóa ban đầu trong một quá trình được gọi là **key expansion**. Kích thước của khóa càng dài thì AES càng có thể tạo ra nhiều khóa phụ và do đó có thể thực hiện nhiều vòng hơn cho thuật toán mã hóa, dẫn đến sự mã hóa mạnh hơn.

Thuật toán này bao gồm 4 thao tác: **Byte Sub**, **Shift Rows**, **Mix Columns** và **Add Round Key**. Để trực quan hóa, khối văn bản đầu vào 128 bits sẽ được biểu diễn bằng một ma trận 4 x 4, mỗi vị trí đại diện cho 8 bits từ khối đầu vào.



Byte Sub

Byte Sub, giống như tên của nó, sẽ thay thế mỗi ô 8 bits bởi một 8 bits khác. Vậy 8 bits là gì? Những 8 bits này sẽ được chọn từ một bảng tra cứu đã được xác định trước, và bất biến qua các vòng. Vậy thực hiện như thế nào? Bốn bits đầu tiên của mỗi ô sẽ quyết định dòng và 4 bits cuối xác định cột, và đó cũng là vị trí trong bảng tra cứu nơi chúng ta có thể tìm thấy 8 bits sẽ thay thế ô đó.



Và đây chính là bảng tra cứu **Byte Sub**:

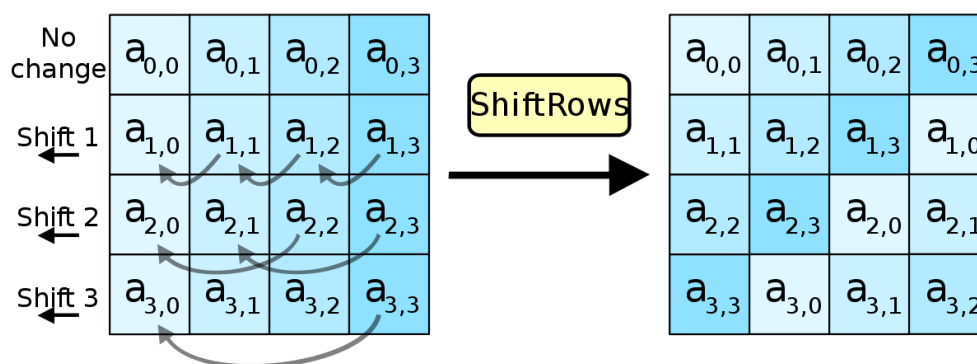
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Các giá trị trong bảng tra cứu được biểu diễn theo hệ thập lục phân, mỗi chữ số tương ứng với 4 bits. Vì vậy, nếu trong mỗi ô có hai chữ số, thì mỗi ô chiếm 8 bits, giống với ma trận đầu vào của chúng ta.

Hãy thực hiện một ví dụ nhanh sau đây. Hãy tưởng tượng tôi có một ô có giá trị là 10100101, tức A5. Nó sẽ được thay thế cho ô ở dòng A hay 10 và cột 10 trong bảng tra cứu, 06 hay 00000110.

Shift Rows

Đơn giản hơn **Byte Sub**, shift rows cũng giống như tên của nó, dịch mỗi dòng của ma trận sang bên trái. Vậy có tất cả bao nhiêu vị trí? Dòng đầu tiên sẽ không bị dịch, dòng thứ hai sẽ dịch 1 vị trí, dòng thứ ba dịch 2 vị trí và dòng thứ tư cũng là dòng cuối cùng sẽ dịch 3 vị trí.



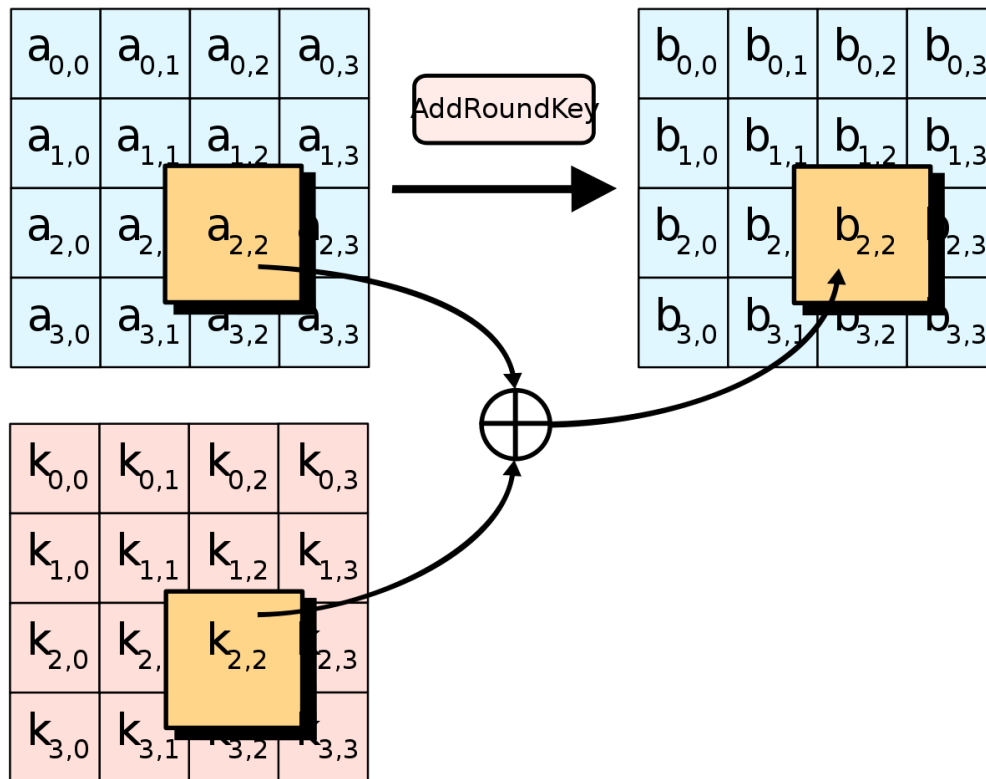
Mix Columns

Trong Mix Columns, chúng ta sẽ thực hiện một phép nhân ma trận giữa ma trận hiện tại và ma trận đã được xác định trước, bất biến qua các vòng. Tuy nhiên đó sẽ là một phép nhân ma trận phức tạp hơn một tí, vì phép **tổng** được thay thế bởi phép **xor** và phép **nhân** được thay thế bởi phép **and**.

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \times \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

Add Round Key

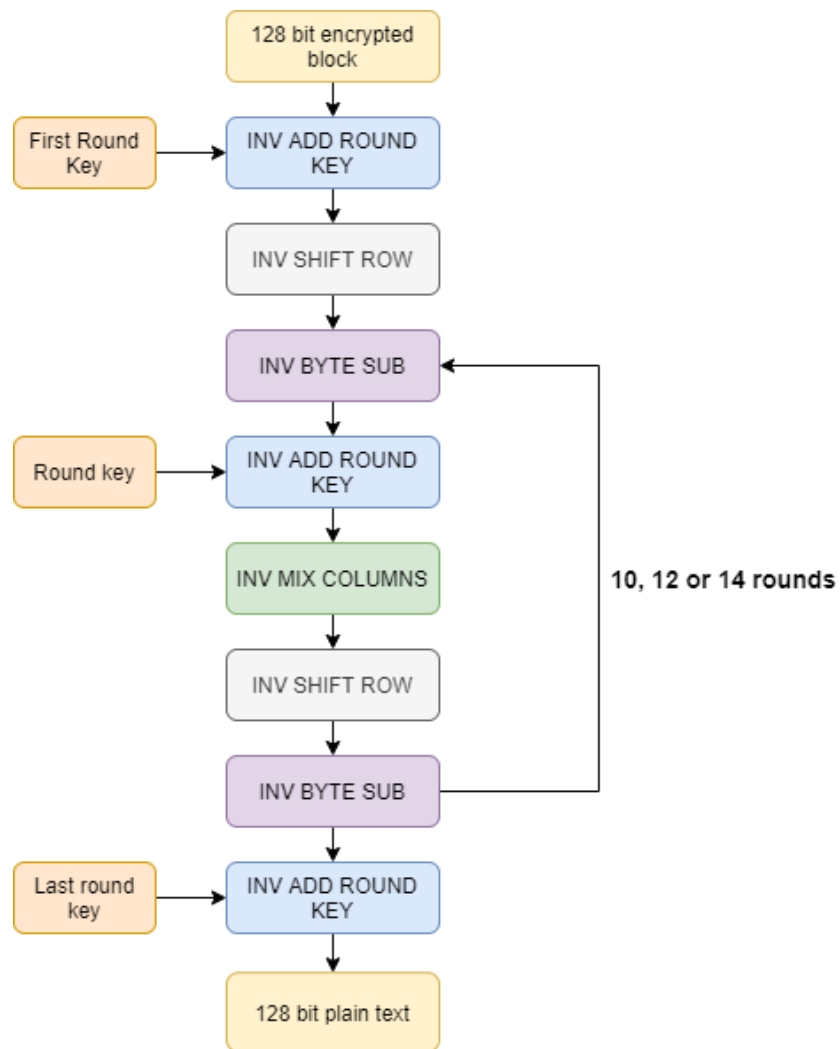
Cho đến lúc này, chúng ta vẫn chưa thực hiện mã hóa thông tin. Tất cả các phép biến đổi trên đều có thể được hoàn tác bởi bất cứ ai, vì chúng ta vẫn chưa đưa khóa vào phương trình. Vì mỗi khóa dài 128 bits, nên chúng ta có thể biểu diễn nó giống như cách biểu diễn khối dữ liệu đầu vào, trong một ma trận. Bây giờ chúng ta sẽ thực hiện phép **xor** giữa mỗi khối của khối dữ liệu đầu vào tương ứng với khối từ ma trận khóa.



Và đó chính là cách mã hóa AES hoạt động! Bây giờ chúng ta đã có thể mã hóa, tôi chắc chắn rằng chúng ta cũng muốn khôi phục lại thông tin đó vào một lúc nào đó: hãy nói về sự giải mã.

2. Decryption

Nếu bạn đã hiểu quá trình mã hóa, bạn sẽ không gặp quá nhiều khó khăn trong quá trình giải mã. Để hoàn tác sự mã hóa, giống như suy nghĩ ban đầu của một người, chúng ta phải đảo ngược quá trình mã hóa. Chúng ta sẽ thực hiện các thao tác tương tự, chỉ theo thứ tự đảo ngược lại. Vì vậy, trái ngược với mã hóa, đây chính là thuật toán giải mã:



Bạn có thể nhận thấy rằng sơ đồ này trông khá giống với thuật toán mã hóa, và trên thực tế là chúng giống nhau! Tuy nhiên nếu bạn chú ý kỹ hơn, bạn sẽ nhận thấy rằng tất cả các thao tác này đều bắt đầu với **INV**, tượng trưng cho các thao tác nghịch đảo. Ví dụ, **Inverse Mix Columns** chính là thao tác nghịch đảo của Mix Columns. Điều này có nghĩa là gì? Nó có nghĩa là để nghịch đảo thao tác Mix Columns, chúng ta phải áp dụng **Inverse Mix Column**.

Inverse Add Round Key

là nghịch đảo của chính nó: nếu bạn áp dụng Add Round Key 2 lần, bạn sẽ nhận lại được những gì lúc bạn bắt đầu. Vì tính chất này, để hoàn tác Add Round Key, chúng ta phải áp dụng nó thêm một lần nữa. Đối với điều này, **Inverse Add Round Key** và **Add Round Key** là cùng một thao tác.

Inverse Shift Row

Nếu bạn nhớ lại, shift rows chỉ dịch mỗi dòng một số lượng vị trí nhất định sang bên trái. Do đó để nghịch đảo, chúng ta phải di chuyển mỗi dòng một số lượng vị trí tương tự nhưng về phía bên phải. Thật đơn giản!

Inverse Sub Bytes

về bản chất là thao tác giống với Sub Bytes: nó sẽ nhận mỗi khối từ ma trận và hoán đổi nó với một khối khác từ một ma trận được xác định trước. Sự khác biệt duy nhất giữa **Inverse Sub Bytes** và **Sub Bytes** ở bảng sau. Đây chính là bảng dành cho **Inverse Sub Bytes**:

	y															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Inverse Mix Columns

Nếu bạn biết về đại số, bạn sẽ nhớ rằng nếu chúng ta nhân một ma trận **C** với nghịch đảo của nó, **C⁻¹**, thì chúng ta được kết quả là ma trận đơn vị. Nếu bạn không quen thuộc với các khái niệm đại số này, hãy nghĩ ma trận đơn vị là 1: nếu chúng ta nhân 5 với nghịch đảo của nó, 1/5, ta được 1.

$$C \cdot C^{-1} = I_n$$

Do đó, gọi ma trận dữ liệu của chúng ta là **X**. Theo Mix Columns, ta đặt **X = X C**. Để hoàn tác phép nhân này ta phải nhân nó với nghịch đảo của **C**, như sau:

$$X \cdot C \cdot C^{-1} = X \cdot I_n = X$$

Chúng ta đã biểu diễn ma trận C theo Mix Columns, vì vậy ở đây chúng ta sẽ biểu diễn ma trận nghịch đảo của C:

$$\begin{array}{ccc} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} & \xleftrightarrow{\text{Inverse}} & \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \\ C & & C^{-1} \end{array}$$

Chương 3 : GIỚI THIỆU SECURE FILE STORAGE APP

I. Giới thiệu

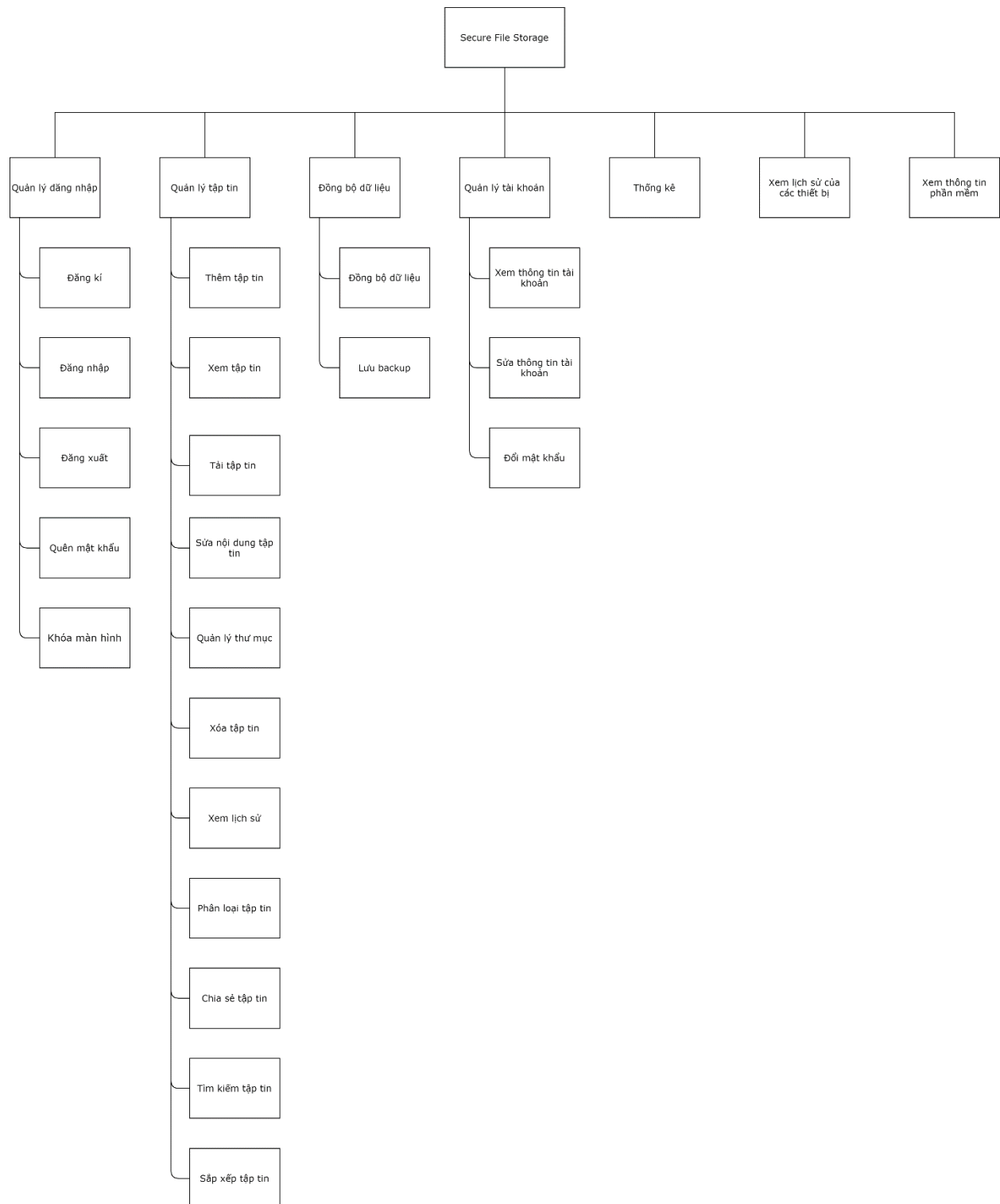
Secure File Storage là phần mềm lưu trữ và bảo vệ các tập tin quan trọng của người dùng một cách đơn giản và nhanh chóng. Hỗ trợ bảo mật với khả năng mã hóa các dữ liệu, cung cấp tài khoản riêng biệt giúp bảo vệ cũng như tạo một môi trường lưu trữ an toàn cho các tập tin của người dùng.

Secure File Storage cung cấp cho người dùng một kết nối bảo mật, là môi trường lý tưởng để bảo vệ các dữ liệu quan trọng của người dùng mà không ai có thể xâm phạm. Chỉ cần cài đặt trực tiếp phần mềm này vào máy tính. Tất cả những dữ liệu sẽ được mã hóa. Phần mềm này cung cấp một phương pháp đơn giản mà hiệu quả để bảo vệ những dữ liệu riêng tư và nhạy cảm của người dùng.

Phần mềm cho phép người dùng có thể thêm tập tin muốn mã hóa bằng thao tác vô cùng đơn giản chỉ với việc kéo thả tập tin vào phần mềm. Ngoài ra phần mềm cho phép người dùng thực hiện các thao tác với tập tin như: xem nội dung tập tin, sửa nội dung, xóa tập tin, tải tập tin, phân loại, tìm kiếm, sắp xếp, khóa màn hình, xem lịch sử các thiết bị đăng nhập, điều này giúp người dùng có thể biết được dữ liệu có bị truy cập trái phép hay không. Không chỉ có vậy, phần mềm còn hỗ trợ đồng bộ dữ liệu lên máy chủ, giúp cho người dùng có thể truy cập dữ liệu ở bất kì đâu bất kì thiết bị nào.

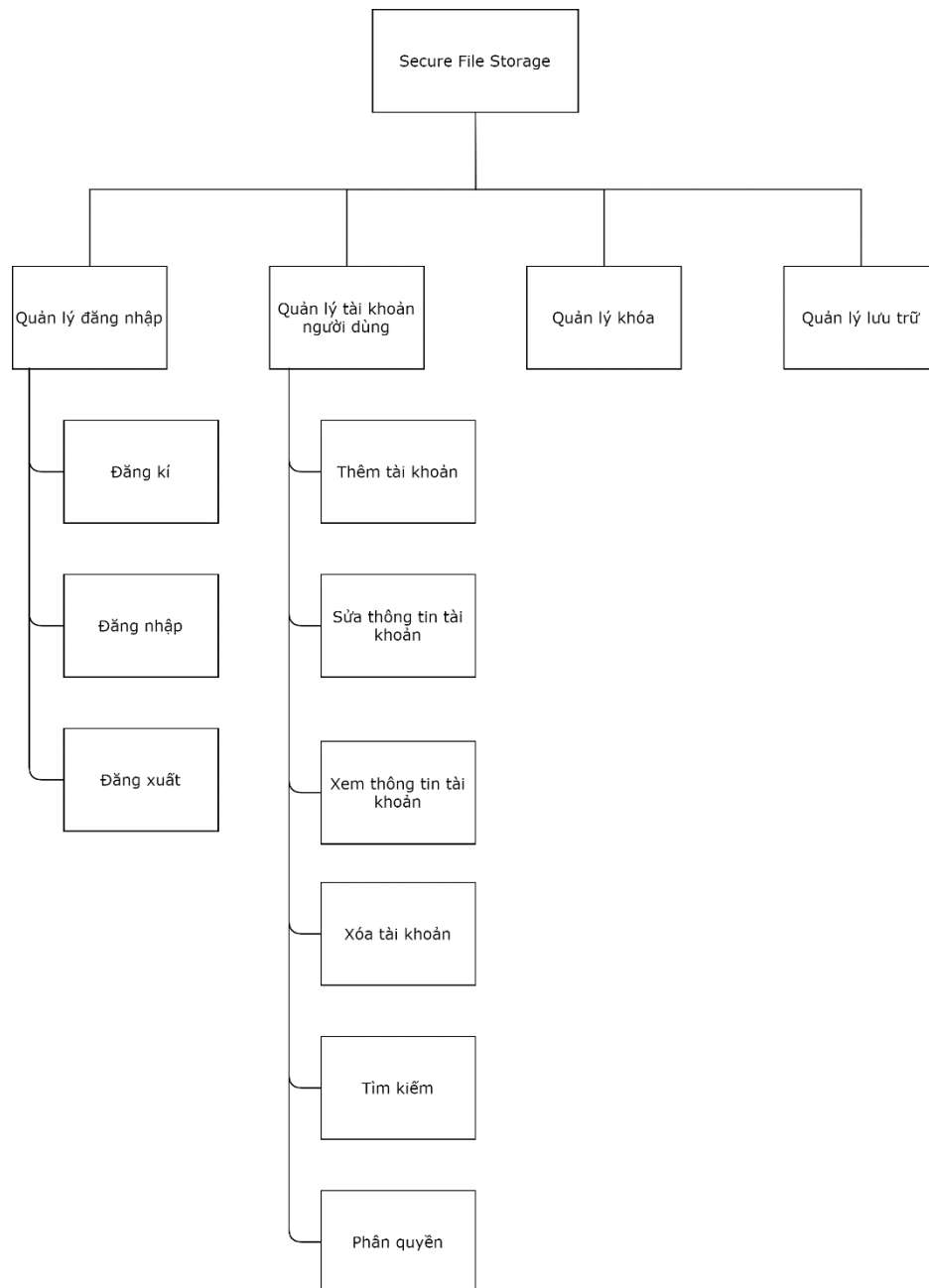
II. Sơ đồ chức năng

User class 1 – User



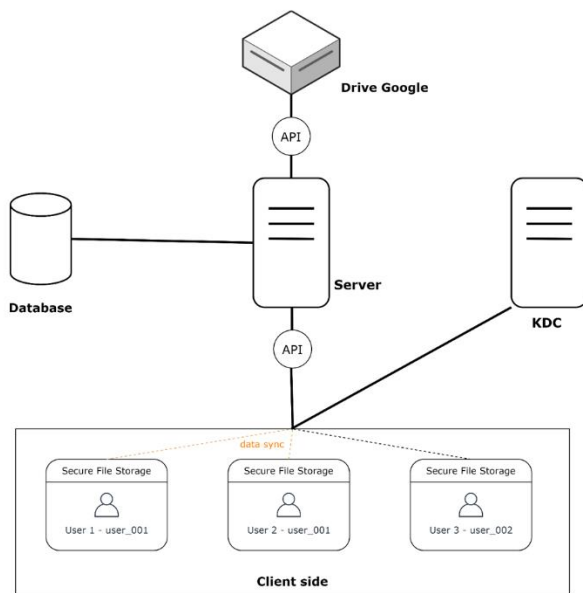
Hình 1: Sơ đồ chức năng của User Class

User Class 2 – Admin



Hình 2: Sơ đồ chức năng của Admin Class

III. Mô tả hệ thống



Front-end:  Electron

Back-end:  django

Database:  SQLite

Storage Server:  Google Drive API

Đối tượng tham gia

Client side

- Mô tả: Phần mềm phía người sử dụng
- Công nghệ sử dụng: **Electronjs**
- Vai trò:
 - Giúp người dùng sử dụng các tính năng của phần mềm (đã nêu trên).
 - Giao tiếp với server side thông qua API của <server> và <KDC>.

Server

- Mô tả: Máy chủ chính thực hiện các thao tác đồng bộ dữ liệu, quản lý dữ liệu người sử dụng và xử lý việc đăng nhập, đăng ký của người sử dụng.
- Công nghệ sử dụng: **Django**.
- Vai trò:
 - Xử lý các request của người dùng về việc đồng bộ dữ liệu (các thao tác về tập tin như: thêm, sửa, xóa tập tin và các thao tác về thư mục).

- Quản lý các dữ liệu người sử dụng bao gồm: thông tin tài khoản, lịch sử truy cập của các thiết bị. Xử lý các thao tác đăng nhập, đăng kí của người sử dụng.

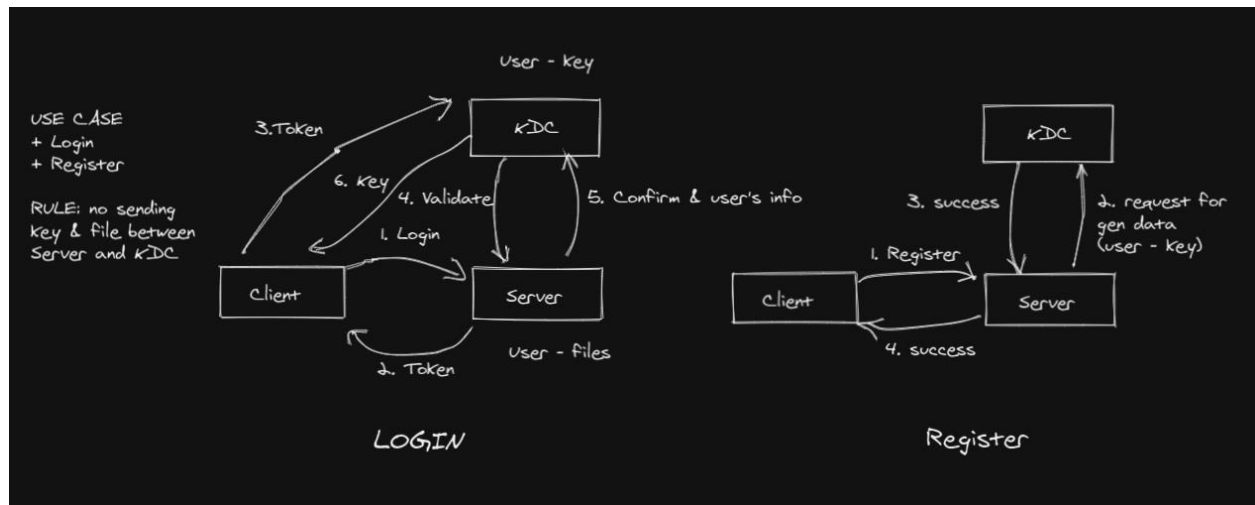
Database

- Mô tả: Cơ sở dữ liệu lưu trữ thông tin của người dùng.
- Công nghệ sử dụng: **SQLite**.
- Vai trò:
 - Lưu trữ thông tin về tài khoản và lịch sử truy cập các thiết bị.

Storage Server

- Mô tả: Hệ thống lưu trữ tập tin mã hóa
- Công nghệ sử dụng: Lưu trữ ở Google Drive và giao tiếp thông qua **Google Drive API**.
- Vai trò:
 - Lưu trữ các tập tin mã hóa và cấu trúc thư mục dùng để đồng bộ dữ liệu trên các thiết bị khác nhau cùng một tài khoản.

IV. Cách thức hoạt động của KDC



Về mặt dữ liệu:

KDC lưu trữ thông tin:

- Thông tin người dùng
- Thông tin khóa

Server lưu trữ thông tin:

- Thông tin người dùng
- Tập tin mã hóa

Cách thức hoạt động:

a. Login

1. Người dùng thực hiện thao tác đăng nhập và sẽ gửi request yêu cầu xác nhận đăng nhập đến server.
2. Server xử lý request từ người dùng, hiện thực xác thực người dùng. Và nếu xác thực thành công, server sẽ gửi thông tin token (bao gồm access token và refresh token) đến phía người dùng.
3. Người dùng sẽ gửi request (kèm access token) đến KDC để lấy được thông tin key.
4. KDC sẽ hỏi server để xác nhận token có hợp lệ hay không và để biết được token của người dùng nào.
5. Server xác nhận và trả về thông tin người dùng nếu token còn hợp lệ.
6. KDC lấy thông tin người dùng được gửi từ server và trả về key cho người dùng.

b. Đăng kí

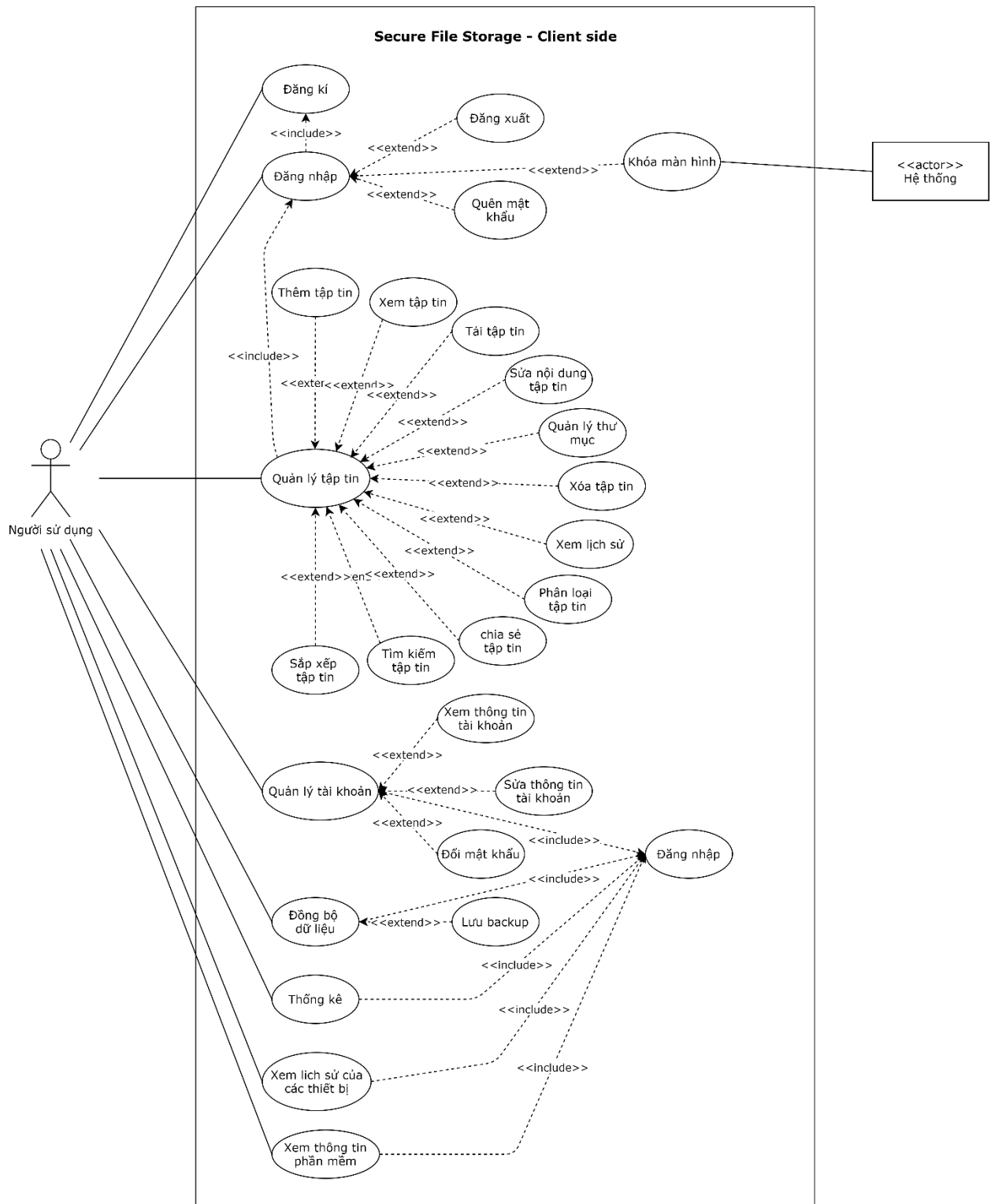
1. Người dùng đăng kí tài khoản mới và sẽ gửi request đến server
2. Server xác nhận tính hợp lệ của thông tin và tạo tài khoản, cấp phát vùng lưu trữ tập tin mới cho tài khoản mới và gửi yêu cầu tạo khóa mới đến KDC.
3. KDC nhận yêu cầu từ server và tạo khóa cho tài khoản mới và phản hồi cho server khi hoàn tất.
4. Server hoàn tất việc đăng kí và gửi phản hồi thành công đến người dùng.

V. Usecase

Người sử dụng

Bảng danh sách Usecase

UC ID	Tên Usecase
UC01	Đăng kí
UC02	Đăng nhập
UC03	Đăng xuất
UC04	Quên mật khẩu
UC05	Khóa màn hình
UC06	Thêm tập tin
UC07	Xem tập tin
UC08	Tải tập tin
UC09	Sửa nội dung tập tin
UC10	Quản lý thư mục
UC11	Xóa tập tin
UC12	Xem lịch sử
UC13	Phân loại tập tin
UC14	Chia sẻ tập tin
UC15	Tìm kiếm tập tin
UC16	Sắp xếp tập tin
UC17	Xem thông tin tài khoản
UC18	Sửa thông tin tài khoản
UC19	Đổi mật khẩu
UC20	Đồng bộ dữ liệu
UC21	Lưu backup
UC22	Thống kê
UC23	Xem lịch sử của các thiết bị
UC24	Xem thông tin phần mềm

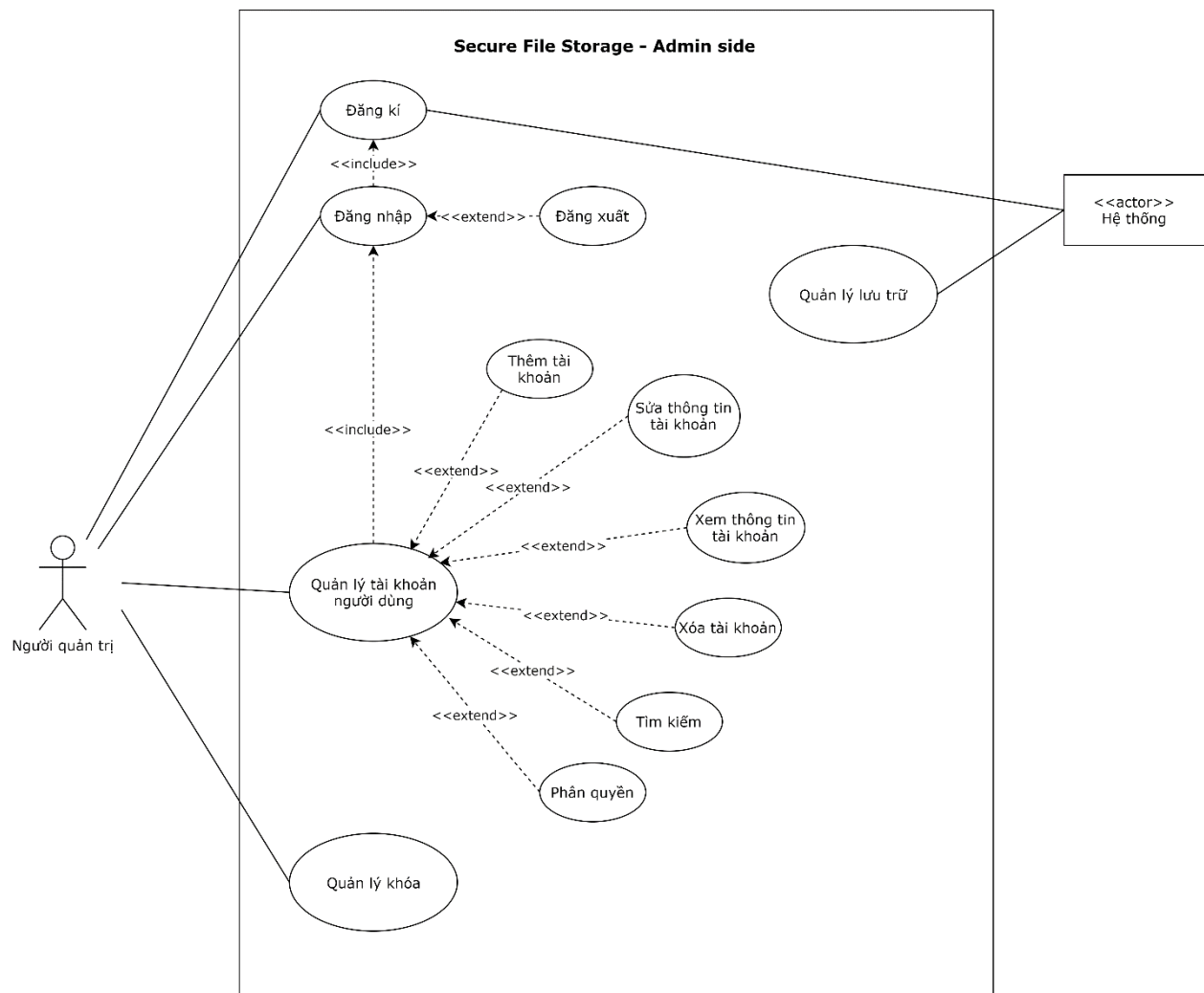


Hình 1: User use-case diagram

Người quản trị

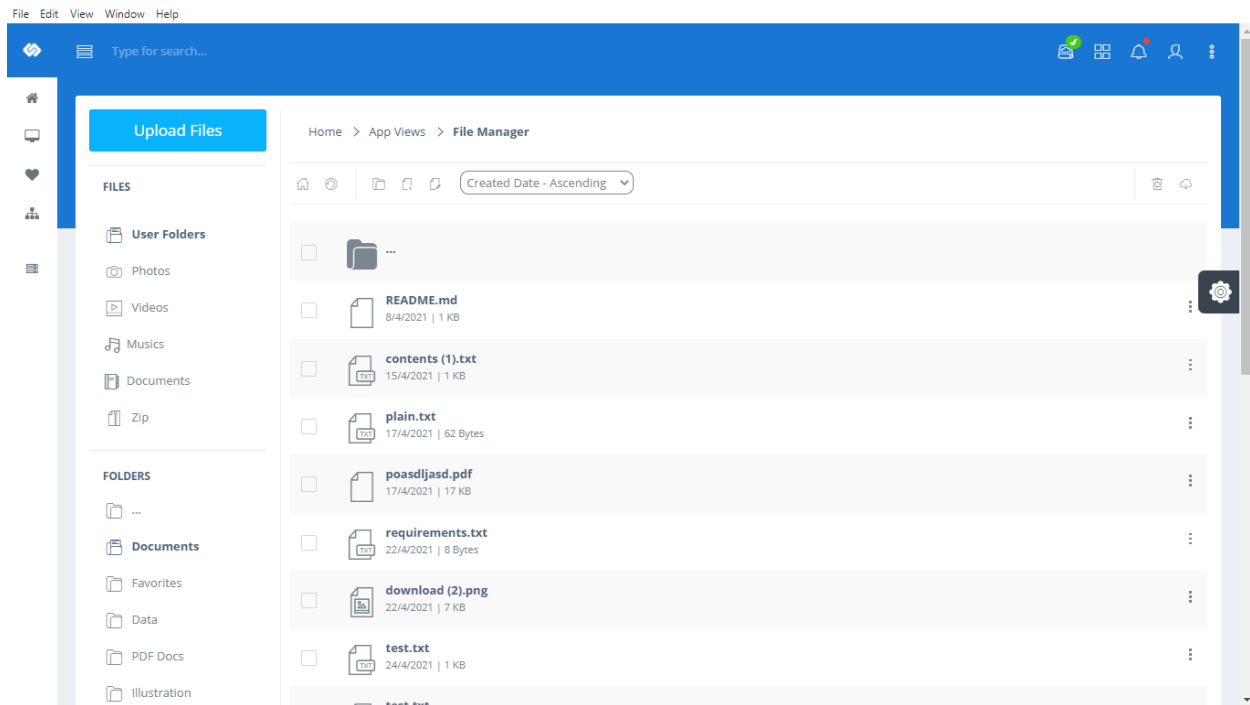
Bảng danh sách Usecase

UC ID	Tên Usecase
UC25	Đăng kí
UC26	Đăng nhập
UC27	Đăng xuất
UC28	Thêm tài khoản
UC29	Sửa thông tin tài khoản
UC30	Xem thông tin tài khoản
UC31	Xóa tài khoản
UC32	Tìm kiếm
UC33	Phân quyền
UC34	Quản lý khóa
UC35	Quản lý lưu trữ

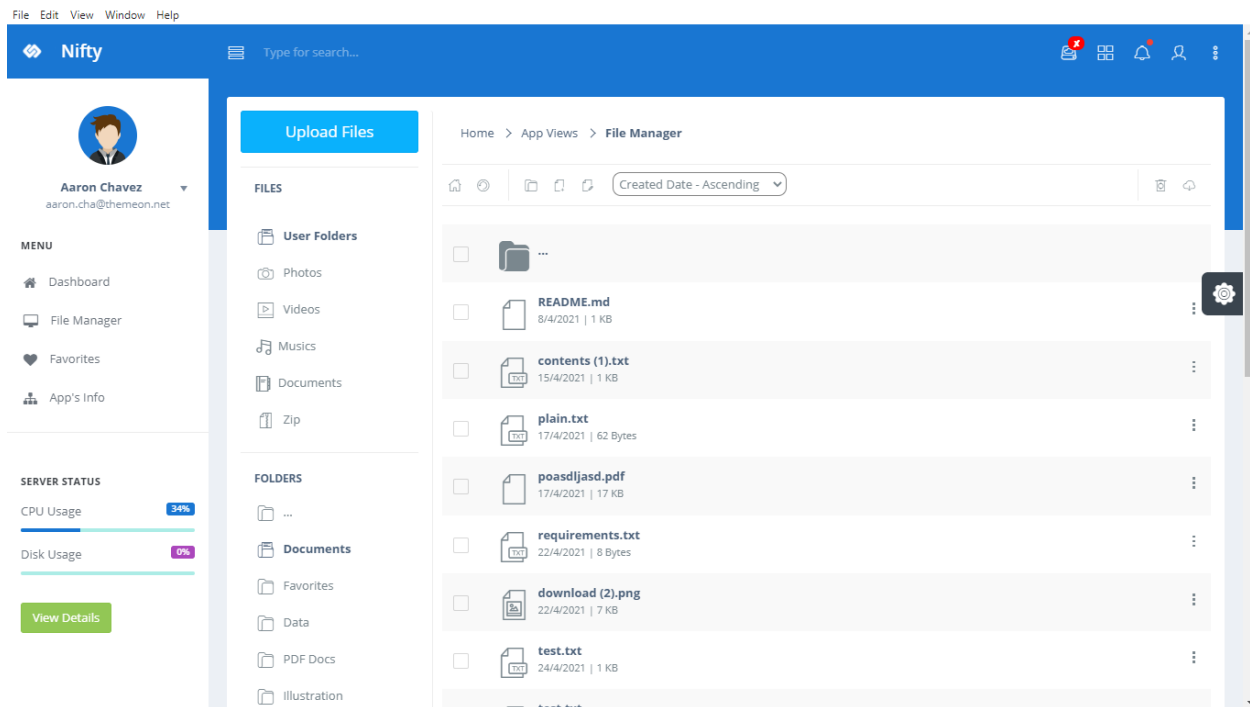


Hình 2: Admin use-case diagram

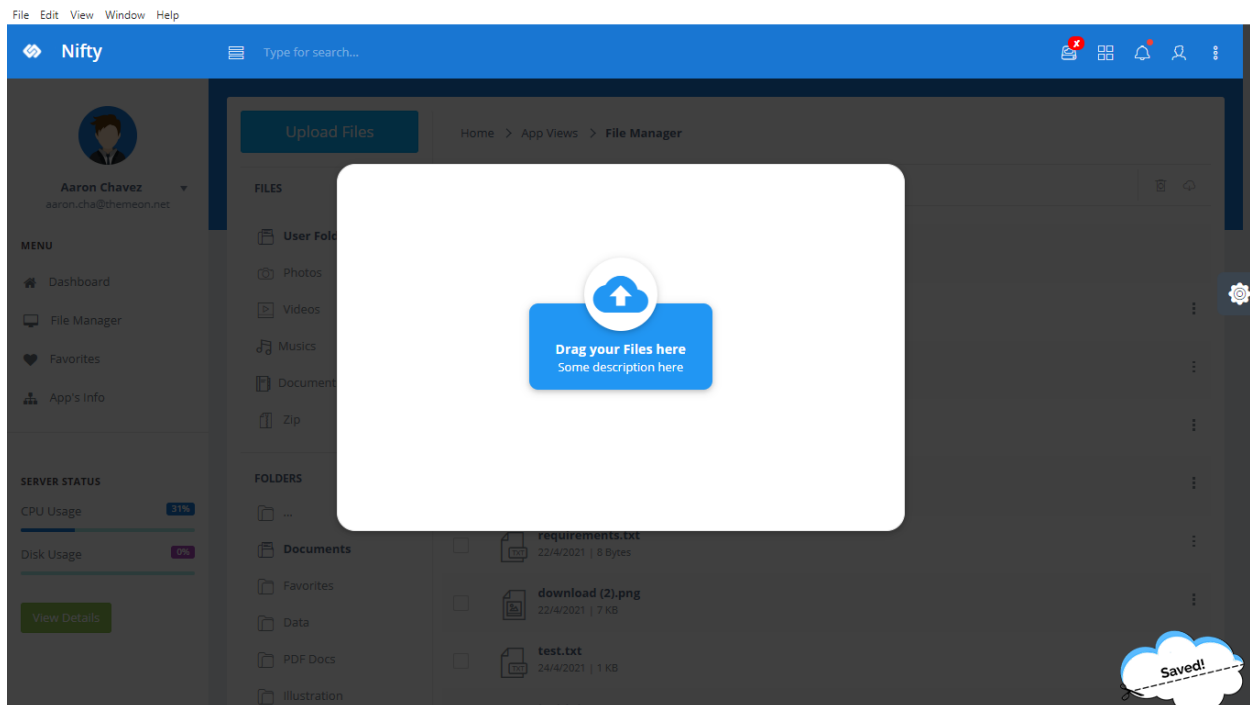
VI. Giao diện phần mềm



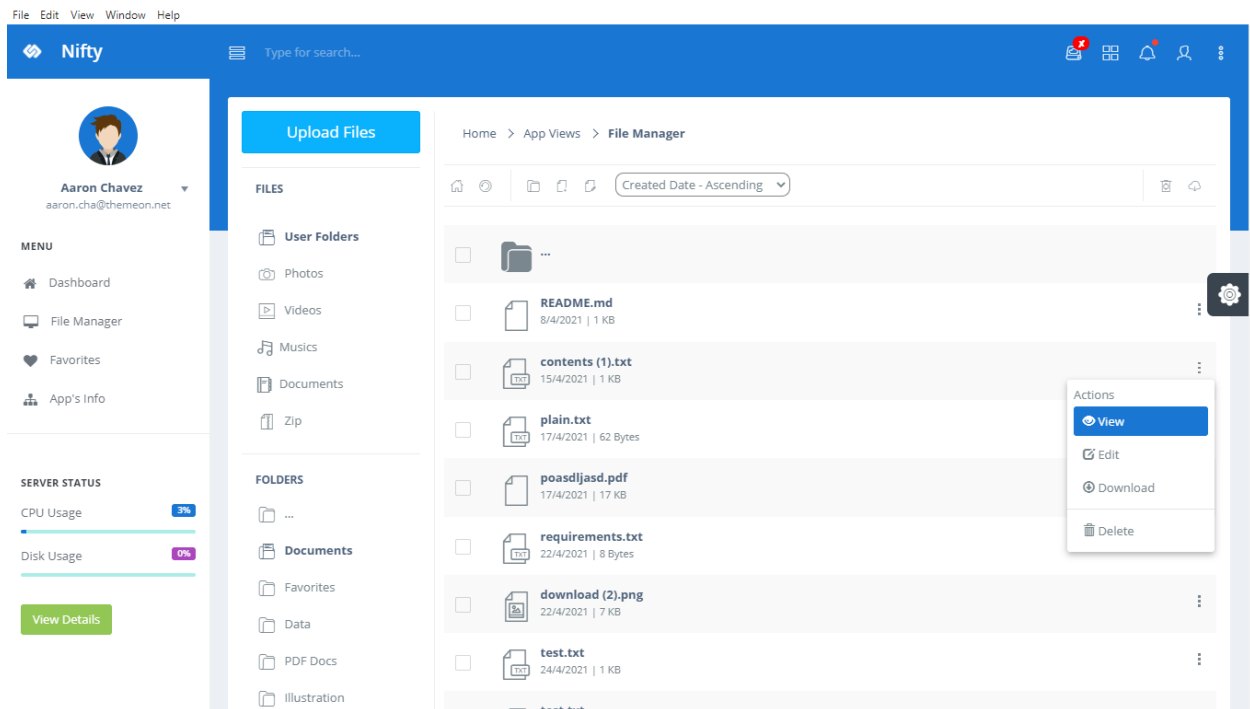
Hình 1: Giao diện chính



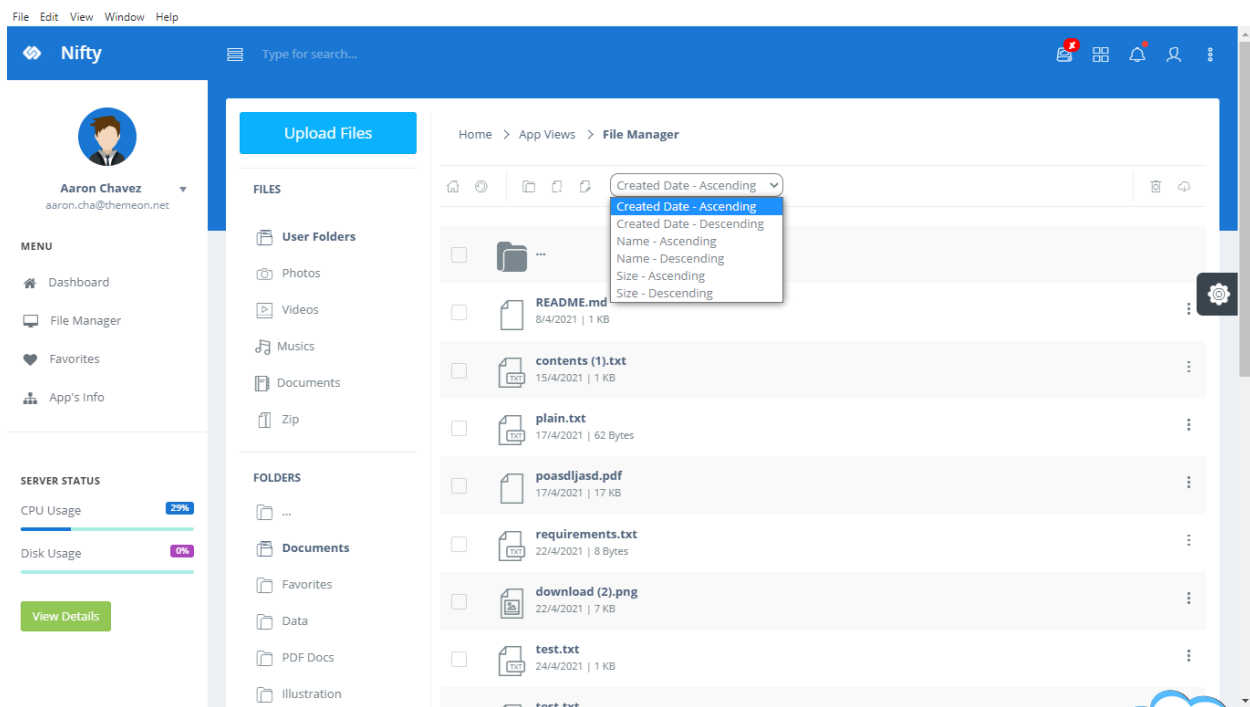
Hình 2: Giao diện chính (+sidebar)



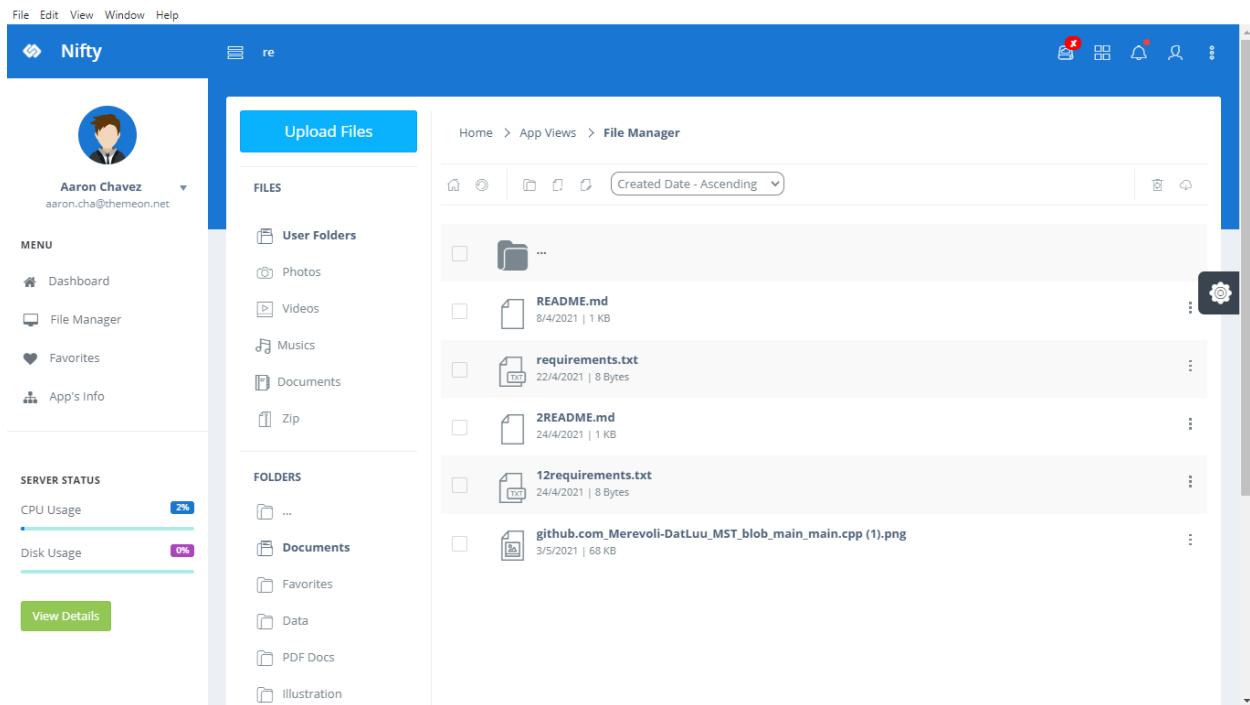
Hình 3: Giao diện thêm tập tin



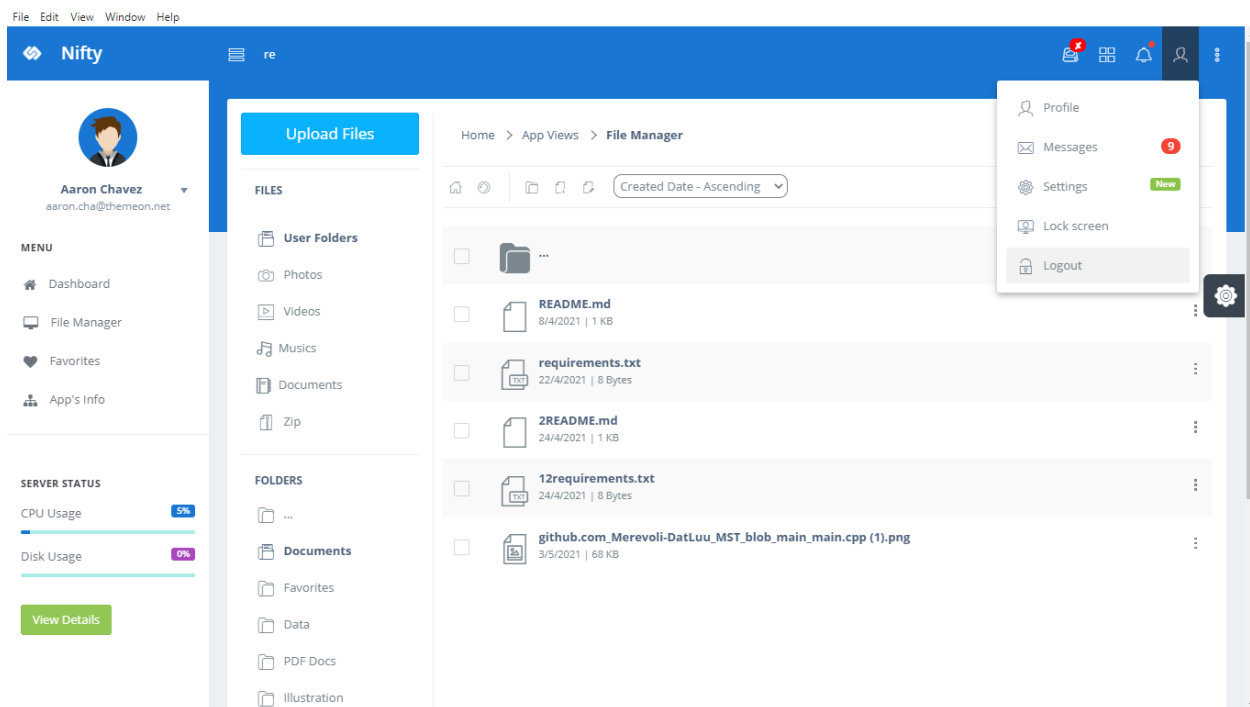
Hình 4: Giao diện more actions



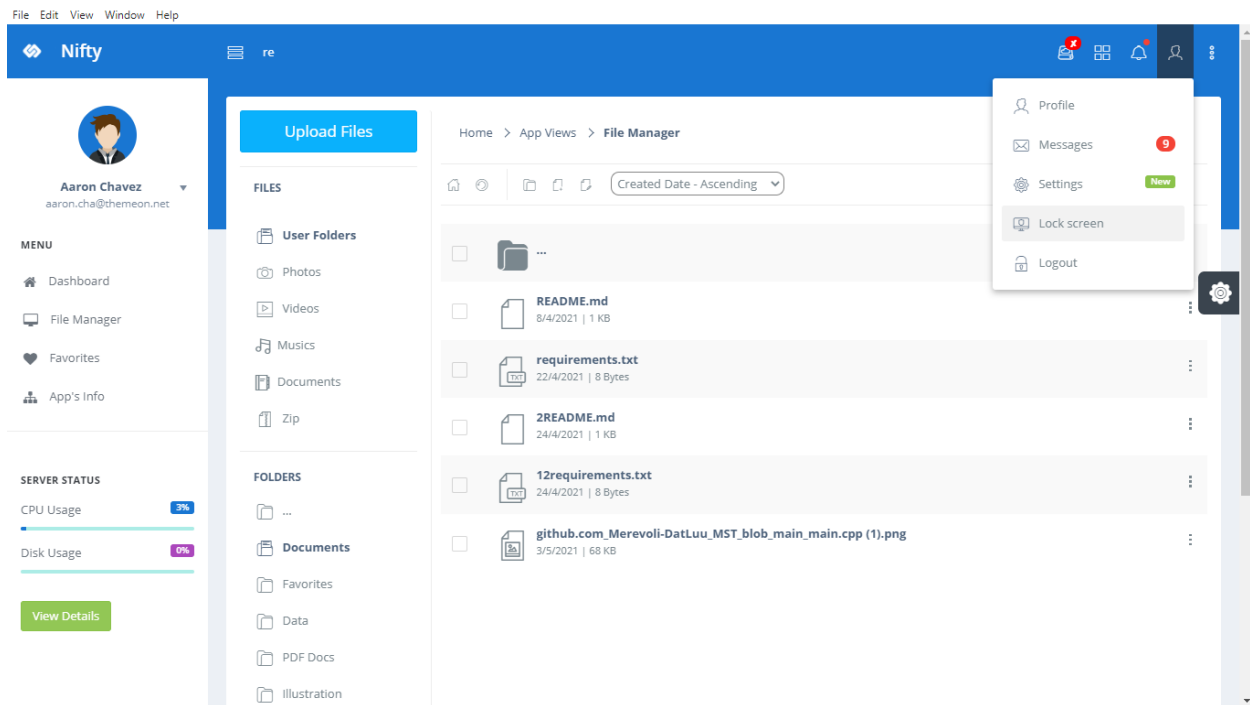
Hình 5: Giao diện tính năng sắp xếp



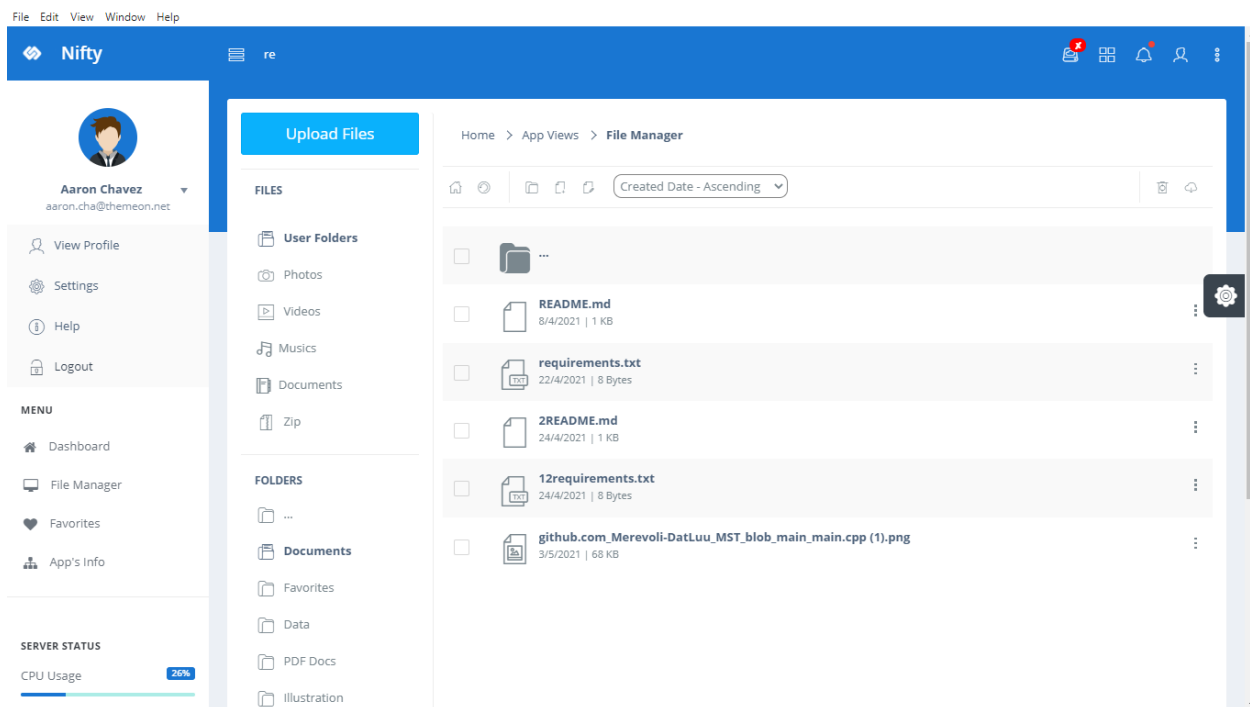
Hình 6: Giao diện tính năng tìm kiếm



Hình 7: Giao diện kích hoạt nút logout



Hình 8: Giao diện kích hoạt nút lockscreen



Hình 9: Giao diện more options ở sidebar

File Edit View Window Help

Account Login

Sign In to your account

Email




Password

☐ Remember me

Sign In

[Forgot password ?](#) [Create a new account](#)

Login with

Hình 10: Giao diện login

File Edit View Window Help

Create a New Account

Come join the Nifty community! Let's set up your account.

E-mail

First Name Password




Last Name Retype password

☐ I agree with the [Terms and Conditions](#)

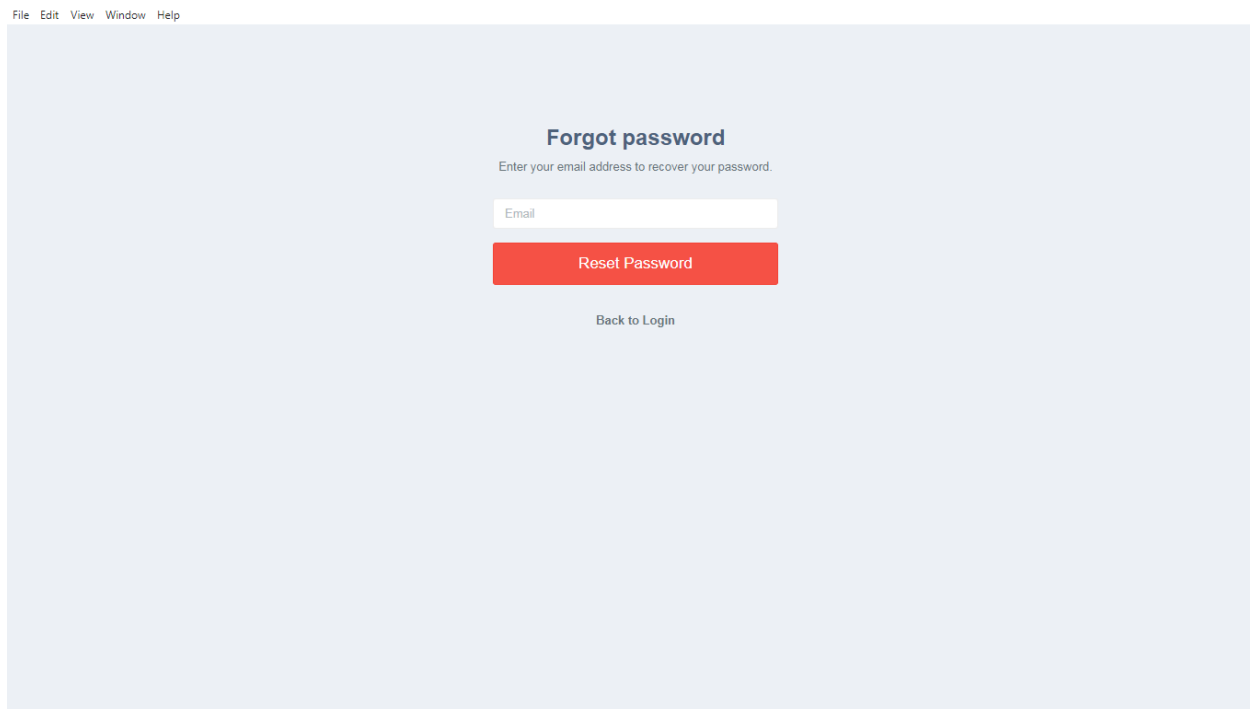
Register

[Already have an account ? Sign In](#)

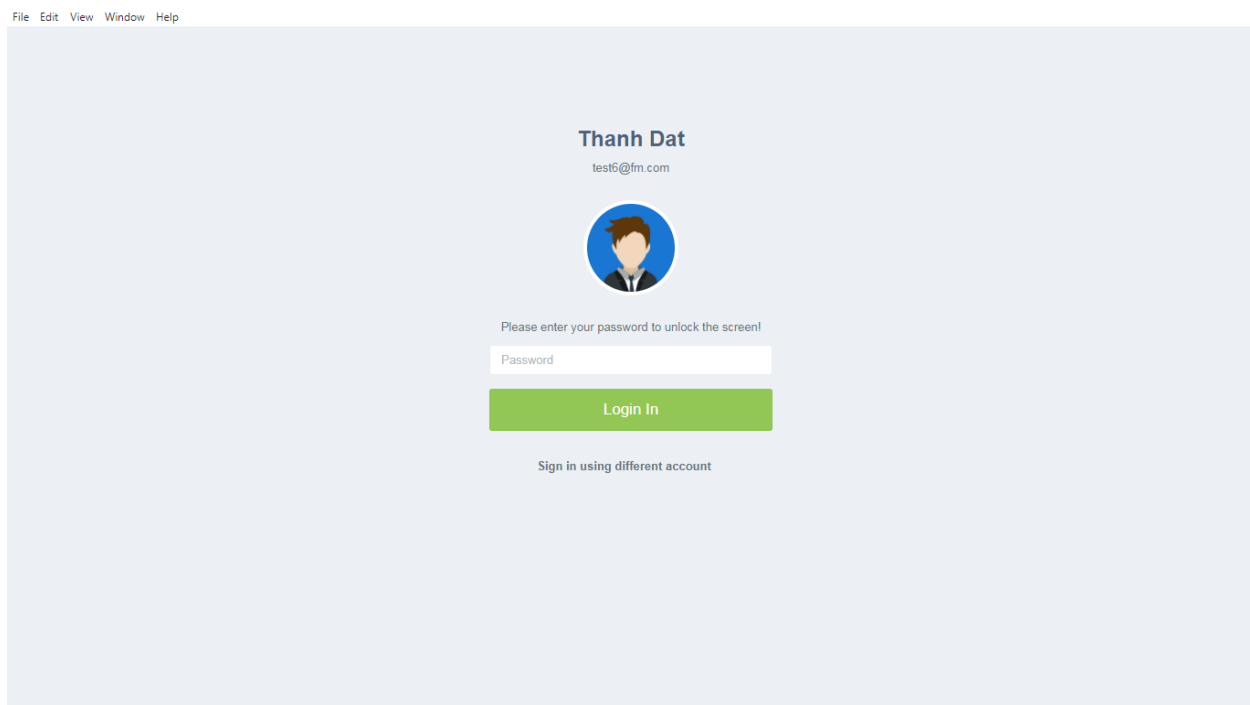
Sign Up with

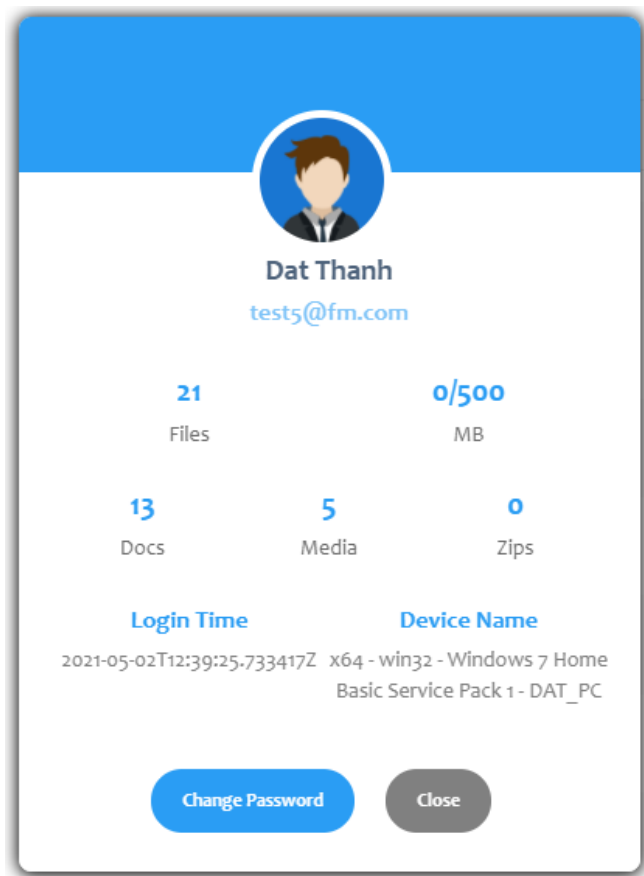
Hình 11: Giao diện Register



Hình 12: Giao diện Forgot Password



Hình 13: Giao diện lockscreen



Hình 14: Thông tin tài khoản

Change Password

Old Password

old password

New Password

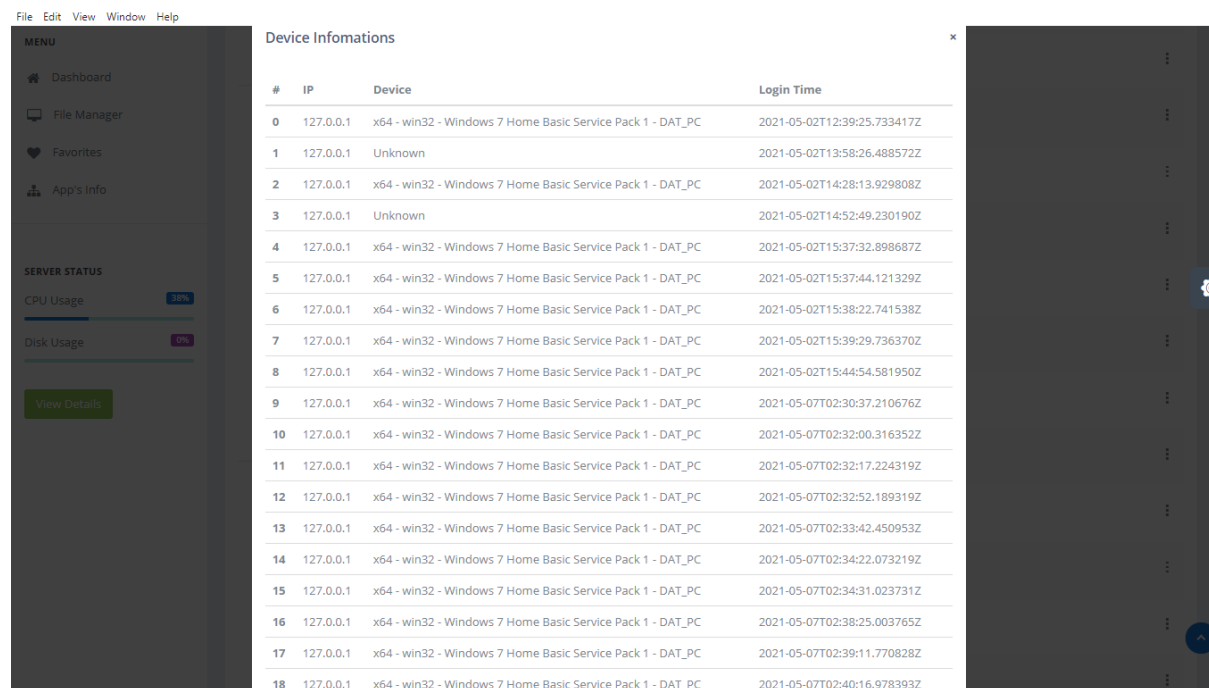
new password

New Password

retype new password

Change

Hình 15: Đổi mật khẩu



Hình 16: Lịch sử truy cập của các thiết bị

Django administration

WELCOME, ADMIN@FM.COM / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > User > Users

AUTHENTICATION AND AUTHORIZATION

Groups + Add

USER

User historys + Add

Users + Add

Select user to change

ADD USER +

Action: Go 0 of 13 selected

<input type="checkbox"/>	EMAIL	FIRST NAME	LAST NAME
<input type="checkbox"/>	test12@fm.com	Thanh	Dat
<input type="checkbox"/>	test11@fm.com	Dat	Thanh
<input type="checkbox"/>	test10@fm.com	Dat	Thanh
<input type="checkbox"/>	test9@fm.com	Dat	Thanh
<input type="checkbox"/>	test8@fm.com	Dat	Thanh
<input type="checkbox"/>	test7@fm.com	Dat	Thanh
<input type="checkbox"/>	test6@fm.com	Dat	Thanh
<input type="checkbox"/>	test5@fm.com	Thanh	Dat
<input type="checkbox"/>	test4@fm.com	Dat	Thanh
<input type="checkbox"/>	test3@fm.com	Dat	Thanh
<input type="checkbox"/>	test2@fm.com	Dat	Thanh
<input type="checkbox"/>	test1@fm.com	Dat	Thanh
<input type="checkbox"/>	admin@fm.com		

Hình 17: Trang Quản lý tài khoản

AUTHENTICATION AND AUTHORIZATION

Groups + Add

USER

User historys + Add

Users + Add

☐ Superuser status
Designates that this user has all permissions without explicitly assigning them.

Groups:

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

admin | log entry | Can add log entry

admin | log entry | Can change log entry

admin | log entry | Can delete log entry

admin | log entry | Can view log entry

auth | group | Can add group

auth | group | Can change group

auth | group | Can delete group

auth | group | Can view group

auth | permission | Can add permission

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

First name:

Last name:

☐ Staff status
Designates whether the user can log into this admin site.

☒ Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

Hình 18: Xem thông tin tài khoản

Django administration

WELCOME, ADMIN@FM.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home - User - User historys

AUTHENTICATION AND AUTHORIZATION

Groups + Add

USER

User historys + Add

Users + Add

Select user history to change

ADD USER HISTORY +

Action: ----- Go 0 of 88 selected

	EMAIL	DEVICE INFO	IP	LOGIN TIME
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 14, 2021, 1:42 a.m.
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 13, 2021, 6:55 a.m.
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 13, 2021, 6:33 a.m.
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 13, 2021, 6:25 a.m.
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 13, 2021, 5:50 a.m.
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 13, 2021, 5:42 a.m.
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 13, 2021, 5:09 a.m.
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 13, 2021, 2:49 a.m.
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 13, 2021, 2:38 a.m.
<input type="checkbox"/>	test5@fm.com	x64 - win32 - Windows 7 Home Basic Service Pack 1 - DAT_PC	127.0.0.1	May 13, 2021, 2:11 a.m.
<input type="checkbox"/>	test10@fm.com	Unknown	127.0.0.1	May 11, 2021, 4:59 a.m.
<input type="checkbox"/>	test11@fm.com	Unknown	127.0.0.1	May 11, 2021, 4:46 a.m.
<input type="checkbox"/>	test11@fm.com	Unknown	127.0.0.1	May 11, 2021, 4:46 a.m.

Hình 19: Quản lý lịch sử thiết bị truy cập

Chương 4 : KẾT LUẬN

I - Các vấn đề đạt được

- Theo yêu cầu đặt ra ban đầu thì cho đến thời điểm hiện tại, đồ án đã đạt được các nội dung sau:

- Tìm hiểu các mối nguy hiểm đến máy chủ web.
- Các biện pháp phòng chống cho máy chủ.
- Nâng cao kiến thức về các lỗ hổng máy chủ.

II- Hạn chế

Trong quá trình tìm hiểu có rất nhiều tài liệu nhóm tìm kiếm tuy có mục đích là giống nhau song lại có phương pháp khác nhau hoàn toàn. Nhóm đã cố gắng tìm hiểu thêm về chúng nhưng không khỏi có nhiều sai sót.