

บทที่ 2

ทฤษฎีและระบบงานที่เกี่ยวข้อง

ทฤษฎีและระบบงานที่นำมาศึกษาประกอบด้วย 2 ส่วนคือ ทฤษฎีที่เกี่ยวข้องเป็นการศึกษาเกี่ยวกับ การใช้งาน Library ของ Flutter และทฤษฎีการจัดการฐานข้อมูล งานที่เกี่ยวข้องเป็นการศึกษาเกี่ยวกับฟังก์ชันการทำงานของระบบ และการออกแบบ UI ให้สวยงาม

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 QR Code

QR Code ย่อมาจาก Quick Response คือ เป็นบาร์โค้ด 2 มิติเป็นสัญลักษณ์แทนข้อมูลต่างๆ ที่มีการตอบสนองที่รวดเร็ว QR Code นำมาประยุกต์ใช้ได้หลากหลายรูปแบบ เช่น แสดง URL ของเว็บไซต์, ข้อความ, เบอร์โทรศัพท์ และข้อมูลที่เป็นตัวอักษรได้ เมื่อนำกล้องของโทรศัพท์มือถือไปถ่าย QR Code เราสามารถอ่าน QR Code ได้โดยใช้โทรศัพท์มือถือที่มีโปรแกรมที่เรียกว่า QR Code reader ติดตั้งอยู่ในเครื่องโทรศัพท์ QR Code Reader สามารถดาวน์โหลดได้ที่ wap.mobilelife.co.th/qr

2.1.2 QR Code Scanner in Flutter

เมื่อเราเลือกใช้ฟังก์ชันสแกน QR Code เพื่อเลือกโต๊ะอาหารบนแอปพลิเคชัน BJ Order ก็จะทำหน้าที่แอปพลิเคชันที่ส่วน method Barcode scanner ซึ่งหลักการทำงานก็คือจะไปเรียก Native library ของแต่ละ platform (Android, iOS) ผลลัพธ์ที่ได้ QR Code ก็จะเป็นข้อมูล URL ของ Server ร้านค้า และ Text String เก็บเลขโต๊ะอาหาร

2.1.3 MySQL

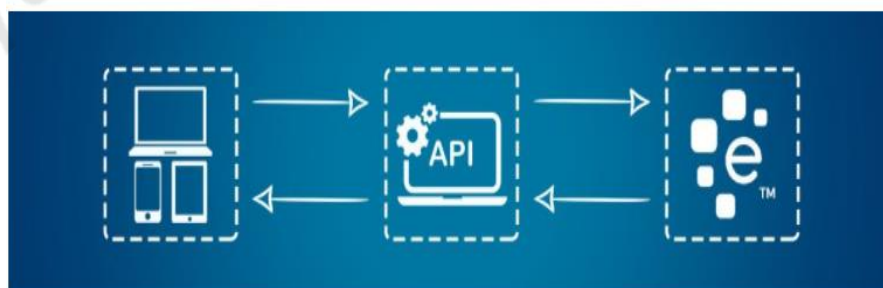
MySQL คือโปรแกรมระบบจัดการฐานข้อมูล ที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่นอย่างบูรณาการ เพื่อให้ได้ระบบงานที่รองรับ ความต้องการของผู้ใช้ เช่นทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) เช่น ภาษา PHP ภาษา asp.net หรือภาษาเจเอสพี เป็นต้น หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น ภาษา Visual Basic.NET ภาษา Java หรือภาษา C# เป็นต้น โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลโอเพนซอร์ส (Open Source) ที่ถูกนำไปใช้งานมากที่สุด

ความสามารถและการทำงานของโปรแกรม MySQL มีดังนี้

MySQL ถือเป็นระบบจัดการฐานข้อมูล Database Management System (DBMS) ฐานข้อมูลมีลักษณะเป็นโครงสร้างของการเก็บรวบรวมข้อมูล การเข้าถึงหรือประมวลผลข้อมูลที่เก็บในฐานข้อมูล จำเป็นจะต้องอาศัยระบบจัดการฐานข้อมูล ซึ่งจะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูลทั้งสำหรับการใช้งานเฉพาะ และรองรับการทำงานของแอปพลิเคชันอื่น ๆ ที่ต้องการใช้งานข้อมูลในฐานข้อมูล เพื่อให้ได้รับความสะดวกในการจัดการกับข้อมูลจำนวนมาก MySQL ทำหน้าที่เป็นทั้งตัวฐานข้อมูล และระบบจัดการฐานข้อมูล MySQL เป็นระบบจัดการฐานข้อมูลแบบ relational ฐานข้อมูลแบบ relational จะทำการเก็บข้อมูลทั้งหมดในรูปแบบของตารางแทนการเก็บข้อมูลทั้งหมดลงในไฟล์เพียงไฟล์เดียว ทำให้ทำงานได้รวดเร็ว และมีความยืดหยุ่น นอกจากนี้ แต่ละตารางที่เก็บข้อมูลสามารถเชื่อมโยงเข้าหากันทำให้สามารถรวมหรือจัดกลุ่มข้อมูลได้ตามความต้องการ โดยอาศัยภาษา SQL ที่เป็นส่วนหนึ่งของโปรแกรม MySQL ซึ่งเป็นภาษามาตรฐานในการเข้าถึงฐานข้อมูล MySQL แจกจ่ายให้ใช้งานแบบ Open Source นั่นคือ ผู้ใช้งาน MySQL ทุกคนสามารถใช้งาน และปรับแต่งการทำงานได้ตามต้องการสามารถดาวน์โหลดโปรแกรม MySQL ได้จากอินเทอร์เน็ต และนำมาใช้งานโดยไม่มีค่าใช้จ่ายใด ๆ

2.1.4 API

API ย่อมาจาก Application Programming Interface คือการเชื่อมต่อจากระบบหนึ่งไปสู่อีกระบบหนึ่ง เพื่อให้ซอฟต์แวร์ภายนอกเข้าถึงและอัปเดตข้อมูลนั้น ๆ ได้ API เป็นตัวกลางที่จะทำให้คอยรับคำสั่งต่าง ๆ ประมวลผลและกระทำข้อมูลส่งกลับคืนไปยังคนสั่งโดยอัตโนมัติ อย่างเช่น Application ต่าง ๆ ที่เราเล่นกันอยู่ในปัจจุบันนั่นเอง ด้วยความสะดวกสบายนี้ ทำให้บริษัทหรืออุตสาหกรรมต่าง ๆ เริ่มนำ API เข้ามาใช้งาน



ภาพประกอบที่ 2.1 ภาพรวมระบบการทำงาน API

รูปแบบการนำเอา API ไปใช้งานมีดังต่อไปนี้

1. Operating Systems API สามารถใช้งานในการสื่อสารระหว่าง application และ operating system เช่น POSIX หรือมาตรฐานการสื่อสารของ OS เองก็มี API เป็น command line เพื่อควบคุมการทำงานของ OS

2. Remote APIs Remote APIs ทำไว้ให้ developer สามารถเข้าควบคุมทรัพยากรผ่านทาง protocol เพื่อให้มีมาตรฐานการสื่อสารเดียวกันถึงแม้ว่าจะเป็นคนละ technology เช่น Database API สามารถอนุญาตให้ developer เข้ามาดึงข้อมูลใน database หลากหลายชนิดได้ ผ่าน function เดียวกัน เพราะฉะนั้น remote API จึงถูกใช้บ่อยในงาน maintenance ด้วยทำงานที่ฝั่ง client ให้ไปดึงข้อมูลจาก server กลับลงมาทำงาน

3. Libraries and frameworks API มักจะเอาไปใช้เป็น software library ซึ่งเขียนขึ้นตาม document ในรูปแบบภาษา program ที่ต่างกันออกไปตามความเหมาะสมกับงาน เพื่อเอาไปทำเป็น framework ให้กับระบบใช้ในการสื่อสารหากัน

ประโยชน์ของการส่ง API มีดังต่อไปนี้

1. ช่วยเป็นสื่อกลางส่งข้อมูลข้าม Server
2. ลดการใช้กำลังคนด้วยการใช้เทคโนโลยี
3. สื่อสารกับลูกค้าได้รวดเร็วขึ้น
4. ไม่ต้องทำงานข้ามเว็บไซต์ เพียงติดตั้ง API และใช้งานในเว็บไซต์ที่มี API เดียวก็พอ
5. ผู้ใช้มีความสะดวกสบายมากขึ้น ไม่ต้องเข้าโปรแกรมหรือ tools ในการทำงานให้ซับซ้อน

2.1.5 Flutter

Flutter คือ SDK (Software Development Kit) สำหรับพัฒนา Application บน Mobile และเป็นโปรแกรมที่สามารถ Build ได้ทั้ง iOS และ Android โดยภาษาที่ใช้ในการเขียน Flutter ก็คือ ภาษา Dart ที่ซึ่งพัฒนาโดย Google เช่นเดียวกับตัวโปรแกรม และยังได้กล่าวถึงจุดเด่นของ Flutter ไว้ว่า Fast Development การพัฒนาอย่างรวดเร็วโดยการแสดงผลภายในไม่กี่นาทีและแก้ไขข้อบกพร่องได้อย่างเร็วขึ้น Expressive and Flexible UI ที่แสดงออกและมีความยืดหยุ่น แสดง widget ที่สวยงามในตัว และสร้างการเคลื่อนไหวที่หลากหลาย Native Performance ประสิทธิภาพ Widget ของ Flutter ได้รวมความแตกต่างของไอคอนและแบบอักษรมาสร้างประสิทธิภาพการทำงานแบบ native ได้อย่างเต็มรูปแบบทั้งบน iOS และ Android

ข้อดีของ Flutter

1. Hot Reload การย่นระยะเวลาที่ใช้ในการ reload ให้เหลือเพียงเสี้ยววินาที
2. Fast Development นอกเหนือจากปุ่ม Hot Reload แล้วยังมีตัวควบคุมที่สามารถปรับแต่งได้ตามความต้องการ
3. Screen reader มีระบบโปรแกรมอ่านหน้าจอ สำหรับนักพัฒนาที่บกพร่องทางสายตา
4. Quick rendering มีการ render ที่รวดเร็ว จึงสามารถสร้างภาพและแบบจำลอง 2D ที่เร็วขึ้น
5. Cross-platform มีโปรแกรมหลากหลายที่สนับสนุน เช่น Android Studio และ Visual Studio Code
6. Flutter goes native มีระบบปลั๊กอินที่จำเป็นต่อการสร้างแอปพลิเคชันให้มีประสิทธิภาพได้อย่างง่ายดาย
7. Open source & free นักพัฒนาแอปพลิเคชันและผู้สนใจสามารถใช้งานได้ฟรี
8. Themes สามารถสร้างความแตกต่างระหว่าง Android และ iOS ในการเขียน/พัฒนาครั้งเดียว

ข้อเสียของ Flutter

1. Mobile only สถานะของ Flutter ตอนนี้สามารถใช้งานได้เฉพาะมือถือเท่านั้น
2. low popularity of Dart คนส่วนมากไม่คุ้นเคยกับภาษา Dart สักเท่าไร ไม่เหมือนกับภาษา Java หรือภาษา Kotlin ซึ่งเป็นภาษาส่วนใหญ่ที่คนนิยมใช้
3. Not all devices are supported Flutter ไม่สามารถใช้ได้ใน iOS 32-bit ดังนั้น iOS รุ่นก่อนๆ เช่น iPhone 5 ลงไปจะไม่สามารถใช้งานได้
4. Limited libraries เนื่องจากยังเป็นโปรแกรมที่ค่อนข้างใหม่ทำให้ยังไม่ค่อยมี libraries ที่เป็นประโยชน์ ซึ่งนักพัฒนาส่วนใหญ่ต้องสร้าง libraries ขึ้นมาเป็นของตนเอง และยังคงใช้เวลาในการสร้างอีกด้วย

2.1.6 RESTful

Representational state transfer หรือ REST คือ การสร้าง Web Service ชนิดหนึ่งที่ใช้สื่อสารกันบน Internet ใช้หลักการแบบ stateless คือไม่มี session ซึ่งต่างจาก Web Service แบบอื่นเช่น WSDL และ SOAP การทำงานของ RESTful Web Service จะอาศัย URI/URL ของ request เพื่อค้นหาและประมวลผลแล้วตอบกลับไปในรูปแบบ XML, HTML, JSON โดย response ที่ตอบกลับจะ

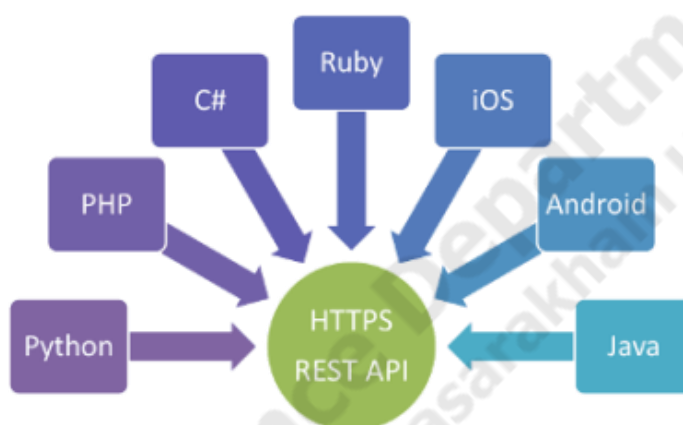
เป็นการยืนยันผลของคำสั่งที่ส่งมา และสามารถพัฒนาด้วยภาษา programming ได้หลากหลาย คำสั่งก็จะมีตาม HTTP Method ที่สำคัญ

Get เป็นการร้องขอข้อมูลจาก resource

Post เป็นการสร้างข้อมูลใหม่ใน resource

Put เป็นการอัปเดตข้อมูลที่มีอยู่แล้ว หรือสร้างใหม่ resource

Delete เป็นการลบข้อมูลที่มีอยู่แล้วใน resource



ภาพประกอบที่ 2.2 REST API

คุณสมบัติของ REST เป็น API อย่างหนึ่ง ซึ่งทุก ๆ system ต่างใช้ resource ซึ่งเป็นได้ทั้ง image, video, web page หรือข้อมูลทางธุรกิจ ก็ได้ที่สามารถแสดงบนระบบ computer วัตถุประสงค์เพื่อให้ user สามารถเข้าถึง, ติดตั้ง, ปรับแต่ง, ขยาย resource เหล่านี้ได้ง่าย ซึ่งทาง RESTful ได้ออกแบบมาให้มีคุณสมบัติต่อไปนี้

1. แสดงผล
2. เก็บข้อมูล
3. มี URIs
4. Stateless ทำงานโดยไม่ต้องมี session
5. เชื่อมต่อระหว่าง web service
6. Caching

Architectural constraints of RESTful API ข้อกำหนดของ RESTful API ซึ่งถือเป็นสิ่งที่สำคัญในการสร้าง RESTful API ตามมาตรฐานซึ่งทำให้ง่ายต่อการพัฒนา และทำให้เป็นที่ยอมรับ มีดังนี้

1. Client-server architecture: Client ไม่จำเป็นต้องรู้อะไรเกี่ยวกับ Business logic ภายใน ไม่มีหน้าที่เกี่ยวกับการจัดเก็บข้อมูล ส่วน Server มีหน้าที่เก็บ Resource และไม่จำเป็นต้องรู้อะไรเกี่ยวกับ UI Frontend หรือสถานะของผู้เรียก
2. Statelessness: ส่ง Request รับ Response จาก Server แล้วจบการทำงาน
3. Cacheability: สามารถ cache response ได้ การ Response จะต้องสามารถกำหนดได้ว่าจะ Cache หรือไม่ เพื่อป้องกันไม่ให้ User หรือ Client ได้รับข้อมูลเก่า
4. Layered system: ปกติ Client ไม่รู้ว่าที่ทำการเชื่อมต่อนั้น ได้เชื่อมต่อโดยตรงกับ Server ปลายทาง หรือไปยังตัวกลางอื่น ๆ ระหว่างทาง, Server ตัวกลางควรสามารถปรับปรุงความสามารถในการขยายระบบได้ โดยการใช้งานการทำ Load balance
5. Code on demand (optional): Server สามารถขยายได้ชั่วคราว หรือปรับแต่งการทำงานของไคลเอนต์ได้ ตัวอย่างเช่น ทำ client-side scripts ใน JavaScript
6. Uniform interface: ถือเป็นข้อสำคัญจะที่แยกระหว่าง REST API และ Non-REST API แสดงให้เห็นถึงวิธีการที่จะคุยกับ Server โดยไม่คำนึงถึงประเภทของอุปกรณ์ หรือประเภทของ application

Uniform interface แยกออกเป็น 4 รูปแบบ

1. Resource-Based: เช่น API/users
2. Manipulation of Resources Through Representations: เช่น User get user_id หรือ Request list of users แล้วทำการ Delete หรือ Modify user
3. Self-descriptive Messages: แต่ละ Message มีข้อมูลเพียงพอที่จะนำมาอธิบายวิธีการ Process message เพื่อให้ Server ทำการวิเคราะห์ได้ง่าย
4. Hypermedia as the Engine of Application State (HATEOAS): จำเป็นต้องมี Links สำหรับทุก ๆ Response เพื่อให้ Client สามารถค้นหาได้ง่าย

HTTP Response Status Code มีดังต่อไปนี้

2xx (success code)

200 OK – มาตรฐาน HTTP response success สำหรับ GET, PUT หรือ POST

201 Create – response สำหรับข้อมูลที่ถูกสร้างขึ้นใหม่ ใช้สำหรับ POST

204 No Content – response สำหรับ request ที่ดำเนินการ success แล้วไม่ return ข้อมูลกลับมา

3xx (Redirection)

304 Not Modified – บอกว่า client ได้รับการ response แล้วอยู่ใน cache และ
ไม่จำเป็นต้องส่งข้อมูลเดิมอีกครั้ง

4xx (Client Error)

400 Bad Request – request ที่ส่งมาโดย client นั้นไม่ถูกดำเนินการ

401 Unauthorized – client ไม่ได้รับอนุญาตในการเข้าถึง resource และควรจะ
request ใหม่ด้วย credential

403 Forbidden – บ่งบอกว่า request นั้นถูกต้อง และ client ได้รับการอนุญาต แต่
client ไม่ได้รับอนุญาตให้เข้าถึง resource ด้วยเหตุผลบางประการ

404 Not Found – resource ที่ request มานั้นไม่ว่างใช้งานตอนนี้

405 Gone – resource ไม่มีอยู่แล้ว หรือถูกย้ายไปที่อื่น

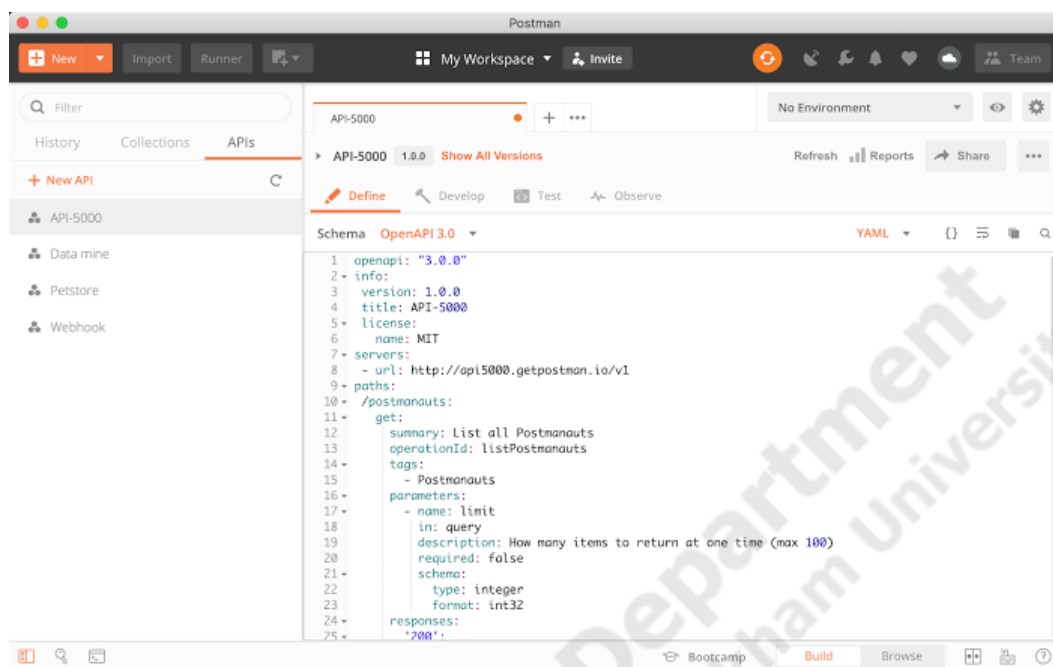
5xx (Server Error)

500 Internal Server Error – request ถูกต้อง แต่ server มีความสับสน และจะ
บริการด้วยเงื่อนไขที่คาดการณ์ไม่ได้

503 Service Unavailable – server ใช้การไม่ได้ (โดยส่วนใหญ่ server อยู่
ในช่วงบำรุงรักษา)

การจะทำ RESTful API สามารถทำได้ในหลายภาษาไม่ว่าจะเป็น PHP Java node ซึ่งแน่นอน
ว่าในแต่ละภาษามีจุดเด่นที่แตกต่างกันไปในแต่ละภาษา

ตัวอย่างเครื่องมือที่ใช้ทดสอบ RESTful API Postman จะเป็นการเครื่องมือที่ใช้ทดสอบการ
เชื่อมต่อ Restful Web Service



ภาพประกอบที่ 2.3 Postman

2.1.7 การเชื่อมต่อ RESTful API

การเชื่อมต่อกับ API จะต้อง import package เพิ่มโดยพิมพ์คำสั่งที่ terminal > cmd ดังภาพประกอบที่ 2.3



ภาพประกอบที่ 2.4 import package ใน terminal cmd

จากนั้นแก้ไขในไฟล์ main.dart โดยเริ่มจากการ import ดังภาพประกอบที่ 2.4

```

1 | import 'dart:async';
2 | import 'dart:convert';
3 |
4 | import 'package:flutter/material.dart';
5 | import 'package:http/http.dart' as http;

```

ภาพประกอบที่ 2.5 import package ใน flutter

ตัวอย่าง Code API สำหรับ Get ข้อมูล


```

01 Future<dynamic> fetchAlbum() async {
02   final response =
03     await http.get(Uri.parse('https://jsonplaceholder.typicode.com/albums'));
04
05   if (response.statusCode == 200) {
06     return jsonDecode(response.body);
07   } else {
08     throw Exception('Failed to load album');
09   }
10 }

```

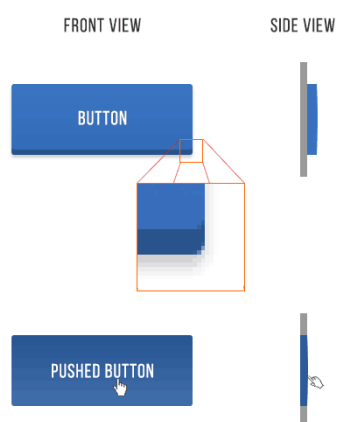
ภาพประกอบที่ 2.6 ตัวอย่างการใช้ API สำหรับ Get ข้อมูล

การเชื่อมต่อ API โดยใช้ `http.get` เพื่อดึงข้อมูลแบบ Get จากนั้น return ค่า response โดยใช้ประเภท Future ประเภทข้อมูลนี้คือ Callback เอาไว้อ่านค่าที่ได้จาก api เมื่อได้ข้อมูลมาแล้ว ซึ่งข้างในเป็นค่าประเภท dynamic ที่ได้จาก `jsonDecode` ก็สามารถนำมาแสดงผลบนแอปพลิเคชันได้ตามความต้องการ

2.1.8 ออกแบบ UI ให้สวยงาม

User interface คือ ส่วนที่ใช้ในการเชื่อมต่อกับผู้ใช้งาน ผู้ใช้งานสามารถโต้ตอบกับการใช้งานผลิตภัณฑ์ได้ ซึ่งจะมุ่งเน้นไปที่เรื่องของหน้าตาการออกแบบ และการดีไซน์ ตัวอย่างเช่น หน้าจอตัวอักษร แบนพิมพ์ รูปภาพ สี หรือแม้แต่กระทั่งสิ่งที่เรามองไม่เห็น เช่น แสง และเสียง เป็นต้น วิธีการออกแบบมีดังนี้

1. แสง และเงา การออกแบบ User Interface ควรต้องมีการใส่เงาให้ปุ่มเพื่อให้เรารู้สึกว่าปุ่มเป็น 3 มิติขึ้นออกมา



ภาพประกอบที่ 2.7 แสงและเงา

2. การออกแบบสี เริ่มต้นด้วยการเริ่มใช้สี ขาว-ดำ ก่อนแล้วค่อยใส่สีเข้าไปทีละนิด เพื่อจะช่วยให้ดีไซน์ UI ที่ซับซ้อนออกมาได้ง่ายขึ้น และนอกจากนั้นยังทำให้เราโฟกัสกับเรื่องการจัดช่องว่าง และวาง Layout สิ่งต่าง ๆ ก่อนด้วยเทคนิค การเลือกใช้สีในงานออกแบบที่มาจากสีหลักไม่เกิน 1-2 สีจะทำให้เราสามารถแบ่งส่วนของดีไซน์ที่ต้องการให้เด่น หรือไม่เด่นได้ง่าย ๆ โดยที่ไม่ทำให้ดีไซน์โดยรวมดูไม่ชัดเจน



ภาพประกอบที่ 2.8 การออกแบบสี

3. การเพิ่ม Whitespace หมายถึงพื้นที่โล่ง ๆ รอบ ๆ ทุกจุดในงานดีไซน์นั่นเอง เพื่อให้การดีไซน์ออกมาแล้วดูสบายตา

4. การวางตัวหนังสือบนรูป แบ่งออกเป็น 4 วิธี

- ใช้สีที่รูปพื้นหลัง เลือกสีที่โปร่งใสมาทับรูปและควรเป็นสีเข้มเท่านั้นเพราะจะทำให้ Text ดูชัดขึ้น
- ใส่ตัวอักษรไว้ในกล่อง โดยใส่ตัวหนังสือขาวในกล่องพื้นหลังสีดำหรือสีดำแบบโปร่งใสเพื่อ ทำให้มีระดับขึ้นอีก
- เบลอรูปพื้นหลัง สามารถเลือกเบลอปื้นหลังทั้งรูปหรือจะใช้วิธีเบลอเฉพาะจุดที่อยู่ใกล้ และ Focus จุดที่อยู่ใกล้
- Floor Fade คือ การทับเฉพาะส่วนล่างที่มีตัวหนังสือแล้วค่อย ๆ ไล่สีออกจนมองไม่เห็น

5. การเพิ่ม และลดความเด่นของตัวหนังสือ มีวิธีดังนี้

- เปลี่ยนขนาดตัวหนังสือ
- เปลี่ยนสีตัวหนังสือ
- ปรับตัวหนา บาง
- ใช้ตัวพิมพ์เล็ก - ตัวพิมพ์ใหญ่
- ใช้ตัวเอียง
- เปลี่ยนระยะห่างตัวหนังสือ
- เปลี่ยนระยะห่างของกล่อง

6. เลือกใช้ฟอนต์ให้เหมาะสม ฟอนต์เป็นตัวกำหนด Mood & Tone ของงานดีไซน์ที่สำคัญมาก ฟอนต์แต่ละแบบให้ความรู้สึกแตกต่างกันไป ควรเลือกใช้ฟอนต์ให้เข้ากับงานเราให้มากที่สุด

2.2 งานวิจัยที่เกี่ยวข้อง

ระบบที่เกี่ยวข้องกับแอปพลิเคชัน BJ Order มีดังนี้

2.2.1 Order QR

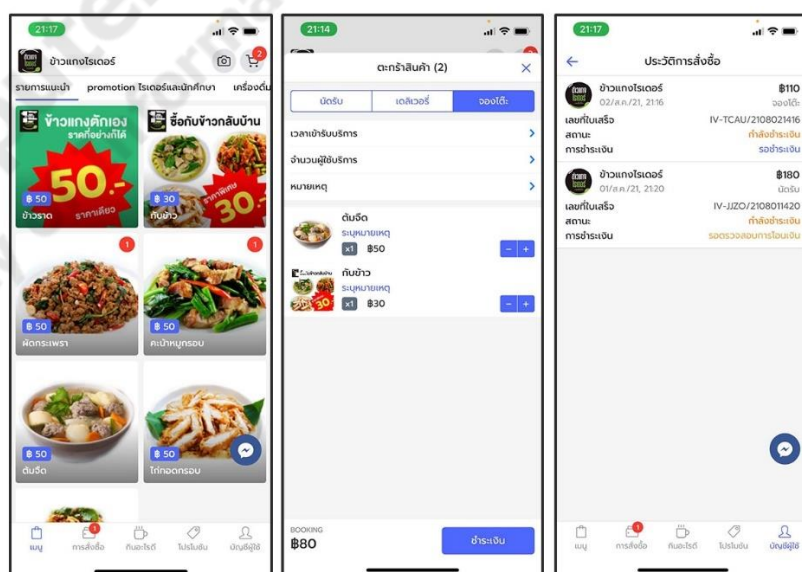
เป็นแอปพลิเคชันที่ช่วยให้ลูกค้าเรียกดูรายการอาหารก่อนไปทานที่ร้าน และสามารถสั่งอาหารแบบเดลิเวอรี่ หรือนัดรับอาหารได้

ข้อดี

1. สามารถเข้าสู่ระบบได้โดยไม่ต้องสมัครสมาชิก
2. เลือกการสั่งอาหารได้หลายรูปแบบเช่น นัดรับ,เดลิเวอรี่,จองโต๊ะ

ข้อเสีย

1. UI เมนูอาหารติดกันมากเกินไปทำให้ดูอึดอัด
2. ตัวหนังสือมีขนาดเล็กมองเห็นได้ไม่ชัดเจน
3. ไม่สามารถเช็คสถานะการนำอาหารได้



ภาพประกอบที่ 2.9 ตัวอย่างแอปพลิเคชัน Order QR

2.2.2 Table Table

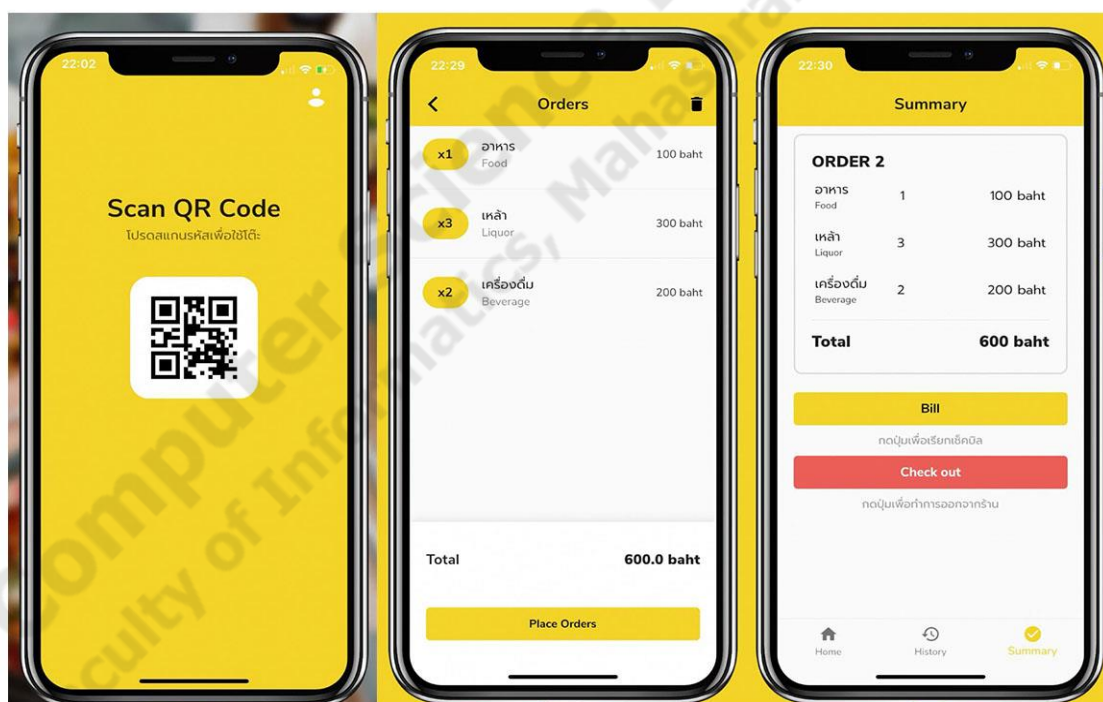
เป็นแอปพลิเคชันสำหรับสั่งอาหารและเครื่องดื่ม จากผับบาร์ และร้านอาหารทั่วไป ผ่านสมาร์ตโฟน โดยจะแสดงผลแบบ real-time ไปยังร้านค้าโดยตรง โดยสามารถสั่งออเดอร์ได้ผ่านการสแกนคิวอาร์โค้ด และรอฟังก์ชันมาเสิร์ฟ แอปพลิเคชันเหมาะสำหรับการใช้งานในผับบาร์ และร้านอาหาร ในช่วงที่ร้านมีลูกค้าหนาแน่นและพนักงานไม่สามารถบริการได้ทั่วถึง

ข้อดี

1. UI มีรูปแบบที่ใช้งานง่าย
2. สีสดใสใส

ข้อเสีย

1. ไม่สามารถเช็คสถานะทำอาหารได้
2. สีเมนูลูกเล่นน้อยเกินไป



ภาพประกอบที่ 2.10 ตัวอย่างแอปพลิเคชัน Table Table

Page365 Store

เป็นเว็บไซต์ที่สามารถจัดการหน้าร้านออนไลน์ผ่าน facebook และจัดการออเดอร์ต่าง ๆ และยังสามารถดูบิลออนไลน์ได้

ข้อดี

1. ดึงรูปสินค้าจาก facebook มาแสดงผลบนแอปพลิเคชันร้านค้า
2. ส่งข้อความไปหาลูกค้าได้

ข้อเสีย

2. UI ปุ่มไม่มีความเด่นเพราะเป็นสีเดียวกันทุกปุ่มแสดงผล



ภาพประกอบที่ 2.11 Page365 Store1

เลขที่ออเดอร์	วันที่สั่งซื้อ	ผู้สั่งซื้อ	ที่อยู่จัดส่ง	เบอร์โทรศัพท์	จำนวนสินค้า	ยอดสุทธิ	สถานะสั่งซื้อ
122	15 มิถุนายน 2015	พลอย สมนาม	309 ถนนปราบ สิบพันสองไร่ ถนน.10100	089-115-1151	1	660 บาท	ยังไม่จ่าย
121	15 มิถุนายน 2015	กอล์ฟ วิทยาพิทักษ์	1109 ซักทองจันทราวงค์อิน12	087-530-6659	1	780 บาท	ยังไม่จ่าย
120	15 มิถุนายน 2015	พิชชา อรุณวงศ์	741 สุรนารี93 บางจาก พระโขนง ถนน. 10240	088-885-6598	1	1,250 บาท	ยังไม่จ่าย
119	15 มิถุนายน 2015	เบงอซี พิธีร์	13 บางขุนเทียน บางขุนเทียน ถนน.10110	088-551-0011	1	660 บาท	ยังไม่จ่าย
118	15 มิถุนายน 2015	สุทนต์ จันทร์สว่าง	1 สุรนารี101/1 บางจาก พระโขนง ถนน.	099-536-4102	2	2,300 บาท	ยังไม่จ่าย
117	15 มิถุนายน 2015	พิชพร อิมเมทรา	13 อานามาว สาทร ถนน. 10120	089-398-8028	2	1,850 บาท	ยังไม่จ่าย
117	15 มิถุนายน 2015	ไพบูลย์ สุกสุระ	96/89 น.เทศบาล บางจาก พระโขนง ถนน.	02-120-1110	1	500 บาท	ยังไม่จ่าย

ภาพประกอบที่ 2.12 Page365 Store2

2.3 ตารางเปรียบเทียบ

ตารางที่ 2.1 ตารางเปรียบเทียบระบบที่เกี่ยวข้องฝั่งลูกค้า Table Table

ฟังก์ชันการทำงาน	Order QR	Table Table	ระบบที่พัฒนา
สแกนคิวอาร์โค้ดเชื่อมต่อร้านอาหาร	✓	✓	✓
สามารถเพิ่มรายการอาหาร	✓	✓	✓
แก้ไขรายการอาหาร	✓		✓
หน้าแสดงใบเสร็จ	✓	✓	✓
เช็คสถานะเมนูอาหาร			✓
ประวัติการสั่งอาหาร	✓		✓
สามารถชำระเงินได้	✓		✓

ตารางที่ 2.2 ตารางเปรียบเทียบระบบที่เกี่ยวข้องฝั่งร้านค้า Page365 Store

ฟังก์ชันการทำงาน	Page365 Store	ระบบที่พัฒนา
สามารถเพิ่มรายการได้	✓	✓
สามารถแก้ไขรายการได้	✓	✓
สามารถยืนยันออเดอร์ได้	✓	✓
สามารถดูรายงานยอดขาย	✓	✓
ดูสถานะการชำระเงิน	✓	✓