



Practical Building a Powerful

# LINE OA

Official Account

## for Developer



Road to LINE Developers  
Hands-on Experience to Develop  
Application with LINE Services

สร้างตัวเองให้เป็น LINE Developers เพื่อพัฒนา LINE OA ที่เหนือกว่า  
เรียนรู้การพัฒนาระบบบนแพลตฟอร์ม LINE ด้วยเทคนิคที่ไม่ควรพลาด

↓ SOURCE CODE :  
SERAZU.COM

ผู้แต่ง ผศ.ดร.วุฒิจพงษ์ ชินศรี  
LINE Certified Coach for API  
บรรณาธิการ กิรพล คชางริญ





มีเพียง “ความรู้” เท่านั้นที่มนุษย์ใช้พลิก “โลก”  
และเปลี่ยน “ชีวิต” เราจึงสร้างสรรค์ และส่งมอบ “ความรู้”  
ในรูปแบบที่ดีที่สุด เพื่อให้คนไทย “เรียนรู้” ได้ตลอดชีวิต

Only “Knowledge” can help human  
change “The World” and “Their Lives”.  
With this truth, it drives us to deliver  
“Knowledge” for Thai being able to  
“Learn” better everyday.



# Practical Building a Powerful LINE OA for Developer

Writer	ผศ.ดร.วุฒิพงษ์ ชินศรี
Editor	ภีรพล คชาเจริญ
Production Manager	วรพล ธนิกุล
Graphic Designer	ชวรัตน์ รัตนะ
Page Layout	สุรัสวดี วงศ์จันทร์สุข
Proofreader	สุนทรี บรรลือศักดิ์
Publishing Coordinators	สุภัตรา อาจปฐุ, สุริย์รัตน์ จิ๋ว
Product Specialist	ศรันย์ชาติสุทธิผล

LINE เป็นเครื่องหมายการค้าของบริษัท LINE Corporation, Visual Studio Code เป็นเครื่องหมายการค้าของบริษัท Microsoft Corporation และเครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอดีซี พรีเมียร์ จำกัด ห้ามลอกเลียนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใดๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้จัดพิมพ์เท่านั้น

สร้างสรรค์โดย



ข้อมูลทางบรรณานุกรม

ผศ.ดร.วุฒิพงษ์ ชินศรี  
Practical Building a Powerful LINE OA for Developer  
นนทบุรี : ไอดีซีฯ, 2566  
232 หน้า

1. โปรแกรมสำหรับระบบการประมวลผลข้อมูลเฉพาะชนิด  
I ชื่อเรื่อง  
005.37

ISBN 978-616-487-396-4

พิมพ์ครั้งที่ 1 มกราคม 2566

ราคา 350 บาท

จัดพิมพ์และจัดจำหน่ายโดย



บริษัท ไอดีซี พรีเมียร์ จำกัด

200 หมู่ 4 ชั้น 19 ห้อง 1901  
อาคารจัสตินอินเตอร์เนชั่นแนลทาวเวอร์  
ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120  
โทรศัพท์ 0-2962-1081 (อัตโนมัติ 10 คู่สาย)  
โทรสาร 0-2962-1084

สมาชิกสัมพันธ์

โทรศัพท์ 0-2962-1081-3 ต่อ 121  
โทรสาร 0-2962-1084

ร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2962-1081-3 ต่อ 112-114  
โทรสาร 0-2962-1084

พิมพ์ที่ บริษัท ดี.เค.ปรี้นติ้ง จำกัด

441/56 หมู่ 2 ต.บางบ่อ อ.บางบ่อ  
จ.สมุทรปราการ 10560  
โทรศัพท์ 0-2115-9105 โทรสาร 0-2115-9044



## PREFACE

หนังสือ “Practical Building a Powerful LINE OA for Developer” สร้างตัวเองให้เป็น LINE Developers เพื่อพัฒนา LINE OA ที่เหนือกว่า ถ่ายทอดจากประสบการณ์จริงในการพัฒนาระบบบันทึกการเข้าร่วมกิจกรรมโดยใช้ LINE Services (LINE APIs) ที่จะช่วยให้ผู้อ่านสามารถก้าวสู่การเป็น LINE Developers ได้จากการเรียนรู้ผ่านหนังสือเล่มนี้

โดยเนื้อหาในหนังสือเล่มนี้จะรวบรวมเทคโนโลยีของ LINE ที่เป็นพื้นฐานที่สำคัญเอาไว้ ได้แก่ LINE Official Account, LINE Messaging API, LINE Login, LINE Front-end Framework (LIFF), LINE Beacon, LINE Notify รวมไปถึงเทคโนโลยีอื่นๆ ที่เกี่ยวข้อง เช่น Firebase, Dialogflow, Postman และ ngrok ซึ่งจะช่วยอำนวยความสะดวกในการพัฒนาแอปพลิเคชันได้

สำหรับส่วนของการนำเสนอเนื้อหาจะเป็นเชิงสรุป และเน้นให้ผู้อ่านสามารถทำตามขั้นตอนที่นำเสนอได้ โดยใช้ตัวอย่างของระบบบันทึกการเข้าร่วมกิจกรรมเป็นกรณีศึกษา ซึ่งจะช่วยให้ผู้อ่านสามารถพัฒนาระบบดังกล่าว และนำไปประยุกต์ใช้งานได้จริง จึงเหมาะสำหรับนักเรียน นักศึกษา นักพัฒนา ที่มีพื้นฐานการเขียนโปรแกรมในระดับเบื้องต้นถึงระดับปานกลางที่ต้องการเป็น LINE Developers

**ผศ.ดร.วุฒิพงษ์ ชินศรี**

LINE Certified Coach for API 2023

ผู้ช่วยอธิการบดีฝ่ายนวัตกรรมและวิสาหกิจ

ผู้อำนวยการศูนย์บริการทางวิชาการ

มหาวิทยาลัยรังสิต

# APPRECIATION



“

หนังสือเล่มนี้ จะพาคุณไปรู้จักกับการพัฒนาบริการขึ้นไปอยู่บนแพลตฟอร์มของ LINE โดยผู้อ่านจะได้เรียนรู้เทคโนโลยี LINE APIs ทั้ง LINE Messaging API (Chatbot), LIFF, LINE Beacon, LINE Login และ LINE Notify ผ่านกรณีศึกษาที่ผู้เขียนได้ทำเป็นตัวอย่างร่วมกับเครื่องมือที่ได้รับความนิยมในการพัฒนาซอฟต์แวร์ เช่น Firebase, Dialogflow และ ngrok เป็นต้น ซึ่งเมื่อคุณอ่านหนังสือเล่มนี้แล้ว ก็สามารถนำไปต่อยอดได้ทันที ”

โดย **จิววัฒน์ กรณียวิทยาการ (ตี)**  
Google Developer Expert – Firebase  
Technology Evangelist, LINE Thailand



“

หนังสือเล่มนี้เป็นสิ่งที่มีติดตัวไว้เลยสำหรับ LINE Developers ทั้งผู้เริ่มต้นก็ดี หรือผู้ที่กำลังทำงานอยู่แล้วก็ดี เพราะเนื้อหาเน้นไปที่การหยิบเทคนิคต่างๆ การใช้ Services ของ LINE ในแง่มุมที่ไม่ควรพลาดไว้ได้ครบครัน ซึ่งจะช่วยให้ผู้อ่านได้ทั้งความรู้และการลงมือปฏิบัติจริง จนสุดท้ายนำไปปรับใช้กับองค์กรหรืองานที่กำลังทำอยู่ได้อย่างแน่นอน ”

โดย **แทนที โพธิ์จันต์ (โต)**  
LINE API Expert



## EDITOR'S NOTE

ตอนหนึ่งของบทความจากเพจการตลาดวันละตอน “ทำไม LINE ถึงตอบโจทย์ธุรกิจออนไลน์ในปี 2022? LINE ประเทศไทยประกาศว่า มียอดผู้ใช้งาน LINE ครบ 50 ล้านคน คนไทยอย่างน้อยถ้ามีโทรศัพท์มือถือในเครื่องต้องมีแอปพลิเคชัน LINE อยู่ในนั้น และเป็นผู้ใช้งานที่มีคุณภาพด้วย เนื่องจากการเปิดตัว LINE ต้องมีหมายเลขโทรศัพท์ จึงเป็นผู้ใช้งานตัวจริง ทุกวันนี้แทนที่จะขอเบอร์ติดต่อก็เปลี่ยนมาเป็นขอแอดไลน์แทนกันแล้ว”

ด้วย LINE App นั้นใช้ฟรี ใช้ง่าย และเข้าถึงง่าย จึงแพร่หลายอย่างมาก ธุรกิจออนไลน์และแบรนด์ต่างก็ใช้ LINE Official Account (LINE OA) ซึ่งเป็น Business Account เพิ่มช่องทางการขาย กระตุ้นยอดขาย ขยายฐานลูกค้า สร้างลูกค้าประจำ ฯลฯ จึงเป็นโอกาสและความท้าทายสำหรับ Developer ที่จะประยุกต์ใช้เพื่อเพิ่มขีดความสามารถ หรือสร้างประสบการณ์ใหม่ๆ ให้แก่ลูกค้า

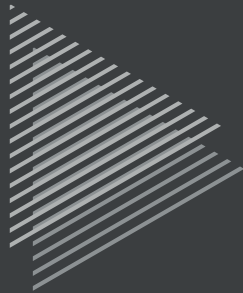
ตัวอย่างของการพัฒนาระบบหรือแอปพลิเคชันบน LINE App ที่พบได้ในชีวิตจริง เช่น ระบบ Chatbot หรือ Chat Commerce ที่ให้ข้อมูลกับลูกค้าได้แทนมนุษย์ ระบบส่งข้อความทักทายหรือส่งคูปองไปยังผู้ใช้งาน LINE ที่อยู่บริเวณใกล้ๆ ร้านค้า ระบบลงทะเบียนอัตโนมัติแบบไร้สัมผัส ระบบยืนยันการเข้า-ออกของพนักงานแทนการตอกบัตร หรือระบบสแกนใบหน้านักเรียนที่จะส่งข้อความแจ้งเตือนไปยัง LINE ของผู้ปกครอง หรือสแกนใบหน้าผู้ปกครองแล้วเชื่อมโยงได้ว่าเป็นพ่อหรือแม่ของเด็กคนไหน การดูเว็บในห้องแชทไลน์ได้เหมือนเว็บแอป

สุดท้ายนี้ ขอขอบพระคุณ ผศ.ดร.วุฒิพงษ์ ชินศรี ที่นำเสนอการพัฒนา LINE OA ได้อย่างครอบคลุม LINE Services หวังเป็นอย่างยิ่งว่าจะเป็นประโยชน์ต่อ Developer ทั้งในระดับผู้เริ่มต้นและมืออาชีพได้นำไปต่อยอดการทำงานและปรับใช้ในธุรกิจต่อไป

ภัสรา คุษาเจริญ  
บรรณาธิการ



# CONTENT



## Chapter 1

### INTRODUCTION เทคโนโลยีและบริการของ LINE

เหตุใด LINE จึงเป็น Platform	
ที่น่าสนใจสำหรับนักพัฒนา.....	10
Messaging API : ระบบโต้ตอบกับผู้ใช้งานผ่าน Chatbot.....	13
LINE Login : ระบบยืนยันตัวตนด้วย LINE Account .....	16
LIFF : เปลี่ยนห้องแชทให้แสดงผลได้ดีเหมือน Web App .....	17
LINE Beacon : ระบบส่งข้อความถึงผู้ใช้ LINE ในพื้นที่.....	18
LINE Notify : ระบบส่งข้อความแจ้งเตือนอัตโนมัติ.....	19
บทวิเคราะห์และสรุป .....	21

## Chapter 2

### DEVELOPER TOOLBOX เทคโนโลยีที่เกี่ยวข้องในการพัฒนา

รู้จักเทคโนโลยีในการพัฒนาระบบหรือแอปเพื่อใช้งานร่วมกับ LINE API.....	26
Visual Studio Code .....	26
Node.js.....	28
Postman.....	29
LINE Bot Designer .....	30
ngrok.....	31
Firebase.....	32
ตรวจสอบการติดตั้งซอฟต์แวร์ที่เกี่ยวข้อง .....	33
บทวิเคราะห์และสรุป .....	39

## Chapter 3

### LINE OA WORKSHOP : การพัฒนาระบบ บันทึกการเข้าร่วมกิจกรรม

การพัฒนาระบบบันทึกการเข้าร่วมกิจกรรม.....	44
สร้าง Chatbot (LINE OA).....	45
การเชื่อมต่อ LINE OA กับ Dialogflow.....	54
การใช้ Inline Editor ใน Dialogflow.....	63
การเขียน Webhook เพื่อประมวลผลข้อมูลจาก Dialogflow.....	67
บทวิเคราะห์และสรุป.....	83

## Chapter 4

### LINE MESSAGING API + LINE OA WORKSHOP : การพัฒนาระบบบันทึก การเข้าร่วมกิจกรรม

Push messages vs Reply messages ต่างกันอย่างไร.....	86
การใช้ Postman กับ LINE Messaging API.....	87
การสร้าง Webhook เพื่อใช้งาน LINE Messaging API.....	103
บทวิเคราะห์และสรุป.....	114

## Chapter 5

### LINE LOGIN WORKSHOP : การพัฒนา ระบบบันทึกการเข้าร่วมกิจกรรม

เตรียมสภาพแวดล้อมสำหรับพัฒนา LINE Login.....	118
การเริ่มต้นใช้งาน Cloud Firestore กับ Web Application.....	129

การใช้ LINE Login กับ Web Application.....	139
การใช้ LINE Login กับ LIFF.....	155
บทวิเคราะห์และสรุป.....	163

## Chapter 6

### LIFF WORKSHOP : การพัฒนาระบบ บันทึกการเข้าร่วมกิจกรรม

การประยุกต์ใช้ LIFF ในหน้าจอ ข้อมูลผู้ใช้งาน.....	175
การประยุกต์ใช้ LIFF ในหน้าจอ ข้อมูลการเข้าร่วมกิจกรรม.....	183
การประยุกต์ใช้ LIFF ในหน้าจอ ขอรับเกียรติบัตร.....	194
บทวิเคราะห์และสรุป.....	202

## Chapter 7

### LINE BEACON WORKSHOP : การพัฒนา ระบบบันทึกการเข้าร่วมกิจกรรม

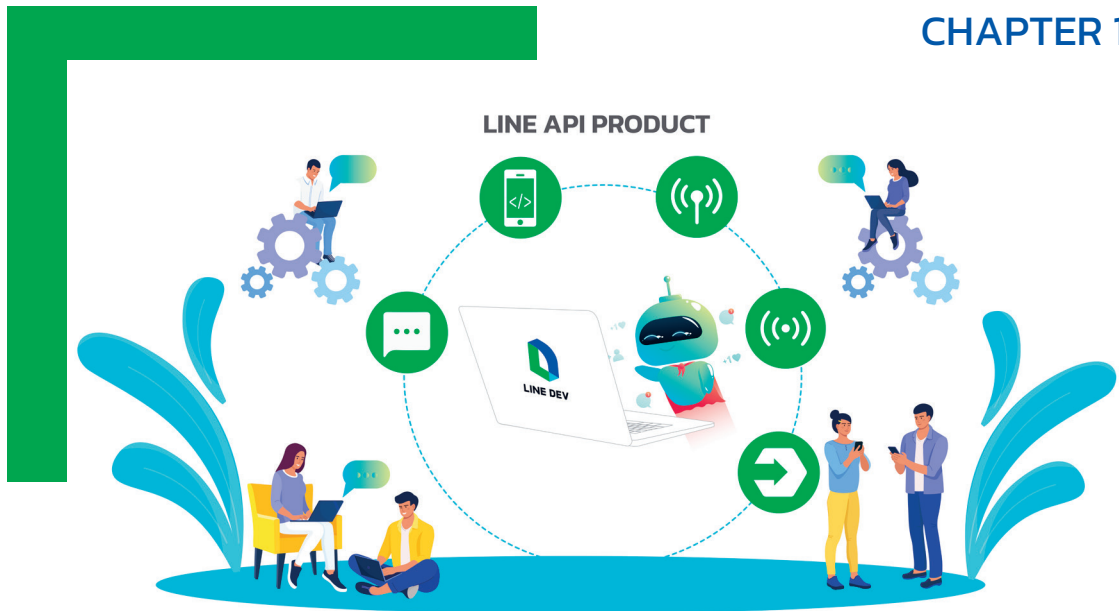
การพัฒนาระบบ LINE Beacon.....	206
การลงทะเบียนอุปกรณ์ Beacon.....	207
การสร้าง Webhook เพื่อใช้งาน LINE Messaging API ร่วมกับ LINE Beacon.....	210
บทวิเคราะห์และสรุป.....	219

## Chapter 8

### LINE NOTIFY WORKSHOP : การพัฒนา ระบบบันทึกการเข้าร่วมกิจกรรม

การสร้าง Token สำหรับใช้บริการ LINE Notify.....	224
บทวิเคราะห์และสรุป.....	230





# INTRODUCTION

## เทคโนโลยีและบริการ ของ LINE

ปัจจุบัน LINE กลายเป็นหนึ่งใน Platform ที่หน่วยงานต่างๆ นำไปใช้ในการให้บริการลูกค้า พนักงาน รวมไปถึงการประชาสัมพันธ์ข้อมูลข่าวสาร เพราะสามารถช่วยสร้างประสบการณ์ที่ดีให้กับผู้ใช้งานได้ ดังนั้นในมุมมองของนักพัฒนา จึงอยากที่จะพัฒนาระบบที่เกี่ยวข้องกับ LINE มากขึ้น ซึ่งทาง LINE เองนั้นได้มีเทคโนโลยีและบริการต่างๆ ที่เตรียมไว้ให้นักพัฒนาเลือกไปประยุกต์ใช้งานได้หลากหลาย สำหรับเนื้อหาในบทนี้จะนำเสนอเกี่ยวกับเทคโนโลยีและบริการของ LINE รวมไปถึงประเด็นที่เกี่ยวข้องโดยสรุป

## เหตุใด LINE จึงเป็น Platform ที่น่าสนใจสำหรับนักพัฒนา

LINE ได้จัดทำเว็บไซต์สำหรับนักพัฒนาในประเทศไทยอย่างเป็นทางการ โดยมีช่องทางในการให้บริการหรือสื่อสารกับนักพัฒนาที่หลากหลาย รวมไปถึงทาง LINE ได้จัดทำเว็บไซต์สำหรับนักพัฒนาในประเทศไทยอย่างเป็นทางการ ซึ่งสามารถเข้าถึงได้จากเว็บไซต์ตามลิงค์ได้รูป 1-1



### API Product

 <b>Messaging API</b> สื่อสารและโต้ตอบกับผู้ใช้งานผ่าน Chatbot	 <b>LINE Login</b> ให้ผู้ใช้ Login ด้วย LINE เพื่อเชื่อมต่อกั...	 <b>LIFF</b> ผลงานการทำงานระหว่างห้องแชทและ...
 <b>LINE Beacon</b> เชื่อมต่อกับ LINE ของคุณกับโลกแห่งความ...	 <b>LINE MINI APP</b> เชื่อมต่อบริการของคุณเข้ากับชีวิตประจำวัน...	 <b>LINE Notify</b> ไม่พลาดทุกการสื่อสารกับผู้ใช้บริการของคุณ...

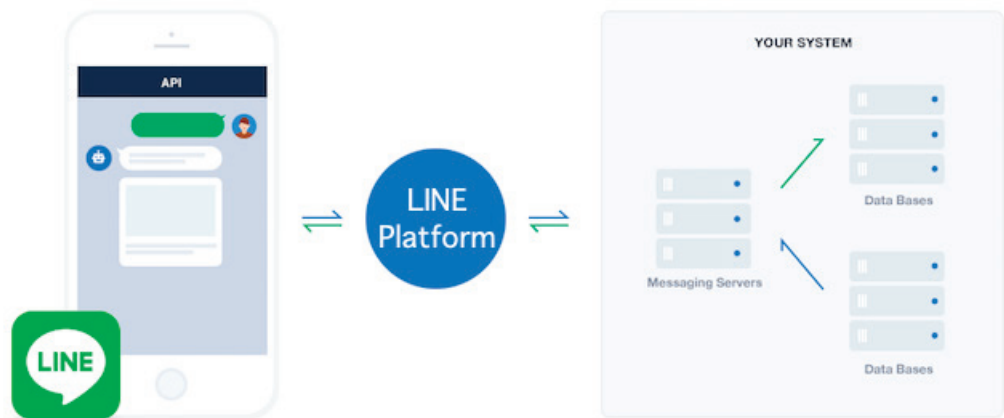
▶ รูปที่ 1-1 เว็บไซต์ LINE Developers Thailand

URL : <https://linedevth.line.me/th/>



## Messaging API : ระบบโต้ตอบกับผู้ใช้งานผ่าน Chatbot

เบื้องต้นคำว่า API (Applications Programming Interface) หรือส่วนต่อประสานโปรแกรมประยุกต์ คือการเขียนโค้ดเพื่อให้โปรแกรมอื่น ๆ หรือแอปพลิเคชันอื่น สามารถเชื่อมต่อกับโปรแกรมหรือฐานข้อมูลของเราได้ ซึ่งการสร้าง API นั้น จำเป็นต้องกำหนดข้อตกลงในการรับส่งข้อมูลให้ชัดเจน เพื่อที่ผู้พัฒนาโปรแกรมเชื่อมต่อกับ API จะสามารถเข้าใจว่าโปรแกรมจะติดต่อสื่อสารกันอย่างไร ข้อมูลที่ส่งไปต้องส่งรูปแบบไหน และ API จะส่งข้อมูลอะไรกลับมาในรูปแบบใด ในช่วงก่อนหน้านี้เรามักทำ API ในรูปแบบของ Web Service โดยใช้ SOAP (Simple Object Access Protocol) ซึ่งจะรับส่งข้อมูลในรูปแบบของ XML (Extensible Markup Language) แต่ปัจจุบัน API มักอยู่ในรูปแบบของ REST (Representational State Transfer) ซึ่งรับส่งข้อมูลในรูปแบบของ JSON (JavaScript Object Notation) LINE ได้ให้บริการ Messaging API สำหรับใช้พัฒนา Chatbot ซึ่งการทำงานของ Messaging API จะอนุญาตให้มีการรับส่งข้อมูลระหว่าง Chatbot กับ LINE Platform ซึ่งจะต้องส่งผ่าน HTTPS (Hypertext Transfer Protocol Secure) ในรูปแบบของ JSON ซึ่งจะมีขั้นตอนการทำงานดังรูป 1-3



▶ รูปที่ 1-3 การทำงานในภาพรวมของ Messaging API

URL : <https://developers.line.biz/en/docs/messaging-api/overview/>

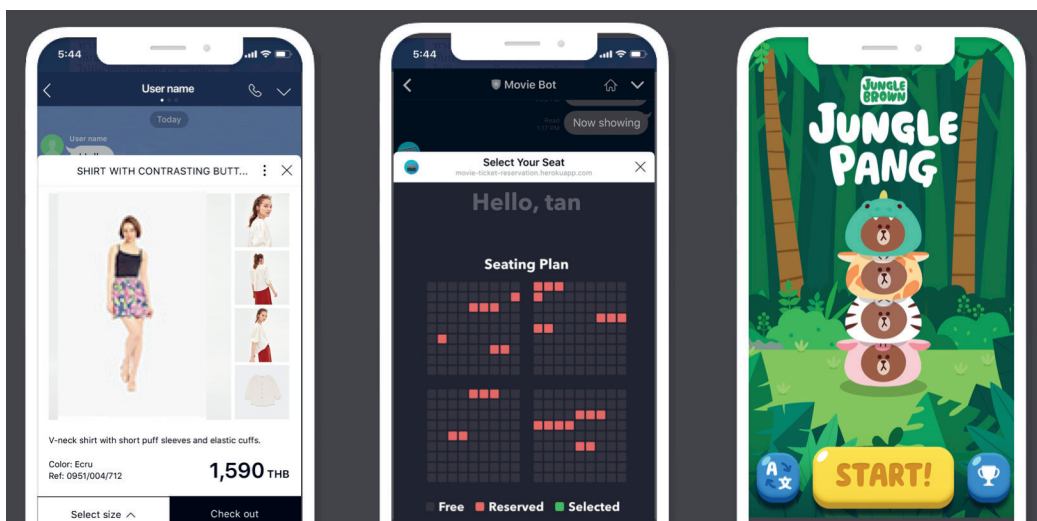


## LIFF : เปลี่ยนห้องแชทให้แสดงผลได้ดีเหมือน Web App

LIFF (LINE Front-end Framework) เป็นเทคโนโลยีที่ช่วยให้นักพัฒนาสามารถใช้งานห้องแชทและเว็บไซต์ร่วมกันได้ดียิ่งขึ้น ซึ่งจะช่วยให้การใช้งานบางอย่างของผู้ใช้ทำได้สะดวกขึ้น เช่น การกรอกข้อมูลลงทะเบียนแบบฟอร์ม หากต้องดำเนินการผ่านการแชทอาจจะทำได้ไม่สะดวกเท่ากับการกรอกข้อมูลผ่านแบบฟอร์ม ซึ่งเราสามารถนำเว็บไซต์ที่มีอยู่แล้วมาแสดงผลผ่าน LIFF ซึ่งเป็น WebView ที่สามารถรันเว็บไซต์อยู่ภายในแอป LINE ได้โดยตรง

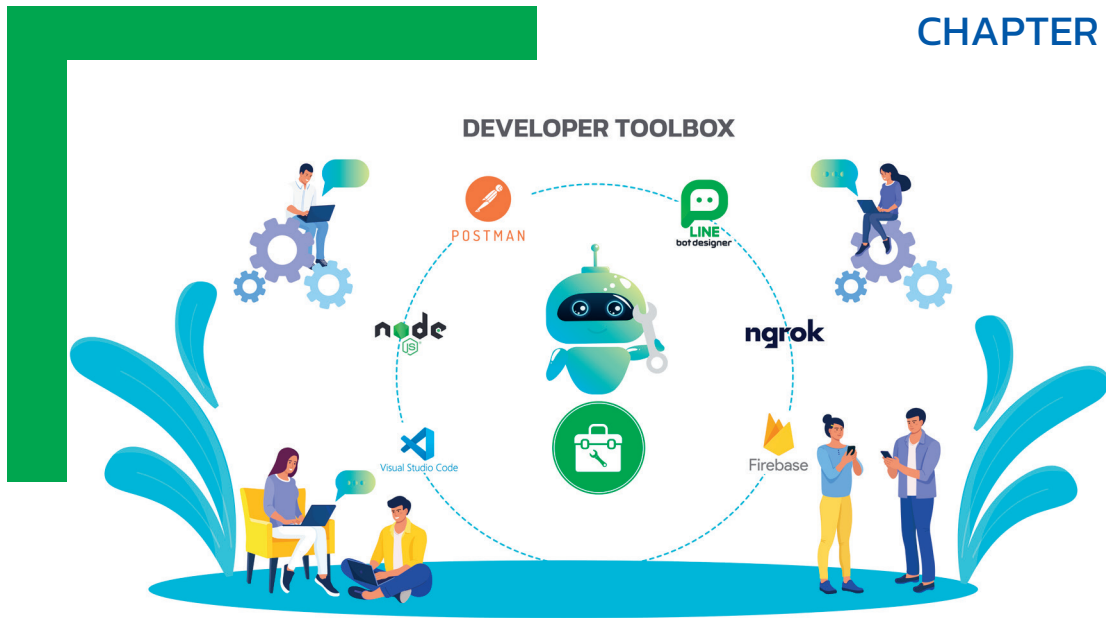
นอกจากการนำเว็บไซต์มาแสดงผลแล้ว LIFF ยังสามารถใช้งานร่วมกับ LINE Login ได้ด้วย โดยนักพัฒนาสามารถเขียนโปรแกรมเพิ่มเติมเพื่อเข้าถึงข้อมูลพื้นฐานของผู้ใช้ เช่น `userId`, `displayImage`, `statusMessage`

สำหรับการใช้งานเว็บไซต์ใน LIFF นั้น นักพัฒนาสามารถเขียนโปรแกรมเพื่อเรียกใช้งานกล้องเพื่อ Scan QR Code ได้ด้วย รวมไปถึงการดำเนินการต่างๆ เพื่อส่งข้อความในรูปแบบต่างๆ กลับไปที่ห้องแชท



รูปที่ 1-6 ตัวอย่างการแสดงผลเว็บไซต์ในแอป LINE ผ่าน LIFF

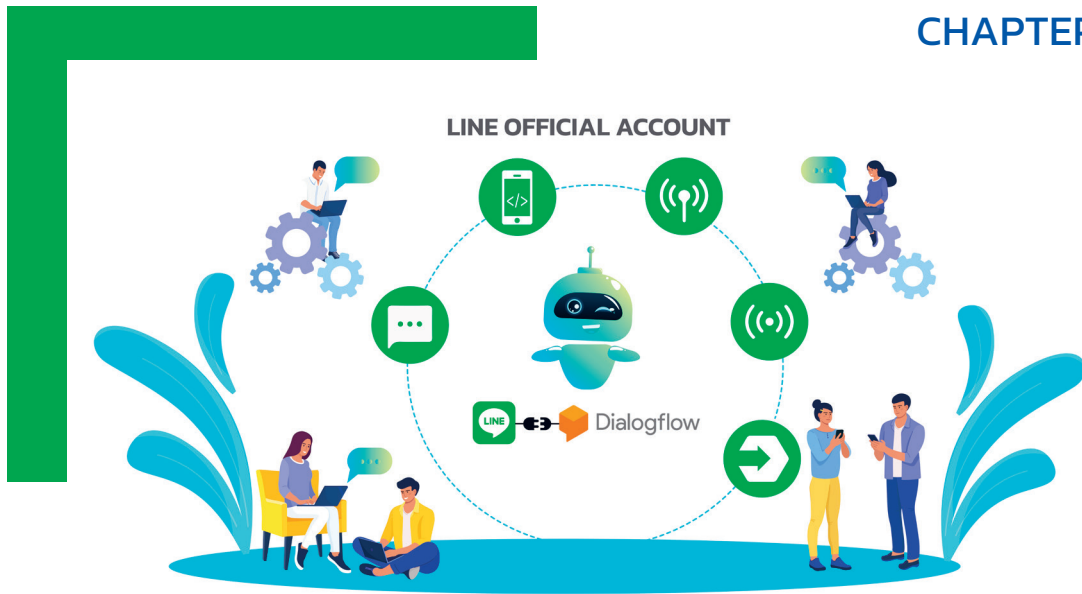
Credit : <https://linedevth.line.me/th/liff>



## DEVELOPER TOOLBOX

# เทคโนโลยีที่เกี่ยวข้อง ในการพัฒนา

- ▶ ในการพัฒนาระบบหรือแอปพลิเคชันเพื่อใช้งานร่วมกับ LINE API นั้น สามารถเลือกใช้เทคโนโลยีได้หลากหลาย ขึ้นอยู่กับความต้องการและความพร้อมของนักพัฒนา สำหรับในหนังสือเล่มนี้จะนำเสนอเทคโนโลยีที่เกี่ยวข้อง ซึ่งผู้เขียนพิจารณาว่าเหมาะสมสำหรับผู้เริ่มต้น และสามารถต่อยอดไปได้ถึงการใช้งานจริงต่อไปในอนาคต



## LINE OA WORKSHOP :

# การพัฒนาระบบบันทึก การเข้าร่วมกิจกรรม

▶ ในบทนี้จะนำเสนอขั้นตอนการพัฒนาบบบันทึกการเข้าร่วมกิจกรรม ซึ่งจะมีการประยุกต์ใช้บริการของ LINE ตามที่ได้แนะนำไว้ก่อนหน้านี้ ที่จะช่วยให้ผู้อ่านได้เรียนรู้การใช้งาน LINE API ได้จากระบบที่พัฒนาขึ้น พร้อมทั้งตัวอย่างในการประยุกต์ใช้งาน



## การพัฒนาระบบบันทึกการเข้าร่วมกิจกรรม

สำหรับการพัฒนาระบบบันทึกการเข้าร่วมกิจกรรม หนังสือเล่มนี้จะแบ่งการนำเสนอเป็น 6 ส่วนคือ

### 1. การสร้าง LINE OA หรือ LINE Chatbot เพื่อเป็นช่องทางในการสื่อสาร :

จะนำเสนอตัวอย่างในการสร้าง Chatbot เพื่อตอบคำถามหรือให้ข้อมูลที่เกี่ยวข้อง โดยใช้ Dialogflow ช่วยในการประมวลผลข้อความที่ได้รับมาจากผู้ใช้

### 2. การใช้ LINE Messaging API เพื่อโต้ตอบกับผู้ใช้งาน :

จะนำเสนอตัวอย่างการประยุกต์ใช้ LINE Messaging API ในการสื่อสารกับผู้ใช้ รวมไปถึงใช้ในการสร้างหรือจัดการเกี่ยวกับ Rich Menu ด้วย

### 3. การใช้ LINE Login ในการยืนยันตัวตนผู้ใช้ :

จะนำเสนอตัวอย่างการใช้ LINE Login ในการยืนยันตัวตนด้วย LINE Account ร่วมกับ Web Application และ LIFF App

### 4. การใช้ LIFF เพื่อแสดงหน้าจอที่เกี่ยวข้องกับการบันทึกการเข้าร่วมกิจกรรม :

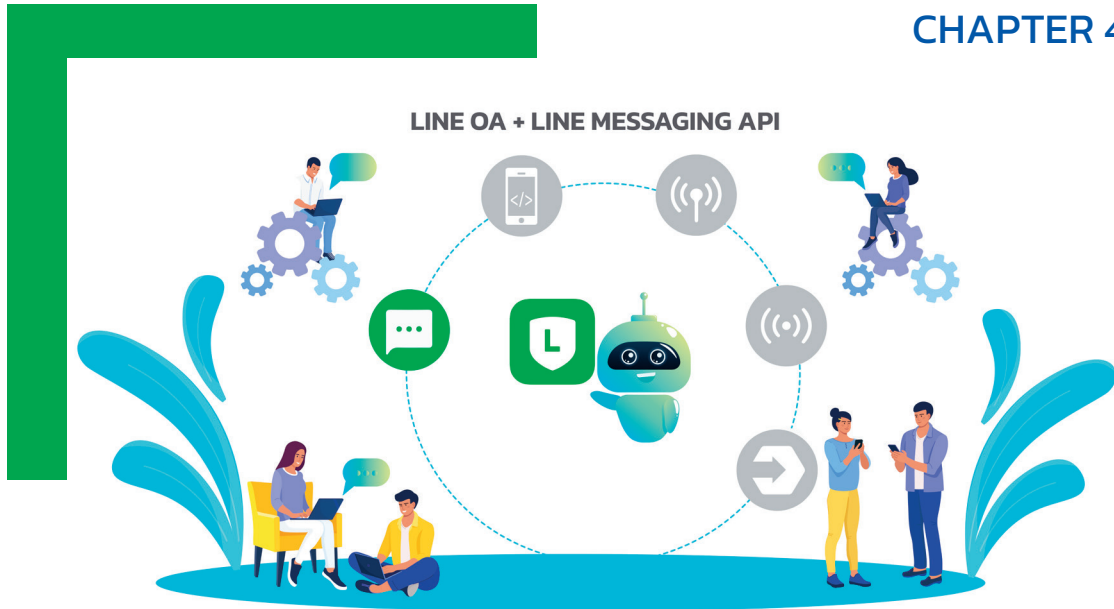
จะนำเสนอตัวอย่างการใช้ LIFF เพื่อสร้างหน้าจอในการบันทึกข้อมูลผู้ใช้ ข้อมูลการเข้าร่วมกิจกรรม และการขอรับเกียรติบัตร

### 5. การใช้ LINE Beacon เพื่อตรวจสอบการเข้ามาในพื้นที่กิจกรรม :

จะนำเสนอตัวอย่างการใช้ LINE Beacon ส่งข้อความเพื่อทักทายผู้ใช้ที่เข้ามาในพื้นที่กิจกรรม พร้อมกับป้อนยืนยันการเข้าร่วมกิจกรรม

### 6. การใช้ LINE Notify เพื่อแจ้งเตือนผู้ดูแลกิจกรรมให้ทราบว่าผู้ใช้เข้ามาในพื้นที่กิจกรรม :

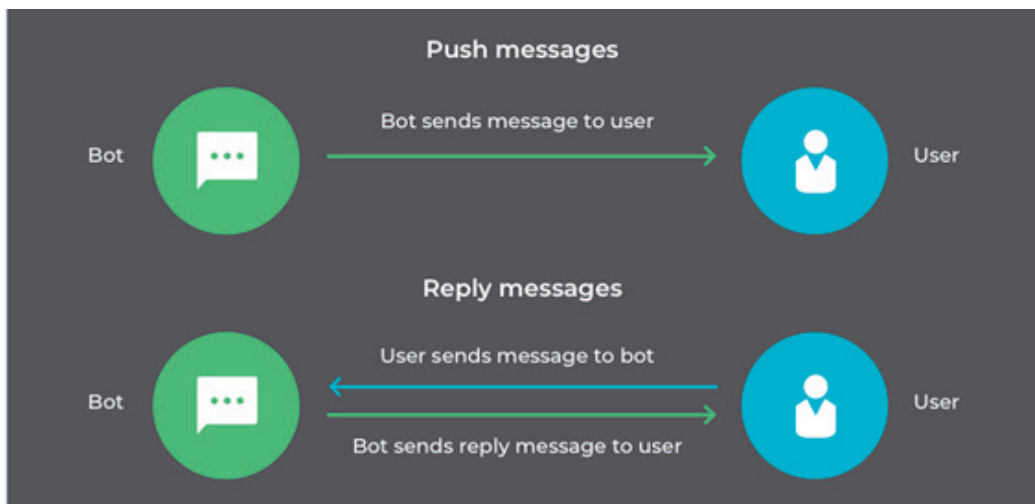
จะนำเสนอตัวอย่างการใช้ LINE Notify ร่วมกับ LINE Beacon เพื่อส่งข้อความแจ้งเตือนไปยังผู้ดูแลกิจกรรมให้ทราบได้ว่ามีผู้ใช้งานเข้ามาในพื้นที่กิจกรรมแล้ว



## LINE MESSAGING API + LINE OA WORKSHOP : การพัฒนาาระบบบันทึก การเข้าร่วมกิจกรรม

- ▶ LINE OA ที่สร้างขึ้นนั้นสามารถตอบโต้การใช้งานสำหรับผู้ใช้งานทั่วไปได้ระดับหนึ่ง ซึ่งหากเราต้องการให้ LINE OA เพิ่มความสามารถบางอย่าง หรือสามารถสร้างประสบการณ์ที่ดีให้กับผู้ใช้งาน เราสามารถใช้ LINE Messaging API มาช่วยได้ ซึ่งเนื้อหาในบทนี้จะนำเสนอเกี่ยวกับการใช้ LINE Messaging API ร่วมกับเครื่องมือหรือซอฟต์แวร์ที่เกี่ยวข้อง รวมไปถึงการเขียนโปรแกรมในส่วนของ Webhook เพื่อใช้งาน LINE Messaging API





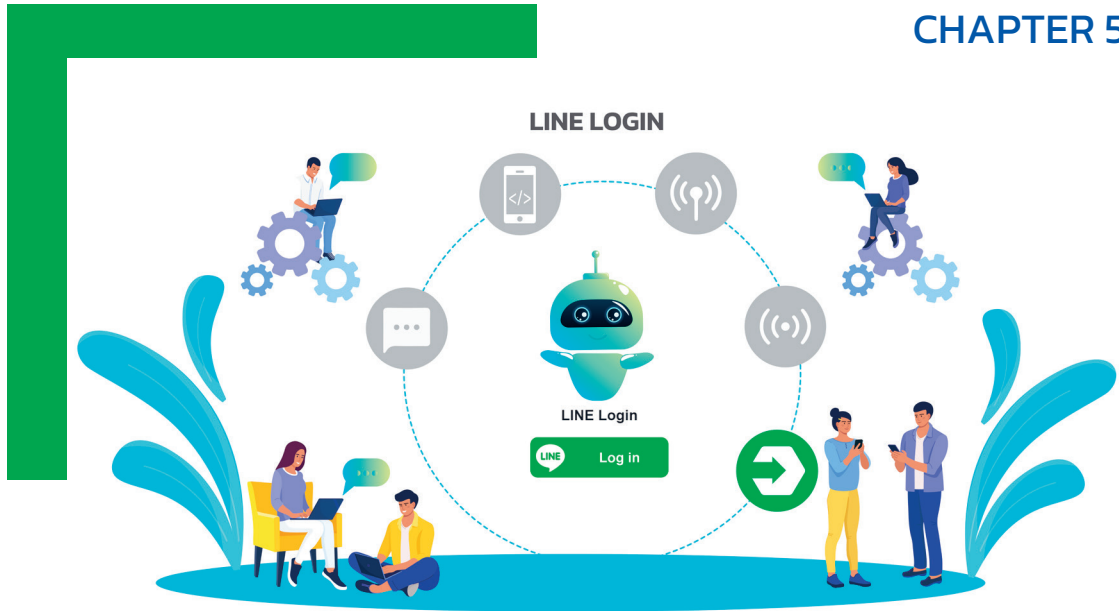
▶ รูปที่ 4-1 การทำงานในภาพรวมของ Push messages และ Reply messages

Credit : <https://developers.line.biz/en/services/messaging-api/>

ในหนังสือเล่มนี้จะอธิบายวิธีการส่งข้อความโดยใช้ LINE Messaging API ผ่าน Postman และการเขียนโปรแกรมเพื่อให้ผู้อ่านสามารถเรียนรู้และใช้งานได้ ส่วนรายละเอียดจะแนะนำให้ศึกษาเพิ่มเติมจากเอกสารฉบับทางการของแต่ละบริการ เนื่องจากเนื้อหาหรือวิธีการอาจจะมีการเปลี่ยนแปลงตามการพัฒนาของ LINE หรือส่วนอื่นๆ ที่เกี่ยวข้อง

## การใช้ Postman กับ LINE Messaging API

หลังจากที่เราติดตั้ง Postman แล้วในบทที่ 2 ในส่วนของ Workspace เราสามารถเพิ่ม Collection ซึ่งเป็นการกำหนดรายละเอียดที่จะเรียกใช้งาน API เอาไว้ ซึ่งในหนังสือเล่มนี้จะนำ Collection ที่คุณโอ้-เทพนที โพธิ์จันทร์ (<https://medium.com/@thepnateephojan>) ซึ่งเป็น LINE API Expert ได้ทำเอาไว้ โดยให้คลิกปุ่ม Import จะปรากฏหน้าจอตั้งรูป 4-2



## LINE LOGIN WORKSHOP :

# การพัฒนาาระบบบันทึก การเข้าร่วมกิจกรรม

- ▶ ระบบส่วนใหญ่จะมีการพัฒนาในส่วนของการบริหารจัดการข้อมูล รวมไปถึงรายงานที่เกี่ยวข้องเอาไว้ ทั้งนี้เพื่อให้ผู้ดูแลระบบเข้าไปใช้งาน ซึ่งเป็นเรื่องปกติที่การเข้าใช้งานในส่วนดังกล่าวจำเป็นต้องมีการยืนยันตัวตน ซึ่ง LINE เองก็มีบริการในส่วนของ LINE Login สำหรับใช้ในการยืนยันตัวตนด้วย LINE Account ซึ่งเนื้อหาในบทนี้จะนำเสนอแนวทางในการใช้ LINE Login ร่วมกับ Web App และ LINE Login ร่วมกับ LIFF

สำหรับในส่วนของ Profile สามารถเรียกดูข้อมูลได้จาก Method `getProfile` ซึ่งจะมีการเรียกใช้คำสั่ง `liff.getProfile` ตามโค้ดในบรรทัดที่ 203 โดยหากได้รับข้อมูลมาแล้วจะนำมาจัดเก็บในตัวแปร `profile` ตามโค้ดในบรรทัดที่ 206 และตรวจสอบข้อมูลว่ามีอยู่ในฐานข้อมูลหรือไม่ โดยการเรียกใช้ Method `checkAdmin` ตามโค้ดในบรรทัดที่ 207 ดังรูป 5-64

```
202         getProfile() {
203             liff
204                 .getProfile()
205                 .then((profile) => {
206                     this.profile = profile;
207                     this.checkAdmin();
208                     console.log(this.profile);
209                 })
210                 .catch((err) => {
211                     console.log("error", err);
212                 });
213         },
```

▶ รูปที่ 5-64 โค้ดในไฟล์ `liff.html` ในส่วนของ Method `getProfile`

ส่วนการ Logout นั้นสามารถเรียกใช้ Method `logout` ซึ่งจะใช้คำสั่ง `liff.logout` ดังรูป 5-65

```
215         logout() {
216             if (liff.isLoggedIn()) {
217                 liff.logout();
218                 this.login = false;
219             }
220         },
221     },
```

▶ รูปที่ 5-65 โค้ดในไฟล์ `liff.html` ในส่วนของ Method `logout`

สำหรับในส่วนของ Method `checkAdmin` ในภาพรวมจะเหมือนเดิมที่เคยใช้ไฟล์ `index.html` ในหัวข้อก่อนหน้านี้ แต่จะมีการเปลี่ยนการอ้างถึงตัวแปรในโค้ดบรรทัดที่ 164 (`this.profile.userId`) ซึ่งหากไม่พบผู้ใช้ในฐานข้อมูลจะใช้คำสั่ง `liff.logout` ตามโค้ดในบรรทัดที่ 174 ดังรูป 5-66



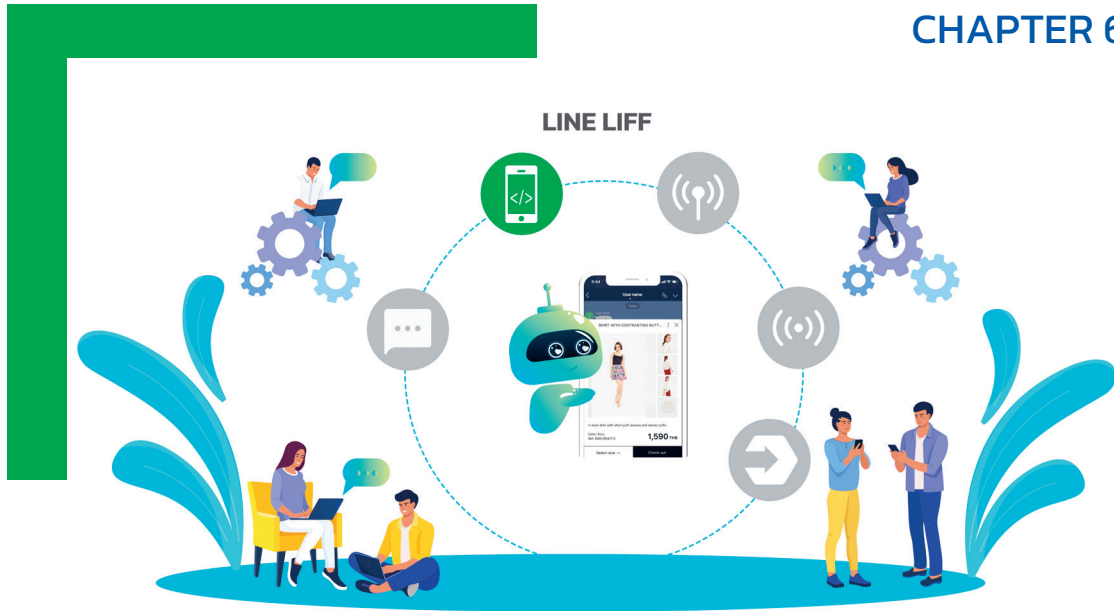
```
160     async checkAdmin() {
161         console.log("Check Admin");
162         const q = query(
163             collection(db, "admin"),
164             where("userId", "=", this.profile.userId)
165         );
166         const unsubscribe = onSnapshot(q, (querySnapshot) => {
167             console.log(querySnapshot.size);
168             if (querySnapshot.size != 0) {
169                 querySnapshot.forEach((doc) => {
170                     console.log(doc.data());
171                 });
172                 this.login = true;
173             } else {
174                 liff.logout();
175                 this.login = false;
176             }
177         });
178     },
```

▶ รูปที่ 5-66 โค้ดในไฟล์ liff.html ในส่วนของ Method checkAdmin

#### ▶ NOTE:

ข้อมูลในรูปแบบ JSON ที่จัดเก็บ Profile ที่ได้จาก liff.getProfile จะมีการกำหนด key ที่แตกต่างกับการเรียกใช้ API เพื่อ getProfile เช่น ข้อมูล UserId ส่วนของ liff.getProfile จะใช้ key userId ในส่วนของการเรียกใช้ API จะใช้ key sub

เมื่อถึงขั้นตอนนี้ เราสามารถใช้ LINE Login ผ่าน LIFF App ได้ ซึ่งหากนำ LIFF URL ไปเปิดใน LINE จะปรากฏหน้าจอจัดรูป 5-67









## LIFF WORKSHOP : การพัฒนาาระบบบันทึก การเข้าร่วมกิจกรรม

- ▶ การพัฒนาแอปพลิเคชันด้วย LIFF (LINE Front-end Framework) ช่วยให้ LINE สามารถเพิ่มขีดความสามารถในการให้บริการต่างๆ กับผู้ใช้ได้อีกมากมาย นอกจากการโต้ตอบกับผู้ใช้ผ่านการแชท ซึ่งจะช่วยให้เราใช้งานห้องแชทและเว็บไซต์ร่วมกันได้ดียิ่งขึ้น โดยในบทนี้จะนำเสนอการใช้งาน LIFF ในการบันทึกข้อมูลผู้ใช้ ข้อมูลการเข้าร่วมกิจกรรม และการขอรับเกียรติบัตร ซึ่งจะช่วยให้ผู้อ่านได้เรียนรู้การใช้งานที่เชื่อมโยงกันระหว่างห้องแชทและเว็บไซต์ได้เป็นอย่างดี



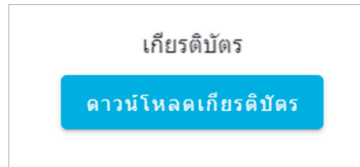
เมื่อมีข้อมูลแล้ว ระบบจะแสดงผลฐานกิจกรรมที่ระบุเอาไว้ ดังรูป 6-23

ข้อมูลการเข้าร่วมกิจกรรมของคุณ	
 Wutthipong	
Sort by <span style="float: right;">▼</span>	
หลักสูตร	สาขาวิชานวัตกรรมดิจิทัล
สถานะการเข้าร่วม	ยังไม่ได้เข้าร่วมกิจกรรม 
หลักสูตร	สาขาวิชาวิทยาการคอมพิวเตอร์
สถานะการเข้าร่วม	ยังไม่ได้เข้าร่วมกิจกรรม 
หลักสูตร	สาขาวิชาคอมพิวเตอร์เกมและอีสปอร์ต
สถานะการเข้าร่วม	ยังไม่ได้เข้าร่วมกิจกรรม 
หลักสูตร	สาขาวิชาสารสนเทศการลงทุน
สถานะการเข้าร่วม	ยังไม่ได้เข้าร่วมกิจกรรม 
หลักสูตร	สาขาวิชาเทคโนโลยีสื่อสังคม
สถานะการเข้าร่วม	ยังไม่ได้เข้าร่วมกิจกรรม 
Rows per page:	10 ▼ 1-5 of 5

▶ รูปที่ 6-23 หน้าจอข้อมูลการเข้าร่วมกิจกรรมเมื่อเปิดครั้งแรก

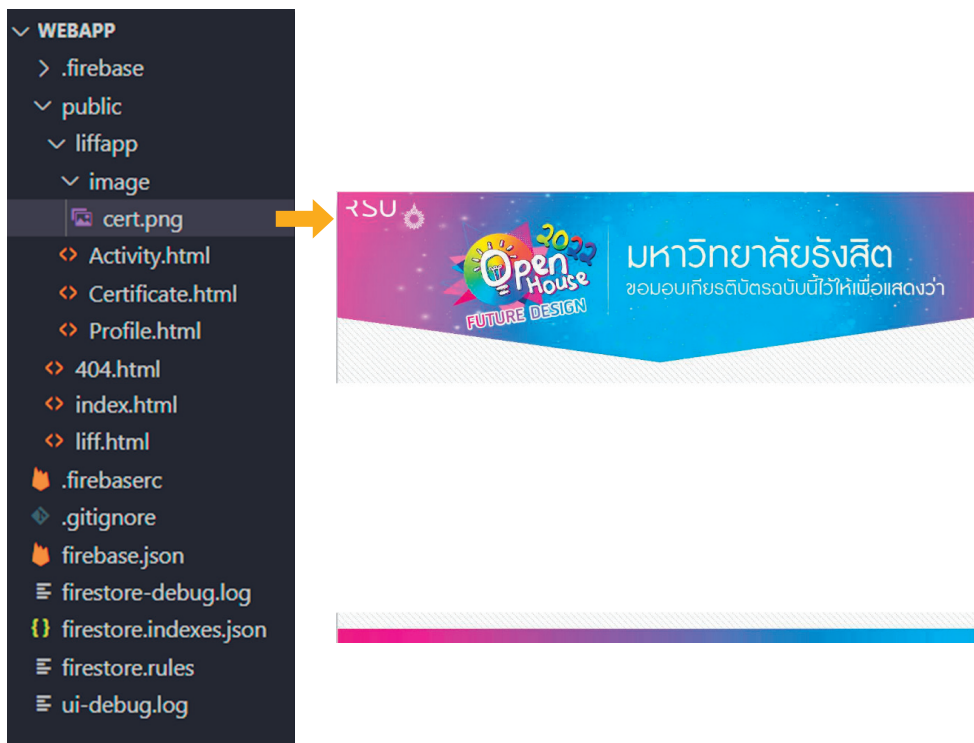


ในกรณีที่ที่มีการเข้าร่วมกิจกรรมแล้ว ระบบจะแสดงปุ่มสำหรับดาวน์โหลดเกียรติบัตร ดังรูป 6-34



▶ รูปที่ 6-34 หน้าจอการขอรับเกียรติบัตรเมื่อเข้าร่วมกิจกรรมแล้ว

เมื่อกดปุ่มดาวน์โหลดเกียรติบัตรแล้ว จะมีการเรียกใช้ Method `genCertAsImage` ซึ่งจะตรวจสอบว่ามีข้อมูล `certURL` อยู่ไหม หากมีอยู่แล้วก็จะเรียกใช้ Method `liffSendImage` เลย แต่หากไม่มีจะใช้ `canvas` เพื่อเปิดรูปเกียรติบัตรที่กำหนด โดยสามารถเปิดดูภาพเกียรติบัตรตัวอย่างได้จาก `LINEDev SourceCode > LINEDev-06 > Page-195` และนำมาเก็บในโฟลเดอร์ `image` ดังรูป 6-35



▶ รูปที่ 6-35 โครงสร้างของไฟล์และโฟลเดอร์ที่ใช้จัดเก็บไฟล์ภาพเกียรติบัตร



```
248     uploadToStorage() {
249         this.loadingUpload = true;
250         const today = new Date();
251         const yyyy = today.getFullYear();
252         let mm = today.getMonth() + 1; // Months start at 0!
253         let dd = today.getDate();
254         if (dd < 10) dd = "0" + dd;
255         if (mm < 10) mm = "0" + mm;
256         const formattedToday = dd + mm + yyyy;
257         var storageRef = ref(
258             storage,
259             "certificate/" + formattedToday + "/" + this.join.name + ".jpg"
260         );
261
262         console.log(this.cert);
263         uploadString(storageRef, this.cert, "data_url").then((snapshot) => {
264             console.log("Uploaded a data_url string!");
265             getDownloadURL(storageRef).then((downloadURL) => {
266                 console.log("File available at", downloadURL);
267                 this.join.certURL = downloadURL;
268                 this.join.genCertDate = Timestamp.fromDate(new Date());
269                 this.updateActivity();
270             });
271         });
272     },
```

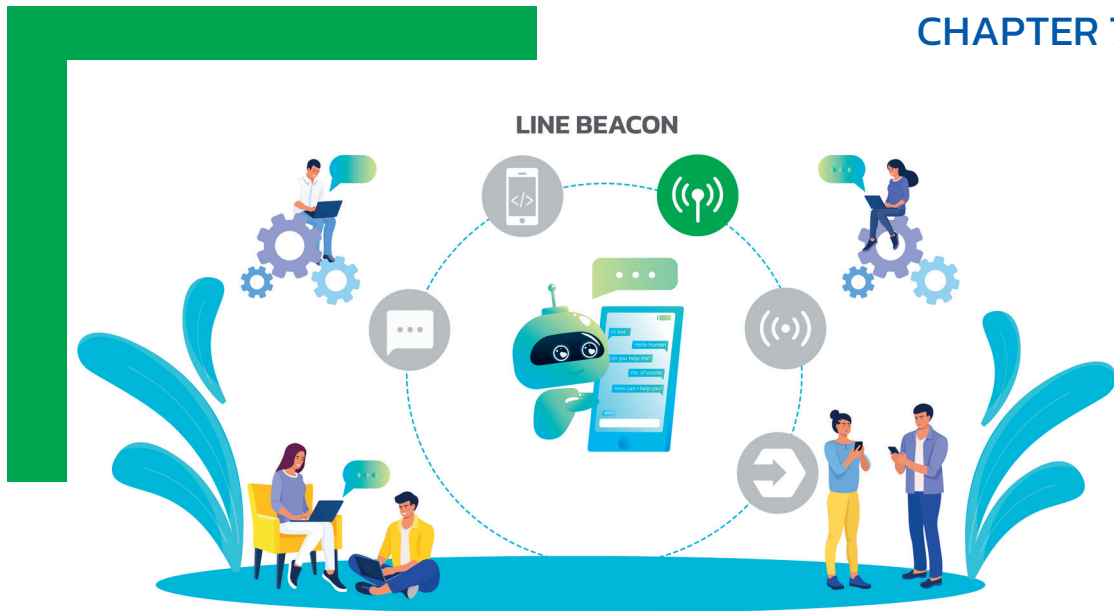
▶ รูปที่ 6-37 โค้ดในส่วนของ Method uploadToStorage ที่ตรวจสอบการเข้าร่วมกิจกรรม

โค้ดในส่วนของ Method updateActivity เมื่ออัปเดตเสร็จแล้ว จะเรียกใช้ Method liffSendImage เพื่อส่งข้อความและภาพกลับไปห้องแชทตามโค้ดในบรรทัดที่ 308 ดังรูป 6-38

```
301     async updateActivity() {
302         const userRef = doc(db, "activitytransaction", this.profile.userId);
303
304         try {
305             await setDoc(userRef, this.join, {
306                 merge: true,
307             });
308             this.liffSendImage(this.join.name, this.join.certURL);
309         } catch (error) {
310             console.error("error", error);
311         }
312     },
313
```

▶ รูปที่ 6-38 โค้ดในส่วนของ Method updateActivity ที่ตรวจสอบการเข้าร่วมกิจกรรม





## LINE BEACON WORKSHOP :

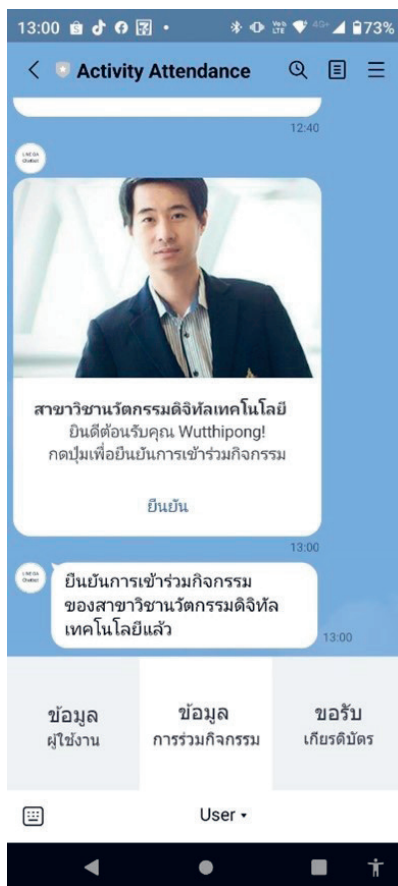
# การพัฒนาาระบบ บันทึกการเข้าร่วมกิจกรรม

การใช้งาน LINE Beacon จะช่วยให้ LINE สามารถเพิ่มขีดความสามารถในการให้บริการ โดยไม่ต้องรอให้ผู้ใช้งานเป็นผู้เริ่มต้นการสื่อสารก่อน และไม่ใช้การใช้ Push message ที่คิดโควตการใช้งาน ซึ่ง LINE Beacon จะกระจายสัญญาณบลูทูธ (Bluetooth Low Energy ; BLE) ออกไป ซึ่งหากผู้ใช้งาน LINE OA เปิดใช้งาน Bluetooth และ Location และอยู่ในรัศมีของอุปกรณ์ Beacon ก็จะมีการส่งสัญญาณให้ LINE Platform ทราบว่ามีผู้ใช้งานเข้ามาในบริเวณนั้น เราจะนำจุดเด่นตรงนี้มาประยุกต์การใช้งานได้ โดยในบทนี้จะนำเสนอแนวทางการส่งข้อความ เพื่อชักจูงผู้ใช้ที่เข้ามาในพื้นที่กิจกรรม พร้อมกับปุ่มยืนยันการเข้าร่วมกิจกรรม เพื่อให้ผู้ใช้สามารถยืนยันการเข้าร่วมกิจกรรมได้ด้วยตนเอง

**NOTE :**

ในกรณีที่ใช้ URL จาก ngrok ระบบจะตรวจสอบข้อมูลจาก Firestore Emulator แต่หากทำการ Deploy แล้วใช้ URL จาก Cloud Functions ระบบจะตรวจสอบข้อมูลจาก Cloud Firestore

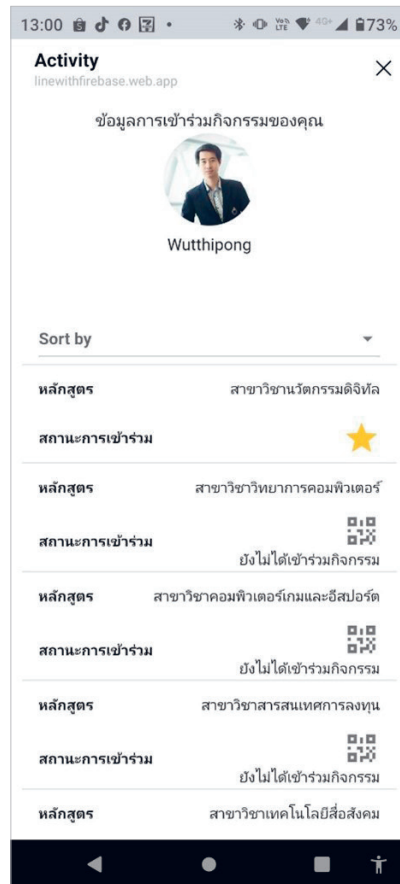
โดยหากมีการกดปุ่ม นอกจากมีการอัปเดตข้อมูลใน Cloud Firestore แล้ว จะมีการเรียกใช้ฟังก์ชัน reply ในไฟล์ util.js ซึ่งจะมีข้อความยืนยันการเข้าร่วมกิจกรรมปรากฏในหน้าจอ LINE OA ดังรูป 7-14



▶ รูปที่ 7-14 หน้าจอ LINE OA ที่ได้รับข้อความยืนยันหลังจาก Webhook ประมวลผลในส่วนของ postback เรียบร้อย



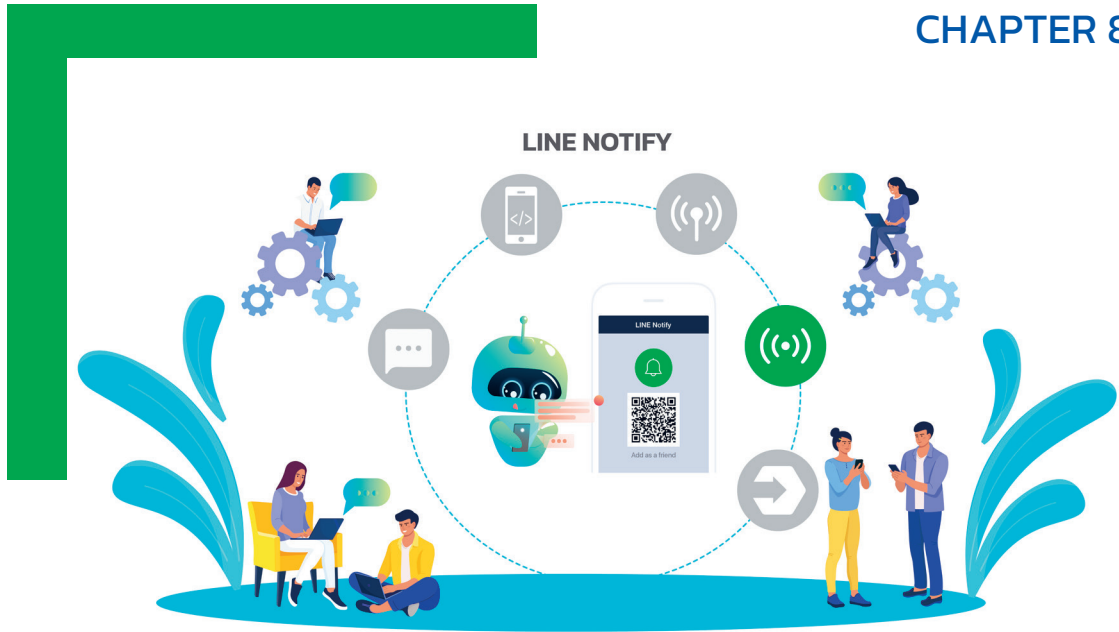
เมื่อเราเปิดหน้าจอข้อมูลการเข้าร่วมกิจกรรม ก็พบว่า มีไอคอนรูปดาวปรากฏที่กิจกรรมที่กำหนดไว้ให้ได้รับการยืนยัน ดังรูป 7-15



▶ รูปที่ 7-15 หน้าจอข้อมูลการเข้าร่วมกิจกรรมที่ได้รับการยืนยัน  
โดยการกดปุ่มยืนยันใน Flex Message

สังเกตว่า บริเวณไอคอนรูปดาวจะไม่ปรากฏวันเวลาที่ยืนยัน เพราะไม่ได้เขียนโปรแกรมเอาไว้ให้บันทึกวันเวลาเหมือนกับเนื้อหาในบทที่แล้ว

สำหรับในกรณีสัญญาณชนิด “banner” นั้น จะเกิดจากการกด Banner ที่ปรากฏอยู่ตรงบริเวณด้านบนของ Friend List หรือ Chat List ดังรูป 7-16



## LINE NOTIFY WORKSHOP :

# การพัฒนาาระบบ บันทึกการเข้าร่วมกิจกรรม

- LINE Notify เป็น LINE Official Account แบบพิเศษที่ทาง LINE มีไว้เพื่อให้บริการการแจ้งเตือน โดย LINE มี APIs ให้นักพัฒนาสามารถนำไปใช้เพื่อสั่งให้ LINE Notify ส่งข้อความในรูปแบบต่างๆ ไปยัง LINE Account หรือ LINE Group ได้ ซึ่งเมื่อพัฒนาเรียบร้อยแล้ว ผู้ใช้งานสามารถเพิ่ม LINE Notify เป็นเพื่อน หรือเพิ่ม LINE Notify เข้าไปในกลุ่มที่กำหนดไว้ โดยเนื้อหาในบทนี้ จะทำระบบที่คอยแจ้งเตือนให้ผู้ที่ดูแลงานกิจกรรมทราบว่า มีผู้ใช้งานเข้ามาในพื้นที่แล้ว ซึ่งจะเป็นการประยุกต์ใช้งานร่วมกับ LINE Beacon นั้นเอง



จากนั้นให้ Copy ไฟล์ในโฟลเดอร์ LINEDev SourceCode > LINEDev-08 > Page-227 ซึ่งจะมี 2 ไฟล์ ได้แก่ index.js และ util.js มาใส่ในโฟลเดอร์ functions แล้วเปิดไฟล์ util.js และนำ Token ที่ Copy ไว้มาวางในโค้ดบรรทัดที่ 100 สำหรับข้อมูลที่ส่งไปนั้นจะอยู่ในรูปของตัว Text ที่ส่งผ่าน Form เราจึงใช้ qs เข้ามาช่วย ตามโค้ดในบรรทัดที่ 112 ดังรูป 8-6

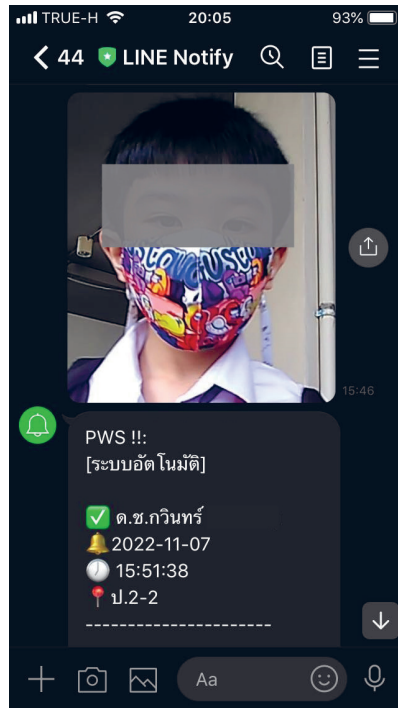
```
99   async notify() {
100     var token = "EwnmPKDkEakX93pm3GN9iuIq2ikgsa6qTwMACS5NwpY";
101     var headers = {
102       "content-type": "application/x-www-form-urlencoded",
103       Authorization: `Bearer ` + token,
104     };
105     var data = { message: "มีผู้ใช้งานเข้ามาในพื้นที่กิจกรรมแล้ว" };
106
107     try {
108       await axios({
109         method: "post",
110         url: "https://notify-api.line.me/api/notify",
111         headers: headers,
112         data: qs.stringify(data),
113       });
114       return true;
115     } catch (error) {
116       functions.logger.error("Utils-notify", error.message);
117       return false;
118     }
119   }
```

▶ รูปที่ 8-6 โค้ดในส่วนของฟังก์ชัน notify ในไฟล์ util.js

#### ▶ NOTE : การ Install Package qs

Package qs (Querystring Parsing) สามารถติดตั้งได้โดยใช้คำสั่งด้านล่าง โดยติดตั้งที่ Path Functions

```
npm install qs --save
```



รูปที่ 8-10 ตัวอย่างการส่งข้อความแจ้งเตือนถึงผู้ปกครองผ่าน LINE Notify

## บทวิเคราะห์และสรุป

LINE Notify ถือเป็นอีกหนึ่งเทคโนโลยีที่มีการนำไปประยุกต์ใช้ในการแจ้งเตือนเรื่องต่างๆ เป็นจำนวนมาก เนื่องจาก LINE Notify มี API ให้เรียกใช้ได้โดยไม่มีค่าใช้จ่าย (LINE, 2565) ทำให้นักพัฒนาสามารถนำไปประยุกต์ใช้ได้ง่าย และช่วยสร้างประสบการณ์ที่ดีให้กับผู้ใช้ได้ เพราะเป็นการแจ้งเตือนผ่านช่องทาง LINE ซึ่งคนไทยนิยมใช้กันอยู่แล้ว และเป็นการแจ้งเตือนแบบอัตโนมัติตามเงื่อนไขที่เขียนโปรแกรมเอาไว้ เมื่อมีเหตุการณ์ตรงตามเงื่อนไข นักพัฒนาสามารถเรียกใช้งาน LINE Notify API เพื่อส่งข้อความแจ้งเตือนไปให้ผู้ใช้ LINE หรือกลุ่ม LINE ได้ (LINE Developers, 2565)



สำหรับตัวอย่างการประยุกต์ใช้ LINE Notify นั้นมีอยู่เป็นจำนวนมาก ทั้งในภาคธุรกิจและผู้ใช้งานทั่วไป จะเห็นได้ว่านักวิจัยเองเมื่อพัฒนาระบบหรืออุปกรณ์ต่างๆ หากต้องการฟังก์ชันในการแจ้งเตือนการใช้งาน LINE Notify ก็เป็นอีกตัวเลือกหนึ่งที่น่าสนใจ เช่น ระบบแจ้งเตือนอัคคีภัยโดยใช้ภาพถ่ายความร้อน และภาพเหตุการณ์จริงแบบประหยัด ซึ่งมีการใช้ LINE Notify แจ้งเตือนเมื่อมีการตรวจจับความร้อน ตรวจจับควัน ตรวจจับก๊าซ ตามเกณฑ์ที่กำหนดไว้ (วุฒิชัย พรพัชรพงศ์ และภัทราวลัย คำปลิว, 2565) หรือระบบแจ้งเตือนปริมาณคงเหลือของแก๊สในครัวเรือนและการสั่งซื้อ เมื่อเซนเซอร์ (Sensor) ตรวจสอบได้ว่า ปริมาณแก๊สอยู่ในระดับ 50%, 25% และ 5% ก็แจ้งเตือนไปยังผู้ใช้และร้านส่งแก๊สต่อไป (เกล้ากล้า ศิลลาจันทร์ และคณะ, 2561)

ผู้เขียนมีความเห็นว่า การแจ้งเตือนโดยใช้ LINE Notify มีความน่าสนใจกว่าการแจ้งเตือนในรูปแบบอื่นๆ ที่นิยมใช้กัน ไม่ว่าจะเป็นการแจ้งเตือนผ่านอีเมล ผ่าน SMS รวมไปถึงผ่านแอปพลิเคชัน สำหรับกรณีอีเมลนั้นจะเหมาะกับการแจ้งเตือนที่เป็นทางการ หากมีการแจ้งเตือนทั่วไปและมีความถี่ในการแจ้งเตือน เช่น การแจ้งเตือนเกี่ยวกับความชื้นหรืออุณหภูมิของอุปกรณ์ IoT ในกรณีนี้การใช้ LINE Notify จะเหมาะสมกว่า ส่วนการแจ้งเตือนผ่าน SMS นั้น ผู้ส่งจะมีค่าใช้จ่ายตามข้อความที่ส่ง แต่ LINE Notify ไม่มีค่าใช้จ่ายในส่วนนี้ และสำหรับการแจ้งเตือนผ่านแอปพลิเคชันนั้น ผู้ใช้งานจำเป็นต้องติดตั้งแอปพลิเคชันนั้น ในส่วนของผู้พัฒนาก็มีค่าใช้จ่ายเกี่ยวกับการแจ้งเตือนด้วย นอกจากนี้ LINE Notify ยังสามารถส่งการแจ้งเตือนไปในกลุ่ม LINE ได้อีกด้วย ทำให้คนที่เกี่ยวข้องสามารถรับการแจ้งเตือนพร้อมๆ กัน และหากต้องการพูดคุยเกี่ยวกับการแจ้งเตือนนั้น ก็สามารถพูดคุยกันได้ทันที

เนื้อหาในบทนี้เราจะสามารถประยุกต์ใช้ LINE Notify ร่วมกับ LINE Beacon เพื่อส่งข้อความแจ้งเตือนไปยังผู้ดูแลกิจกรรมให้ทราบได้ว่า มีผู้ใช้งานเข้ามาในพื้นที่กิจกรรมแล้ว ซึ่ง LINE Notify ยังสามารถนำไปประยุกต์ใช้งานเพื่อการแจ้งเตือนในเรื่องต่างๆ ได้หลากหลายตามความต้องการของนักพัฒนา