ระบบส่งรายชื่อเพลงสำหรับนักร้องผ่านแอปพลิเคชันไลน์

A Song Request System for Singers via the LINE Application

นายทศพร คึขุนทด 664230040

หมู่เรียน 66/46

โครงงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา 7204903

โครงงานด้านเทคโนโลยีสารสนเทศ 2

สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม

ภาคเรียนที่ 1 ปีการศึกษา 2568

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในสถานบันเทิง เช่น บาร์หรือผับ การเปิดเพลงตามคำขอของลูกค้าเป็นส่วนสำคัญในการสร้างบรรยากาศและความพึงพอใจให้กับผู้มาใช้บริการ รูปแบบดั้งเดิมที่ใช้กันอย่างแพร่หลายคือการจดชื่อเพลงลงบนกระดาษแล้วนำไปส่งให้นักร้องหรือดีเจ ซึ่งกระบวนการดังกล่าวมีข้อจำกัดและปัญหาหลายประการ ปัญหาเหล่านี้ส่งผลโดยตรงต่อประสบการณ์ของลูกค้าและความราบรื่นในการทำงานของนักร้องและทีมงาน เพื่อแก้ไขปัญหาดังกล่าวและยกระดับการให้บริการให้ทันสมัยและมีประสิทธิภาพมากขึ้น ผู้จัดทำจึงมีความสนใจที่จะพัฒนาระบบที่นำเทคโนโลยีดิจิทัลเข้ามาประยุกต์ใช้

1.2 แนวคิดในการแก้ไขปัญหา

เพื่อแก้ไขปัญหาที่เกิดขึ้นจากการขอเพลงในรูปแบบเดิม โครงการนี้จึงนำเสนอแนวคิดในการพัฒนาระบบส่งรายชื่อเพลงแบบดิจิทัลโดยใช้แอปพลิเคชัน LINE ซึ่งเป็นที่นิยมและเข้าถึงง่ายเป็นช่องทางหลักในการสื่อสารระหว่างลูกค้าและนักร้องแนวคิดหลักคือการสร้าง LINE Bot ที่ทำหน้าที่เป็นผู้ช่วยรับคำขอเพลงจากลูกค้า ผู้ใช้สามารถค้นหาเพลงที่ต้องการผ่านการเชื่อมต่อกับ Spotify API และส่งคำขอเข้ามาในระบบได้ทันที จากนั้นคำขอเพลงจะถูกส่งไปยังเว็บแอปพลิเคชันที่แสดงผลคิวเพลงแบบเรียลไทม์สำหรับนักร้องหรือดีเจโดยเฉพาะ พร้อมกันนี้ ระบบจะบันทึกประวัติการเล่นเพลงลงใน Firebase Firestore เพื่อใช้ตรวจสอบเพลงที่เคยเล่นไปแล้วและนำข้อมูลไปใช้ประโยชน์ต่อไปได้ ซึ่งแนวทางนี้จะช่วยลดการใช้กระดาษ ลดข้อผิดพลาดในการสื่อสาร และสร้างระบบการจัดการคิวเพลงที่เป็นระเบียบ

1.3 วัตถุประสงค์ของระบบ

1.3.1 เพื่อพัฒนาระบบรับคำขอเพลงจากลูกค้าผ่านแอปพลิเคชัน LINE โดยใช้ LINE Bot API

1.3.2 เพื่อพัฒนาระบบค้นหาเพลงโดยเชื่อมต่อกับ Spotify API ทำให้ผู้ใช้สามารถค้นหาเพลงได้อย่างแม่นยำ

1.3.3 เพื่อเพิ่มประสิทธิภาพและความพึงพอใจในการให้บริการขอเพลงภายในสถานบันเทิง

1.4 ขอบเขตการศึกษา

1.4.1 ขอบเขตของระบบ

1.4.1.1 ผู้ใช้งาน

1) สามารถใช้แอปพลิเคชัน LINE เพื่อเพิ่มเพื่อนและโต้ตอบกับ Chatbot ของระบบ

2) สามารถค้นหาและส่งคำขอเพลงผ่านการพิมพ์ชื่อเพลงหรือชื่อศิลปิน

3) ได้รับการตอบกลับ ข้อความยืนยัน และการแจ้งเตือนสถานะเพลงจาก Bot

1.4.2 ฮาร์ดแวร์ที่ใช้ในการพัฒนา

1.4.2.1 คอมพิวเตอร์: หน่วยประมวลผลกลาง (CPU): Intel Core i5-13400F เมนบอร์ด(Mainboard): Gigabyte B760M DS3H หน่วยความจำ (RAM): 32 GB (DDR5) หน่วยประมวลผลกราฟิก (GPU): NVIDIA GeForce RTX 4060 Ti (16 GB) พื้นที่จัดเก็บข้อมูล (Storage): 1 TB

1.4.2.2 สมาร์ทโฟน : OPPO A94 ใช้สำหรับทดสอบการใช้งานในฝั่งผู้ใช้ผ่านแอปพลิเคชัน LINE

1.4.3 ซอฟต์แวร์ที่ใช้ในการพัฒนา

1.4.3.1 ระบบปฏิบัติการ: Microsoft Windows 11

1.4.3.2 โปรแกรมแก้ไขโค้ด (Code Editor): Visual Studio Code

1.4.3.3 Node.js: สภาพแวดล้อมสำหรับการรันโค้ด JavaScript ฝั่งเซิร์ฟเวอร์

1.4.3.4 Express.js: เฟรมเวิร์กสำหรับสร้างเว็บแอปพลิเคชันและ API

1.4.3.5 Socket.IO: ไลบรารีสำหรับการสื่อสารแบบสองทิศทางระหว่างเซิร์ฟเวอร์และไคลเอนต์

1.4.3.6 Gemini CLI: เครื่องมือ Command-Line Interface สำหรับการโต้ตอบและจัดการโมเดล Generative AI ของ Google

1.4.3.7 HTML: ภาษามาร์กอัปสำหรับสร้างโครงสร้างเว็บเพจ

1.4.3.8 CSS: ภาษาสำหรับจัดรูปแบบและตกแต่งเว็บเพจ

1.4.3.9 JavaScript: ภาษาโปรแกรมสำหรับสร้างการโต้ตอบบนเว็บเพจ

1.4.4 บริการคลาวด์และ API ที่ใช้

1.4.4.1 LINE Messaging API: ใช้สำหรับสร้าง Chatbot เพื่อรับและส่งข้อความโต้ตอบกับผู้ใช้

1.4.4.2 Spotify API: ใช้สำหรับค้นหาข้อมูลเพลง ศิลปิน และหน้าปกอัลบั้ม

1.4.4.3 Firebase Firestore: ใช้เป็นฐานข้อมูลแบบ NoSQL สำหรับจัดเก็บข้อมูลผู้ใช้ ข้อความ และผลการวิเคราะห์อารมณ์ของเพลง เพื่อให้สามารถเรียกใช้และอัปเดตข้อมูลได้แบบเรียลไทม์

1.4.4.4 Firebase: แพลตฟอร์มสำหรับพัฒนาแอปพลิเคชันพร้อมบริการหลังบ้าน เช่น โฮสติ้ง, ฐานข้อมูล และการยืนยันตัวตน

1.5 ประโยชน์ที่ได้คาดว่าจะได้รับ

1.5.1 ลดขั้นตอนและความผิดพลาดจากการใช้กระดาษ ทำให้ข้อมูลคำขอเพลงถูกต้องและไม่ตกหล่น

1.5.2 เพิ่มความสะดวกสบายให้ลูกค้าสามารถขอเพลงได้ง่ายและรวดเร็วผ่านสมาร์ทโฟน

1.5.3 สามารถนำข้อมูลประวัติการเล่นเพลงไปวิเคราะห์เพื่อดูแนวโน้มหรือเพลงยอดนิยมของร้านได้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

การพัฒนาระบบส่งรายชื่อเพลงสำหรับนักร้องผ่านแอปพลิเคชันไลน์ เป็นโครงการที่มุ่งเน้นการนำ เทคโนโลยีสมัยใหม่มาประยุกต์ใช้เพื่อเพิ่มประสิทธิภาพและแก้ไขปัญหาของกระบวนการขอเพลง ในสถานบันเทิงแบบดั้งเดิม โดยมีหัวใจสำคัญคือการสร้าง Chatbot บนแอปพลิเคชัน LINE เพื่อเป็นช่องทางการสื่อสารหลัก, เชื่อมต่อกับบริการภายนอกอย่าง Spotify API เพื่อความถูกต้อง ของข้อมูลเพลง, และพัฒนาระบบแสดงผลคิวเพลงแบบเรียลไทม์สำหรับนักร้อง ซึ่งจะช่วยลดขั้นตอน ลดความผิดพลาด และสร้างประสบการณ์ที่น่าประทับใจให้กับผู้ใช้บริการ

2.1 ระบบงานเดิม

ในระบบงานปัจจุบัน การขอเพลงในสถานบันเทิงส่วนใหญ่ยังคงใช้วิธีการแบบดั้งเดิมคือการที่ลูกค้าเขียนชื่อเพลงที่ต้องการลงบนเศษกระดาษ แล้วนำไปยื่นให้กับนักร้องโดยตรง หรือฝากพนักงานเสิร์ฟไปส่งให้ซึ่งกระบวนการดังกล่าวมีข้อจำกัดและปัญหาที่ส่งผลกระทบต่อประสิทธิภาพการทำงานและความพึงพอใจของลูกค้าหลายประการ เนื่องจากข้อมูลบนกระดาษอาจเกิดความผิดพลาดได้ง่าย เช่น ลายมือที่อ่านไม่ออก การสะกดชื่อเพลงผิด หรือกระดาษเปียกน้ำจนข้อความเลือนหาย นอกจากนี้ กระดาษที่ใช้ส่งคำขอยังอาจตกหล่นหรือสูญหายระหว่างการส่งต่อได้ง่ายทำให้คำขอของลูกค้าไม่ถูกส่งไปถึงนักร้อง อีกทั้งกระบวนการส่งต่อผ่านพนักงานยังทำให้เกิดความล่าช้า และที่สำคัญคือไม่มีระบบที่ชัดเจนในการจัดลำดับเพลงที่ขอเข้ามา ทำให้เกิดความซ้ำซ้อนหรือเกิดความไม่พอใจจากลูกค้าที่ไม่ทราบว่าเพลงของตนเองอยู่ในลำดับใด

2.2 ระบบงานอื่นที่เกี่ยวข้อง

จากการศึกษาค้นคว้า พบว่ามีผู้พัฒนาระบบที่มีแนวคิดคล้ายคลึงกันอยู่บ้าง โดยเป็นการนำเทคโนโลยีมาใช้จัดการการขอเพลงในร้านอาหารและสถานบันเทิง

ณัฐพล ดวงจันทร์และคณะ (2562) ได้ทำการพัฒนาระบบขอเพลงสำหรับร้านอาหารและสถานบันเทิงผ่านเว็บแอปพลิเคชัน โดยมีวัตถุประสงค์เพื่อลดปัญหาการสื่อสารที่ผิดพลาดและเพิ่มความสะดวกในการจัดการเพลง ระบบดังกล่าวให้ผู้ใช้บริการสแกน QR Code ที่โต๊ะเพื่อเข้าไปยังหน้าเว็บไซต์สำหรับส่งคำขอเพลงจากนั้นข้อมูลจะถูกส่งไปยังหน้าจอของผู้ดูแลระบบเพื่อจัดการคิวเพลง

แม้ว่าระบบดังกล่าวจะสามารถแก้ไขปัญหาของระบบงานเดิมได้ แต่ก็ยังมีข้อจำกัดในการที่ผู้ใช้จำเป็นต้องเปิดเบราว์เซอร์เพื่อเข้าใช้งาน ซึ่งอาจไม่สะดวกสำหรับผู้ใช้ทุกคน ในขณะที่โครงการนี้ได้เลือกใช้ แอปพลิเคชัน LINE ซึ่งเป็นแพลตฟอร์มที่ผู้ใช้ส่วนใหญ่คุ้นเคยและใช้งานเป็นประจำอยู่แล้วมาเป็นช่องทางหลักในการสื่อสาร การใช้ LINE Bot ทำให้ผู้ใช้สามารถส่งคำขอเพลงได้ทันทีโดยไม่ต้องติดตั้งแอปพลิเคชันเพิ่มเติมหรือสลับไปใช้งานเบราว์เซอร์ ซึ่งเป็นแนวทางที่ช่วยเพิ่มความสะดวกและลดขั้นตอนในการใช้งานได้ดียิ่งขึ้น

2.3 องค์ความรู้และเทคโนโลยีที่เกี่ยวข้อง

2.3.1 ไลน์ เมสเสจจิ้ง เอพีไอ (LINE Messaging API)

เป็น API สำหรับสื่อสารระหว่าง Chatbot และผู้ใช้บนแอปพลิเคชัน LINE ทำให้ระบบสามารถรับข้อความจากผู้ใช้, ตอบกลับอัตโนมัติ, และส่งการแจ้งเตือนสถานะเพลงได้โดยตรงผ่าน LINE การใช้ LINE Messaging API ช่วยให้การโต้ตอบเป็นไปอย่างรวดเร็ว สะดวก และอยู่บนแพลตฟอร์มที่ผู้ใช้คุ้นเคย

2.3.2 สปอทิฟาย เว็บ เอพีไอ (Spotify Web API)

เป็นบริการที่เปิดให้นักพัฒนาสามารถเข้าถึงฐานข้อมูลขนาดใหญ่ของ Spotify เพื่อใช้ในการค้นหาและดึงข้อมูลเพลง, ศิลปิน, อัลบั้ม และหน้าปกอัลบั้ม ซึ่งช่วยให้ระบบได้ข้อมูลที่เป็นมาตรฐานสากล มีความถูกต้อง และลดปัญหาการสะกดผิดได้

2.3.3 โหนดดอทเจเอส (Node.js)

เป็นสภาพแวดล้อมการทำงานสำหรับภาษา JavaScript ที่ทำให้สามารถรันโค้ดนอกเว็บเบราว์เซอร์บนฝั่งเซิร์ฟเวอร์ได้ มีจุดเด่นคือการทำงานแบบ Non-blocking I/O ซึ่งทำให้สามารถจัดการคำขอจำนวนมากพร้อมกันได้อย่างมีประสิทธิภาพ เหมาะสำหรับการสร้างเซิร์ฟเวอร์ที่ต้องรับ Webhook Events, ประมวลผลคำขอ และติดต่อสื่อสารกับ API ภายนอก

2.3.4 เอ็กซ์เพรดอทเจเอส (Express.js)

เป็นเฟรมเวิร์กสำหรับ Node.js ที่ใช้สำหรับสร้างเว็บแอปพลิเคชันและ API ได้อย่างรวดเร็วและเป็นระบบ สามารถใช้เพื่อสร้าง API, จัดการเส้นทาง (Routing) ของข้อมูล และสร้าง Webhook Endpoint เพื่อรับข้อมูลจากเซิร์ฟเวอร์ภายนอก

2.3.5 ซ็อกเก็ตดอทไอโอ (Socket.IO)

เป็นไลบรารี JavaScript สำหรับการสื่อสารแบบสองทิศทางและเรียลไทม์ระหว่างไคลเอนต์และเซิร์ฟเวอร์ ทำให้เว็บแอปพลิเคชันสามารถอัปเดตข้อมูลได้ทันทีโดยไม่ต้องทำการรีเฟรชหน้าจอเหมาะสำหรับการสร้างระบบที่ต้องการการแสดงผลข้อมูลล่าสุดแบบทันที

2.3.6 ไฟร์เบส (Firebase)

เป็นแพลตฟอร์มจาก Google สำหรับการพัฒนาแอปพลิเคชัน ที่มาพร้อมกับบริการหลังบ้าน(Backend-as-a-Service) โดยมีความสามารถหลากหลาย เช่น Firebase Hosting สำหรับโฮสต์เว็บแอปพลิเคชัน

2.3.7 ไฟร์เบส ไฟร์สโตร์ (Firebase Firestore)

เป็นฐานข้อมูลแบบ NoSQL ที่โฮสต์บนคลาวด์สำหรับการจัดเก็บและซิงค์ข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์แบบเรียลไทม์ มีความยืดหยุ่นและสามารถปรับขนาดได้สูง เหมาะสำหรับแอปพลิเคชันที่ต้องการการอัปเดตข้อมูลที่รวดเร็วและเชื่อถือได้ เช่น แอปพลิเคชันแชท หรือระบบที่ต้องแสดงสถานะล่าสุดของผู้ใช้

2.3.8 เจมิไน คอมมานด์ไลน์อินเทอร์เฟซ (Gemini Command Line Interface)

เป็นเครื่องมือแบบบรรทัดคำสั่งสำหรับโต้ตอบและจัดการโมเดล Generative AI ของ Google ซึ่งมีความสามารถในการเข้าใจและสร้างข้อความได้อย่างเป็นธรรมชาติ สามารถนำมาประยุกต์ใช้เพื่อสร้างข้อความตอบกลับที่มีความสร้างสรรค์ หรือพัฒนาฟีเจอร์ที่ต้องใช้การวิเคราะห์และสร้างเนื้อหาจาก AI

2.3.9 เทคโนโลยีสำหรับหน้าเว็บแอป

2.3.9.1 เอชทีเอ็มแอล (HTML)

เป็นภาษามาร์กอัปหลักที่ใช้ในการสร้างและจัดโครงสร้างเนื้อหาบนเว็บเพจ ใช้สำหรับกำหนดองค์ประกอบต่างๆ เช่น หัวข้อ, ย่อหน้า, รายการ, และรูปภาพ เพื่อให้เว็บเบราว์เซอร์สามารถแสดงผลได้อย่างถูกต้อง

2.3.9.2 ซีเอสเอส (CSS)

เป็นภาษาที่ใช้สำหรับออกแบบและจัดรูปแบบการแสดงผลของเอกสาร HTML ทำหน้าที่ควบคุมการแสดงผลด้านภาพ เช่น สี, ขนาดตัวอักษร, การจัดวางองค์ประกอบ และการตอบสนองต่อขนาดหน้าจอที่แตกต่างกัน เพื่อให้เว็บแอปพลิเคชันมีความสวยงามและใช้งานง่าย

2.3.9.3 จาวาสคริปต์ (JavaScript)

เป็นภาษาสคริปต์ที่ใช้เพื่อเพิ่มการโต้ตอบและความสามารถแบบไดนามิกให้กับเว็บเพจทำหน้าที่จัดการกับเหตุการณ์ต่างๆ ที่เกิดจากผู้ใช้, จัดการข้อมูลที่ได้รับจากเซิร์ฟเวอร์แบบเรียลไทม์,และปรับเปลี่ยนเนื้อหาบนหน้าเว็บโดยไม่ต้องโหลดหน้าใหม่ทั้งหมด

บทที่ 3

วิธีการดำเนินงาน

บทนี้เป็นการอธิบายขั้นตอนการดำเนินงานของโครงงาน “ระบบส่งรายชื่อเพลงสำหรับนักร้องผ่านแอปพลิเคชัน LINE” เพื่อให้ได้ระบบที่สามารถตอบสนองต่อความต้องการของผู้ใช้งานได้อย่างเหมาะสมและมีประสิทธิภาพ โดยผู้พัฒนาได้ศึกษาข้อมูล วิเคราะห์ ออกแบบ และพัฒนาระบบตามขั้นตอนที่วางไว้ ซึ่งมีรายละเอียดดังต่อไปนี้

3.1 การศึกษาเบื้องต้น

การศึกษาเบื้องต้นเป็นขั้นตอนแรกของการดำเนินโครงงาน โดยผู้พัฒนาได้ทำการศึกษาข้อมูลที่เกี่ยวข้องกับระบบการส่งรายชื่อเพลงผ่านแอปพลิเคชัน LINE ทั้งในส่วนของการทำงานของระบบที่คล้ายคลึงกัน การทำงานของ LINE Messaging API และการใช้งาน Spotify API เพื่อให้เข้าใจถึงหลักการทำงานของแต่ละระบบ รวมถึงศึกษาปัญหาและข้อจำกัดของการส่งรายชื่อเพลงแบบเดิมที่ผู้ใช้งานต้องค้นหาเพลงด้วยตนเอง ซึ่งทำให้เสียเวลาและไม่สะดวกในการใช้งาน

จากการศึกษา พบว่าการนำเทคโนโลยี Chatbot ของ LINE มาประยุกต์ร่วมกับข้อมูลจาก Spotify API สามารถช่วยให้ผู้ใช้งานค้นหาเพลงได้รวดเร็วและสะดวกยิ่งขึ้น โดยผู้พัฒนาจึงได้กำหนดวัตถุประสงค์หลักของระบบ คือ

3.1.1 เพื่อพัฒนาระบบที่ช่วยให้นักร้องสามารถค้นหาและส่งรายชื่อเพลงได้สะดวกผ่าน LINE

3.1.2 เพื่อเพิ่มความถูกต้องและรวดเร็วในการค้นหาเพลงด้วยการเชื่อมต่อกับฐานข้อมูล Spotify

3.1.3 เพื่อให้ระบบสามารถบันทึกและจัดการข้อมูลเพลงได้อย่างมีประสิทธิภาพด้วย Firebase Firestore

3.2 การกำหนดความต้องการของระบบ

การพัฒนาระบบนี้เริ่มจากการเก็บรวบรวมความต้องการจากผู้ใช้ (นักร้องและผู้ควบคุมรายการเพลง) รวมถึงผู้ดูแลระบบ เพื่อให้เข้าใจฟังก์ชันการทำงานที่จำเป็นและขอบเขตของระบบได้อย่างครบถ้วน

3.2.1 ขอบเขตของระบบ

ระบบนี้ถูกออกแบบให้ทำงานผ่าน LINE Application โดยใช้ LINE Chatbot เป็นตัวกลางในการสื่อสารกับผู้ใช้ ผู้ใช้สามารถพิมพ์ชื่อเพลง ศิลปิน หรือคำสั่งค้นหา จากนั้นระบบจะเชื่อมต่อกับ Spotify API เพื่อดึงข้อมูลเพลงที่ตรงกับคำค้น พร้อมแสดงชื่อเพลง ศิลปิน และภาพปกอัลบั้มกลับมาให้ผู้ใช้เลือกส่งต่อในห้องแชต ระบบจะบันทึกประวัติการส่งเพลงและรายชื่อเพลงทั้งหมดไว้ใน Firebase Firestore เพื่อใช้ในการจัดการภายหลัง

3.2.2 ฮาร์ดแวร์ที่ใช้กับระบบงาน

คอมพิวเตอร์หรือเซิร์ฟเวอร์ที่รองรับการรัน Node.js สำหรับประมวลผล Chatbot โทรศัพท์มือถือของผู้ใช้งานที่ติดตั้งแอปพลิเคชัน LINE อุปกรณ์เครือข่ายที่เชื่อมต่ออินเทอร์เน็ต

3.2.3 ซอฟต์แวร์ที่ใช้กับระบบงาน

3.2.3.1 LINE Messaging API: ใช้สำหรับเชื่อมต่อและรับ–ส่งข้อความระหว่างผู้ใช้กับระบบ

3.2.3.1 Spotify API: ใช้สำหรับค้นหาข้อมูลเพลง ศิลปิน และอัลบั้ม

3.2.3.1 Firebase Firestore: ใช้จัดเก็บข้อมูลผู้ใช้และประวัติการส่งรายชื่อเพลง

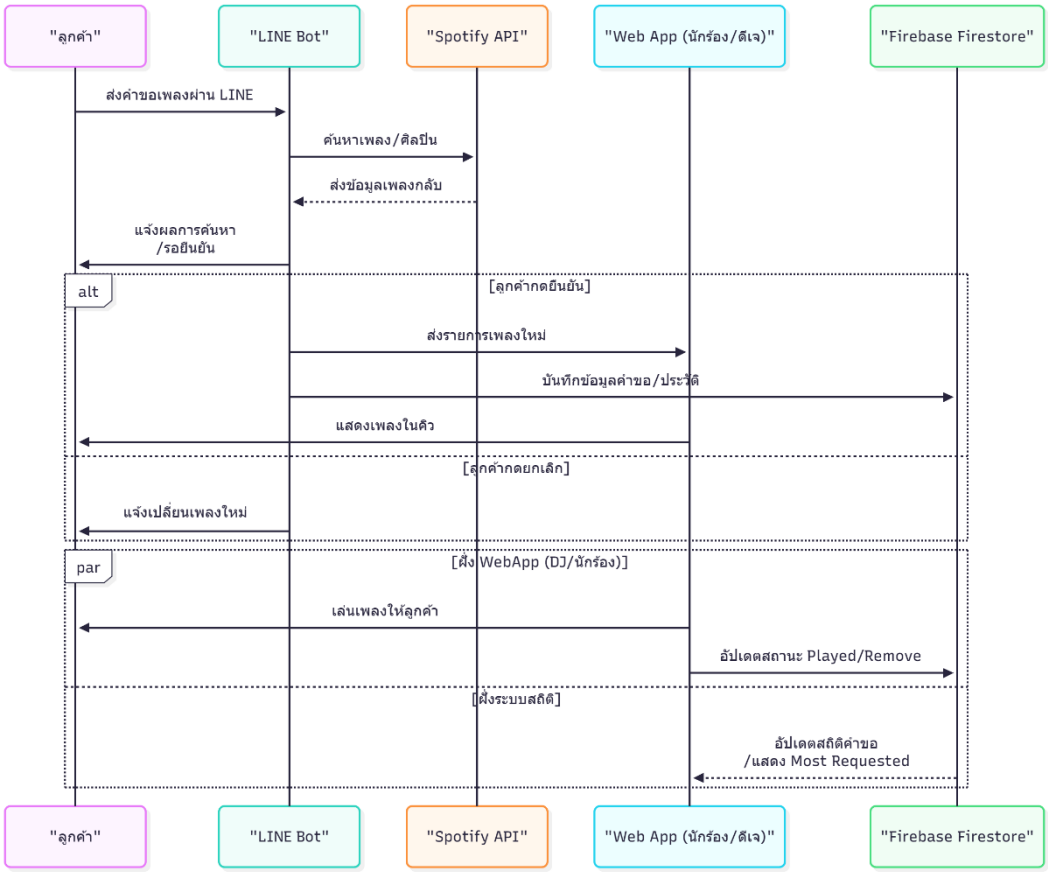
3.2.3.1 Node.js: สำหรับพัฒนาเซิร์ฟเวอร์ของระบบ

3.2.3.1 Visual Studio Code: สำหรับพัฒนาและทดสอบโค้ดโปรแกรม

3.2.3.1 GitHub: สำหรับจัดเก็บและควบคุมเวอร์ชันของโค้ด

3.3 การออกแบบระบบ

3.3.1 การออกแบบระบบ



ภาพที่ 3.1 แผนภาพกระแสข้อมูลระดับภาพรวมของระบบ

1) ลูกค้าส่งคำขอ: ลูกค้าพิมพ์ชื่อเพลงหรือศิลปินที่ต้องการ แล้วส่งเป็นข้อความไปหา "LINE Bot"

2) บอทค้นหาเพลง: "LINE Bot" นำข้อความนั้นไปค้นหาใน "Spotify API" เพื่อหาข้อมูลเพลงที่ตรงกัน

3) Spotify ส่งข้อมูลกลับ: "Spotify API" ส่งผลการค้นหา (เช่น ชื่อเพลง, ชื่อศิลปิน) กลับมาให้ "LINE Bot"

4) บอทส่งผลลัพธ์ให้ลูกค้ายืนยัน: "LINE Bot" นำข้อมูลเพลงที่ได้มาแสดงให้ลูกค้าดู เพื่อให้ลูกค้ายืนยันว่าเป็นเพลงที่ต้องการหรือไม่

กรณีที่ 1: ลูกค้ากดยืนยันเพลง

5) บอทส่งเพลงใหม่เข้าระบบ: เมื่อลูกค้ากดยืนยัน "LINE Bot" จะส่งข้อมูลเพลงนั้นไปยัง "Web App" ของนักร้อง/ดีเจ

6) Web App บันทึกข้อมูล: "Web App" จะทำการบันทึกข้อมูลเพลงที่ขอใหม่นี้ลงในฐานข้อมูล "Firebase Firestore" เพื่อเก็บเป็นประวัติและจัดการคิว

7) แสดงผลที่ Web App: เพลงที่ขอใหม่จะไปปรากฏขึ้นในหน้าจอคิวเพลงของ "Web App" เพื่อให้นักร้อง/ดีเจเห็น

8) แจ้งเตือนลูกค้า: "LINE Bot" ส่งข้อความกลับไปหาลูกค้าเพื่อแจ้งว่า "เพลงของคุณถูกเพิ่มในคิวเรียบร้อยแล้ว"

กรณีที่ 2: ลูกค้ากดยกเลิก

9. บอทแจ้งลูกค้ายกเลิก: "LINE Bot" จะส่งข้อความกลับไปหาลูกค้าเพื่อแจ้งว่า "การขอเพลงถูกยกเลิกแล้ว" หรือให้ลูกค้า "แจ้งเปลี่ยนเพลงใหม่"

ฝั่งนักร้อง/ดีเจ

10. เล่นเพลง: นักร้อง/ดีเจเห็นคิวเพลงใน "Web App" และทำการ "เล่นเพลงให้ลูกค้า"

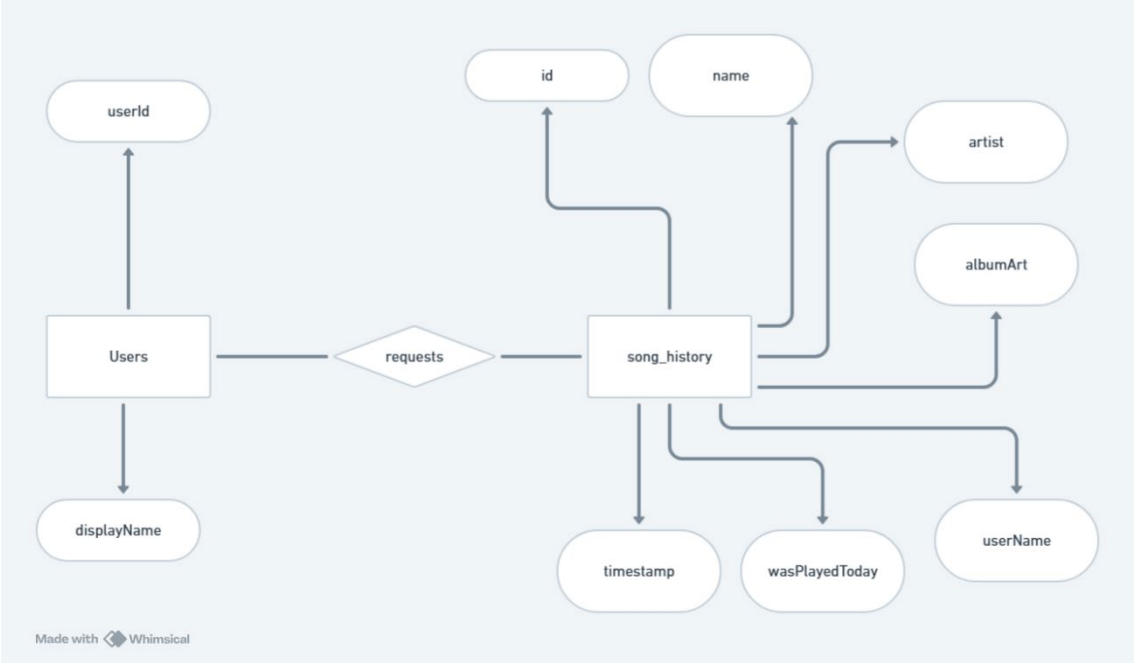
11. อัปเดตสถานะ: หลังจากเล่นเพลงนั้นจบ หรือลบเพลงออกจากคิว "Web App" จะอัปเดตสถานะของเพลงนั้นใน "Firebase Firestore" เป็น "Played" หรือ "Remove"

ฝั่งระบบสถิติ

12. อัปเดตสถิติ: ในขณะเดียวกัน ระบบจะนำข้อมูลการขอเพลงไปอัปเดตสถิติใน "Firebase Firestore" เช่น เพลงไหนถูกขอบ่อยที่สุด (Most Requested)

3.3.2 การออกแบบฐานข้อมูล (Database Design)

3.3.2.1แผนภาพแสดงความสัมพันธ์ของข้อมูล (Entity-Relationship Diagram)



ภาพที่ 3.2 แผนภาพแสดงความสัมพันธ์ของข้อมูล

3.3.2.2 พจนานุกรมข้อมูล (Data Dictionary)

ตารางที่ 3.1 ตารางการจัดข้อมูลเพลงที่ส่งมาจากการขอเพลง (pending_songs)

ลำดับ (No)	คุณสมบัติ (Attribute)	คำอธิบาย (Description)	ขนาด (Width)	ประเภท (Type)	ประเภทคีย์ (Key Type)
1	id	รหัสเฉพาะของแต่ละคำขอเพลงที่สร้างขึ้นโดยระบบ	36	String	Primary Key
2	name	ชื่อเพลง	150	String	
3	artist	ชื่อศิลปิน	100	String	
4	albumArt	URL ของภาพหน้าปกอัลบั้มจาก Spotify	255	String	
5	userName	ชื่อผู้ใช้ LINE ที่ทำการขอเพลง	50	String	Foreign Key
6	timestamp	ประทับเวลาที่ทำการขอเพลง	-	Timestamp	
7	wasPlayedToday	สถานะที่บ่งบอกว่าเพลงนี้ถูกเล่นในวันนี้แล้วหรือยัง	-	Boolean	

ตารางที่ 3.2 ตารางการจัดข้อมูลเพลงที่ส่งมาจากการกดเล่นเพลง (song_history)

ลำดับ (No)	คุณสมบัติ (Attribute)	คำอธิบาย (Description)	ขนาด (Width)	ประเภท (Type)	ประเภทคีย์ (Key Type)
1	id	รหัสเฉพาะของแต่ละคำขอเพลงที่สร้างขึ้นโดยระบบ	36	String	Primary Key
2	name	ชื่อเพลง	150	String	
3	artist	ชื่อศิลปิน	100	String	
4	albumArt	URL ของภาพหน้าปกอัลบั้มจาก Spotify	255	String	
5	userName	ชื่อผู้ใช้ LINE ที่ทำการขอเพลง	50	String	Foreign Key
6	playedAt	ประทับเวลาที่ทำการกดเล่นเพลง	-	Timestamp	
7	wasPlayedToday	สถานะที่บ่งบอกว่าเพลงนี้ถูกเล่นในวันนี้แล้วหรือยัง	-	Boolean	

3.3.3 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface Design)

การออกแบบโครงร่างหน้าจอของการพัฒนาระบบ "Line Music Bar" ซึ่งเป็นการออกแบบส่วนประกอบของหน้าจอสำหรับการขอเพลง การจัดการคิวเพลง และการแสดงสถิติผ่านหน้าเว็บ โดยมีรายละเอียดในการออกแบบดังต่อไปนี้

3.3.3.1 ผลการทำงานที่ได้จากการออกแบบหน้าจอการขอเพลง

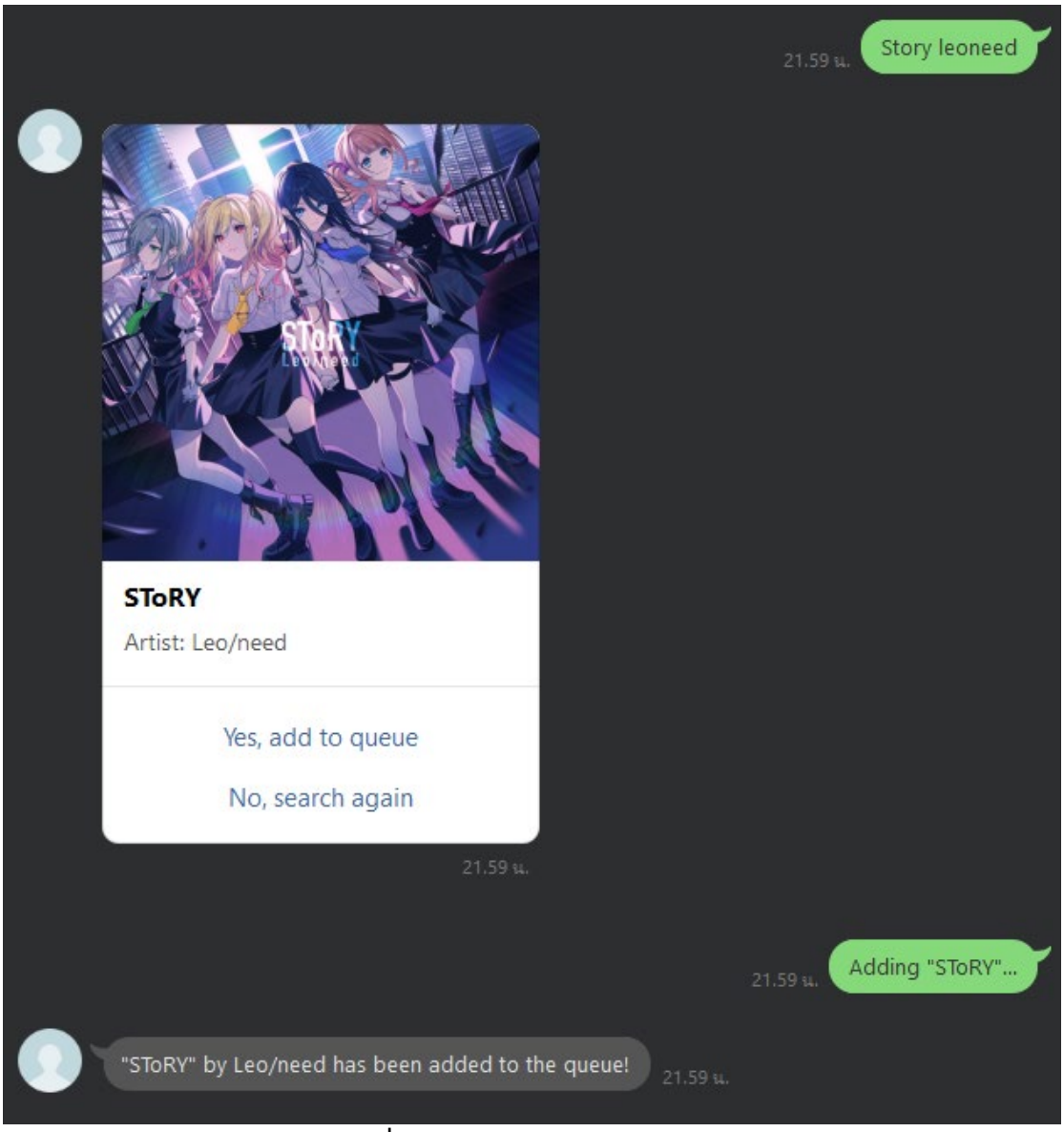
หน้าจอการขอเพลง (Song Request Page) เป็นส่วนที่ผู้ใช้โต้ตอบผ่านแอปพลิเคชันไลน์ โดยหน้านี้จะเริ่มต้นเมื่อผู้ใช้ส่งชื่อเพลงที่ต้องการ จากนั้นบอทจะนำชื่อเพลงไปค้นหาผ่าน SPOTIFY API และแสดงผลการค้นหาพร้อมปุ่มให้ผู้ใช้เลือก โดยมีขั้นตอนดังนี้

1) ผู้ใช้ส่งชื่อเพลงที่ต้องการเข้ามาในแชท

2) ระบบแสดงผลการค้นหาที่ได้จาก SPOTIFY API ซึ่งประกอบด้วยรูปภาพ ชื่อเพลง และชื่อศิลปิน

3) ผู้ใช้สามารถกดยอมรับเพื่อส่งเพลงไปให้นักร้อง ("Yes, add to queue") หรือกดปฏิเสธเพื่อค้นหาเพลงใหม่ ("No, search again")

4) เมื่อกดยอมรับ ระบบจะส่งข้อความเพื่อยืนยันการเพิ่มเพลงเข้าคิว



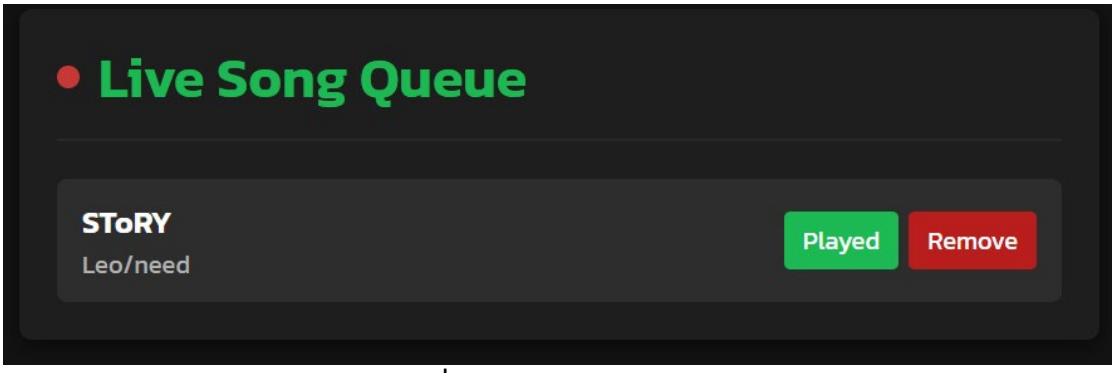
ภาพที่ 3.3 หน้าจอการขอเพลงใน Line

3.3.3.2 ผลการทำงานที่ได้จากการออกแบบหน้าจอคิวเพลงแสดงสด

หน้าคิวเพลงแสดงสด (Live Song Queue Page) เป็นหน้าจอสำหรับให้นักร้องหรือผู้ดูแลใช้จัดการเพลงที่ถูกขอเข้ามา โดยจะแสดงรายการเพลงที่ผู้ใช้ได้ขอไว้ ในแต่ละรายการเพลงจะมีปุ่มสำหรับจัดการ 2 ปุ่ม คือ

Played: เมื่อกดปุ่มนี้หมายความว่าเพลงได้ถูกเล่นแล้ว และข้อมูลจะถูกนำไปเก็บเพื่อใช้จัดอันดับสถิติ

Remove: เมื่อกดปุ่มนี้หมายความว่าเพลงจะไม่ถูกเล่น และข้อมูลนั้นจะไม่ถูกเก็บไว้



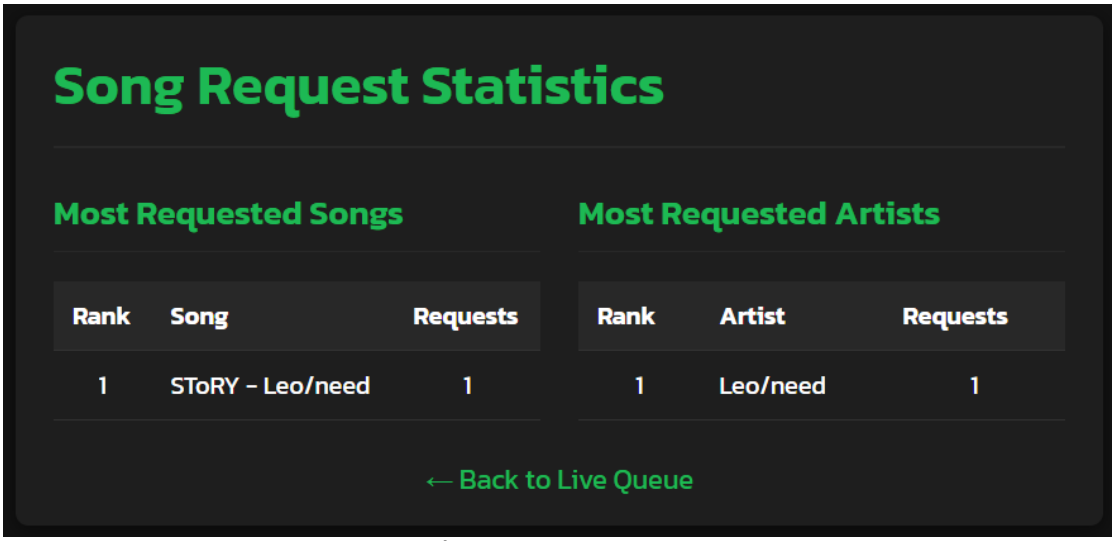
ภาพที่ 3.3 หน้าคิวเพลงแสดงสด

3.3.3.3 ผลการทำงานที่ได้จากการออกแบบหน้าจอสถิติการขอเพลง

หน้าสถิติการขอเพลง (Song Request Statistics Page) จะแสดงข้อมูลสรุปการขอเพลง ซึ่งรวบรวมมาจากเพลงที่ถูกกด "Played" ในหน้าคิวเพลงแสดงสด โดยแบ่งการจัดอันดับออกเป็น 2 ประเภทคือ

แสดงอันดับเพลงที่ถูกขอมากที่สุด กับ แสดงอันดับศิลปินที่ถูกขอเพลงมากที่สุด

นอกจากนี้ยังมีปุ่มสำหรับย้อนกลับไปหน้าคิวเพลงแสดงสด (Back to Live Queue)



ภาพที่ 3.4 หน้าสถิติการขอเพลง

3.4 การพัฒนาระบบ

ผู้พัฒนาได้วางแผนและดำเนินการพัฒนาระบบตามลำดับขั้นตอนดังนี้

3.4.1 ศึกษาเอกสารและเครื่องมือ เช่น LINE Developer, Spotify Developer, Firebase

3.4.2 กำหนดความต้องการของระบบจากผู้ใช้งานจริง

3.4.3 วิเคราะห์กระบวนการทำงานและข้อมูล

3.4.4 ออกแบบระบบและฐานข้อมูล

3.4.5 พัฒนาโปรแกรมโดยใช้ Node.js เชื่อมต่อกับ LINE Messaging API และ Spotify API

3.4.6 จัดเก็บข้อมูลบน Firebase Firestore

3.4.7 ทดสอบระบบโดยใช้งานจริงในกลุ่มผู้ใช้จำลอง

3.4.8 ปรับปรุงและจัดทำเอกสารคู่มือการใช้งาน

3.5 การทดสอบระบบ

หลังจากพัฒนาระบบเสร็จสมบูรณ์ ผู้พัฒนาได้ดำเนินการทดสอบระบบในหลายระดับ เพื่อประเมินความถูกต้องและประสิทธิภาพการทำงาน

3.5.1 การทดสอบแต่ละส่วน (Unit Test)

ทดสอบฟังก์ชันย่อย เช่น การค้นหาเพลง การดึงข้อมูลจาก Spotify API และการบันทึกข้อมูลลง Firestore

3.5.2 การทดสอบแบบเพิ่มเติม (Integration Test)

ทดสอบการทำงานร่วมกันระหว่าง LINE Messaging API และ Spotify API เพื่อให้แน่ใจว่าระบบสามารถรับ–ส่งข้อมูลได้ถูกต้อง

3.5.3 การทดสอบระบบบรวม (System Test)

ทดสอบระบบโดยรวมจากการใช้งานจริงในสภาพแวดล้อมจำลอง เพื่อดูการตอบสนอง ความเร็ว และความเสถียร

3.5.4 การทดสอบระบบเพื่อส่งมอบงาน (Acceptance Test)

นำระบบให้ผู้ใช้งานจริงทดลองใช้และให้ข้อเสนอแนะ เพื่อปรับปรุงก่อนส่งมอบงานอย่างสมบูรณ์

บรรณานุกรม

ณัฐพล ดวงจันทร์และคณะ. (2562). การพัฒนาระบบขอเพลงสำหรับร้านอาหารและสถานบันเทิงผ่านเว็บแอปพลิเคชัน. สารนิพนธ์ปริญญาตรี, สาขาวิชาการจัดการทั่วไป คณะวิทยาการจัดการ มหาวิทยาลัยราชภัฏนครปฐม.วารสารวิทยาการจัดการ ปีที่ 6 ฉบับที่ 2 พ.ศ. 2562