# 111

# ชื่อโครงงานภาษาไทย

# ชื่อโครงงานภาษาอังกฤษ

# ชื่อ-นามสกุลผู้พัฒนา รหัสประจำตัว....

# ชื่อ-นามสกุลผู้พัฒนา รหัสประจำตัว....

# หมู่เรียน.....

# โครงงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา 720XXX

# โครงงานด้านเทคโนโลยีสารสนเทศ 2

# สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี

# มหาวิทยาลัยราชภัฏนครปฐม

# ภาคเรียนที่ 1 ปีการศึกษา 2567

# บทที่ 1

# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน บริการฝากและแชร์ไฟล์ส่วนใหญ่มักบังคับให้ผู้ใช้ต้องผ่านกระบวนการที่ซับซ้อนและใช้เวลา ตั้งแต่การสมัครสมาชิก การยืนยันตัวตน ไปจนถึงการลงชื่อเข้าใช้ ทุกครั้งที่ต้องการอัปโหลดหรือดาวน์โหลดไฟล์ ขั้นตอนเหล่านี้ก่อให้เกิดความยุ่งยากและเป็นอุปสรรคต่อการใช้งานที่ต้องการความรวดเร็ว โดยเฉพาะในกรณีที่ผู้ใช้มีความประสงค์เพียงเพื่อแชร์ไฟล์ชั่วคราวหรือใช้งานเพียงไม่กี่ครั้ง ซึ่งไม่ก่อให้เกิดความคุ้มค่าในการสร้างบัญชีผู้ใช้ถาวรเพียงเพื่อทำธุรกรรมเล็กน้อยดังกล่าว

ด้วยเหตุนี้ โครงงานจึงมุ่งเน้นการพัฒนาระบบฝากและแชร์ไฟล์ที่ ไม่จำเป็นต้องมีการลงทะเบียนหรือเข้าสู่ระบบ อันเป็นการลดขั้นตอนอันซับซ้อนและเพิ่มความสะดวกแก่ผู้ใช้งาน การออกแบบลักษณะนี้มีความสำคัญในหลายประการ กล่าวคือ ประการแรก เป็นการลดอุปสรรคในการเข้าถึงบริการ ผู้ใช้สามารถใช้งานได้โดยไม่ต้องผ่านขั้นตอนการสมัครหรือจดจำรหัสผ่าน ประการที่สอง เป็นการเพิ่มความสะดวกและความรวดเร็วในการใช้งาน ตอบสนองต่อพฤติกรรมผู้ใช้ในยุคดิจิทัลที่ให้ความสำคัญกับความฉับไว ประการที่สาม เป็นการเสริมสร้างความเป็นส่วนตัว เนื่องจากผู้ใช้ไม่จำเป็นต้องเปิดเผยข้อมูลส่วนบุคคลใด ๆ ต่อระบบ จึงลดความเสี่ยงต่อการรั่วไหลของข้อมูล และประการที่สี่ เป็นการเปิดโอกาสให้กลุ่มผู้ใช้ที่มีข้อจำกัดด้านเทคโนโลยี หรือไม่ต้องการผูกพันกับแพลตฟอร์มใดแพลตฟอร์มหนึ่ง สามารถเข้าถึงและใช้งานได้อย่างเท่าเทียม

จากปัญหาที่กล่าวมาข้างต้น ผู้จัดทำโครงงานจึงได้พัฒนาระบบให้บริการฝากและแชร์ไฟล์ออนไลน์ที่มุ่งเน้นความรวดเร็ว สะดวก และไม่จำเป็นต้องสมัครสมาชิกหรือเข้าสู่ระบบก่อนใช้งาน ระบบถูกออกแบบให้รองรับการอัปโหลดไฟล์จากหลายแพลตฟอร์ม ได้แก่ เว็บเบราว์เซอร์ การส่งข้อความส่วนตัวในโปรแกรมแชทที่ได้รับความนิยม

## 1.2 แนวคิดในการแก้ไขปัญหา

จากปัญหาที่กล่าวมาข้างต้น ผู้จัดทำโครงงานได้นำเทคโนโลยีสารสนเทศและเทคโนโลยีเครือข่ายเข้ามาประยุกต์ใช้ในการพัฒนาระบบฝากและแชร์ไฟล์ออนไลน์ โดยมุ่งเน้นให้ผู้ใช้สามารถ

อัปโหลดและแชร์ไฟล์ได้ทันทีโดยไม่ต้องสมัครสมาชิกหรือเข้าสู่ระบบ ระบบนี้ใช้ เว็บเซิร์ฟเวอร์และฐานข้อมูลแบบคลาวด์ในการจัดเก็บข้อมูลเซสชันและไฟล์แบบชั่วคราว พร้อมทั้งใช้ เทคโนโลยีการเข้ารหัสเพื่อรักษาความปลอดภัยของข้อมูล

นอกจากนี้ ระบบยังรองรับการอัปโหลดไฟล์จากหลายแพลตฟอร์ม เช่น เว็บเบราว์เซอร์และโปรแกรมแชทที่ได้รับความนิยม โดยผู้ใช้จะได้รับ รหัสเข้าถึงสำหรับดาวน์โหลดไฟล์ ทำให้สามารถแชร์ไฟล์ได้อย่างรวดเร็วและปลอดภัย มีการจัดการเซสชันแบบอัตโนมัติและการตรวจสอบชนิดของไฟล์ เพื่อช่วยลดความเสี่ยงของไฟล์อันตราย และเพิ่มประสิทธิภาพในการจัดการข้อมูลโดยไม่ต้องพึ่งพาการจัดเก็บแบบถาวร

ด้วยแนวคิดนี้ โครงงานจึงสามารถแก้ไขปัญหาความยุ่งยากของบริการฝากไฟล์ทั่วไป พร้อมทั้งให้ผู้ใช้เข้าถึงไฟล์ได้สะดวก รวดเร็ว และปลอดภัยผ่านเทคโนโลยีสารสนเทศและระบบเครือข่ายอย่างเต็มประสิทธิภาพ

## 1.3 วัตถุประสงค์ของระบบ

โครงงานนี้มีวัตถุประสงค์หลักในการพัฒนาระบบฝากและแชร์ไฟล์ออนไลน์ที่สามารถใช้งานได้ทันทีโดยไม่ต้องสมัครสมาชิกหรือเข้าสู่ระบบ เพื่อลดความซับซ้อนและเพิ่มความสะดวกในการใช้งาน โดยมีวัตถุประสงค์เฉพาะดังนี้

1.3.1 เพื่อให้ผู้ใช้สามารถอัปโหลดและแชร์ไฟล์ได้อย่างรวดเร็ว

1.3.2 เพื่อรักษาความปลอดภัยของข้อมูลผู้ใช้

1.3.3 เพื่อรองรับการใช้งานจากหลายแพลตฟอร์ม

## 1.4 ขอบเขตการศึกษา

โครงงานนี้มีขอบเขตการศึกษาเพื่อกำหนดแนวทางการพัฒนาระบบฝากและแชร์ไฟล์ออนไลน์อย่างชัดเจน โดยแบ่งออกเป็นหัวข้อดังนี้

1.4.1 ขอบเขตของระบบ

1.4.1.1 ผู้ดูแลระบบ

ก) จัดการและควบคุมเซิร์ฟเวอร์และฐานข้อมูล Cloud MongoDB

ข) ตรวจสอบสถานะเซสชันและไฟล์ที่อัปโหลด

ค) ดูแลความปลอดภัยของระบบและการเข้ารหัสไฟล์

* + - 1. ผู้ใช้งานระบบ

ก) ผู้ใช้ทั่วไปสามารถอัปโหลดไฟล์และรับรหัสเข้าถึงเพื่อแชร์ไฟล์ได้ทันทีโดยไม่ต้องสมัครสมาชิกหรือเข้าสู่ระบบ

ข) ผู้ใช้สามารถดาวน์โหลดไฟล์ที่ได้รับลิงก์จากผู้แชร์ และระบบรองรับการใช้งานจากหลายแพลตฟอร์ม ได้แก่ เว็บเบราว์เซอร์, Discord Direct Messages และ LINE Messages

1.4.2 ฮาร์ดแวร์ที่ใช้ในการพัฒนา

1.4.2.1 เครื่องคอมพิวเตอร์แบบตั้งโต๊ะ จำนวน 1 เครื่อง

1.4.2.2 สมาร์ทโฟน รุ่น Xiaomi Note 13 จำนวน 1 เครื่อง

1.4.3 ซอฟต์แวร์ที่ใช้ในการพัฒนา

1.4.3.1 ระบบปฏิบัติการ Microsoft Windows 11 Pro

1.4.3.2 Visual Studio Code เวอร์ชัน 1.103.2 ใช้เป็นเครื่องมือในการพัฒนา

1.4.3.4 HTML5, CSS, Javascript ใช้สำหรับการเขียนหน้าเว็บแอพ

1.4.3.5 Node.js เวอร์ชัน 22.14.0 ใช้สำหรับการเขียนระบบการทำงานเบื้องหลัง

1.4.3.6 Discord Developer เครื่องมือในการสร้างบัญชีสำหรับแชทบอทบนดิสคอร์ด

1.4.3.7 LINE Developer เครื่องมือในการสร้างบัญชีสำหรับแชทบอทบนไลน์

1.4.4 บริการแบบคราวด์ที่ใช้ในการพัฒนา

1.4.4.1 Railway บริการแบบคราวด์สำหรับใช้ในการเป็นเว็บเซิร์ฟเวอร์

1.4.4.2 Cloud MongoDB บริการแบบคราวด์สำหรับใช้ในการเก็บข้อมูลชั่วคราว

1.4.4.3 GitHub บริการแบบคราวด์สำหรับใช้ในการจัดเก็บโค้ดของโปรเจค

## 1.6 ประโยชน์ที่ได้คาดว่าจะได้รับ

1.6.1 ผู้ใช้สามารถอัปโหลดและแชร์ไฟล์ได้อย่างรวดเร็ว

1.6.2 ผู้ใช้งานสามารถลดความเสี่ยงในการถูกเข้าถึงโดยไม่ได้รับอนุญาต

1.6.3 ได้ระบบรับฝากและส่งไฟล์ออนไลน์ที่ปลอดภัยและใช้งานได้หลายแพลตฟอร์ม

# บทที่ 2

# หลักการและทฤษฎีที่เกี่ยวข้อง

ในการพัฒนาระบบฝากและแชร์ไฟล์ออนไลน์แบบไม่ต้องเข้าสู่ระบบ ผู้จัดทำได้ศึกษาข้อมูล แนวคิด และเทคโนโลยีที่เกี่ยวข้องเพื่อใช้เป็นพื้นฐานในการออกแบบและพัฒนาระบบ โดยสามารถสรุปได้ดังนี้

## 2.1 ระบบงานเดิม

โดยทั่วไป บริการฝากไฟล์ในปัจจุบัน เช่น กูเกิลไดรฟ์, ดรอปบ็อกซ์ หรือบริการฝากไฟล์ชั่วคราวมักบังคับให้ผู้ใช้ต้องสมัครสมาชิกและเข้าสู่ระบบก่อนใช้งาน ซึ่งขั้นตอนเหล่านี้สร้างความยุ่งยากและเสียเวลา โดยเฉพาะผู้ใช้ที่ต้องการแชร์ไฟล์เพียงชั่วคราวและไม่ต้องการเก็บข้อมูลถาวร ดังนั้น การใช้เทคโนโลยีสารสนเทศและเทคโนโลยีเครือข่ายเข้ามาประยุกต์ใช้ในการพัฒนาระบบฝากและแชร์ไฟล์ออนไลน์ จึงเป็นทางออกที่ดีในการการลดความยุ่งยากในการใช้งานการอัปโหลดไฟล์

## 2.2 ระบบงานอื่นที่เกี่ยวข้อง

การฝากและแลกเปลี่ยนไฟล์ผ่านเครือข่ายอินเทอร์เน็ตมีการพัฒนามาอย่างต่อเนื่องตั้งแต่อดีต โดยในระยะแรกนิยมใช้การส่งไฟล์แนบผ่านอีเมลและการอัปโหลดไฟล์ผ่านโปรโตคอลเอฟทีพี (File Transfer Protocol) ซึ่งแม้จะใช้งานได้ แต่มีข้อจำกัดด้านขนาดไฟล์ ความสะดวกในการเข้าถึง และความปลอดภัยของข้อมูล เมื่อเทคโนโลยีอินเทอร์เน็ตและระบบคลาวด์ได้รับการพัฒนา การให้บริการฝากไฟล์ออนไลน์จึงเริ่มแพร่หลายมากขึ้น โดยสามารถแบ่งประเภทของบริการฝากไฟล์ได้ดังนี้

1) บริการฝากไฟล์ชั่วคราวเป็นบริการที่อนุญาตให้ผู้ใช้อัปโหลดไฟล์และแชร์ลิงก์ดาวน์โหลดให้ผู้อื่น โดยไฟล์จะถูกลบออกจากระบบโดยอัตโนมัติเมื่อครบระยะเวลาที่กำหนด ตัวอย่างเช่น WeTransfer, File.io, SendGB บริการเหล่านี้มีข้อดีคือไม่จำเป็นต้องสมัครสมาชิก และมีความสะดวกในการใช้งาน แต่ก็ยังมีข้อจำกัด เช่น จำกัดขนาดไฟล์สูงสุด, ระยะเวลาเก็บรักษาสั้น และไม่มีระบบการเข้ารหัสไฟล์ที่รัดกุม

2) บริการฝากไฟล์ถาวรหรือระบบคลาวด์สตอเรจ กูเกิลไดรฟ์, ดรอปบ็อกซ์, วันไดรฟ์ และเมก้า ซึ่งบริการเหล่านี้ได้รับความนิยมสูงเพราะสามารถเก็บรักษาไฟล์ได้ถาวร มีพื้นที่จัดเก็บจำนวนมาก และมักมีฟังก์ชันเสริม เช่น การทำงานร่วมกันและการแชร์ไฟล์ด้วยสิทธิ์เข้าถึงที่แตกต่างกัน อย่างไรก็ตาม บริการเหล่านี้ส่วนใหญ่จำเป็นต้องสมัครสมาชิกและเข้าสู่ระบบก่อนใช้งาน และอาจมีข้อจำกัดเรื่องพื้นที่จัดเก็บฟรี รวมถึงมีความเสี่ยงต่อความเป็นส่วนตัวหากข้อมูลรั่วไหล

3) ระบบที่เน้นความปลอดภัยของข้อมูลบางชิ้นได้พัฒนาระบบฝากไฟล์ที่เน้นการเข้ารหัสไฟล์ก่อนจัดเก็บ เช่น การใช้การเข้ารหัสเออีเอส, อาร์เอสเอ หรือแม้แต่การนำบล็อกเชนมาใช้จัดเก็บข้อมูลเพื่อป้องกันการแก้ไขหรือเข้าถึงโดยไม่ได้รับอนุญาต แม้จะช่วยเพิ่มความปลอดภัย แต่ระบบเหล่านี้มักซับซ้อนเกินไปสำหรับผู้ใช้ทั่วไป

จากการศึกษาเปรียบเทียบจะเห็นได้ว่าระบบงานที่มีอยู่ในปัจจุบันยังไม่สามารถตอบโจทย์ผู้ใช้ทุกกลุ่มได้ โดยเฉพาะผู้ใช้ที่ต้องการฝากไฟล์อย่างรวดเร็ว ไม่ต้องการสมัครสมาชิก และยังคงคำนึงถึงความปลอดภัยของข้อมูล ดังนั้นโครงงานนี้จึงถูกพัฒนาขึ้นเพื่อแก้ไขข้อจำกัดเหล่านี้

## 2.3 องค์ความรู้ที่เกี่ยวข้อง

### 2.3.1 โหนดเจเอส (Node.js) และ เอ็กซ์เพรสเจเอส (Express.js)

โหนดเจเอสเป็นสภาพแวดล้อมการทำงานสำหรับภาษาจาวาสคริปต์ (JavaScript) ที่สามารถทำงานฝั่งเซิร์ฟเวอร์ได้ โดยใช้กลไกการขับเคลื่อนด้วยเหตุการณ์ และการประมวลผลอินพุตเอาต์พุตที่ไม่ต้องรอให้คำสั่งก่อนหน้าทำเสร็จ ทำให้มีประสิทธิภาพสูงในการประมวลผลคำร้องจำนวนมากพร้อมกันเหมาะสำหรับการสร้างเว็บเซิร์ฟเวอร์และระบบเรียลไทม์

เอ็กซ์เพรสเจเอส เป็น เว็บแอปพลิเคชันเฟรมเวิร์กที่ทำงานบนโหนดเจเอส ออกแบบมาให้มีโครงสร้างแบบเรียบง่ายแต่ยืดหยุ่น โดยใช้แนวคิดมิดเดิลแวร์ช่วยจัดการคำร้องและการตอบสนองอย่างเป็นระบบ ผู้จัดทำเลือกใช้เอ็กซ์เพรสเจเอสเป็นโครงสร้างหลักของระบบ เนื่องจากรองรับการสร้างเรสต์เอพีไอได้อย่างมีประสิทธิภาพ ง่ายต่อการจัดการไฟล์ การเชื่อมต่อฐานข้อมูล และการขยายระบบในอนาคต

### 2.3.2 มองโกดีบี (MongoDB)

มองโกดีบีเป็นฐานข้อมูลเชิงเอกสารที่อยู่ในกลุ่มโนเอสคิวแอลใช้รูปแบบการจัดเก็บข้อมูลเป็นเอกสารแบบคล้ายเจสันทำให้มีความยืดหยุ่นในการออกแบบโครงสร้างข้อมูลสามารถรองรับข้อมูลที่มีโครงสร้างไม่ตายตัวได้ดี อีกทั้งยังมีประสิทธิภาพสูงในการทำงานกับข้อมูลขนาดใหญ่

ในโครงงานนี้มองโกดีบีถูกนำมาใช้จัดเก็บข้อมูลเซสชันและเมตาดาต้าไฟล์ เช่น เซสชันไอดี, แอคเซสคีย์, ชื่อไฟล์ และคีย์การเข้ารหัส โดยใช้กลไก ทีทีแอล (Time-To-Live Index) ในการลบข้อมูลอัตโนมัติเมื่อครบระยะเวลาที่กำหนด เพื่อรักษาความปลอดภัยและป้องกันการสะสมของข้อมูลเก่า

### 2.3.3 การเข้ารหัสไฟล์ด้วย เอเอส-256-จีซีเอ็ม (AES-256-GCM)

แอดวานซ์ เอนคริปชัน สแตนดาร์ด (Advanced Encryption Standard - AES) เป็นมาตรฐานการเข้ารหัสแบบสมมาตร (Symmetric Encryption) ที่ได้รับการยอมรับอย่างกว้างขวาง โดยใช้คีย์เดียวกันทั้งในการเข้ารหัสและถอดรหัส โครงงานนี้เลือกใช้โหมดการเข้ารหัสแบบจีซีเอ็ม (GCM - Galois/Counter Mode) ซึ่งเป็นโหมดที่ให้ทั้งการเข้ารหัสและการตรวจสอบความถูกต้องของข้อมูลทำให้มั่นใจได้ว่าไฟล์ไม่ได้ถูกแก้ไขระหว่างทาง

ในระบบ ทุกไฟล์จะถูกเข้ารหัสก่อนจัดเก็บลงในเซิร์ฟเวอร์ โดยใช้กุญแจเข้ารหัสที่สร้างขึ้นจากเซสชันไอดีและค่าลับที่กำหนดในระบบ ช่วยเพิ่มความปลอดภัยและลดความเสี่ยงจากการเข้าถึงไฟล์โดยไม่ได้รับอนุญาต

### 2.3.4 เว็บซ็อกเก็ต (WebSocket)

เว็บซ็อกเก็ตเป็นโปรโตคอลการสื่อสารแบบสองทิศทาง ทำงานบนทีซีพีโดยสามารถเปิดการเชื่อมต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์อย่างต่อเนื่อง ทำให้สามารถแลกเปลี่ยนข้อมูลแบบเรียลไทม์โดยไม่ต้องสร้างการร้องขอใหม่ทุกครั้ง

ในโครงงานนี้ เว็บซ็อกเก็ตถูกนำมาใช้เพื่อแจ้งเตือนผู้ใช้งานทันทีเมื่อไฟล์ถูกอัปโหลดสำเร็จและพร้อมให้ดาวน์โหลด ช่วยเพิ่มประสบการณ์การใช้งานที่สะดวกและทันสมัย

### 2.3.5 คิวอาร์โค้ด (QR Code)

คิวอาร์โค้ดเป็นบาร์โค้ดสองมิติที่สามารถเก็บข้อมูลได้มากกว่าบาร์โค้ดแบบหนึ่งมิติ โดยสามารถอ่านได้อย่างรวดเร็วผ่านกล้องถ่ายรูปของสมาร์ทโฟนหรืออุปกรณ์อื่น ๆ ในโครงงานผู้จัดทำใช้คิวอาร์โค้ดเพื่อสร้างลิงก์สำหรับการอัปโหลดไฟล์ ทำให้ผู้ใช้สามารถเข้าถึงระบบได้อย่างสะดวกโดยไม่ต้องพิมพ์ยูอาร์แอล (URL)

### 2.3.6 การเชื่อมต่อกับแพลตฟอร์มโซเชียล ดิสคอร์ด (Discord) และ ไลน์ (LINE)

ดิสคอร์ดเจเอส (Discord.js) เป็นไลบรารีที่พัฒนาเพื่อเชื่อมต่อและสื่อสารกับ ดิสคอร์ดเอพีไอ ช่วยให้สามารถพัฒนาบอทที่ตอบสนองต่อข้อความและไฟล์ที่ผู้ใช้ส่งมาได้ ผู้จัดทำใช้เพื่อให้ระบบสามารถรับไฟล์ผ่านข้อความส่วนตัวบนดิสคอร์ด



**ภาพที่ 2.1** โปรแกรมแชทดิสคอร์ด

ที่มา https://discord.com/

ไลน์เมสเสจจิงเอพีไอ ผ่านไลบรารีสำหรับสร้างบอทไลน์ ทำให้ระบบสามารถรับไฟล์ที่ผู้ใช้งานส่งเข้ามาทางแชทไลน์ได้โดยตรง เป็นการเพิ่มช่องทางการใช้งานให้ผู้ใช้เข้าถึงระบบได้หลากหลายยิ่งขึ้น

### 2.3.7 โมดูลส่วนเสริม (Dependencies) ที่ใช้ในการพัฒนา

ในการพัฒนาระบบ ผู้จัดทำได้เลือกใช้เทคโนโลยีและไลบรารีของโหนดเจเอส หลายชนิด เพื่อรองรับความต้องการของระบบที่ต้องมีความปลอดภัยสูง ใช้งานได้หลายแพลตฟอร์ม และสามารถทำงานแบบเรียลไทม์ได้ โดยรายละเอียดของเทคโนโลยีที่สำคัญมีดังนี้

### 2.3.7.1 ไลน์/บอท-เอสดีเค (line/bot-sdk) เวอร์ชัน 10.0.0

ไลน์/บอท-เอสดีเค เป็นไลบรารีอย่างเป็นทางการที่พัฒนาโดย บริษัทไลน์สำหรับการเชื่อมต่อกับไลน์เมสเสจจิงเอพีไอ โดยช่วยให้ผู้พัฒนาสามารถสร้างบอทที่โต้ตอบกับผู้ใช้งานผ่านแอปพลิเคชันไลน์ได้อย่างสะดวก ภายในโครงงานนี้ ผู้จัดทำเลือกใช้เพื่อรองรับการอัปโหลดไฟล์ผ่านข้อความในไลน์ทำให้ผู้ใช้งานสามารถส่งไฟล์เข้ามาเก็บไว้ในระบบได้โดยตรงโดยไม่ต้องเข้าหน้าเว็บ ซึ่งตอบโจทย์ความสะดวกและการเข้าถึงที่หลากหลายของผู้ใช้

### 2.3.7.2 แอ็กซิโอส (Axios) เวอร์ชัน 1.10.0

แอ็กซิโอส เป็นไลบรารีสำหรับส่งคำร้องขอเอชทีทีพี (HTTP) ที่ทำงานได้ทั้งฝั่งเบราว์เซอร์และโหนดเจเอสโดยรองรับการทำงานแบบอะซิงโครนัส (Asynchronous) ผ่าน พรอมิส (Promise) ซึ่งทำให้โค้ดอ่านง่ายและจัดการได้สะดวก ในโครงงานนี้แอ็กซิโอสถูกใช้สำหรับดาวน์โหลดไฟล์ที่ผู้ใช้อัปโหลดผ่านดิสคอร์ดและสำหรับการเรียกใช้งานเอพีไอภายนอกบางส่วน เนื่องจากมีประสิทธิภาพสูงและรองรับการสตรีมข้อมูลขนาดใหญ่

### 2.3.7.3 คอร์ส (CORS) เวอร์ชัน 2.8.5

คอร์ส เป็นกลไกที่ช่วยให้เว็บเซิร์ฟเวอร์สามารถระบุได้ว่าจะอนุญาตให้โดเมนใดเข้าถึงข้อมูลได้บ้าง ไลบรารีคอร์สของโหนดเจเอสจึงถูกใช้เพื่อกำหนดสิทธิ์การเข้าถึงเอพีไอของระบบ โดยมีความสำคัญอย่างยิ่งในโครงงานนี้ เพราะระบบมีการเรียกใช้งานจากหลายแพลตฟอร์ม หากไม่มีการจัดการคอร์สที่ถูกต้อง อาจทำให้เกิดปัญหาในการร้องขอข้อมูลจากภายนอก

### 2.3.7.4 ดิสคอร์ดเจเอส เวอร์ชัน 14.21.0

ดิสคอร์ดเจเอส เป็นไลบรารียอดนิยมสำหรับสร้างบอทและเชื่อมต่อกับ ดิสคอร์ดเอพีไอโดยสามารถจัดการข้อความ ไฟล์ และกิจกรรมต่าง ๆ ภายในดิสคอร์ดได้ ในโครงงานนี้ ผู้จัดทำใช้ดิสคอร์ดเจเอสเพื่อทำให้ระบบรองรับการอัปโหลดไฟล์ผ่านข้อความส่วนตัวบนดิสคอร์ด ทำให้ระบบมีความยืดหยุ่นและสามารถเข้าถึงผู้ใช้ที่นิยมใช้ดิสคอร์ดได้อย่างสะดวก

### 2.3.7.5 ดอทอีเอ็นวี (dotenv) เวอร์ชัน 17.0.1

ดอทอีเอ็นวี เป็นไลบรารีที่ใช้สำหรับโหลดตัวแปรสภาพแวดล้อม (Environment Variables) จากไฟล์ ดอทอีเอ็นวี เข้าสู่โปรแกรมโหนดเจเอสเพื่อเก็บข้อมูลสำคัญ เช่น คีย์สำหรับเข้ารหัสไฟล์ รหัสผ่านฐานข้อมูลและโทเคน สำหรับดิสคอร์ดและไลน์โดยการใช้ดอทอีเอ็นวีจะช่วยเพิ่มความปลอดภัยและความยืดหยุ่น เพราะสามารถแยกค่าการตั้งค่าออกจากโค้ดโปรแกรม ทำให้โค้ดสะอาดและปรับแต่งได้ง่าย

### 2.3.7.6 เอ็กซ์เพรสเจเอส เวอร์ชัน 5.1.0

เอ็กซ์เพรสเจเอส เป็นเฟรมเวิร์กที่ทำงานบนโหนดเจเอส ถูกออกแบบมาเพื่อสร้างเว็บเซิร์ฟเวอร์และเอพีไอได้อย่างง่ายดาย มีซอฟต์แวร์กลางที่หลากหลายและรองรับการทำงานกับเรสต์เอพีไอได้อย่างมีประสิทธิภาพ ในโครงงานนี้ เอ็กซ์เพรสเจเอสทำหน้าที่เป็นแกนหลักของระบบทั้งหมด ทั้งการอัปโหลดไฟล์ ดาวน์โหลดไฟล์ การสร้างเซสชัน และการจัดการเว็บซ็อกเก็ต

### 2.3.7.7 เอ็กซ์เพรสเรทลิมิต (express-rate-limit) เวอร์ชัน 7.5.1

เอ็กซ์เพรสเรทลิมิต เป็นมิดเดิลแวร์ของเอ็กซ์เพรสเจเอส ที่ช่วยจำกัดจำนวนคำร้องขอ (Request) จากผู้ใช้ เพื่อป้องกันการใช้งานเกินกำหนดหรือการโจมตีแบบ ดอส ดีดอส ซึ่งในระบบฝากไฟล์สาธารณะถือว่ามีความจำเป็น เพราะผู้ไม่หวังดีอาจส่งคำร้องจำนวนมากเพื่อทำให้ระบบล่มได้

### 2.3.7.8 ไฟล์ไทป์ (file-type) เวอร์ชัน 15.0.1

ไลบรารีไฟล์ไทป์ใช้สำหรับตรวจสอบชนิดไฟล์ (MIME Type) โดยอิงจากข้อมูลจริงของไฟล์ ไม่ใช่เพียงนามสกุลไฟล์ จึงช่วยเพิ่มความปลอดภัยจากผู้ใช้ที่อาจพยายามอัปโหลดไฟล์อันตราย เช่น ไฟล์ดอทอีเอ็กซ์อี หรือดอทเจเอส ที่ปลอมตัวเป็น ดอทเจพีจี ซึ่งการตรวจสอบด้วยไฟล์ไทป์ จะช่วยป้องกันปัญหานี้ได้

### 2.3.7.9 ฟอร์มดาต้า (form-data) เวอร์ชัน 4.0.3

ฟอร์มดาต้าเป็นไลบรารีที่ช่วยจัดการข้อมูลในรูปแบบ มัลติพาร์ท/ฟอร์ม-ดาต้า (multipart/form-data) ซึ่งเป็นรูปแบบมาตรฐานสำหรับการอัปโหลดไฟล์ผ่านเอชทีทีพี โพส ผู้จัดทำใช้ฟอร์มดาต้า เพื่อทำงานร่วมกับการส่งไฟล์ไปยังเอพีไอ หรือบริการภายนอกได้ง่ายขึ้น

### 2.3.7.10 ไมม์ไทป์ส (mime-types) เวอร์ชัน 3.0.1

ไมม์ไทป์ส เป็นไลบรารีที่ใช้สำหรับค้นหาและจัดการไมม์ไทป์สตามนามสกุลไฟล์ (Extension) ซึ่งถูกนำมาใช้ควบคู่กับไฟล์ไทป์ เพื่อเพิ่มความแม่นยำในการตรวจสอบไฟล์ที่ผู้ใช้อัปโหลดเข้ามา

### 2.3.7.11 มองกูส (mongoose) เวอร์ชัน 8.16.2

มองกูส เป็นการสร้างแบบจำลองข้อมูลเชิงวัตถุสำหรับมองโกดีบี ที่ช่วยให้การทำงานกับฐานข้อมูลสะดวกขึ้น โดยสามารถกำหนด สคีมา และ โมเดล ของข้อมูลได้ชัดเจน ในโครงงานนี้ มองกูสถูกใช้สำหรับจัดการข้อมูลเซสชันและเมตาดาต้าไฟล์ เช่น เซสชันไอดี แอคเซสคีย์และชื่อไฟล์ที่อัปโหลด

### 2.3.7.12 multer เวอร์ชัน 2.0.1

multer เป็น middleware ของ เอ็กซ์เพรสเจเอส ที่ออกแบบมาเพื่อจัดการการอัปโหลดไฟล์โดยเฉพาะ สามารถกำหนดโฟลเดอร์ปลายทาง การตั้งชื่อไฟล์ และการกรองชนิดไฟล์ได้ตามต้องการ ในโครงงานนี้ multer ใช้เป็นตัวกลางในการรับไฟล์จากผู้ใช้ก่อนเข้าสู่กระบวนการเข้ารหัส

### 2.3.7.13 node-cron เวอร์ชัน 4.2.0

node-cron เป็นไลบรารีสำหรับตั้งเวลาและจัดการงาน (Scheduled Tasks) โดยใช้รูปแบบ Cron Syntax ผู้จัดทำใช้ node-cron ในการตั้งเวลาลบโฟลเดอร์อัปโหลดและข้อมูลที่หมดอายุ เพื่อให้ระบบสะอาดและไม่กินพื้นที่จัดเก็บมากเกินไป

### 2.3.7.14 qrcode เวอร์ชัน 1.5.4

qrcode เป็นไลบรารีที่ใช้สร้าง QR Code ในรูปแบบภาพ bitmap หรือ Data URL ผู้จัดทำใช้ qrcode เพื่อสร้าง QR Code สำหรับอัปโหลดไฟล์ ทำให้ผู้ใช้งานเข้าถึงระบบได้ง่าย เพียงสแกน QR Code ก็สามารถเริ่มใช้งานได้ทันที

### 2.3.7.15 uuid เวอร์ชัน 11.1.0

uuid เป็นไลบรารีที่ใช้สร้างรหัสเฉพาะ (Universally Unique Identifier) โดยรหัสนี้จะไม่ซ้ำกันและเหมาะสำหรับใช้เป็น sessionId หรือชื่อไฟล์เข้ารหัส ในโครงงานนี้ uuid ใช้เพื่อสร้างรหัสเซสชันที่ไม่ซ้ำกันในแต่ละครั้งที่มีการอัปโหลด

### 2.3.7.16 ws เวอร์ชัน 8.18.3

ws เป็นไลบรารีที่ช่วยให้ Node.js สามารถทำงานกับ WebSocket ได้อย่างสะดวก ผู้จัดทำใช้ ws เพื่อสร้างการเชื่อมต่อแบบเรียลไทม์ระหว่างผู้ใช้กับระบบ เช่น การแจ้งเตือนเมื่อไฟล์อัปโหลดสำเร็จและพร้อมสำหรับการดาวน์โหลด

# บทที่ 3

## วิธีการดำเนินงาน

3.1 การศึกษาเบื้องต้น

3.1.1 การศึกษาปัญหาของระบบการแชร์ไฟล์แบบดั้งเดิม

ปัจจุบันการส่งถ่ายไฟล์ระหว่างอุปกรณ์ต่างๆ มักประสบปัญหาหลายประการ ดังนี้

1. ปัญหาความซับซ้อนในการใช้งาน: ผู้ใช้ต้องติดตั้งแอปพลิเคชันหลายตัว เช่น Google Drive, Dropbox, หรือต้องใช้ email ในการส่งไฟล์ ซึ่งมีขั้นตอนที่ยุ่งยาก

2. ปัญหาด้านความปลอดภัย: การส่งไฟล์ผ่าน email หรือบริการ cloud storage บางประเภทไม่มีการเข้ารหัสแบบ end-to-end หรือมีการเก็บข้อมูลบนเซิร์ฟเวอร์เป็นระยะเวลานาน

3. ปัญหาข้อจำกัดของขนาดไฟล์: หลายบริการมีการจำกัดขนาดไฟล์หรือจำนวนไฟล์ที่สามารถส่งได้ในแต่ละครั้ง

4. ปัญหาการเข้าถึงข้ามแพลตฟอร์ม: การส่งไฟล์จากอุปกรณ์มือถือไปยังคอมพิวเตอร์ หรือระหว่างระบบปฏิบัติการที่ต่างกันมักมีความยุ่งยาก

3.1.2 การวิเคราะห์ความต้องการของผู้ใช้

จากการศึกษาปัญหาดังกล่าว ผู้พัฒนาได้วิเคราะห์ความต้องการของผู้ใช้งาน ดังนี้

1. ต้องการความง่ายในการใช้งาน: ผู้ใช้ต้องการวิธีการส่งไฟล์ที่รวดเร็วและไม่ซับซ้อน

2. ต้องการความปลอดภัย: ไฟล์ที่ส่งควรมีการเข้ารหัสและมีอายุการใช้งานที่จำกัด

3. ต้องการการเข้าถึงข้ามแพลตฟอร์ม: สามารถส่งไฟล์จากอุปกรณ์ใดก็ได้ไปยังอุปกรณ์ใดก็ได้

4. ต้องการความรวดเร็ว: การถ่ายโอนไฟล์ควรเป็นแบบทันทีและไม่ต้องรอนาน

3.1.3 แนวทางการแก้ปัญหาและวัตถุประสงค์

จากการวิเคราะห์ปัญหาและความต้องการ ผู้พัฒนาได้กำหนดแนวทางการแก้ปัญหาดังนี้

1. พัฒนาระบบการแชร์ไฟล์แบบ QR Code-based เพื่อความง่ายในการใช้งาน

2. ใช้เทคโนโลยี Session-based architecture ที่มีอายุการใช้งานสั้น (5 นาที) เพื่อความปลอดภัย

3. รองรับการอัปโหลดไฟล์จากหลายแพลตฟอร์ม (Web, Discord, LINE)

4. ใช้ระบบเข้ารหัสไฟล์แบบ AES-256-GCM สำหรับความปลอดภัย

5. ใช้ระบบ Access Key แบบ 5 ตัวอักษรสำหรับการเข้าถึงไฟล์ที่ง่ายต่การจำ

3.1.4 การศึกษาความเป็นไปได้ทางเทคนิค

การศึกษาความเป็นไปได้ในการพัฒนาระบบ Fileora ได้พิจารณาจากแง่มุมต่างๆ ดังนี้

1. ความเป็นไปได้ทางเทคนิค: ใช้เทคโนโลยี Node.js, เอ็กซ์เพรสเจเอส, MongoDB ที่มีความเสถียรและรองรับการพัฒนาอย่างครอบคลุม

2. ความเป็นไปได้ทางด้านการรักษาความปลอดภัย: ใช้ระบบ crypto ในตัวของ Node.js สำหรับการเข้ารหัส และ TTL (Time To Live) ใน MongoDB สำหรับการหมดอายุอัตโนมัติ

3. ความเป็นไปได้ทางด้าน Scalability: สถาปัตยกรรมของระบบรองรับการขยายตัวและการจัดการ load ที่เพิ่มขึ้น

4. ความเป็นไปได้ทางด้านการบำรุงรักษา: ใช้ Clean Code principles และมีระบบทดสอบที่ครอบคลุม

3.2 การกำหนดความต้องการของระบบ

การพัฒนาระบบ Fileora ต้องศึกษาความต้องการของเจ้าของระบบและผู้ใช้ระบบ เพื่อให้ได้ข้อกำหนดความต้องการที่สมบูรณ์ เพื่อใช้ในการพัฒนาระบบ

3.2.1 ขอบเขตของระบบ

ระบบ Fileora มีขอบเขตการทำงานดังนี้

3.2.1.1 ขอบเขตที่ระบบสามารถทำได้

1. การสร้าง Session สำหรับการอัปโหลดไฟล์พร้อม QR Code

2. การอัปโหลดไฟล์จากหลายแพลตฟอร์ม (Web UI, Discord Bot, LINE Bot)

3. การเข้ารหัสไฟล์ด้วยระบบ AES-256-GCM

4. การสร้าง Access Key แบบ 5 ตัวอักษรสำหรับการเข้าถึงไฟล์

5. การค้นหาและดาวน์โหลดไฟล์ด้วย Access Key

6. การแจ้งเตือนแบบเรียลไทม์ผ่าน WebSocket

7. การจัดการการหมดอายุของ Session (5 นาที)

8. การทำความสะอาดไฟล์ที่หมดอายุอัตโนมัติ

3.2.1.2 ขอบเขตที่ระบบไม่สามารถทำได้

1. การจัดเก็บไฟล์ถาวร (ระบบมีการหมดอายุ 5 นาที)

2. การจัดการบัญชีผู้ใช้หรือระบบ Authentication

3. การแชร์ไฟล์แบบ Public URL

4. การรองรับไฟล์ที่มีขนาดใหญ่กว่า 10 MB

5. การรองรับไฟล์ประเภทที่เป็นอันตราย (.exe, .bat, .js เป็นต้น)

3.2.2 ฮาร์ดแวร์ที่ใช้กับระบบงาน

3.2.2.1 ฮาร์ดแวร์ฝั่งเซิร์ฟเวอร์ (Server-side Hardware Requirements)

- CPU: Intel/AMD x64 หรือ ARM64 ที่รองรับ Node.js

- RAM: อย่างน้อย 512 MB (แนะนำ 1 GB ขึ้นไป)

- Storage: อย่างน้อย 1 GB สำหรับระบบและ temporary file storage

- Network: การเชื่อมต่ออินเทอร์เน็ตที่เสถียร

- Operating System: Windows 10+, macOS 10.15+, หรือ Linux (Ubuntu 18.04+)

3.2.2.2 ฮาร์ดแวร์ฝั่งผู้ใช้ (Client-side Hardware Requirements)

- อุปกรณ์ที่รองรับ Web Browser หรือ Mobile Application

- กล้องถ่าย QR Code สำหรับ Mobile Device

- การเชื่อมต่ออินเทอร์เน็ต

- หน่วยความจำเพียงพอสำหรับการเก็บไฟล์ที่ต้องการส่ง

3.2.3 ซอฟต์แวร์ที่ใช้กับระบบงาน

3.2.3.1 ซอฟต์แวร์ฝั่งเซิร์ฟเวอร์ (Server-side Software)

1. \*\*Runtime Environment\*\*

- Node.js version 16.0 หรือสูงกว่า

- npm (Node Package Manager)

2. \*\*Framework และ Libraries\*\*

- Express.js v5.1.0: Web application framework

- Mongoose v8.16.2: MongoDB object modeling

- Multer v2.0.1: File upload middleware

- WebSocket (ws) v8.18.3: Real-time communication

3. \*\*Security และ Encryption\*\*

- Node.js Crypto module: File encryption/decryption

- CORS v2.8.5: Cross-origin resource sharing

- Express Rate Limit v7.5.1: Request rate limiting

4. \*\*Database\*\*

- MongoDB: Document-based database

- MongoDB Atlas (Cloud) หรือ Local MongoDB installation

5. \*\*External API Integration\*\*

- Discord.js v14.21.0: Discord bot integration

- @line/bot-sdk v10.0.0: LINE bot integration

- QRCode v1.5.4: QR code generation

- Axios v1.10.0: HTTP client

6. \*\*Development และ Testing Tools\*\*

- Jest v30.0.5: Testing framework

- Supertest v7.1.4: HTTP assertions

- dotenv v17.0.1: Environment variables management

- node-cron v4.2.0: Scheduled tasks

3.2.3.2 ซอฟต์แวร์ฝั่งผู้ใช้ (Client-side Software)

1. \*\*Web Browser Requirements\*\*

- Chrome 80+, Firefox 75+, Safari 13+, Edge 80+

- รองรับ JavaScript ES6+

- รองรับ WebSocket

- รองรับ File API

2. \*\*Mobile Application Requirements\*\*

- Discord Application (สำหรับการใช้งาน Discord Bot)

- LINE Application (สำหรับการใช้งาน LINE Bot)

- QR Code Scanner หรือ Camera App

3. \*\*Operating System Support\*\*

- Web: Windows 10+, macOS 10.15+, Linux, iOS 13+, Android 8+

- Mobile: iOS 13+ และ Android 8+ สำหรับ Discord/LINE integration

3.3 การออกแบบระบบ

การออกแบบระบบ Fileora ประกอบไปด้วยการออกแบบระบบ การออกแบบฐานข้อมูล และการออกแบบส่วนติดต่อกับผู้ใช้

3.3.1 การออกแบบระบบ

3.3.1.1 แผนภาพกระแสข้อมูลระดับภาพรวม (Data Flow Diagram : Context Diagram)

แผนภาพกระแสข้อมูลระดับภาพรวมของระบบ Fileora แสดงให้เห็นการโต้ตอบระหว่างผู้ใช้งานกับระบบ:

\*\*ภาพที่ 3.1\*\* แผนภาพกระแสข้อมูลระดับภาพรวมของระบบ Fileora

External Entities:

- Web User: ผู้ใช้งานผ่าน Web Browser

- Discord User: ผู้ใช้งานผ่าน Discord Bot

- LINE User: ผู้ใช้งานผ่าน LINE Bot

Main System: ระบบ Fileora

Data Flows:

- QR Code Request/Response

- File Upload

- Access Key Generation

- File Search/Download

- Real-time Notifications

3.3.1.2 แผนภาพกระแสข้อมูลระดับที่ 1 (Data Flow Diagram Level 1)

แผนภาพระดับ 1 แสดงกระบวนการทำงานภายในระบบ Fileora โดยแบ่งออกเป็นกระบวนการหลัก:

\*\*ภาพที่ 3.2\*\* แผนภาพกระแสข้อมูลระดับที่ 1 ของระบบ Fileora

Process 1.0: Session Management

- สร้าง Session ID และ Encryption Key

- สร้าง QR Code

- จัดการ Session Expiry (TTL 5 minutes)

Process 2.0: File Upload Processing

- รับไฟล์จากหลายแพลตฟอร์ม

- ตรวจสอบ File Type และ Security

- เข้ารหัสไฟล์ด้วย AES-256-GCM

- บันทึกข้อมูลในฐานข้อมูล

Process 3.0: Access Key Management

- สร้าง Access Key (5 characters)

- การค้นหาด้วย Access Key

- การจัดการการหมดอายุ

Process 4.0: File Download & Decryption

- ตรวจสอบสิทธิ์การเข้าถึง

- ถอดรหัสไฟล์

- ส่งไฟล์ให้ผู้ใช้

Process 5.0: Real-time Communication

- WebSocket Connection Management

- การแจ้งเตือนการอัปโหลดสำเร็จ

- การอัปเดต UI แบบเรียลไทม์

Data Stores:

- D1: Session Database (MongoDB)

- D2: Encrypted Files Storage (File System)

3.3.1.3 แผนภาพลำดับการทำงาน (Sequence Diagram)

แผนภาพลำดับการทำงานแสดงขั้นตอนการทำงานของระบบ Fileora ในแต่ละ Scenario:

\*\*Scenario 1: Web QR Code Upload Workflow\*\*

\*\*ภาพที่ 3.3\*\* แผนภาพลำดับการทำงานของระบบ Fileora (Web QR Code Workflow)

\*\*Scenario 2: Discord Bot Upload Workflow\*\*

\*\*ภาพที่ 3.4\*\* แผนภาพลำดับการทำงานของระบบ Fileora (Discord Bot Workflow)

3.3.1.4 แผนภาพสถาปัตยกรรมระบบ (System Architecture Diagram)

แผนภาพสถาปัตยกรรมระบบแสดงโครงสร้างและองค์ประกอบทางเทคนิคของระบบ Fileora:

\*\*ภาพที่ 3.5\*\* แผนภาพสถาปัตยกรรมระบบ Fileora

\*\*คำอธิบายองค์ประกอบระบบ:\*\*

\*\*1. User Interfaces (ส่วนติดต่อผู้ใช้)\*\*

- Web Browser: ใช้งานผ่าน REST API และ WebSocket

- Discord App: ใช้งานผ่าน Discord Bot

- LINE App: ใช้งานผ่าน LINE Bot

\*\*2. Fileora Server Components (ส่วนประกอบเซิร์ฟเวอร์)\*\*

- REST API: จัดการ session, upload, download และค้นหาไฟล์

- WebSocket Server: แจ้งเตือนแบบเรียลไทม์

- Bot Integrations: รองรับ Discord และ LINE Bot

- Middleware: ตรวจสอบความปลอดภัยและจำกัดการใช้งาน

- Encryption Service: เข้ารหัสไฟล์ด้วย AES-256-GCM

- QR Code Generator: สร้าง QR Code สำหรับ upload

- Cleanup Service: ทำความสะอาดไฟล์หมดอายุ

\*\*3. Data Storage (การจัดเก็บข้อมูล)\*\*

- MongoDB: เก็บข้อมูล session และ metadata พร้อม TTL

- File System: เก็บไฟล์ที่เข้ารหัสในโฟลเดอร์ uploads

\*\*4. External APIs (API ภายนอก)\*\*

- Discord API: สื่อสารกับ Discord Bot และดาวน์โหลดไฟล์

- LINE Messaging API: รับ webhook และส่งข้อความ

3.3.2 การออกแบบฐานข้อมูล (Database Design)

3.3.2.1 แผนภาพแสดงความสัมพันธ์ของข้อมูล (Entity-Relationship Diagram)

ระบบ Fileora ใช้ฐานข้อมูล MongoDB แบบ Document-oriented โดยมี Entity หลักคือ:

\*\*Session Entity:\*\*

- sessionId (String, Primary Key, Unique)

- accessKey (String, Unique, Index)

- encryptionKey (String)

- files (Array of File Objects)

- createdAt (Date, TTL Index - expires in 5 minutes)

\*\*File Sub-document:\*\*

- filename (String) - ชื่อไฟล์ที่เข้ารหัสแล้ว

- originalname (String) - ชื่อไฟล์ต้นฉบับ

- iv (String) - Initialization Vector สำหรับการเข้ารหัส

- authTag (String) - Authentication Tag สำหรับการตรวจสอบความถูกต้อง

- sender (Object) - ข้อมูลผู้ส่ง

- platform (String: 'Web', 'Discord', 'LINE')

- name (String) - ชื่อผู้ส่ง

\*\*Relationships:\*\*

- Session → Files: One-to-Many (1 Session สามารถมีได้ 1 File เท่านั้น based on business logic)

- TTL Relationship: Sessions จะถูกลบอัตโนมัติหลังจาก 5 นาที

3.3.2.2 พจนานุกรมข้อมูล (Data Dictionary)

\*\*ตาราง Session\*\*

| Field Name | Data Type | Length | Constraint | Description |

|------------|-----------|---------|------------|-------------|

| sessionId | String | Variable | Primary Key, Unique, Required | รหัสเซสชันที่ไม่ซ้ำกัน ใช้ UUID format |

| accessKey | String | 5 | Unique, Required, Index | รหัสเข้าถึง 5 ตัวอักษร A-Z, 0-9 |

| encryptionKey | String | 64 | Required | คีย์เข้ารหัส SHA-256 ในรูปแบบ Hex |

| files | Array | Variable | Default: [] | อาร์เรย์ของข้อมูลไฟล์ |

| createdAt | Date | - | Default: Date.now, TTL: 5m | วันที่สร้าง พร้อม TTL 5 นาที |

\*\*ตาราง File (Sub-document)\*\*

| Field Name | Data Type | Length | Constraint | Description |

|------------|-----------|---------|------------|-------------|

| filename | String | Variable | Required | ชื่อไฟล์ที่เข้ารหัส (UUID.enc) |

| originalname | String | Variable | Required | ชื่อไฟล์ต้นฉบับ |

| iv | String | 24 | Required | Initialization Vector (Hex) |

| authTag | String | 32 | Required | Authentication Tag (Hex) |

| sender.platform | String | Variable | Enum: Web/Discord/LINE | แพลตฟอร์มที่ส่งไฟล์ |

| sender.name | String | Variable | Required | ชื่อผู้ส่งไฟล์ |

3.3.3 การออกแบบส่วนติดต่อกับผู้ใช้

3.3.3.1 ออกแบบผลลัพธ์ (Output Design)

\*\*1. QR Code Output\*\*

- Format: Data URL (data:image/png;base64,...)

- Size: 350x350 pixels

- Error Correction: Medium level

- Content: Upload URL with Session ID

\*\*2. Access Key Output\*\*

- Format: 5-character alphanumeric string

- Character Set: A-Z, 0-9 (excluding ambiguous characters)

- Display: Large, bold font for easy reading

- Validation: Real-time input validation

\*\*3. File Download Output\*\*

- HTTP Response with proper Content-Disposition header

- Original filename preservation

- Streaming download for large files

- Progress indication during download

\*\*4. Status Messages\*\*

- Success notifications: Green color scheme

- Error messages: Red color scheme

- Loading states: Animated indicators

- Real-time updates via WebSocket

3.3.3.2 ออกแบบรายงาน (Report Design)

เนื่องจากระบบ Fileora เป็นระบบแชร์ไฟล์ชั่วคราว จึงไม่มีการออกแบบรายงานแบบดั้งเดิม แต่มีการออกแบบ Monitoring และ Logging:

\*\*1. System Performance Metrics\*\*

- Number of active sessions

- File upload/download statistics

- Error rates and response times

- Storage usage monitoring

\*\*2. Security Audit Logs\*\*

- File type validation failures

- Suspicious upload attempts

- Session creation/expiry tracking

- Cleanup operation logs

3.3.3.3 ออกแบบส่วนนำเข้า (Input Design)

\*\*1. Web Interface Input\*\*

- File Selection: Drag & drop area with click-to-browse

- File Type Validation: Client-side pre-validation

- Size Limitation: 10MB maximum with progress indication

- Upload Progress: Real-time progress bar

\*\*2. Access Key Search Input\*\*

- Input Field: 5-character limit with uppercase auto-conversion

- Validation: Real-time format checking

- Search Trigger: Enter key or search button

- Error Handling: Visual feedback for invalid keys

\*\*3. Multi-platform Input Support\*\*

- Discord Bot: Attachment handling via DM

- LINE Bot: File message processing

- Web Upload: Standard HTTP multipart form

\*\*4. Security Input Validation\*\*

- MIME Type Detection: Server-side verification using file-type library

- Extension Blacklist: Forbidden file types (.exe, .bat, .js, etc.)

- Content Scanning: File header analysis

- Size Limits: Configurable file size restrictions

3.4 การพัฒนาระบบ

ในการศึกษาและพัฒนาระบบ Fileora นั้น ผู้พัฒนาระบบได้มีการออกแบบขั้นตอนการพัฒนาระบบ ดังต่อไปนี้

3.4.1 ศึกษาข้อมูลเอกสารจากการสัมภาษณ์ การสอบถาม และสังเกตการทำงานของบุคคลที่เกี่ยวข้องกับระบบงาน

3.4.2 นำข้อมูลที่ได้มาทำการกำหนดความต้องการของระบบ

3.4.3 วิเคราะห์ระบบ

3.4.4 ออกแบบระบบ

3.4.5 พัฒนาระบบ

3.4.6 ทดสอบระบบ ด้วยการติดตั้งและทดสอบใช้งานจริง เพื่อให้ทราบถึงข้อผิดพลาดต่างๆ ที่อาจเกิดขึ้นของระบบ เพื่อเพิ่มความมั่นใจและความน่าเชื่อถือของระบบ

3.4.7 สรุปการประเมินผลการทดสอบ

3.4.8 จัดทำเอกสารคู่มือการใช้งาน

3.5 การทดสอบระบบ

เมื่อโปรแกรมได้พัฒนาขึ้นมาแล้ว จำเป็นต้องดำเนินการทดสอบระบบก่อนที่จะนำระบบไปใช้งานจริง

ในการศึกษาและพัฒนาระบบ Fileora นั้น ผู้พัฒนาระบบได้มีการออกแบบขั้นตอนการทดสอบระบบ ดังต่อไปนี้

3.5.1 การทดสอบแต่ละส่วน (Unit Testing)

3.5.2 การทดสอบแบบเพิ่มเติม (Integration Testing)

3.5.3 การทดสอบระบบรวม (System Testing)

3.5.4 การทดสอบระบบเพื่อส่งมอบงาน (User Acceptance Testing)