

แอปพลิเคชันค้นหายาเพื่อคุณ
(DRUGIDEN)

นายกฤษณะ ชินโคตร

โครงการนี้เป็นส่วนหนึ่งของการศึกษา
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์ สติ๊ต และคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยอุบลราชธานี
ปีการศึกษา 2560
ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

โครงงาน : แอปพลิเคชันค้นหายาเพื่อคุณ
(DRUGIDEN)
โดย : นายกฤษณะ ชินโคตร
อาจารย์ที่ปรึกษา : ดร.เกรียงศักดิ์ ตรีประพิณ
ระดับการศึกษา : วิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์
ปีการศึกษา : 2560

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

คณะกรรมการสอบประเมินความรู้โครงงานคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา
(ดร.เกรียงศักดิ์ ตรีประพิณ)

..... กรรมการ
(ดร. สุภาวดี หรรษพงศ์สิน)

..... กรรมการ
(ดร. ไพบูลย์ คงไชย)

..... หัวหน้าภาควิชา
(ดร. ชัชวิน นามมั่น)

วันที่ . . . / . . . / . . .

กิตติกรรมประกาศ

การพัฒนาโครงงานค้นหาญเพื่อคุณ สำเร็จลุล่วงได้ด้วยความช่วยเหลือจาก ดร.เกรียงศักดิ์ ตรีประพิน อาจารย์ที่ปรึกษาโครงงานที่ได้ให้คำแนะนำ ทฤษฎี และแนวคิดตลอดจนช่วยแก้ไขข้อบกพร่องต่างๆ โดยตลอด โครงงานนี้สำเร็จสมบูรณ์ ผู้จัดทำโครงงานจึงได้ขอรับรางวัลชมเชย
เป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณอาจารย์ประจำภาควิชาคณิตศาสตร์ สกิติ และคุณพิวเตอร์ คณะวิทยาศาสตร์ทุกท่าน ที่เคยให้คำแนะนำและช่วยเหลือในการศึกษาตลอดปีการศึกษา

ขอขอบพระคุณหน่วยข้อมูลฯและสุขภาพ คณะเภสัชศาสตร์ มหาวิทยาลัยอุบลราชธานี ที่ให้เชื้อรานข้อมูลการพิสูจน์เอกสารนี้เอกสารนี้ได้รับการประเมินโดยคณะกรรมการพิจารณาและอนุมัติให้ดำเนินการ

ขอขอบพระคุณบิดาและมารดาที่ค่อยให้กำลังใจและการสนับสนุนในการพัฒนาโครงงาน คอยสอบตามและคอยห่วงใยเสมอมา

ขอขอบคุณเพื่อนสาขาวิชาการคอมพิวเตอร์ ชั้นปี 4 รุ่นที่ 18 มหาวิทยาลัยอุบลราชธานี ที่ช่วยให้คำแนะนำ คำปรึกษาและเป็นกำลังใจในการทำโครงงานจนสำเร็จลุล่วงไปได้ด้วยดี

นายกฤษณะ ชินโคตร
วันที่ 18 เมษายน 61

โครงงาน	: แอปพลิเคชันค้นหาฯเพื่อคุณ
โดย	: นายกฤษณะ ชินโคตร
อาจารย์ที่ปรึกษา	: ดร.เกรียงศักดิ์ ตรีประพิน
ระดับการศึกษา	: วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา	: 2560

บทคัดย่อ

โครงงานแอปพลิเคชันค้นหาฯเพื่อคุณเป็นการพัฒนาแอปพลิเคชันบนโทรศัพท์สมาร์ทโฟน (Smartphone) ระบบปฏิบัติการแอนดรอยด์ (Android) เพื่อช่วยในการพิสูจน์เอกสารหลักฐาน ยาเม็ดหรือแคปซูลสำหรับเภสัชกรและประชาชนทั่วไป ด้วยการระบุลักษณะทางกายภาพของยาได้แก่ สี ขนาดด้านยาว รูปทรงและตัวอักษรหรือตัวอักษรระบุเลขประจำยา เม็ดยา หรือข้อมูลทางด้านการผลิต เช่น ชื่อทางการค้า ชื่อสามัญ บริษัทที่จำหน่าย สถานะของยา เป็นต้น โครงงานใช้ Node.js พัฒนาเว็บเซอร์วิส (Web Service) ส่วนเว็บเซอร์วิสทำหน้าที่ติดต่อระหว่างแอปพลิเคชันกับฐานข้อมูลการพิสูจน์เอกสารหลักฐานยาเม็ดของหน่วยข้อมูลยาและสุขภาพ คณะเภสัชศาสตร์ มหาวิทยาลัยอุบลราชธานี และใช้ Ionic Framework ใน การพัฒนาแอปพลิเคชันโดยแอปพลิเคชันที่พัฒนามาสามารถระบุลักษณะรูปทรงยาเม็ดหรือแคปซูลด้วยการค้นหาแบบทั่วไปและแบบขั้นสูง และการประมวลผลภาพ (OpenCV) ที่มีแบบจำลอง Support Vector Machines (SVM) ที่ให้ค่าความถูกต้องของแบบจำลองแบบร้อยละ 94.17 และการระบุลักษณะสีด้วยการดูค่า RGB จากจุดพิกเซล (Pixel) และการวัดขนาดด้านยาวจะใช้วัดถูกที่รู้ขนาดด้านกว้างและด้านยาวมาช่วยอ้างอิงขนาดด้านยาวของยาเม็ดได้ คำสำคัญ: การพิสูจน์เอกสารหลักฐานยาเม็ด เทคนิคการประมวลผลภาพ แอปพลิเคชันบนโทรศัพท์スマาร์ทโฟน

สารบัญ

	หน้า
กิตติกรรมประกาศ	ค
บทคัดย่อภาษาไทย	ง
 สารบัญ	 จ
สารบัญตาราง	ช
 สารบัญภาพ	 ณ
บทที่	
 1 บทนำ	 1
1.1 ที่มาและเหตุผล	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 เครื่องมือที่ใช้ในการพัฒนา	2
1.6 ขั้นตอนการดำเนินงาน	4
 2 ทฤษฎีที่เกี่ยวข้อง	 5
2.1 ความรู้พื้นฐานเกี่ยวกับ Ionic Framework	5
2.1.1 รายละเอียดของ Cordova	6
2.1.2 Hybrid Mobile Application	7
2.1.3 Lifecycle events	7
2.1.4 ข้อดีและข้อเสียของ Ionic Framework	8
2.2 ความรู้พื้นฐานเกี่ยวกับภาษา TypeScript	8
2.3 ความรู้พื้นฐานเกี่ยวกับ Docker	8
2.3.1 รายละเอียดของ Docker Concepts	9
2.3.2 ข้อดีของ Docker	10
2.3.3 คำสั่งเบื้องต้น	10
2.3.4 การใช้งาน Docker ใน Ubuntu	11
2.4 ความรู้พื้นฐานเกี่ยวกับ HTTP Protocol	11
2.4.1 HTTP Method	12
2.4.2 รายละเอียดของ HTTP Request	12
2.4.3 รายละเอียดของ HTTP Response	13
2.4.4 รายละเอียดของ HTTP Response	13
2.5 ความรู้พื้นฐานเกี่ยวกับ RESTful API	14
2.6 การทำ Authentication ด้วย JSON Web Token	14
2.6.1 การพิสูจน์ตัวตน (Authentication)	14

บทที่	หน้า
2.6.2 รายละเอียดของ JSON Web Token (JWT)	15
2.6.3 การใช้งาน JSON Web Token	16
2.6.4 ข้อดีและข้อเสียของ JWT	17
2.7 ความรู้พื้นฐานเกี่ยวกับ OpenCV (Open Source Computer Vision)	17
2.7.1 การประมวลผลภาพ (Image Processing)	18
2.7.2 การทำ Thresholding รูปภาพ	20
2.7.3 การตรวจหา Canny Edge	21
2.7.3.1 ลดสัญญาณรบกวน (Noise Reduction)	21
2.7.3.2 การหาการไล่ระดับสีความเข้มของภาพ (Gradient)	21
2.7.3.3 Nonmaxima Suppression	22
2.7.4 การหา Contour Approximation	22
2.7.5 Support Vector Machines (SVM)	24
2.7.6 K-Nearest Neighbour Algorithm	25
2.7.6.1 ขั้นตอนวิธีการดำเนินการของอัลกอริทึมแบบ K-Nearest Neighbour Algorithm มีดังนี้	25
2.7.6.2 การดำเนินการหลักของอัลกอริทึมแบบ K-Nearest Neighbour Algorithm ประกอบไปด้วยการทำงาน 2 พังค์ชัน ได้แก่	26
2.7.7 Random Forest Classifier	26
2.8 การขนาดของยาด้วยการใช้วัตถุอ้างอิง	26
2.9 เครื่องมือที่ใช้ในการพัฒนา	27
 3 การวิเคราะห์และออกแบบระบบ	 28
3.1 รายละเอียดการออกแบบแอปพลิเคชัน	28
3.1.1 ส่วนแอปพลิเคชัน	28
3.1.2 ส่วนเว็บเซอร์ฟิส	30
3.2 Use Case Diagram	31
3.3 Class Diagram	34
3.4 Sequence Diagram	37
3.4.1 Sequence Diagram ของการค้นหายาแบบทั่วไป	38
3.4.2 Sequence Diagram ของการค้นหายาแบบขั้นสูง	39
3.4.3 Sequence Diagram ของการถ่ายรูปภาพเพื่อพิสูจน์เอกสารยามาเม็ด	41
3.4.4 Sequence Diagram ของการดูรายการบุ๊กมาร์ก	42
3.5 State Diagram	44
3.6 การประมวลผลภาพยาเม็ดเพื่อการพิสูจน์เอกสารยามาเม็ด	47
3.6.1 ทดลองการจำแนกรูปทรงของยาเม็ดด้วยการเรียนรู้ของเครื่อง (Machines Learning)	47
3.6.2 การขนาดด้านยาวของเม็ดยาโดยใช้วัตถุอ้างอิงที่รู้ขนาด	53

บทที่		หน้า
	3.6.3 การหาลักษณะสีของยาเม็ด	54
4	การสร้างระบบ	56
4.1	การค้นหายาแบบทั่วไป	56
4.2	การค้นหายาแบบขั้นสูง	58
4.3	การถ่ายรูปภาพ	59
4.4	การเรียกใช้งานเว็บเซอร์วิส	60
4.5	การติดต่อฐานข้อมูลของเว็บเซอร์วิส	62
4.6	การสร้างและการตรวจสอบ JSON Web Token	62
4.7	การถอดรหัสรูปภาพ BASE64	64
4.8	การ routing ของเว็บเซอร์วิส	66
4.9	การประมวลผลภาพ	69
5	การทดสอบระบบ	77
5.1	ผลการทดสอบการค้นหายาแบบทั่วไป	78
5.2	ผลการทดสอบการค้นหายาแบบขั้นสูง	78
5.3	ผลการทดสอบเลือกคุณรายละเอียดยา	79
5.4	ผลการทดสอบถ่ายรูปภาพเพื่อการพิสูจน์เอกสารลักษณ์ยาเม็ด	80
5.5	ผลการทดสอบการ routing ของเว็บเซอร์วิส	82
6	สรุปและข้อเสนอแนะ	84
6.1	สรุปความสามารถของระบบ	84
6.2	ปัญหาและอุปสรรคในการพัฒนา	85
6.3	แนวทางการพัฒนาต่อ	86
	บรรณานุกรม	89
	ภาคผนวก	92
ภาคผนวก ก	การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม	92
ก.1	การติดตั้ง Node.js	92
ก.2	การติดตั้ง Ionic Framework	95
ก.3	การติดตั้ง OpenCV	95
ภาคผนวก ข	การติดตั้งแอปพลิเคชันค้นหายาเพื่อคุณ	100
ภาคผนวก ค	คู่มือการใช้งานแอปพลิเคชันค้นหายาเพื่อคุณ	101
	ประวัติผู้เขียน	108

สารบัญตาราง

ตารางที่	หน้า
1.1 ขั้นตอนการดำเนินงาน	4
3.1 สัญลักษณ์ของ Use case Diagram	31
3.2 สัญลักษณ์ของ Class Diagram	34
3.3 สัญลักษณ์ของ Sequence Diagram	37
3.4 สัญลักษณ์ของ State Diagram	44
3.5 ตารางแสดงจำนวนของข้อมูลรูปทรงযามีเดที่ใช้ฝึกและทดสอบ	48
3.6 แสดงความถูกต้องของแบบจำลอง SVM	50
3.7 แสดงความถูกต้องของแบบจำลอง KNN	51
3.8 แสดงความถูกต้องของแบบจำลอง RFC	52
3.9 สรุปผลการทดลอง	53
3.10 ตารางแสดงรหัสสีจากฐานข้อมูลพิสูจน์เอกสารลักษณ์ยามีเดท	55
5.1 ผลการทดสอบค้นหายาแบบทั่วไป	78
5.2 ผลการทดสอบการค้นหายาแบบทึ่นสูง	78
5.3 ผลการทดสอบเลือกตัวรายละเอียดยา	79
5.4 ผลการทดสอบถ่ายรูปภาพเพื่อการพิสูจน์เอกสารลักษณ์ยามีเดท	80
5.5 ผลการทดสอบการ rounding ของเว็บเซอร์วิส	82

สารบัญภาพ

รูปที่		หน้า
2.1	โครงสร้างของ Ionic Framework	6
2.2	โครงสร้างของ Cordova	6
2.3	ความแตกต่างสถาปัตยกรรมระหว่าง Containers และ Virtual Machines . .	9
2.4	คำสั่งในการใช้งาน Docker	11
2.5	รูปแบบโครงสร้างของ HTTP Request	13
2.6	รูปแบบโครงสร้างของ HTTP response	14
2.7	ตัวอย่างรูปแบบของ JSON	15
2.8	ตัวอย่างรูปแบบของ JSON Web Token	16
2.9	ตัวอย่างการสร้าง JSON Web Token	16
2.10	ตัวอย่างการตรวจสอบ JSON Web Token	17
2.11	ภาพแบบ Binary	18
2.12	ภาพแบบ Grayscale	19
2.13	ภาพแบบ RGB	19
2.14	สมการการทำ Thresholding รูปภาพ	20
2.15	ตัวอย่างการทำ Thresholding รูปภาพต้นฉบับขนาด 10x10 พิกเซล	20
2.16	ตัวอย่างการทำ Thresholding รูปภาพผลลัพธ์ขนาด 10x10 พิกเซล	21
2.17	ตัวอย่างการตรวจหา Canny Edge	22
2.18	ตัวอย่างการใช้งานฟังก์ชัน cv2.findContours() โดยกำหนดค่าพารามิเตอร์ เป็น cv2.CHAIN_APPROX_NONE	23
2.19	ตัวอย่างการใช้งานฟังก์ชัน cv2.findContours() โดยกำหนดค่าพารามิเตอร์ เป็น cv2.CHAIN_APPROX_SIMPLE	23
2.20	ผลลัพธ์ของการหา Contours	24
2.21	รูปแบบ kernel function ในแบบต่างๆ	24
3.1	ภาพรวมของแอปพลิเคชันค้นหายาเพื่อคุณ	29
3.2	Use Case Diagram ของแอปพลิเคชันค้นหายาเพื่อคุณ	32
3.3	Class Diagram ของแอปพลิเคชันค้นหายาเพื่อคุณ	35
3.4	Sequence Diagram ของการค้นหายาแบบทั่วไป	38
3.5	Sequence Diagram ของการค้นหายาแบบขั้นสูง	39
3.6	Sequence Diagram ของการถ่ายรูปภาพเพื่อพิสูจน์เอกสารลักษณ์ยาเม็ด	41
3.7	Sequence Diagram ของการดูรายการบุ๊กมาร์ก	42
3.8	State Diagram ส่วนการค้นหายาและการแสดงรายละเอียดยา	45
3.9	State Diagram ส่วนการถ่ายรูปเพื่อการพิสูจน์เอกสารลักษณ์ยา	46
3.10	ตัวอย่างข้อมูลสำหรับการฝึกแบบจำลองรูปทรงสี่เหลี่ยม	48
3.11	แสดงการขั้นตอนการทำความสะอาดข้อมูลรูปสำหรับใช้ฝึกและสอน	49
3.12	การกำหนดค่าพารามิเตอร์ของแบบจำลอง SVM	49

3.13	การกำหนดค่าพารามิเตอร์ของแบบจำลอง KNN	50
3.14	การกำหนดค่าพารามิเตอร์ของแบบจำลอง RFC	51
3.15	แสดงการวัดขนาดวัตถุด้วยการใช้วัตถุอ้างอิง	54
4.1	การค้นหาแบบทั่วไป	57
4.2	การค้นหาแบบขั้นสูง	58
4.3	การถ่ายรูปภาพ	60
4.4	การเรียกใช้งานเว็บเซอร์วิส	61
4.5	การติดต่อฐานข้อมูลของเว็บเซอร์วิส	62
4.6	การสร้างและการตรวจสอบ JSON Web Token	63
4.7	การถอดรหัสรูปภาพ BASE64	65
4.8	การ routing ของเว็บเซอร์วิส	67
4.9	การประมวลผลภาพ	70
4.10	การประมวลผลภาพ(ต่อ)	71
4.11	การประมวลผลภาพ(ต่อ)	72
4.12	การประมวลผลภาพ(ต่อ)	73
4.13	การประมวลผลภาพ(ต่อ)	74
ก.1	หน้าเว็บดาวน์โหลด Node.js	92
ก.2	ไฟล์ติดตั้งสำหรับติดตั้ง Node.js	93
ก.3	หน้าต่างตอนรับของ Node.js	93
ก.4	หน้าต่างข้อตกลงในการใช้ Node.js	94
ก.5	หน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง Node.js	94
ก.6	หน้าต่างติดตั้ง Node.js	95
ก.7	คำสั่งสำหรับติดตั้ง ionic framework	95
ก.8	หน้าเว็บสำหรับดาวน์โหลด OpenCV	96
ก.9	ไฟล์ติดตั้ง OpenCV	96
ก.10	เลือกโฟรเดอร์สำหรับติดตั้ง OpenCV	96
ก.11	เปิด properties ของ This PC	97
ก.12	หน้าต่างของ System	97
ก.13	หน้าต่าง System Properties	98
ก.14	หน้าต่าง Environment Variables	99
ข.1	หน้าดาวน์โหลดแอปพลิเคชันค้นหาเพื่อคุณผ่านบริการ Google Play	100
ค.1	หน้าแรกของแอปพลิเคชันค้นหาเพื่อคุณ	101
ค.2	หน้าค้นหาของแอปพลิเคชันค้นหาเพื่อคุณ	102
ค.3	หน้ารายการการค้นหา	103
ค.4	หน้าแสดงรายละเอียดยา	105
ค.5	หน้าถ่ายรูปภาพ	106
ค.6	หน้าแสดงผลลัพธ์การพิสูจน์เอกสารลักษณะยา	107

บทที่ 1

บทนำ

1.1 ที่มาและเหตุผล

ปัญหาจากการได้รับประทานยาปลอมหรือยาที่ถูกเพิกถอนทะเบียนเป็นปัญหาใหญ่ในปัจจุบัน เพื่อป้องกันปัญหานี้ก่อนจะมีการใช้ยาได ๆ ที่ไม่รู้จัก จึงต้องมีการตรวจสอบยาให้ละเอียด ก่อน แต่ในปัจจุบันการพิสูจน์เอกลักษณ์ยาเม็ดหรือแคปซูลเป็นไปได้ยาก เนื่องมาจากความคล้ายคลึงกันของยาเม็ดหรือแคปซูลในหลายชนิด ทั้งขนาด สี รูปทรง ชื่อยา เป็นต้น ดังนั้นการพิสูจน์เอกลักษณ์ยาเม็ดหรือแคปซูลจำเป็นต้องอาศัยเภสัชกรหรือผู้เชี่ยวชาญที่มีประสบการณ์และความสามารถในการจดจำในการพิสูจน์เอกลักษณ์ยาเม็ด และการจดจำลักษณะของเม็ดยาจำนวนมาก และข้อมูลรายละเอียดของยาแต่ละชนิดจำเป็นต้องใช้การอ้างอิงข้อมูลในการพิสูจน์เอกลักษณ์ยาเม็ดที่เชื่อถือได้ เช่น คันหาในเว็บไซต์พิสูจน์ยาโดยเฉพาะที่ได้รับการรับรองจากหน่วยงานราชการ เป็นต้น

หน่วยข้อมูลยาและสุขภาพ คณะเภสัชศาสตร์ มหาวิทยาลัยอุบลราชธานี ได้เริ่มทำงานวิจัย ที่รวบรวมข้อมูลได้แก่ ชื่อการค้า ชื่อสามัญทางยา รูปแบบผลิตภัณฑ์ รูปร่างลักษณะ สัญลักษณ์ หรือตัวอักษรบนยาเม็ดหรือแคปซูล สี ขนาดของยาเม็ดและแคปซูล บริษัทผู้ผลิต และบริษัทผู้จำหน่ายโดยเริ่มจากการจัดทำฐานข้อมูลการพิสูจน์เอกลักษณ์ของยาเม็ดและแคปซูลก่อนและได้สร้างเว็บไซต์ Drug Identification Database [13] เพื่อใช้ในการเป็นฐานข้อมูลในการพิสูจน์เอกลักษณ์ยา

เนื่องจากการใช้เว็บไซต์ Drug Identification Database ยังไม่สามารถรองรับการใช้งานผ่านโทรศัพท์สมาร์ทโฟนและเพื่อความสะดวกของการใช้งาน ผู้พัฒนาจึงได้ทำการแอปพลิเคชันค้นหายาเพื่อคุณที่ผู้ใช้งานสามารถสืบค้นหารายละเอียดยาได้และสามารถถ่ายรูปยาเม็ดเพื่อการพิสูจน์เอกลักษณ์ของยาเม็ดหรือแคปซูลได้โดยไม่ต้องอาศัยเภสัชกรหรือผู้เชี่ยวชาญในการจดจำในการพิสูจน์เอกลักษณ์ยาและสามารถตอบโต้กับผู้ใช้งานได้ทันที โดยใช้ความเป็นโทรศัพท์

สมาร์ทโฟนที่สามารถใช้งานได้ที่ทุกที่ทุกเวลา

1.2 วัตถุประสงค์

1. เพื่อพัฒนาแอปพลิเคชันที่สามารถการสืบค้นข้อมูล และ การพิสูจน์เอกสารยังไง เมื่อยาหรือแคปชัลได้
2. เพื่อให้การพิสูจน์เอกสารของยาเม็ดหรือแคปชัลสามารถทำงานและคุณลักษณะได้ด้วยผู้ใช้งานเอง

1.3 ขอบเขตของโครงการ

1. แอปพลิเคชันสามารถทำงานบนระบบทำงานบนโทรศัพท์สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ได้
2. แอปพลิเคชันสามารถสืบค้นหาข้อมูลยาเม็ดหรือแคปชัลได้ โดยใช้ฐานข้อมูลพิสูจน์เอกสารยามีเดที่จัดทำโดยประเทศไทย
3. แอปพลิเคชันสามารถถ่ายรูปภาพและประมวลผลรูปภาพเพื่อพิสูจน์เอกสารยามีเดท และแคปชัลได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

เป็นแอปพลิเคชันที่ใช้ในการพิสูจน์เอกสารยามีเดทหรือแคปชัล การให้ข้อมูลผู้ใช้งานเกี่ยวกับข้อมูลการผลิตยาเม็ดหรือแคปชัล และเพื่อให้เป็นเครื่องมือในการเฝ้าระวังปัญหาที่เกิดจากการได้รับยาปลอมหรือยาที่ถูกเพิกถอนทะเบียน

1.5 เครื่องมือที่ใช้ในการพัฒนา

1. ด้านฮาร์ดแวร์
 - (a) ความต้องการของระบบสำหรับโทรศัพท์สมาร์ทโฟน

- ระบบปฏิบัติการแอนดรอยด์ 4.1 ขึ้นไป หรือ ระบบปฏิบัติการไอโอเอส 7 ขึ้นไป
- จอแสดงผล 768 x 1366 พิกเซล
- หน่วยความจำหลัก 4GB
- หน่วยความจำรอง 8GB
- ความละเอียดของกล้องถ่ายรูป 12 MP

(b) ความต้องการของระบบสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล

- ระบบปฏิบัติการ windows 7 ขึ้นไป
- หน่วยประมวลผล Intel Pentium 4, 3GHz ขึ้นไป หรือ AMD Athlon 64 3000 ขึ้นไป
- หน่วยความจำหลัก 4GB
- หน่วยความจำลอง 128GB

2. ด้านซอฟต์แวร์

- (a) Node.js ใช้สำหรับสร้างเว็บเซอร์วิสที่ติดต่อกับระหว่างผู้ใช้งานกับฐานข้อมูลด้วย RESTful API
- (b) IONIC 3 Framework เป็นเครื่องมือที่ใช้ในการพัฒนา Mobile Application แบบ Hybrid (Hybrid Mobile App คือการพัฒนาแอปพลิเคชันครั้งเดียวแล้วสามารถทำงานได้หลาย Platform)
- (c) OpenCV ย่อมาจาก Open Source Computer Vision ซึ่งเป็นไลบรารีที่รวมฟังก์ชันต่างๆสำหรับการประมวลผลภาพและคอมพิวเตอร์วิศวกรรมศาสตร์เอาไว้เป็นจำนวนมาก ไลบรารีนี้อยู่ภายใต้ใบอนุญาต BSD ซึ่งเราสามารถใช้ได้ฟรีทั้งทางด้านการศึกษาและทางการค้า นอกจากนั้น OpenCV ยังมีอินเตอร์เฟสที่หลากหลายรองรับการพัฒนาโปรแกรมบนภาษาโปรแกรมต่างๆ เช่น C/C++, Python, Java, Javascript เป็นต้น และยังสามารถรันได้ทั้งบน Window, Linux, Android, และ Mac

(d) Android SDK (Software Development Kit) เป็นชุดเครื่องมือที่เอาไว้สำหรับพัฒนาแอปพลิเคชัน Android OS

1.6 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1: ขั้นตอนการดำเนินงาน

บทที่ 2

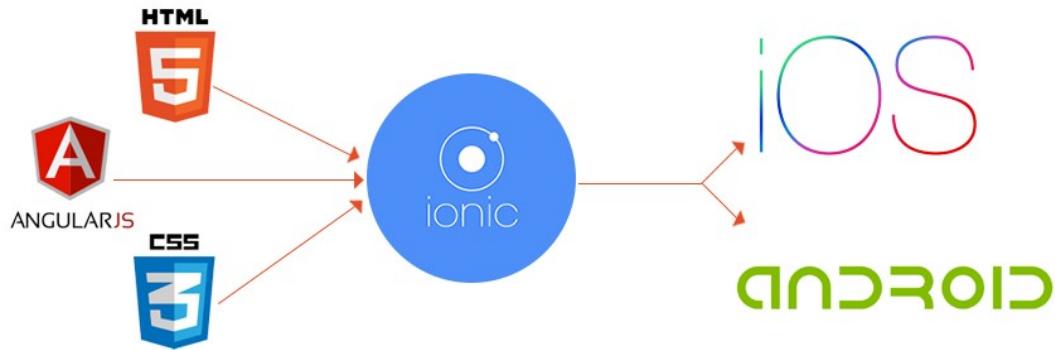
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะอธิบายถึงองค์ความรู้และทฤษฎีต่างๆ ที่จำเป็นต่อการพัฒนาแอปพลิเคชันและเว็บเซอร์วิส โดยประกอบด้วยเนื้อหาดังต่อไปนี้

- ความรู้พื้นฐานเกี่ยวกับ Ionic Framework
- ความรู้พื้นฐานเกี่ยวกับ Docker
- ความรู้พื้นฐานเกี่ยวกับ HTTP Protocol
- ความรู้พื้นฐานเกี่ยวกับ RESTful API
- การทำ Authentication ด้วย JSON Web Token
- ความรู้พื้นฐานเกี่ยวกับ OpenCV
- การหาขนาดของยาด้วยการใช้วัตถุอ้างอิง
- เครื่องมือที่ใช้ในการพัฒนา

2.1 ความรู้พื้นฐานเกี่ยวกับ Ionic Framework

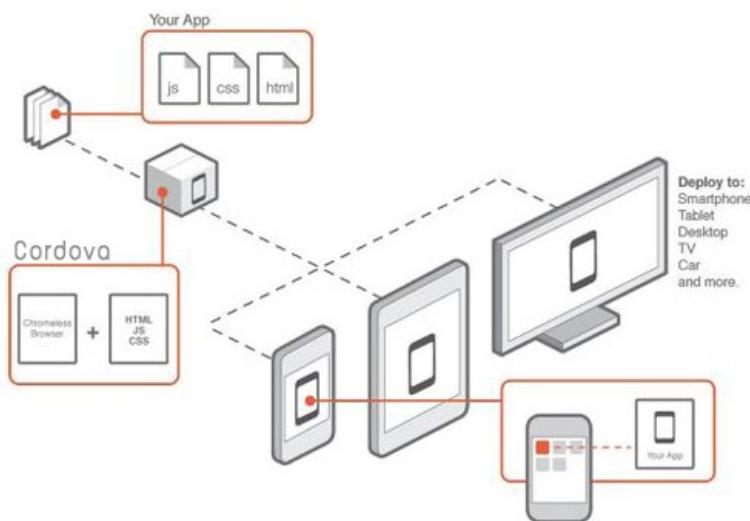
Ionic Framework [16] เป็นเครื่องมือที่ใช้ในการพัฒนา Mobile Application แบบ Hybrid Mobile Application ที่สามารถทำงานได้หลาย Platforms ทั้งระบบปฏิบัติการไอโอเอосและระบบปฏิบัติการแอนดรอยด์ โดยจะใช้เทคโนโลยีในการพัฒนาคือ HTML CSS และ JavaScript โดยเป็นภาษาหลักในการใช้พัฒนาแอปพลิเคชันเนื่องจากใช้แกนหลังเป็น AngularJS และมีการใช้งาน Command-line interface (CLI) สำหรับจัดการดูแลบริการต่างๆ ใน เช่น การสร้างหน้า การติดตั้ง เป็นต้น ผู้พัฒนาพัฒนาแอปพลิเคชันค้นหายาเพื่อคุณด้วย Ionic Framework เนื่องจาก Ionic Framework สามารถพัฒนาได้หลาย Platforms และใช้เทคโนโลยีพื้นฐานของการพัฒนาเว็บไซต์ คือ HTML CSS และ JavaScript



รูปที่ 2.1: โครงสร้างของ Ionic Framework

2.1.1 รายละเอียดของ Cordova

Cordova ถูกพัฒนาจาก Nitobi ในปี ค.ศ.2009 เป็น Open Source ที่ช่วยให้เทคโนโลยีเว็บสามารถใช้งานกับมือถือได้ ซึ่งก่อนหน้านี้มีชื่อว่า “PhoneGap” และ Cordova เป็นตัวจัดการเทคโนโลยีเว็บให้เข้าถึงการทำงานของระบบปฏิบัติการ เช่น การถ่ายรูป การเรียกไฟล์ จากตัวเครื่อง จีพีเอส เป็นต้น ด้วยการทำงานผ่านไลบรารี (Library) ซึ่งสามารถใช้ได้ทุกระบบปฏิบัติการทั้ง Window Phone, Blackberry, IOS หรือ Android



รูปที่ 2.2: โครงสร้างของ Cordova

2.1.2 Hybrid Mobile Application

Ionic Framework เป็นแอปพลิเคชันประเภท Hybrid Mobile เป็นการเขียนแอปพลิเคชันระหว่าง Native Application [11] และ Web Application [12] เพื่อแก้ไขปัญหาในการทำงานซ้ำซ้อนระหว่างระบบปฏิบัติการ ซึ่งการเขียนแอปพลิเคชันสามารถใช้งานได้กับทุกระบบปฏิบัติการ และยังสามารถเรียกใช้งานทรัพยากรของระบบปฏิบัติการของโทรศัพท์สมาร์ทโฟนนั้นได้อย่างอิสระ

2.1.3 Lifecycle events

การเปลี่ยนหน้าเพจของ Ionic framework จะเป็นการสแตก (Stack) ของหน้าเพจ มีคำสั่ง pop สำหรับดึงหน้าเพจออกจากสแตกและคำสั่ง push สำหรับซ่อนทับหน้าเพจใหม่เข้ามาในสแตก

Lifecycle events [10] จะเกิดขึ้นในระหว่างการเปลี่ยนหน้าเพจ การแสดงเพิ่มเข้ามา และการปิดหรือลบออกไปของ Component ที่มีรูปแบบการใช้งานคล้ายเพจ ในบางครั้งการกำหนดการทำงานแทรกเข้าไปในเพจ ดังต่อไปนี้

1. ionViewDidLoad: เริ่มทำงานเมื่อโหลดหน้าเพจและถูกเก็บข้อมูลในหน่วยความจำ กิจกรรมนี้จะไม่เกิดขึ้นเมื่อหน้าเพจมีการแคช (Cache) เกิดขึ้น
2. ionViewWillEnter: เริ่มทำงานเมื่อหน้าเพจถูกเรียกขึ้นมาแสดงและกลายเป็นหน้าเพจที่ใช้งานอยู่
3. ionViewDidEnter: เริ่มทำงานเมื่อหน้าเพจที่ใช้งานอยู่ถูกเรียกมาอย่างสมบูรณ์ กิจกรรมนี้จะไม่เกิดขึ้นเมื่อหน้าเพจมีการแคชเกิดขึ้น
4. ionViewWillLeave: เริ่มทำงานเมื่อกำลังจะออกจากหน้าเพจและไม่ใช่หน้าเพจที่ใช้งานอยู่
5. ionViewDidLeave: เริ่มทำงานเมื่อออกจากหน้าเพจเสร็จสิ้นแล้วและไม่ใช่หน้าเพจที่ใช้งานอยู่

6. ionViewWillUnload: เริ่มทำงานเมื่อหน้าเพจนั้นถูกทำลายหรือถูกลบหน้าเพจออกไป
7. ionViewDidEnter: เริ่มการทำงานก่อนจะโหลดหน้าเพจ ใช้สำหรับกำหนดสิทธิการเข้าไปยังหน้าเพจ
8. ionViewCanLeave: เริ่มการทำงานก่อนมีการออกจากหน้าเพจ ใช้สำหรับกำหนดเงื่อนไขในการออกจากหน้าเพจ

2.1.4 ข้อดีและข้อเสียของ Ionic Framework

1. ข้อดีของ Ionic Framework

- สามารถทำงานใช้งานได้หลายระบบปฏิบัติการ
- มีส่วนติดต่อผู้ใช้งาน (User Interface) ที่ถูกออกแบบมาสวยงาม
- ใช้เทคโนโลยี HTML CSS และ JavaScript ที่ได้รับการยอมรับและใช้งานอย่างแพร่หลาย

2. ข้อเสียของ Ionic Framework

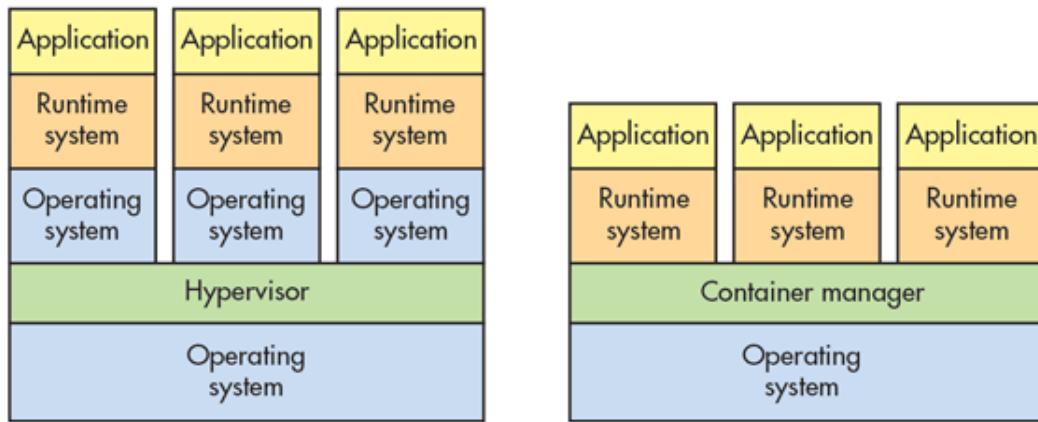
- มีข้อจำกัดการทำงานใน WebView [6] ของระบบปฏิบัติการแอนดรอยด์ 4.0 ใน การเล่น Animation รวมถึงการทำงานของ JavaScript ค่อนข้างช้า

2.2 ความรู้พื้นฐานเกี่ยวกับภาษา TypeScript

TypeScript เป็นภาษาโปรแกรมที่พัฒนาโดย Microsoft

2.3 ความรู้พื้นฐานเกี่ยวกับ Docker

Docker [9] เป็นซอฟแวร์ Open-source สำหรับสร้างแพ็จเกจของ application ที่เก็บรวมเข้าไว้ด้วยกันใน Container ที่มีการทำงานในลักษณะจำลองสภาพแวดล้อมขึ้นมาบนเครื่องเซิฟเวอร์ เพื่อใช้ในการเริ่มการทำงานของเซอร์วิสที่ต้องการ มีการทำงานคล้ายคลึงกับ Virtual Machine แต่มีข้อแตกต่างกันในการจำลองสภาพแวดล้อมทั้ง OS เพื่อใช้งาน แต่



รูปที่ 2.3: ความแตกต่างสถาปัตยกรรมระหว่าง Containers และ Virtual Machines

สำหรับ Docker จะใช้ Container ในการจำลองสภาพแวดล้อมขึ้น เพื่อใช้งานสำหรับเซอร์วิสที่ต้องการใช้งานเท่านั้น โดยไม่มีส่วนของ OS เข้าไปเกี่ยวข้องเหมือนกับ Virtual Machine แสดงดังรูปที่ 2.3

ในการติดตั้งเว็บเซอร์วิสที่เครื่องเซิฟเวอร์ที่มีสภาพแวดล้อมไม่เหมือนกับเครื่องที่ใช้พัฒนา จำเป็นจะต้องติดตั้งส่วนเสริมเพิ่มเติมเข้ามาในเครื่องเซิฟเวอร์อีก และเพื่อป้องกันปัญหานี้ ผู้พัฒนาได้พัฒนาเว็บเซอร์วิสของแอปพลิเคชันหลายเพื่อคุณไว้ใน Docker image สำหรับพร้อมติดตั้งด้วย Docker Engine

2.3.1 รายละเอียดของ Docker Concepts

1. **Images** เป็นเหมือนกับตัวตนแบบของ container ภายใต้ปกอบด้วย application ที่มีการติดตั้งเพื่อใช้งานสำหรับเซอร์วิส มีการกำหนดการตั้งค่าไว้เรียบร้อย และนำมาสร้างเป็น docker images เก็บไว้บน docker registry เพื่อนำไปใช้งาน
2. **Container** ถูกสร้างมาจาก images และเมื่อสร้างขึ้นมาจะเป็นการเริ่มทำงานของเซอร์วิส ซึ่งภายใน container แต่ละตัวจะมีการใช้งาน RAM CPU และไฟล์ config เป็นต้น เปรียบเสมือนเป็นเครื่องเซิฟเวอร์ผู้ใช้งานสามารถควบคุมและจัดการกับ binaries, dependencies ทั้งหมดได้ใน container
3. **Registries and Repositories** เป็นคลาวด์เซิร์ฟเวอร์สำหรับเก็บ image ไว้บน DockerHub

[15] และภายใน registry จะมี repositories ที่เอาไว้จัดเก็บ image ที่อยู่ใน registry

2.3.2 ข้อดีของ Docker

1. สามารถใช้งานได้บน Linux Mac และ Windows
2. มีขนาดข้อมูลเล็กและการติดตั้งได้อย่างเร็วรวด เพราะไม่มีส่วน OS
3. มีความต้องการในการใช้ CPU RAM และพื้นที่น้อยกว่า Virtual Machine
4. ลดปัญหาสภาพแวดล้อมที่ต่างกัน ระหว่างติดตั้งที่เครื่องเซิฟเวอร์
5. ผู้ใช้งานสามารถ pull image จาก docker registry ที่มีการสร้างไว้ให้แล้วมาใช้งาน

2.3.3 คำสั่งเบื้องต้น

มีคำสั่งเบื้องต้นดังต่อไปนี้

1. docker images แสดงรายการรายละเอียด images ทั้งหมดบนเครื่องผู้ใช้งาน
2. docker pull คำสั่งดึง images จาก registry มาไว้ที่เครื่องผู้ใช้งาน
3. docker ps แสดงรายการรายละเอียดของ container ที่กำลังทำงานอยู่ทั้งหมด
4. docker stop CONTAINER ID คำสั่งหยุดการทำงานของ container
5. docker start CONTAINER ID คำสั่งเริ่มการทำงานของ container
6. docker rm CONTAINER ID คำสั่งลบ container
7. docker logs CONTAINER ID คำสั่งเพื่อดูข้อมูลการทำงานของ container
8. docker run IMAGES NAME ใช้ในการสร้าง container ใหม่ ในกรณีที่ไม่มี images นั้นๆ บนเครื่องผู้ใช้งาน docker จะ pull images มาไว้ที่เครื่องผู้ใช้งานโดยอัตโนมัติ

2.3.4 การใช้งาน Docker ใน Ubuntu

การใช้งาน Docker ใน Ubuntu [17] สามารถอธิบายได้ดังรูปที่ 2.4 ดังต่อไปนี้

- บรรทัดที่ 1 – 2 เป็นการติดตั้ง Docker สามารถเรียกใช้ผ่าน Command line
- บรรทัดที่ 3 ทดสอบการทำงานของ Docker ด้วยการเรียกใช้งาน Image ที่ชื่อ “hello-world”
- บรรทัดที่ 4 – 8 ถ้าที่เครื่องผู้ใช้งานไม่มี image นั้น Docker จะทำการดาวโหลดจาก Docker Hub มาที่เครื่องผู้ใช้งาน
- บรรทัดที่ 9 Docker สามารถทำงานได้ตามปกติ และเรียกใช้งาน Image hello-world ได้สำเร็จ

```

1 sudo apt - get update
2 sudo apt - get install - y docker - ce
3 docker run hello-world
4 Unable to find image 'hello-world:latest'
5 locally latest: Pulling from library / hello - world
6 78445 dd45222: Pull complete Digest: sha256:
    c5515758d4c5e1e838e9cd307f6c6a0d 620
    b5e07e6f927b07d05f6d12a1ac8d7
7 Status: Downloaded newer image
8 for hello - world: latest
9 Hello from Docker! This message shows that your
    installation appears to be working correctly.

```

รูปที่ 2.4: คำสั่งในการใช้งาน Docker

2.4 ความรู้พื้นฐานเกี่ยวกับ HTTP Protocol

HTTP (Hypertext Transfer Protocol) เป็นโปรโตคอลที่อยู่ในส่วนของ Application Layer และเป็นโปรโตคอลสื่อสารสำหรับการแลกเปลี่ยนสารสนเทศอินเทอร์เน็ตของ World Wide Web (WWW) ซึ่งมีโครงสร้างเป็นตัวอักษรและตัวเลข (Text) เรียกว่า ทรัพยากร (Resource) โดยทรัพยากรเป็นข้อมูลต่างๆ เช่น HTML ไฟล์ รูปภาพ และวิดีโอ เป็นต้น

แอปพลิเคชันค้นหาฯเพื่อคุณอาศัย HTTP ในการสื่อสารกับเว็บเซอร์วิส โดยใช้ HTTP Method สำหรับร้องขอทรัพยากร (HTTP Request) จากเว็บเซอร์วิสและคืนทรัพยากร (HTTP Response) จากเว็บเซอร์วิสตามายังเครื่องผู้ใช้งาน

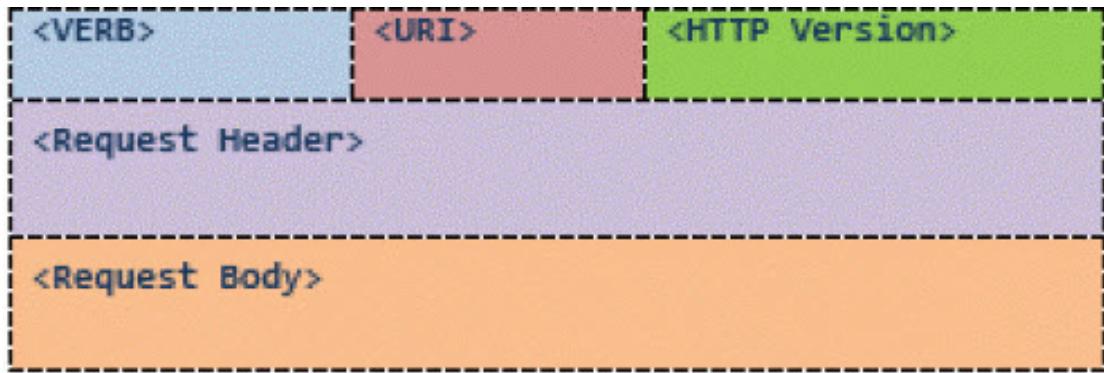
2.4.1 HTTP Method

HTTP ได้กำหนดคำสั่งร้องขอไว้ทั้งหมด 8 คำสั่ง แสดงการกระทำที่ต้องการ เพื่อที่จะดำเนินการกับทรัพยากรที่ถูกระบุ ดังต่อไปนี้

1. HEAD ร้องขอการตอบรับจากทรัพยากรที่ระบุ คล้ายกับ GET แต่จะไม่มีส่วนเนื้อหาที่ร้องขอกลับมา คำสั่งนี้ใช้ประโยชน์ในการตรวจสอบข้อมูลส่วนหัวของการตอบรับ โดยไม่จำเป็นต้องส่งเนื้อหาเต็มมาทั้งหมด
2. GET ร้องขอการนำเสนอดอกทรัพยากรที่ระบุและมีส่วนของเนื้อหาที่ร้องขอ
3. POST ส่งข้อมูลไปยังทรัพยากรที่ระบุเพื่อให้นำไปประมวล
4. PUT อัปโหลดการนำเสนอดอกทรัพยากรที่ระบุ
5. DELETE ลบทรัพยากรที่ระบุ
6. TRACE ส่งข้อมูลร้องขอกลับมา เครื่องลูกข่ายจะเห็นว่ามีข้อมูลที่สื่อถึงเพิ่มหรือเปลี่ยนแปลงข้อความร้องขอ ก่อนไปถึงทรัพยากรปลายทาง
7. OPTIONS คืนค่าเป็นรายชื่อคำสั่ง HTTP ที่เครื่องแม่ข่ายนั้นรองรับสำหรับทรัพยากรที่ระบุ
8. CONNECT แปลงการเชื่อมต่อของการร้องขอไปเป็น TCP/IP [14]

2.4.2 รายละเอียดของ HTTP Request

รูปแบบโครงสร้างของ HTTP Request เป็นการข้อมูลโดยผู้ใช้งานไปยังเซิฟเวอร์เพื่อให้ส่งข้อมูลตอบกลับมาที่ผู้ใช้งานแสดงดังรูปที่ 2.5 และรายละเอียดใน HTTP Request มีดังนี้



รูปที่ 2.5: รูปแบบโครงสร้างของ HTTP Request

2.4.3 รายละเอียดของ HTTP Response

1. <VERB> เป็นส่วนของ HTTP method
2. <URL> เป็นตำแหน่งของสถานที่ข้อมูลที่ต้องการให้ระบบทำงาน
3. <HTTP Version> เป็นเวอร์ชันของ HTTP
4. <Request Header> เป็นส่วนของ Metadata ที่ใช้เก็บค่า key-value ของ Header เพื่อบอกข้อมูลผู้สั่ง
5. <Request Body> เป็นส่วนข้อมูล Content ใน REST

2.4.4 รายละเอียดของ HTTP Response

รูปแบบโครงสร้างของ HTTP Response คือการส่งข้อมูลที่ทางเซิฟเวอร์ตอบรับกลับไปยังผู้ใช้งานตามที่ได้อ่านมาแสดงดังรูปที่ 2.6 และรายละเอียดใน HTTP Response มีดังนี้

1. <HTTP Version> เป็นเวอร์ชันของ HTTP
2. <Response Code> เป็นผลลัพธ์การทำงานในระบบ HTTP เป็นตัวเลข 3 หลัก เช่น 2XX การร้องขอสำเร็จ, 3XX การเปลี่ยนทาง, 4XX ความผิดพลาดจากเครื่องผู้ใช้งาน, 5XX ความผิดพลาดจากเครื่องเซิฟเวอร์



รูปที่ 2.6: รูปแบบโครงสร้างของ HTTP response

3. <Request Body> เป็นส่วนข้อมูลผลลัพธ์ Content ใน REST ที่เชิฟเวอร์ตอบกลับมาที่ผู้ใช้งาน

2.5 ความรู้พื้นฐานเกี่ยวกับ RESTful API

REST (Representational state transfer) คือการสร้างเว็บเซอร์วิสชนิดหนึ่งที่อาศัย HTTP Method GET, POST, PUT และ DELETE ในการทำงาน ใช้หลักการแบบ stateless คือไม่มีการใช้งาน session และส่งผลกลับมาในรูปแบบของ JSON หรือ XML มีขนาดข้อมูลที่เล็ก สามารถส่งผลให้สามารถรับส่งข้อมูลไปมาข้าม Platform ได้อย่างสะดวก เนื่องจากเป็นการเรียกใช้งานผ่าน HTTP Protocol ที่ใช้ในการเรียกเว็บไซต์ จึงเป็นที่นิยม และแอปพลิเคชันค้นหาฯเพื่อคุณกับเว็บเซอร์วิสติดต่อสื่อสารผ่าน HTTP โดยใช้ HTTP method ในการร้องขอทรัพยากรจากเว็บเซอร์วิสและส่งทรัพยากรกลับมาที่แอปพลิเคชันในรูปแบบของ JSON ดังแสดงในรูปที่ 2.7

2.6 การทำ Authentication ด้วย JSON Web Token

2.6.1 การพิสูจน์ตัวตน (Authentication)

ในบางแอปพลิเคชันจะสามารถใช้งานได้จำเป็นต้องรู้จักผู้ใช้งานที่กำลังใช้งานอยู่ ซึ่งระบบที่ทำหน้าที่ในการตรวจสอบ เรียกว่า “Authentication”

ผู้พัฒนาได้พัฒนาการพิสูจน์ตัวตนก่อนใช้งานเว็บเซอร์วิสด้วย JSON Web Token [1]

```
{
  - results: [
    - {
      drugid: 14133,
      path: "dimg/14133/14133_1.JPG",
    },
    - {
      drugid: 6385,
      path: "dimg/6385/6385_1.Medroxyprogesteron acetate_1.JPG",
    },
    - {
      drugid: 6385,
      path: "dimg/6385/6385_1.Medroxyprogesteron acetate_1.JPG",
    },
    - {
      drugid: 10968,
      path: "dimg/10968/10968_1.jpg",
    },
  ],
}
```

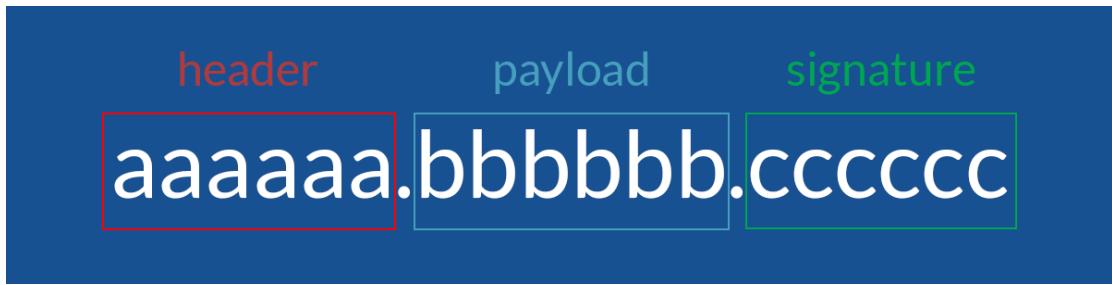
รูปที่ 2.7: ตัวอย่างรูปแบบของ JSON

เพื่อป้องการเข้าถึงข้อมูลจากผู้ที่ไม่พึงประสงค์ โดยการร้องขอทรัพยากรไปยังเว็บเซอร์วิสจากแอปพลิเคชันค้นหายาเพื่อคุณ จะฝัง JSON Web Token ไว้ใน Header ของ HTTP Request เพื่อไว้สำหรับการพิสูจน์ตัวตนของเว็บเซอร์วิส และรายละเอียดของ JSON Web Token มีดังนี้

2.6.2 รายละเอียดของ JSON Web Token (JWT)

JSON Web Token เป็นมาตรฐานเปิด (RFC 7519) ที่เข้ามาแก้ปัญหาการส่งข้อมูลอย่างปลอดภัยระหว่างแอปพลิเคชันและเว็บเซอร์วิส โดยถูกออกแบบให้มีขนาดที่กระหัด (Compact) และเก็บข้อมูลภายในตัว (Self-contained) JWT เป็น Token หรือ ชุดตัวอักษร โดยมีโครงสร้างแบ่งออกเป็น 3 ส่วน ได้แก่

1. Header สำหรับเก็บประเภทการเข้ารหัสของโทเคน
2. Payload สำหรับเก็บข้อมูล
3. Signature ส่วน Digital Signed ซึ่งเหมือนกับลายเซ็นที่ท้ายไว้ตรวจสอบว่าเป็นโทเคนที่ถูกสร้างอย่างถูกต้องหรือไม่ถูกต้อง และถ้าหากมีผู้เขียนแปลงข้อมูลของโทเคนจะทำให้



รูปที่ 2.8: ตัวอย่างรูปแบบของ JSON Web Token

การตรวจสอบ Signature ไม่ถูกต้อง

2.6.3 การใช้งาน JSON Web Token

ผู้พัฒนาเลือกพัฒนา JWT ด้วย Node.js ที่มีแพ็กเกจรองรับการใช้งาน JWT

1. การสร้าง JSON Web Token จากรูปที่ 2.9 สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 เรียกแพ็กเกจ jsonwebtoken มาเก็บที่ตัวแปร jwt
- บรรทัดที่ 2 เรียกใช้งานฟังก์ชัน sign สำหรับการสร้าง JSON WebToken โดยกำหนดให้ data คือ“foobar” คำลับ คือ “secret” และเวลาหมดอายุของ Token คือ 1 ชั่วโมง

```

1 var jwt = require('jsonwebtoken');
2 var token = jwt.sign({
3   data: 'foobar'
4 }, 'secret', {
5   expiresIn: '1h'
6 });

```

รูปที่ 2.9: ตัวอย่างการสร้าง JSON Web Token

2. การตรวจสอบ JSON Web Token ใน การตรวจสอบความถูกต้องของโทเคน คำลับในการสร้างและการตรวจสอบต้องเหมือนกัน โทเคนจะต้องไม่หมดอายุและโทเคนจะต้องไม่ถูกเปลี่ยนตัวอักษร จากรูปที่ 2.10 สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 เรียกใช้งานฟังก์ชัน verify สำหรับตรวจสอบความถูกต้องของ token โดยกำหนดค่าลับ คือ ‘secret’
- บรรทัดที่ 2 เมื่อการตรวจสอบ token เสร็จแล้ว จะได้ผลลัพธ์ในตัวแปรชื่อว่า decoded ถ้าหากการตรวจสอบถูกต้องตัวแปร err จะมีค่าเป็น null และถ้าหากการตรวจสอบไม่ถูกต้องตัวแปร err มีค่าเป็นข้อความผิดพลาดจากการตรวจสอบ

```

1  jwt.verify(token, 'secret', function(err, decoded) {
2      console.log(decoded)
3  });

```

รูปที่ 2.10: ตัวอย่างการตรวจสอบ JSON Web Token

2.6.4 ข้อดีและข้อเสียของ JWT

1. ข้อดีของใช้งาน JWT สำหรับการ Authentication

- JWT สามารถใช้งานได้กับทุกภาษาที่รองรับข้อมูลแบบ JSON
- สามารถส่ง JWT ผ่าน HTTP ใน Header POST และ URL ได้ง่าย รวดเร็ว
- มีความปลอดภัยในตัวของมันเอง

2. ข้อเสียของใช้งาน JWT สำหรับการ Authentication

- JWT สามารถเก็บข้อมูลในตัวเอง แต่ข้อมูลนั้นต้องเป็นข้อมูลที่สามารถเปิดเผยได้
- ถ้าหาก JWT ถูกขโมยไปใช้กับเว็บเซอร์วิส ก็จะสามารถทำ Authentication API ได้
- ข้อมูลที่ถูกเก็บไว้ใน JWT จะไม่สามารถถูกอัพเดทข้อมูลได้

2.7 ความรู้พื้นฐานเกี่ยวกับ OpenCV (Open Source Computer Vision)

OpenCV เป็นไลบรารี (Library) ที่รวมฟังก์ชันสำหรับการประมวลผลภาพและคอมพิวเตอร์วิทศนศาสตร์ไว้เป็นจำนวนมาก อยู่ภายใต้ใบอนุญาต BSD ซึ่งสามารถใช้ได้ฟรีทั้งทางด้าน

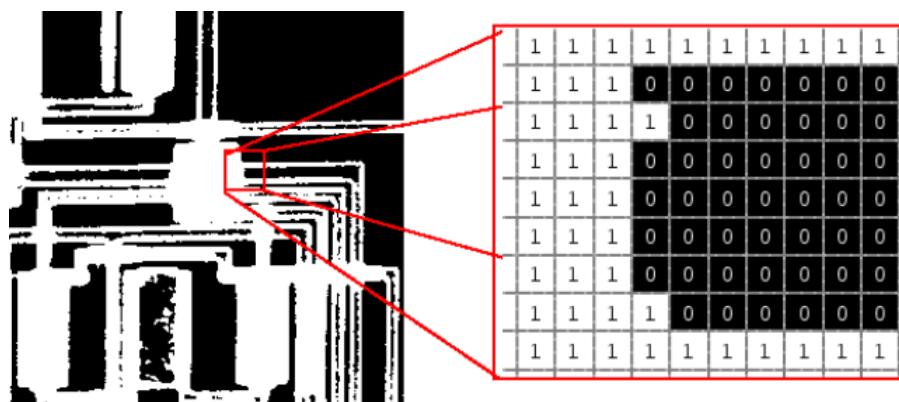
การศึกษาและทางการค้า เนื่องจากนี้ยังมีการประยุกต์ใช้งาน OpenCV ในแบบต่างๆ ได้แก่ ระบบตรวจจับใบหน้า (Face Detection) การจำจำใบหน้า (Face Recognition) การติดตามวัตถุ (Object Tracking) การเรียนรู้ของเครื่อง (Machine Learning) เป็นต้น

2.7.1 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพคือการนำรูปที่มีอยู่แล้ว หรือรูปที่รับเข้ามาจากอุปกรณ์ต่างๆ หรือรูปที่มีอยู่มาประมวลผลเพื่อหาลักษณะเด่นบางประการของรูปที่มีอยู่ หรืออาจจะเป็นการตีความหมายของภาพ รวมถึงการปรับคุณลักษณะของภาพให้เป็นไปตามต้องการ โดยใช้กระบวนการทางคณิตศาสตร์

การประมวลผลภาพแนวคิดทางคณิตศาสตร์ที่ใช้ในการประมวลผล Signal processing มาทำการประยุกต์ใช้กับสัญญาณภาพ และภาพจะเก็บอยู่ในรูปของอาร์เรย์ (Array) โดยกลุ่มของของอาร์เรย์กลุ่มนึงจะเป็นค่าของภาพหนึ่งพิกเซล เช่น ภาพแบบ RGB ใช้อาร์เรย์ 3 ช่องเพื่อกำกับค่าสีของ RGB ในหนึ่งพิกเซล รูปแบบการจัดเก็บภาพแต่ละชนิดจะแตกต่างกัน ขึ้นอยู่กับระบบสีของภาพ โดยแบ่งชนิดของภาพได้ดังนี้

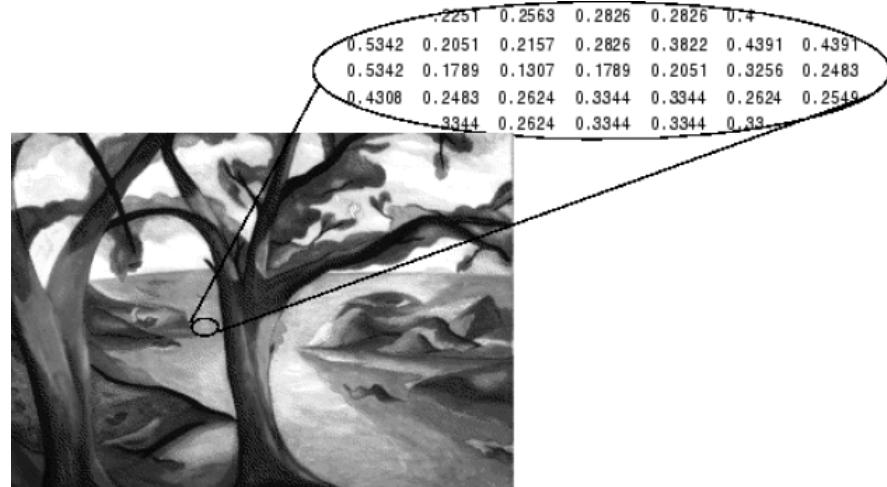
- Binary Image เป็นรูปที่มีสีเพียงสองระดับคือสีขาวและสีดำ โดยในอาร์เรย์ซึ่งนั้นมีค่าคือ 0 หมายความว่าดำ และ 255 หมายความว่าขาว ดังรูปที่ 2.11



รูปที่ 2.11: ภาพแบบ Binary

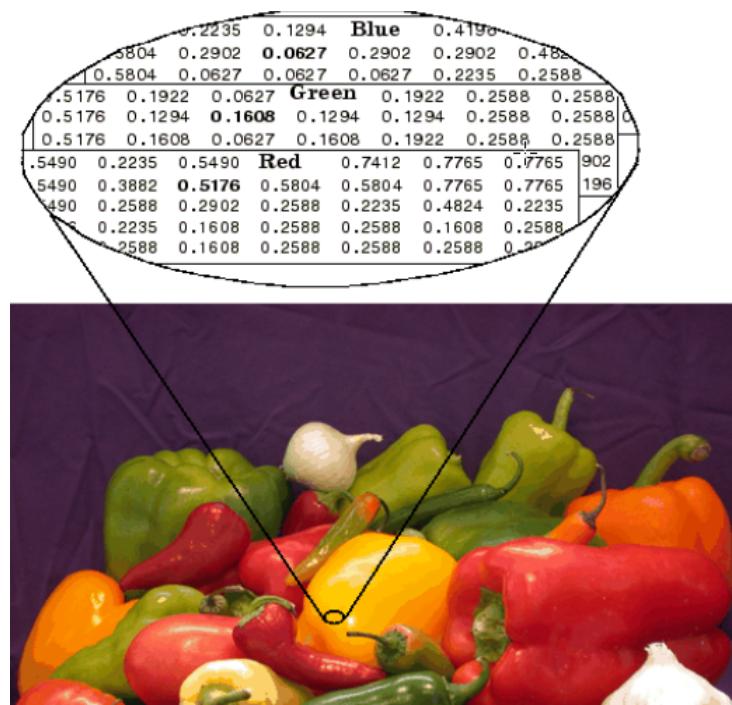
- Grayscale Image เป็นรูปที่มี channel เดียว โดยเก็บเป็นอาร์เรย์คล้ายกับภาพ Binary

Image แต่ค่าที่อยู่ในอาร์เรย์เป็นค่าความสว่าง ซึ่งมีค่าได้ตั้งแต่ 0 ถึง 255 ดังรูปที่ 2.12



รูปที่ 2.12: ภาพแบบ Grayscale

- RGB Image เป็นรูปที่มี 3 channel โดยภาพจะเก็บอยู่ในรูปภาพโดยเรียงตามลำดับ BGR แต่ถ้าอยู่ในไฟล์ภาพจะเรียงตามปกติ คือ RGB ดังรูปที่ 2.13



รูปที่ 2.13: ภาพแบบ RGB

2.7.2 การทำ Thresholding รูปภาพ

การทำ Thresholding เป็นวิธีการแยกบริเวณรูปภาพ (Image Segmentation) ของภาพสีเทา (Grayscale) จากรูปที่ 2.14 ผู้พัฒนากำหนดค่า threshold เท่ากับ 120 ในรูปภาพค่าพิกเซลแต่ละพิกเซลของภาพต้นฉบับที่มีค่าน้อยกว่า 120 จะถูกกำหนดเป็นค่าพิกเซลของภาพผลลัพธ์เป็น 0 (สีขาว) และถ้าในรูปภาพค่าพิกเซลแต่ละพิกเซลของภาพต้นฉบับนั้นมีค่ามากกว่าเท่ากับ 120 จะถูกกำหนดพิกเซลของภาพผลลัพธ์เป็น 255 (สีดำ) จะได้สมการดังรูปที่ เมื่อ $f(x, y)$ คือ ตำแหน่งพิกเซลของภาพต้นฉบับ และ $g(x, y)$ คือ ตำแหน่งพิกเซลของภาพผลลัพธ์

ตัวอย่างการทำ Thresholding ได้แก่ การรูปภาพต้นฉบับนำมาเข้าสมการตามรูปที่ 2.15 และกำหนดค่า threshold เท่ากับ 120 ดังนั้นผลลัพธ์จะได้ดังรูปที่ 2.16

$$\begin{aligned} g(x, y) &= 0 && \text{if } f(x, y) \geq \text{threshold value} \\ g(x, y) &= 255 && \text{if } f(x, y) < \text{threshold value} \end{aligned}$$

รูปที่ 2.14: สมการการทำ Thresholding รูปภาพ

230	255	210	231	255	170	200	150	230	255
240	230	199	222	255	240	230	230	240	230
245	240	0	150	170	20	120	119	245	240
244	244	20	230	240	15	150	90	244	244
210	231	20	200	150	16	230	80	210	231
199	222	20	230	230	89	100	100	199	222
200	150	15	17	170	90	210	231	200	150
230	230	16	18	240	95	199	222	230	230
255	170	255	170	210	231	255	170	255	170
255	240	255	240	199	222	255	240	255	240

รูปที่ 2.15: ตัวอย่างการทำ Thresholding รูปภาพต้นฉบับขนาด 10×10 พิกเซล

255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	0	255	255	0	0	0	255	255
255	255	0	255	255	0	255	0	255	255
255	255	0	255	255	0	255	0	255	255
255	255	0	255	255	0	0	0	255	255
255	255	0	255	255	0	255	255	255	255
255	255	0	0	255	0	255	255	255	255
255	255	0	0	255	0	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255

รูปที่ 2.16: ตัวอย่างการทำ Thresholding รูปภาพผลลัพธ์ขนาด 10×10 พิกเซล

2.7.3 การตรวจหา Canny Edge

การหาขอบภาพเป็นการหาเส้นรอบวัตถุที่อยู่ในภาพ เมื่อทราบเส้นรอบวัตถุจะสามารถคำนวณหาพื้นที่หรือจำแนกชนิดของวัตถุนั้นได้ ขอบภาพเกิดจากความแตกต่างของความเข้มแสงจากจุดหนึ่งไปยังอีกจุดหนึ่ง การตรวจหา Canny Edge [5] เป็นอัลกอริทึมการตรวจหาขอบที่เป็นที่นิยม ได้รับการพัฒนาโดย John F. Canny เป็นอัลกอริทึมแบบหลายขั้นตอนและจะดำเนินการผ่านแต่ละขั้นตอน

2.7.3.1 ลดสัญญาณรบกวน (Noise Reduction)

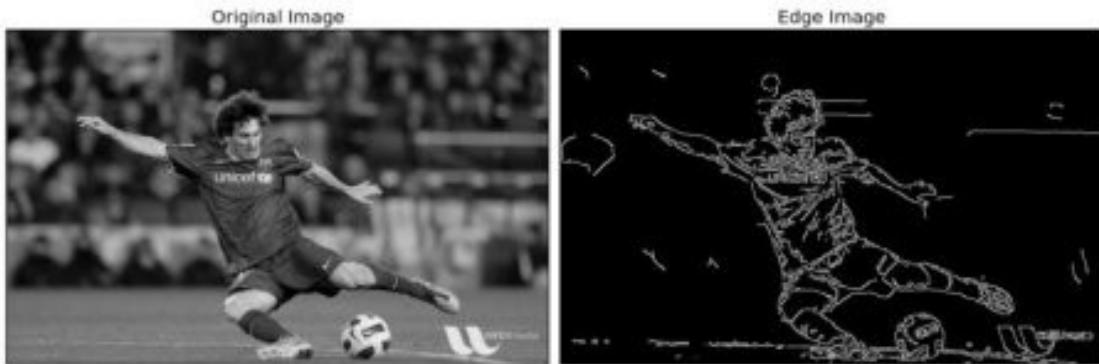
เนื่องจากการตรวจหาขอบมีความอ่อนไหวต่อสัญญาณรบกวนในภาพ ขั้นตอนแรกคือการขจัดสัญญาณรบกวนด้วยกรองแบบเกาส์ (Gaussian) 5×5

2.7.3.2 การหาการไล่ระดับสีความเข้มของภาพ (Gradient)

รูปที่ Smoothing จะถูกกรองด้วย Sobel kernel ทั้งในแนวอนและแนวตั้งจากจุดหนึ่งไปยังอีกจุด โดยการหาจุดในแนวอน (G_x) และในแนวตั้ง (G_y) จากทั้งสอง ภาพนี้สามารถทำการไล่ระดับสีและทิศทางของขอบสำหรับแต่ละพิกเซล

2.7.3.3 Nonmaxima Suppression

หลังจากที่ผ่านขั้นตอนแรกมาแล้ว รูปที่ได้อาจจะมีเส้นขอบที่ไม่ใช่ขอบที่แท้จริงปรากฏอยู่เนื่องจากสัญญาณรบกวนหรือลักษณะของวัตถุในภาพเป็นพื้นผิวที่มีลวดลายหรือมีรายละเอียดภายในมาก ดังนั้นเพื่อลดปัญหาดังกล่าวจึงได้มีการกำหนดค่า threshold ขึ้นมา 2 ค่า คือ High threshold (T1) และ Low threshold (T2) โดยพิกเซลที่มีค่ามากกว่า T1 จะถูกปรับเป็น 1 (ให้เป็นพิกเซลที่เป็นขอบ) แต่ถ้า้อยกว่า T2 จะถูกปรับเป็น 0 ส่วนค่าที่อยู่ระหว่างค่า threshold ทั้งสอง การปรับเป็นค่า 0 หรือ 1 ขึ้นอยู่กับพิกเซลที่อยู่รอบข้าง หากพบว่าพิกเซลที่อยู่รอบข้างของพิกเซลที่เป็นขอบมีค่ามากกว่า T2 แล้ว จะปรับค่าพิกเซลดังกล่าวให้มีค่าเป็น 1 และถ้าเป็นสมาชิกหนึ่งในภาพขอบด้วย



รูปที่ 2.17: ตัวอย่างการตรวจหา Canny Edge

2.7.4 การหา Contour Approximation

รูปทรง (Contours) สามารถอธิบายได้ง่ายๆว่าเป็นเส้นโค้งที่เชื่อมต่อกันจุดต่อเนื่องทั้งหมดตามแนวขอบที่มีสีเดียวกัน และรูปทรงเป็นเครื่องมือที่มีประโยชน์สำหรับการวิเคราะห์รูปทรง การตรวจจับวัตถุและการจดจำ

การใช้ภาพใบหนารีจิได้ผลลัพธ์ที่มีความถูกต้องที่ดีกว่า ดังนั้นก่อนที่จะหารูปทรงควรใช้ threshold และ canny edge จากภาพก่อน ใน OpenCV การหารูปทรงคือการหาวัตถุสีขาวจากพื้นหลังสีดำ ดังนั้นวัตถุที่จะตรวจจับควรเป็นสีขาวและพื้นหลังควรเป็นสีดำ

```

1 import numpy as np
2 import cv2
3 im = cv2.imread('test.jpg')
4 imgrey = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
5 ret, thresh = cv2.threshold(imgrey, 127, 255, 0)
6 im2, contours, hierarchy = cv2.findContours(thresh,
7 cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

```

รูปที่ 2.18: ตัวอย่างการใช้งานฟังก์ชัน cv2.findContours() โดยกำหนดค่าพารามิเตอร์เป็น cv2.CHAIN_APPROX_NONE

```

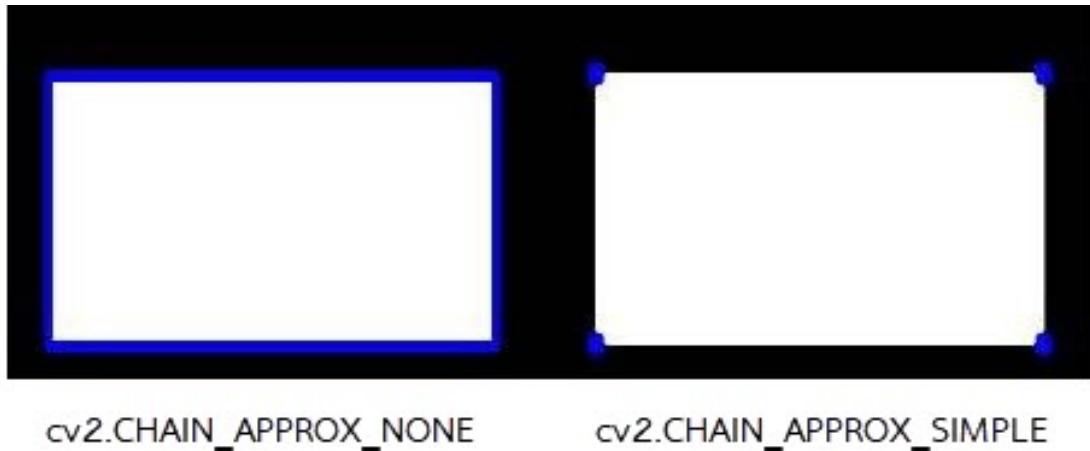
1 import numpy as np
2 import cv2
3 im = cv2.imread('test.jpg')
4 imgrey = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
5 ret, thresh = cv2.threshold(imgrey, 127, 255, 0)
6 im2, contours, hierarchy = cv2.findContours(thresh,
7 cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

```

รูปที่ 2.19: ตัวอย่างการใช้งานฟังก์ชัน cv2.findContours() โดยกำหนดค่าพารามิเตอร์เป็น cv2.CHAIN_APPROX_SIMPLE

จากรูปที่ 2.18 ในการใช้ฟังก์ชัน cv2.findContours() โดยกำหนดค่าพารามิเตอร์เป็น cv2.CHAIN_APPROX_NONE จะขอบทั้งหมดจะถูกเก็บไว้ ซึ่งแตกต่างจากรูปที่ 2.19 ที่กำหนดค่าพารามิเตอร์เป็น cv2.CHAIN_APPROX_SIMPLE จะทำการลบจุดที่ซ้ำซ้อนทั้งหมด และบีบอัดเส้นขอบทำให้ประยัดหน่วยความจำ และจากรูปที่ 2.20 รูปสี่เหลี่ยมผืนผ้าแสดงให้เห็นถึงเทคนิคนี้ เพียงจุดบนพิกัดทั้งหมดในอาร์เรย์เส้น (เส้นสีน้ำเงิน)

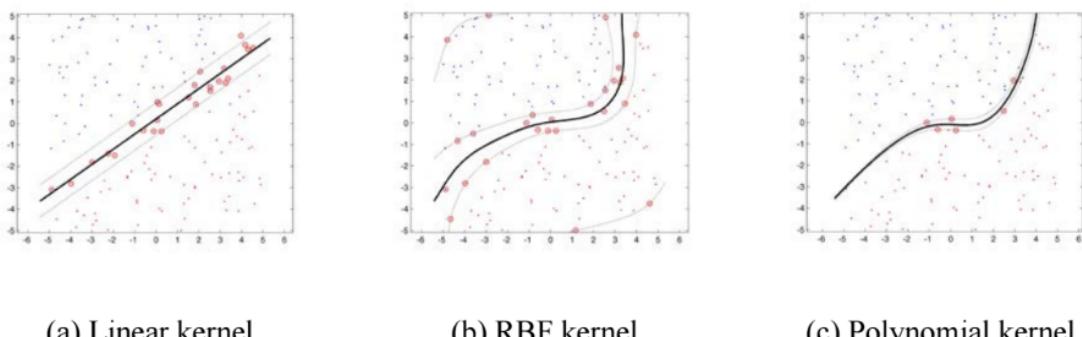
จากรูปที่ 2.20 ภาพด้านซ้ายกำหนดค่าพารามิเตอร์เป็น cv2.CHAIN_APPROX_NONE ได้ทั้งหมด 734 จุด และภาพด้านขวากำหนดค่าพารามิเตอร์เป็น cv2.CHAIN_APPROX_SIMPLE ได้ทั้งหมด 4 จุด นั้นเป็นเหตุผลที่ผู้พัฒนาจะเลือกใช้ cv2.CHAIN_APPROX_SIMPLE แทน cv2.CHAIN_APPROX_NONE เพื่อความประยัดหน่วยความจำ



รูปที่ 2.20: ผลลัพธ์ของการหา Contours

2.7.5 Support Vector Machines (SVM)

SVM เป็นอัลกอริทึมในการคัดแยกที่มีการนำมาใช้กันในด้านการประมวลผลเป็นดิจิตอล หลักการของ SVM คือการให้อินพุทที่ใช้ฝึกเป็นเวคเตอร์ในสเปช N มิติ เช่นถ้าในกรณีของ 2 มิติ และ 3 มิติ จะเป็นจุดที่อยู่ในระบบ XY และ XYZ ตามลำดับ จากนั้นทำการสร้างไฮเปอร์เพลน (Hyperplane) ที่จะแยกกลุ่มของเวคเตอร์อินพุทออกเป็นประเภทต่างๆ ในกรณีที่เป็น 2 มิติ และ 3 มิติ ไฮเปอร์เพลน คือเส้นตรงและระบบ



รูปที่ 2.21: รูปแบบ kernel function ในแบบต่างๆ

ไฮเปอร์เพลนที่ได้จากการเลือกใช้ kernel function ที่ต่างกันก็จะให้ประสิทธิภาพการ

ทำงานของโมเดลที่ต่างกัน นอกจากการเลือกใช้ kernel function ที่เหมาะสมแล้ว การกำหนดค่าต่างๆ (initial parameters) ของ kernel function ก็มีผลตัวยเช่นกันจากไฮเปอร์เพลนที่แสดงในรูปข้างบนเกิดจากการกำหนดค่าของ initial parameter ที่ต่างกันดังรูปที่ 2.21

2.7.6 K-Nearest Neighbour Algorithm

K-Nearest Neighbour Algorithm (KNN) [18] เป็นวิธีที่ใช้ในการจัดแบ่งคลาส (Classification) มีลักษณะการทำงานแบบ Supervised learning (ข้อมูลที่นำมาเรียนรู้ต้องมีคลาสกำกับไว้แล้ว) โดยเทคนิคนี้จะตัดสินใจว่า คลาสใดที่จะแทนเงื่อนไขหรือกรณีใหม่ๆ ได้บ้าง โดยการตรวจสอบจำนวนบางจำนวน (“K” ในขั้นตอนวิธีการ kNN) ของกรณีหรือเงื่อนไขที่เหมือนกันหรือใกล้เคียงกันมากที่สุด โดยจะหาผลรวม (Count Up) ของจำนวนเงื่อนไข หรือกรณีต่างๆ สำหรับแต่ละคลาส และกำหนดเงื่อนไขใหม่ๆ ให้คลาสที่เหมือนกันกับคลาสที่ใกล้เคียงกันมากที่สุด

2.7.6.1 ขั้นตอนวิธีการดำเนินการของอัลกอริทึมแบบ K-Nearest Neighbour Algorithm มีดังนี้

1. กำหนดขนาดของค่า K
2. คำนวณระยะห่าง (Distance) ของข้อมูลที่ต้องการพิจารณากับกลุ่มข้อมูลตัวอย่าง
3. จัดเรียงลำดับของระยะห่าง และเลือกพิจารณาชุดข้อมูลที่ใกล้จุดที่ต้องการพิจารณาตามจำนวนของค่า K ที่กำหนดไว้
4. พิจารณาข้อมูลจำนวน K ชุด และสังเกตกลุ่มคลาสที่มีจุดใกล้กับจุดพิจารณาเป็นจำนวนมากที่สุด
5. กำหนดคลาสให้กับจุดพิจารณาที่ใกล้จุดพิจารณามากที่สุด

2.7.6.2 การดำเนินการหลักของอัลกอริทึมแบบ K-Nearest Neighbour Algorithm ประกอบไปด้วยการทำงาน 2 ฟังก์ชัน ได้แก่

1. ฟังก์ชันระยะทาง (Distance Function) เป็นการคำนวณค่าระยะห่างระหว่างสองจุด เพื่อที่นำมารวัดความคล้ายคลึงของข้อมูล
2. ฟังก์ชันการแจกแจง (Combination Function) เป็นการรวมกันของผลลัพธ์ที่ได้จาก การคำนวณค่าระยะห่างโดยทำการเรียงลำดับค่าระยะห่างจากน้อยไปมาก จากนั้นนำมา เปรียบเทียบกับค่า K เพื่อหาคลาสของเป้าหมาย

2.7.7 Random Forest Classifier

Random Forest Classifier (RFC) [19] เป็นวิธีที่ใช้ในการจัดแบ่งคลาส โดยหลักการ สุ่มข้อมูล (Random sampling) เพื่อสร้างต้นไม้ตัดสินใจขึ้นมาจำนวนมาก ในการจำแนกคลาส ของวัตถุโดยจะอินพุทข้อมูลเวกเตอร์ (Vecter) ใส่ให้แต่ละต้นไม้ตัดสินใจที่ถูกสร้างขึ้นมา ต้นไม้ แต่ละต้นจะให้คะแนนสำหรับข้อมูลเวกเตอร์ที่ถูกป้อนเข้ามาในแต่ละคลาส วิธีการ Random Forest จะเลือกคลาสที่มีคะแนนมากที่สุดและทำนายคลาสของวัตถุ

2.8 การหาขนาดของยาด้วยการใช้วัตถุอ้างอิง

การหาขนาดยาด้วยการใช้วัตถุอ้างอิง ด้วยวิธีการเทียบบัญญติไตรยางศ์ [20] เนื่องจาก รูปภาพมีหน่วยเป็นพิกเซลและรู้ขนาดด้านยาวและกว้างของวัตถุอ้างอิงแล้ว ถ้าหากวัตถุอ้างอิงมี ขนาดด้านยาวเท่ากับ 15 เซนติเมตร ขนาดด้านกว้างเท่ากับ 10 เซนติเมตร และวัตถุอ้างอิงในรูป ที่ตรวจจับได้ด้วยการหา Contours ได้ขนาดด้านยาวเท่ากับ 98 พิกเซล ขนาดด้านยาวเท่ากับ 65 พิกเซล และขนาด 1 พิกเซล จะเท่ากับกี่เซนติเมตร ด้วยการเทียบบัญญติไตรยางศ์ จะได้ว่า 15 (เซนติเมตร) / 98 (พิกเซล) เท่ากับอัตราส่วน 0.153 เซนติเมตร/พิกเซล ดังนั้น 1 พิกเซล จะ เท่ากับ 0.153 เซนติเมตร และสามารถหาขนาดของยาด้วยการนำ 0.153 ไปคูณกับขนาดพิกเซล ของรูปภาพยา

2.9 เครื่องมือที่ใช้ในการพัฒนา

Nginx [8] มาจากคำว่า “Engine-X” เป็นเว็บเซิร์ฟเวอร์ที่มีประสิทธิภาพดีและนิยมอยู่ในปัจจุบัน ถูกคิดค้นขึ้นมาเพื่อให้สามารถที่จะรองรับการทำงานได้มากกว่า Apache[6] และนอกจักนี้ Nginx ยังมีโมดูลเสริมเข้ามาที่เพียงพอต่อการใช้งานทั่วไป และเป็นซอฟแวร์แบบ Open Source ที่สามารถใช้งานได้ฟรีโดยรองรับระบบปฏิบัติการ Linux และ ระบบปฏิบัติการ Windows

MySQL [7] เป็นโปรแกรมจัดการฐานข้อมูลที่รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช่ว่ำกับเครื่องมือหรือโปรแกรมอื่นอย่างบูรณาการ เพื่อให้ได้ระบบงานที่รองรับความต้องการของผู้ใช้ เป็นฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ผู้พัฒนาได้นำมาใช้งานกับเว็บเซอร์วิสในการจัดเก็บฐานข้อมูลการพิสูจน์เอกสารลักษณ์ยาเม็ดหรือแคปซูล

Express – Node.js [3] ถูกสร้างขึ้นบนฐานของระบบ Node.js ซึ่งมีความเร็วของการทำงานและมีความครบถ้วนของระบบ ครอบคลุมในส่วนของการทำงานพื้นฐานในการทำเว็บเซอร์วิส ใช้ในการทำ routing[2] middleware การจัดการ request และ response ที่ถูกส่งมาจากแอปพลิเคชัน

บทที่ 3

การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และการออกแบบแอปพลิเคชันค้นหาฯเพื่อคุณ เนื่องจากคณะเภสัชศาสตร์ได้สร้างฐานข้อมูลพิสูจน์เอกสารกักษณ์ยาเม็ดและแคปซูลในประเทศไทยเรียบร้อย ทางผู้พัฒนาจึงได้ดำเนินการเขียนเว็บเซอร์วิส ใช้สำหรับติดต่อฐานข้อมูลพิสูจน์เอกสารกักษณ์ยาเม็ดและแคปซูลของคณะเภสัชศาสตร์เพื่อดึงข้อมูลพิสูจน์เอกสารกักษณ์ยาเม็ดและแคปซูลส่งออกให้แอปพลิเคชันในรูปแบบ JSON และการแผนภาพในการออกแบบระบบมีดังนี้

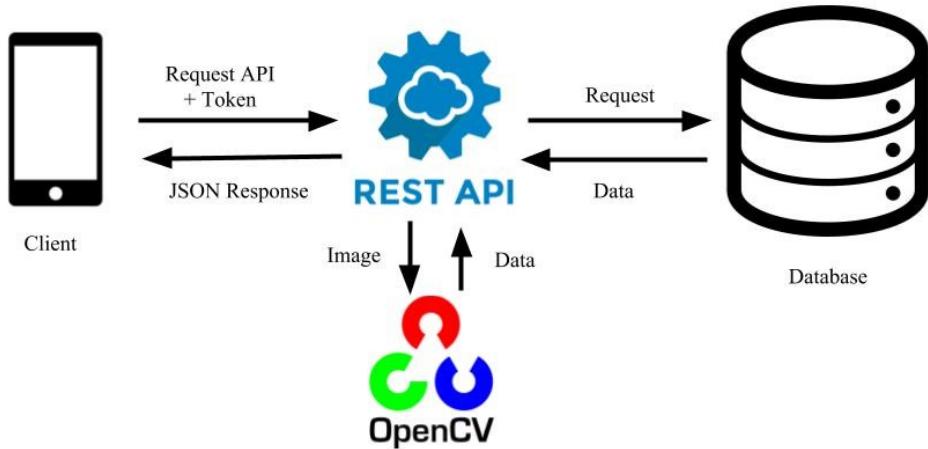
- รายละเอียดการออกแบบแอปพลิเคชัน
- Use Case Diagram เป็นแผนภาพที่ใช้แสดงให้ทราบว่าระบบทำงานหรือมีหน้าที่ใดบ้าง
- Class Diagram เป็นแผนภาพที่ใช้แสดง Class และความสัมพันธ์ระหว่าง Class
- Sequence Diagram เป็นแผนภาพที่ใช้แสดงให้เห็นถึงการตอบโต้ข้อมูลระหว่างคลาส เรียงตามลำดับของเวลาที่เกิดเหตุการณ์จากน้อยไปมาก
- State Diagram ใช้เพื่อแสดงสถานะของวัตถุ รวมไปถึงเหตุการณ์ในแอปพลิเคชัน
- การประมวลผลภาพยาเม็ดเพื่อการพิสูจน์เอกสารกักษณ์ยาเม็ด

3.1 รายละเอียดการออกแบบแอปพลิเคชัน

โครงการแอปพลิเคชันค้นหาฯเพื่อคุณ Drugiden แบ่งส่วนการทำงานออกเป็น 2 ส่วนหลัก ได้แก่ ส่วนแอปพลิเคชัน และส่วนเว็บเซอร์วิส

3.1.1 ส่วนแอปพลิเคชัน

ส่วนแอปพลิเคชันผู้ใช้งานสามารถใช้งานฟังก์ชันดังต่อไปนี้



รูปที่ 3.1: ภาพรวมของแอปพลิเคชันค้นหายาเพื่อคุณ

- สามารถค้นหายาเม็ดแบบทั่วไป เป็นการค้นหาที่ใช้ key word ใน การค้นหา เช่น para, น้ำเงิน, ส้ม, CAPSULE เป็นต้น
- สามารถค้นหายาเม็ดแบบขั้นสูง เป็นการค้นหาที่ต้องใช้การระบุเอกสารลักษณ์ของยาเม็ดหรือ แคปซูลแบบเจาะจง ลักษณะทางกายภาพหรือข้อมูลรายละเอียดต่างๆ
- สามารถดูรายละเอียดยาเม็ด เป็นการดูรายละเอียดของยาเม็ดหรือแคปซูลนั้นๆ ได้ทั้งดู รูปภาพ ข้อมูลยาต่างๆ เช่น ผู้ผลิต ผู้รับอนุญาติ ผู้จำหน่าย ขนาด กว้าง ยาว รูปแบบ ผลิตภัณฑ์ รูปร่าง ประเภทของยา เป็นต้น
- สามารถดูรายการยา เป็นการแสดงรายการยาที่เป็นผลลัพธ์จากการค้นหายาเม็ดแบบทั่ว การค้นหายาแบบขั้นสูง และรายการบัญชีการรักษา
- สามารถกดบันทึกบัญชีการรักษา เพื่อบันทึกลงบนเครื่องผู้ใช้งานและดูย้อนหลังได้
- สามารถดูรายการบัญชีการรักษา เป็นการดูรายการบัญชีการรักษาที่ผู้ใช้งานกดบันทึกบัญชีก
- ถ่ายรูปภาพยาเม็ดเพื่อพิสูจน์เอกสารลักษณ์ยา เป็นการเปิดกล้องเพื่อถ่ายรูปภาพยาเม็ดและส่ง รูปภาพไปประมวลผลที่ API service

3.1.2 ส่วนเว็บเซอร์วิส

เว็บเซอร์วิสใช้เป็นสื่อในการแลกเปลี่ยนข้อมูลกันระหว่างแอปพลิเคชันและฐานข้อมูลในรูปแบบ JSON โดยจะเรียกว่า API (Application Programming Interface) ที่จะคอยกระทำการต่างๆ เช่น การดึงข้อมูลยา การประมวลผลรูปภาพ เป็นต้น และส่วนเว็บเซอร์วิสมีหน้าที่การทำงานดังต่อไปนี้

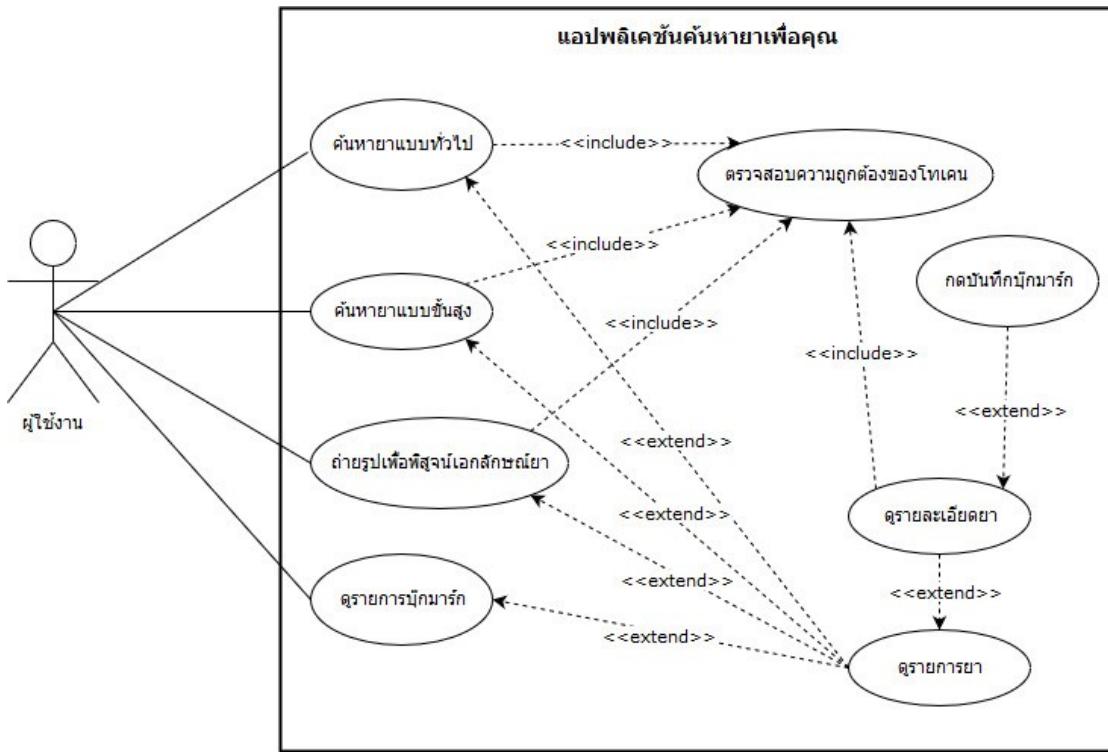
1. สามารถติดต่อกับฐานข้อมูลการพิสูจน์เอกสารลักษณ์ยาเม็ดหรือแคปซูลของหน่วยข้อมูลยาและสุขภาพ คณะเภสัชศาสตร์ มหาวิทยาลัยอุบลราชธานี
2. สามารถค้นหาเม็ดยา กับฐานข้อมูลการพิสูจน์เอกสารลักษณ์ยาเม็ดหรือแคปซูล โดยการใช้พารามิเตอร์จากการร้องขอทรัพยากร (Request) ของแอปพลิเคชัน เช่น สี รูปทรง ขนาด ชื่อการค้า ชื่อสามัญ เป็นต้น
3. สามารถพิสูจน์ตัวตนของผู้ใช้งาน เว็บเซอร์วิสมีการพิสูจน์ตัวตนผู้ใช้งานก่อนเรียกใช้งานการร้องขอด้วย JSON Web Token เพื่อป้องกันการเรียกใช้งานจากผู้ที่ไม่พึงประสงค์
4. สามารถอ่านรูปภาพแบบ base64
5. สามารถประมวลผลภาพเพื่อหาลักษณะของเม็ดยา ได้แก่ รูปทรง ขนาดและสี

3.2 Use Case Diagram

Use Case Diagram เป็นแผนผังเพื่อแสดงฟังก์ชันแสดงการทำงานของระบบโดยรวม แสดงส่วนประกอบในระบบและกิจกรรมที่เกิดขึ้นในระบบ สัญลักษณ์ที่ใช้ในการเขียน Use Case Diagram แสดงในตารางที่ 3.1

ตารางที่ 3.1: สัญลักษณ์ของ Use case Diagram

สัญลักษณ์	การใช้งาน
	Use case คือส่วนย่อยของระบบงาน แทนด้วยวงรีและชื่อของ Use case ภายในวงรี
	Actor คือบุคคลหรือระบบงานอื่นที่ใช้งานระบบหรือได้รับประโยชน์จากการซึ่งอยู่ภายนอกระบบ แทนด้วยรูปคนและมีชื่อบทบาทการใช้งานระบบ
	เส้นตรงที่แสดงถึงการใช้งาน Use case ของผู้กระทำ
	กรอบ สีเหลี่ยม แสดงถึง ขอบเขต ของ ระบบ โดย แสดง ชื่อ ระบบ ภายในหรือด้านนอกกรอบสีเหลี่ยม Use case อยู่ภายนอกกรอบสีเหลี่ยม และ actor อยู่ภายนอกกรอบสีเหลี่ยม
	ความสัมพันธ์แบบ «includes» แสดงว่า Use case หนึ่งดำเนินการตามขั้นตอนของ Use case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประ ซึ่ง Use case ที่หัวลูกศรเรียกใช้งาน Use case ที่หัวลูกศรทุกครั้งที่มีการทำงาน
	ความสัมพันธ์แบบ «extend» แสดงว่า Use case หนึ่งดำเนินการตามขั้นตอนของ Use case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประ ซึ่ง Use case ที่หัวลูกศรเรียกใช้งาน Use case ที่หัวลูกศร แต่การใช้งานไม่จำเป็นต้องเกิดขึ้นทุกครั้งที่มีอยู่กับเงื่อนไขระหว่างการทำงาน



รูปที่ 3.2: Use Case Diagram ของแอปพลิเคชันค้นหายาเพื่อคุณ

ในการพัฒนาแอปพลิเคชันค้นหายาเพื่อคุณสามารถอธิบายขั้นตอนการทำงานที่สำคัญและความสัมพันธ์ต่างๆ ของระบบด้วย Use Case Diagram แสดงดังรูปที่ 3.2 ซึ่งประกอบด้วย Use case ต่างๆ ดังนี้

1. ค้นหายาแบบทั่วไป : ทำหน้าที่การสืบค้นแบบให้ผู้ใช้งานกรอกคำสำคัญ เช่น ชื่อบริษัท ชื่อ การค้า รูปร่าง สี ขนาด เป็นต้น
2. ค้นหายาแบบขั้นสูง : ทำหน้าที่การสืบค้นแบบให้ผู้ใช้งานกรอกรายละเอียดของยาเม็ด เช่น รูปร่าง สี ขนาด เป็นต้น
3. ถ่ายรูปเพื่อพิสูจน์เอกสารลักษณ์ยา : ทำหน้าที่ให้ผู้ใช้งานถ่ายรูปภาพยาเม็ดเพื่อการพิสูจน์เอกสารลักษณ์ยาเม็ด
4. ดูรายการยา : ทำหน้าที่แสดงรายการยาจากการค้นหายาแบบทั่วไป ค้นหายาแบบขั้นสูง การถ่ายรูปเพื่อพิสูจน์เอกสารลักษณ์ยา และการดูรายการบุ๊กมาร์ก

5. ดูรายละเอียดยา : ทำหน้าที่แสดงรายละเอียดของยาเม็ดให้แก่ผู้ใช้งาน
6. กดบันทึกบุกมาრ์กยา : ทำหน้าที่บันทึกรายการยาเอาไว้ในเครื่องผู้ใช้งาน
7. ดูรายการบุกมาრ์ก : ทำหน้าที่แสดงรายการบุกมาาร์กทั้งหมดที่ผู้ใช้กดบันทึกในเครื่องผู้ใช้งาน
8. ตรวจสอบความถูกต้องของโทเคน : ทำหน้าที่ตรวจสอบความถูกต้องของโทเคนของแอปพลิเคชันกับเว็บเซอร์วิส ก่อนจะสามารถใช้งานแอปพลิเคชันได้

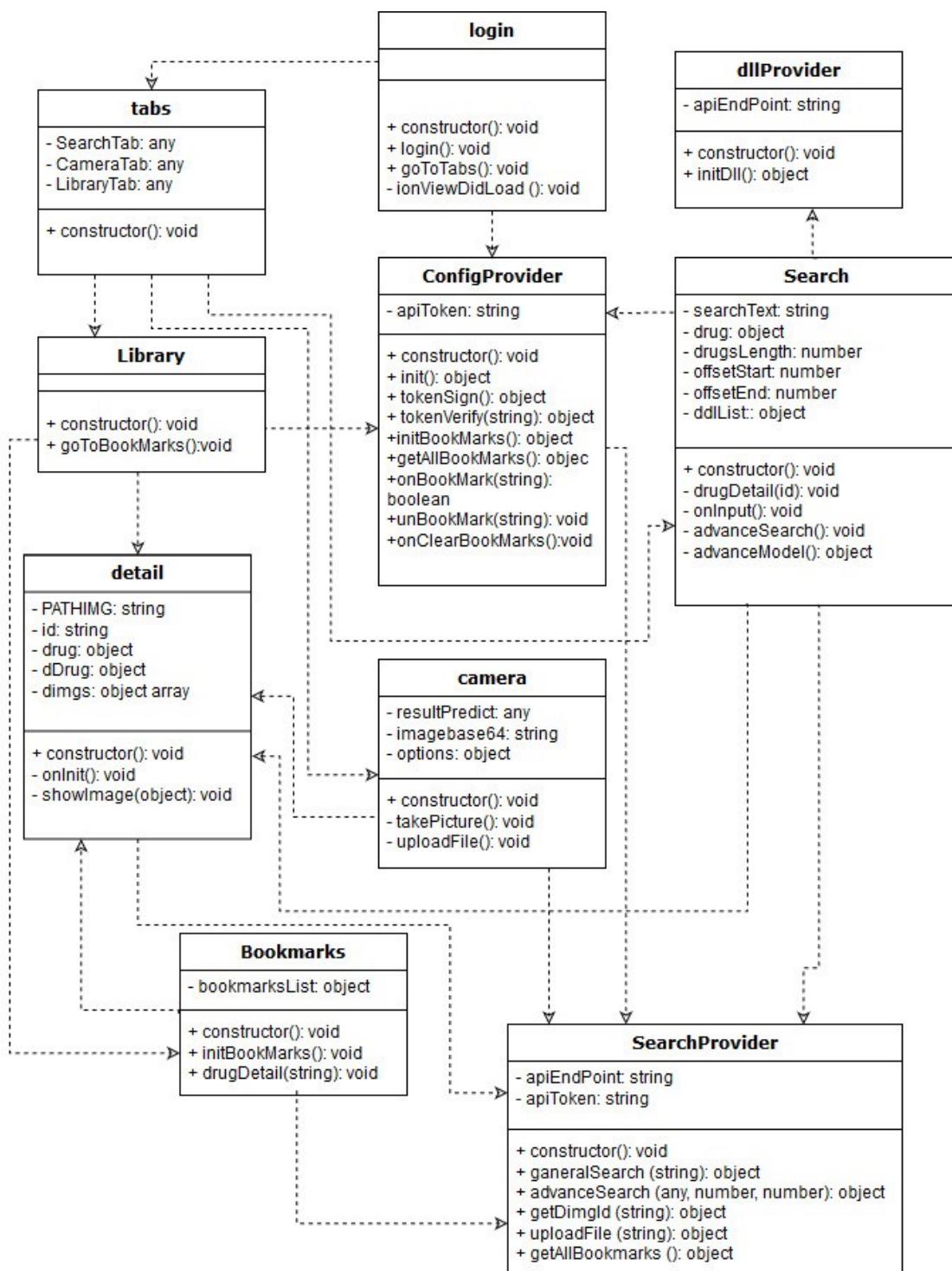
3.3 Class Diagram

Class Diagram คือแผนภาพที่ใช้แสดงคลาสและความสัมพันธ์ในแบบต่างๆ ระหว่างคลาส สัญลักษณ์ที่ใช้ในการเขียน Class Diagram แสดงในตารางที่ 3.2

ตารางที่ 3.2: สัญลักษณ์ของ Class Diagram

สัญลักษณ์	การใช้งาน			
<table border="1"> <tr> <td>Class Name</td> </tr> <tr> <td>Attribute Name</td> </tr> <tr> <td>Operation Name()</td> </tr> </table>	Class Name	Attribute Name	Operation Name()	Class สัญลักษณ์แทนด้วยสี่เหลี่ยมแบ่งเป็น 3 ส่วน ส่วนบนเป็นชื่อ Class ส่วนกลางเป็น Attribute และส่วนล่างเป็น Operation Name หรือ Method ซึ่งคลาสเป็นสิ่งที่เก็บรวบรวมข้อมูลที่แสดงถึงบุคคล สถานที่ เหตุการณ์หรือสิ่งต่างๆ ที่มีความเกี่ยวข้องกับระบบ Method เป็นการกระทำหรือฟังก์ชันที่คลาสนั้นสามารถทำได้
Class Name				
Attribute Name				
Operation Name()				
Method Name()	Method สามารถแบ่งการมองเห็น (Visibility) ได้ 3 ชนิดได้แก่ <ol style="list-style-type: none"> Public แทนสัญลักษณ์ด้วยเครื่องหมายบวก (+) Private แทนสัญลักษณ์ด้วยเครื่องหมายลบ (-) Protected แทนสัญลักษณ์ด้วยเครื่องหมายบวก (#) 			
	Dependency Relationship หมายความว่า คลาสที่อยู่ผึ้งต้นลูกสามารถเรียกใช้คลาสที่อยู่ผึ้งหัวลูกศร			
	Generalization หมายความว่า คลาสที่อยู่ผึ้งต้นลูกจะทำการสืบทอดคลาสที่อยู่หัวลูกศร			
	Association Relationship หมายความว่า คลาสที่อยู่ผึ้งต้นลูกจะทำการกำหนดคลาสอื่นในรูป Attribute ภายในคลาส และสามารถเรียกใช้ Method จากคลาสนั้นได้			

Class Diagram แสดงความสัมพันธ์ในรูปแบบต่างๆ ระหว่างคลาสของแอปพลิเคชัน คำนหายาเพื่อคุณ อธิบายได้ตามรูปที่ 3.3 ดังต่อไปนี้



รูปที่ 3.3: Class Diagram ของแอปพลิเคชันค้นหายาเพื่อคุณ

จากรูปที่ 3.3 สามารถอธิบายแผนภาพ Class Diagram ได้ดังต่อไปนี้

1. Class Login เป็นคลาสที่แสดงหน้าแรกของแอปพลิเคชัน เพื่อเข้าสู่ระบบอัตโนมัติโดยจะทำงานร่วมกับคลาส ConfigProvider สำหรับการตรวจสอบความถูกต้องของโทเคนหรือการร้องขอโทเคนจากเว็บเซอร์วิสในการใช้แอปพลิเคชัน
2. Class Tabs เป็นคลาสที่จัดการและแสดงหน้าแท็บทั้ง 2 คลาส ได้แก่ คลาส Search และคลาส Camera
3. Class ConfigProvider เป็นคลาสที่กำหนดค่าเริ่มต้นของแอปพลิเคชัน ให้แก่ การร้องขอโทเคนจากเว็บเซอร์วิส และการตรวจสอบความถูกต้องของโทเคน
4. Class Camera เป็นคลาสที่เปิดการทำงานของกล้องถ่ายรูปเพื่อให้ถ่ายรูปและประมวลผลภาพ
5. Class Search เป็นคลาสที่แสดงหน้าการค้นหาแบบทั่วไปและการค้นหาแบบละเอียด โดยจะทำงานร่วมกับคลาส SearchProvider สำหรับใช้ร้องขอไปที่เว็บเซอร์วิสเพื่อค้นหา และทำงานร่วมกับคลาส dllProvider สำหรับใช้ร้องขอไปที่เว็บเซอร์วิสเพื่อดึงรายการ dropdown มาแสดง
6. Class Library เป็นคลาสที่แสดงรายการคำสั่งในการจัดการบุ๊กマーกภายในเครื่อง
7. Class Bookmarks เป็นคลาสที่แสดงรายการบุ๊กマーกของผู้ใช้งานที่ถูกเก็บไว้ในเครื่องผู้ใช้งาน
8. Class SearchProvider เป็นคลาสที่จัดการการเชื่อมต่อกับเว็บเซอร์วิส ได้แก่ การร้องขอไปยังเว็บเซอร์วิสเพื่อการค้นหาทั่วไปกับการค้นหาแบบละเอียด การร้องขอไปยังเว็บเซอร์วิสเพื่อดึงข้อมูลฯลฯแบบละเอียด
9. Class Detail เป็นคลาสที่แสดงข้อมูลฯลฯแบบละเอียดที่ได้มาจากการคลาส SearchProvider
10. Class dllProvider เป็นคลาสที่จัดการกับรายการ dropdown ที่ร้องขอไปยังเว็บเซอร์วิส

3.4 Sequence Diagram

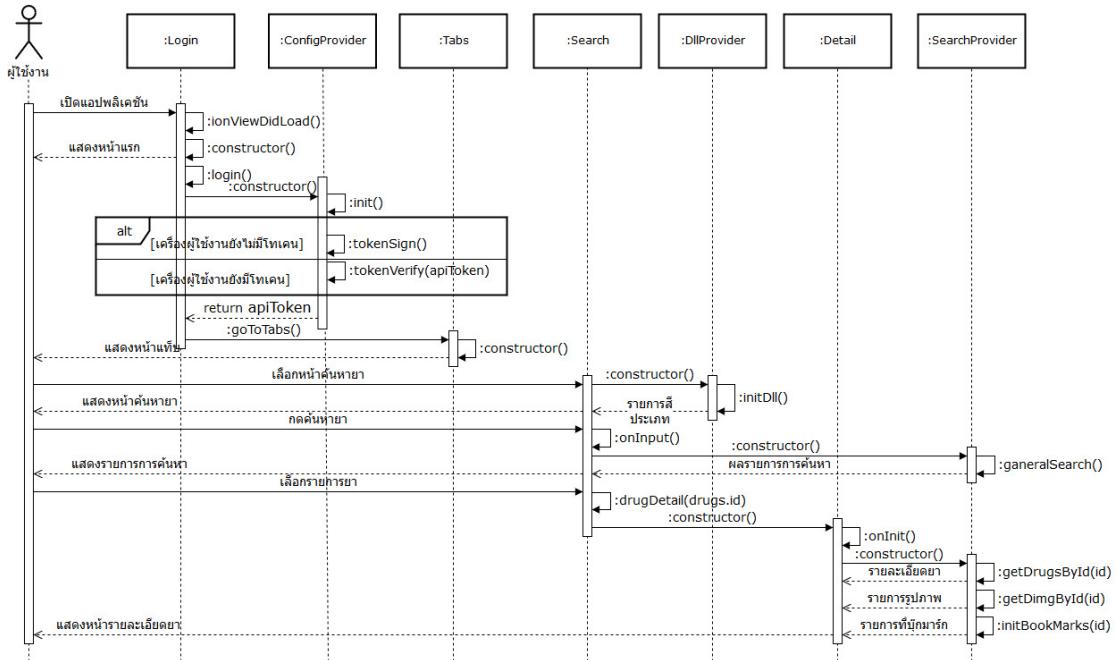
Sequence Diagram เป็น Diagram ที่แสดงขั้นตอนการทำงานของแต่ละ Use Case ระหว่าง Object ต่างๆ ที่ส่งข้อความถึงกันและกัน โดย Sequence Diagram จะช่วยให้มองเห็นการทำงานของภาพรวมของระบบ ส่วนประกอบสัญลักษณ์ที่ใช้ในการเขียน Sequence Diagram แสดงดังตารางที่ 3.3

ตารางที่ 3.3: สัญลักษณ์ของ Sequence Diagram

สัญลักษณ์	การใช้งาน
	Class แสดงถึงการทำงานของ Use Case ใน การส่งหรือรับข้อความ แทนด้วยสัญลักษณ์สี่เหลี่ยมมีชื่อคลาสอยู่ภายใน
	Lifeline หรือเส้นอายุขัย แสดงช่วงเวลาตั้งแต่เริ่มสร้าง object ในคลาสนั้น จนกระทั่ง object นั้นถูกทำลาย สัญลักษณ์แทนด้วยเส้นประ
	Focus of control หรือจุดควบคุม เป็นจุดควบคุมที่ object ใช้ทำการส่งหรือรับข้อความ สัญลักษณ์แทนด้วยสี่เหลี่ยม
	Message คือ ข้อความที่รับส่งระหว่าง Object สัญลักษณ์แทนด้วยลูกศรและประกอบด้วย 2 ส่วน คือ ข้อมูล (Data) และฟังก์ชัน (Function)
	Return Message เป็นข้อมูลที่ส่งกลับหลังจากทำงานเสร็จ

Sequence Diagram ที่ใช้อธิบายการทำงานของแอปพลิเคชันค้นหาฯเพื่อคุณซึ่งประกอบไปด้วย การค้นหาฯแบบทั่วไป การค้นหาฯแบบขั้นสูง การถ่ายรูปเพื่อพิสูจน์เอกสารลักษณ์ฯ และการดูรายละเอียดฯ โดยมีรายละเอียดดังต่อไปนี้

3.4.1 Sequence Diagram ของการค้นหายาแบบทั่วไป

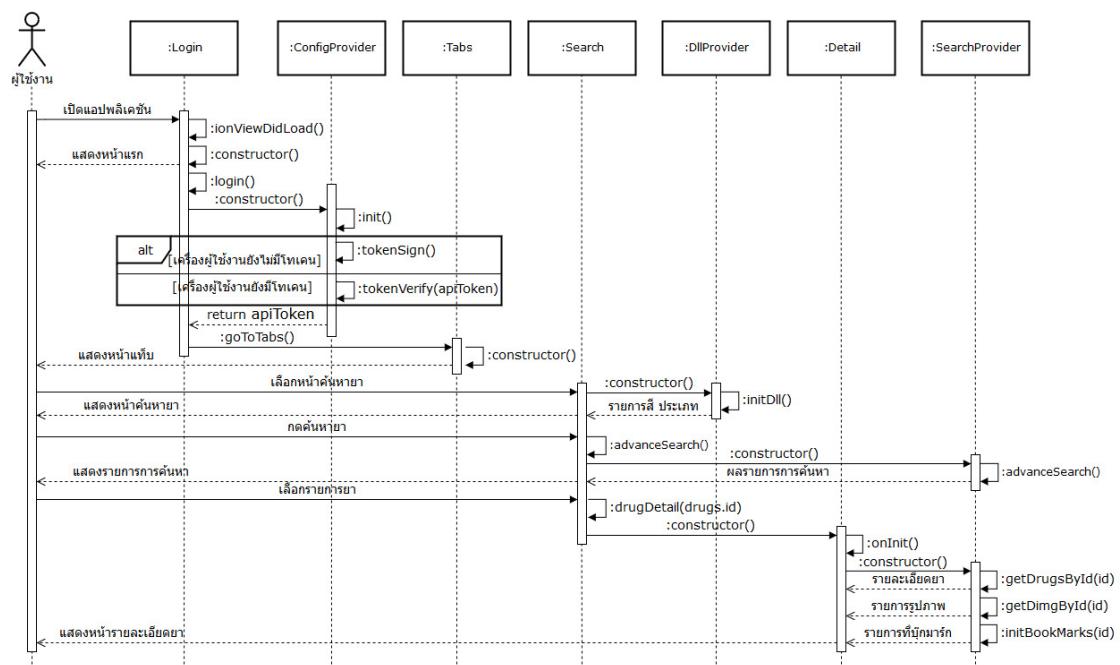


รูปที่ 3.4: Sequence Diagram ของการค้นหายาแบบทั่วไป

จากรูปที่ 3.4 Sequence Diagram ของการค้นหายาแบบทั่วไป สามารถอธิบายได้ดังนี้ เมื่อผู้ใช้งานเปิดแอปพลิเคชันค้นหายาเพื่อคุณ ระบบจะเริ่มต้นการทำที่คลาส Login ด้วยฟังก์ชัน ionViewDidLoad และ constructor เพื่อสร้างหน้าแรกและแสดงหน้าแรกให้ผู้ใช้งาน หลังจากนั้นระบบจะเรียกใช้งานฟังก์ชัน Login และในขณะเดียวกันก็จะเรียกใช้งานคลาส ConfigProvider ฟังก์ชัน Constructor เพื่อเรียกใช้งานฟังก์ชัน init สำหรับกำหนดค่าเริ่มต้นของโทเคน โดยจะมีเงื่อนไขในตรวจสอบเครื่องผู้ใช้งานมีโทเคนหรือไม่ ถ้าหากไม่มีโทเคนจะเรียกใช้งานฟังก์ชัน tokenSign และถ้าหากมีโทเคนจะเรียกใช้งานฟังก์ชัน tokenVerify หลังจากนั้นจะคืน apiToken ที่เป็นผลลัพธ์กลับมาที่คลาส Login และดำเนินการต่อไปด้วยฟังก์ชัน goToTabs สำหรับเรียกใช้งานคลาส Tabs ฟังก์ชัน constructor เพื่อสร้างหน้าแท็บและแสดงหน้าแท็บให้ผู้ใช้งาน จากนั้นเรียกใช้งานคลาส Search ฟังก์ชัน constructor และเรียกใช้งาน DllProvider ฟังก์ชัน initDll เพื่อสร้างหน้าค้นหาและแสดงหน้าค้นหาให้ผู้ใช้งาน เมื่อผู้ใช้งานกดค้นหายาที่หน้าค้นหายา จะเรียกใช้งานคลาส Search ฟังก์ชัน onInput หลังจากนั้นจะเรียกใช้งานคลาส SearchProvider ฟังก์ชัน constructor และฟังก์ชัน generalSearch

เพื่อร้องขอการค้นหาไปที่เว็บเซอร์วิสและส่งข้อมูลรายการการค้าขายกลับมายังคลาส Search จากนั้นแสดงที่หน้ารายการค้าขายให้ผู้ใช้งาน เมื่อผู้ใช้งานเลือกรายการยา จะเรียกใช้งานคลาส Search พังก์ชัน drugDetail พร้อมกับส่งหมายเลขไปกับพังก์ชัน drugDetail ไปที่คลาส Detail พังก์ชัน constructor และพังก์ชัน onInit เพื่อร้องขอข้อมูลรายการจากเว็บเซอร์วิสที่คลาส SearchProvider ผ่านทางพังก์ชัน getDrugById เพื่อร้องขอข้อมูลรายละเอียดยา และผ่านทางพังก์ชัน getDimgById เพื่อร้องขอข้อมูลรูปภาพ และพังก์ชัน initBookMarks เพื่อตรวจสอบรายการที่ถูกบุกมาไว้ และแสดงหน้ารายละเอียดยาแก่ผู้ใช้งาน

3.4.2 Sequence Diagram ของการค้นหายาแบบขั้นสูง

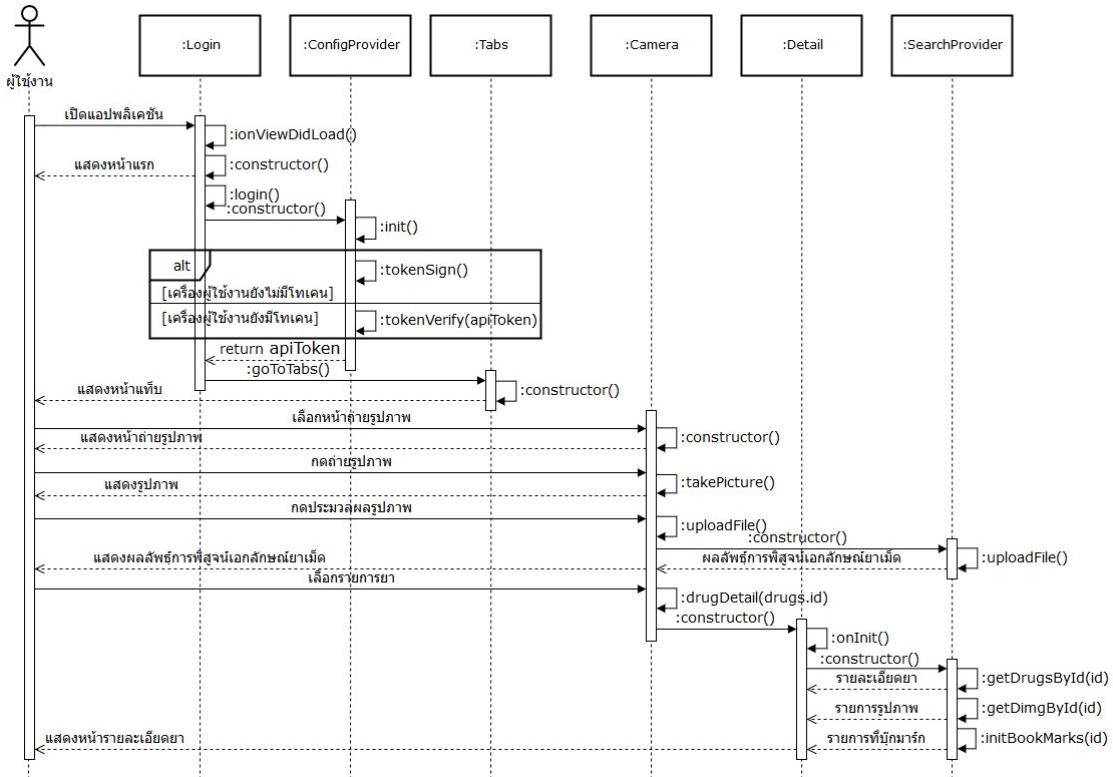


รูปที่ 3.5: Sequence Diagram ของการค้นหายาแบบขั้นสูง

จากรูปที่ 3.5 Sequence Diagram ของการค้นหายาแบบขั้นสูง สามารถอธิบายได้ดังนี้ เมื่อผู้ใช้งานเปิดแอปพลิเคชันค้นหายาเพื่อคุณ ระบบจะเริ่มต้นการทำที่คลาส Login ด้วยพังก์ชัน ionViewDidLoad และ constructor เพื่อสร้างหน้าแรกและแสดงหน้าแรกให้ผู้ใช้งาน หลังจากนั้นระบบจะเรียกใช้งานพังก์ชัน Login และในขณะเดียวกันก็จะเรียกใช้งานคลาส ConfigProvider พังก์ชัน Constructor เพื่อเรียกใช้งานพังก์ชัน init สำหรับกำหนดค่าเริ่ม

ต้นของໂທເຄນ ໂດຍຈະມີເງື່ອນໄຂໃນຕຽບສອບເຄື່ອງຜູ້ໃຊ້ຈານມີໂທເຄນຫຼືວ່າມີ ຄ້າທາກໄມ້ມີໂທເຄນຈະເຮັດໃຊ້ຈານພັກໆໜັນ tokenSign ແລະ ຄ້າທາກມີໂທເຄນຈະເຮັດໃຊ້ຈານພັກໆໜັນ tokenVerify ລັ້ງຈາກນັ້ນຈະຄືນ apiToken ທີ່ເປັນຜລລັບຮົກລັບມາທີ່ຄລາສ Login ແລະ ດຳເນີນກາຣຕ່ອໄປດ້ວຍພັກໆໜັນ goToTabs ສໍາໜັບເຮັດໃຊ້ຈານຄລາສ Tabs ພັກໆໜັນ constructor ເພື່ອສ້າງໜ້າແທັບແລະ ແສດງໜ້າແທັບໃຫ້ຜູ້ໃຊ້ຈານ ຈາກນັ້ນເຮັດໃຊ້ຈານຄລາສ Search ພັກໆໜັນ constructor ແລະ ເຮັດໃຊ້ຈານ DLLProvider ພັກໆໜັນ initDll ເພື່ອສ້າງໜ້າຄັ້ນຫາແລະ ແສດງໜ້າຄັ້ນຫາໃຫ້ຜູ້ໃຊ້ຈານ ເມື່ອຜູ້ໃຊ້ຈານດັດຄັ້ນຫາຍາ ຄລາສ Search ພັກໆໜັນ advanceSearch ໃນຂ່າຍະເດີຍກັນຈະເຮັດໃຊ້ຈານຄລາສ SearchProvider ພັກໆໜັນ constructor ແລະ ພັກໆໜັນ advanceSearch ເພື່ອຮ້ອງຂອງກາຮັ້ນຫາໄປທີ່ເວັບເຊອຣິວິສແລະ ສັງຂໍອມູນລາຍກາຮັ້ນຫາຢ່າງລັບມາຍັງຄລາສ Search ຈາກນັ້ນແສດງທີ່ໜ້າຮາຍກາຮັ້ນຫາໃຫ້ຜູ້ໃຊ້ຈານ ເມື່ອຜູ້ໃຊ້ຈານເລືອກຮາຍກາຮັ້ນຫາ ຈະເຮັດໃຊ້ຈານຄລາສ Search ພັກໆໜັນ drugDetail ພຣອມກັບສັງໝາຍເລຂຍາໄປກັບພັກໆໜັນ ໄປທີ່ຄລາສ Detail ພັກໆໜັນ constructor ແລະ ພັກໆໜັນ onInit ເພື່ອຮ້ອງຂອງຂໍອມູນລາຍຈາກເວັບເຊອຣິວິສທີ່ຄລາສ SearchProvider ຜ່ານທາງ ພັກໆໜັນ getDrugById ເພື່ອຮ້ອງຂອງຂໍອມູນລາຍລະເອີຍດຍາ ແລະ ຜ່ານທາງພັກໆໜັນ getDimgById ເພື່ອຮ້ອງຂອງຂໍອມູນລາຍປາກ ແລະ ພັກໆໜັນ initBookMarks ເພື່ອຕຽບສອບຮາຍກາຮັ້ນທີ່ຄູກບຸກມາຮັກໄວ້ ແລະ ແສດງໜ້າຮາຍລະເອີຍດຍາແກ່ຜູ້ໃຊ້ຈານ

3.4.3 Sequence Diagram ของการถ่ายรูปภาพเพื่อพิสูจน์เอกสารยามีเดียเมด

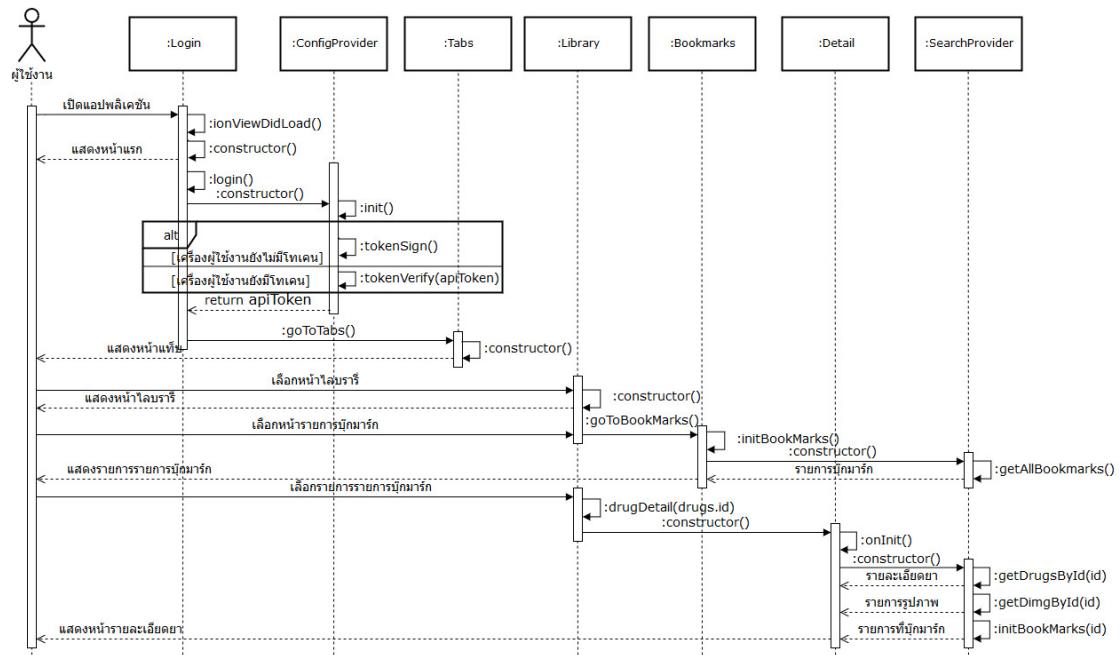


รูปที่ 3.6: Sequence Diagram ของการถ่ายรูปภาพเพื่อพิสูจน์เอกสารยามีเดียเมด

จากรูปที่ 3.6 Sequence Diagram ของการถ่ายรูปภาพเพื่อพิสูจน์เอกสารยามีเดียเมด สามารถอธิบายได้ดังนี้ เมื่อผู้ใช้งานเปิดแอปพลิเคชันค้นหารายเพื่อคุณ ระบบจะเริ่มต้นการทำที่คลาส Login ด้วยฟังก์ชัน ionViewDidLoad และ constructor เพื่อสร้างหน้าแรกและแสดงหน้าแรกให้ผู้ใช้งาน หลังจากนั้นระบบจะเรียกใช้งานฟังก์ชัน LoginPage และในขณะเดียวกัน ก็จะเรียกใช้งานคลาส ConfigProvider ฟังก์ชัน Constructor เพื่อเรียกใช้งานฟังก์ชัน init สำหรับกำหนดค่าเริ่มต้นของโทเคน โดยจะมีเงื่อนไขในตรวจสอบเครื่องใช้งานมีโทเคนหรือไม่ ถ้าหากไม่มีโทเคนจะเรียกใช้งานฟังก์ชัน tokenSign และถ้าหากมีโทเคนจะเรียกใช้งานฟังก์ชัน tokenVerify หลังจากนั้นจะคืน apiToken ที่เป็นผลลัพธ์กลับมาที่คลาส Login และดำเนินการต่อไปด้วยฟังก์ชัน goToTabs สำหรับเรียกใช้งานคลาส Tabs ฟังก์ชัน constructor เพื่อสร้างหน้าแท็บและแสดงหน้าแท็บให้ผู้ใช้งาน จากนั้นเรียกใช้งานคลาส Camera ฟังก์ชัน constructor เพื่อสร้างหน้าการถ่ายรูปเพื่อพิสูจน์เอกสารยามีเดีย เมื่อผู้ใช้งานกดถ่ายรูปภาพ

ยาจะเรียกใช้งานฟังก์ชัน takePicture และจะแสดงรูปประที่ถ่ายได้ให้ผู้ใช้งาน เมื่อผู้ใช้งาน กดประมวลผลภาพจะเรียกใช้งานฟังก์ชัน uploadFile ในขณะเดียวกันจะเรียกใช้งานคลาส SearchProvider ฟังก์ชัน constructor และฟังก์ชัน uploadFile เพื่อร้องขอไปประมวล ผลภาพที่เว็บเซอร์วิสและส่งผลลัพธ์กลับมาที่คลาส Camera เพื่อแสดงผลลัพธ์การพิสูจน์เอกสาร ลักษณ์ยาให้ผู้ใช้งาน เมื่อผู้ใช้งานเลือกรายการยา จะเรียกใช้งานคลาส Search ฟังก์ชัน drugDetail พร้อมกับส่งหมายเลขยาไปกับฟังก์ชัน ไปที่คลาส Detail ฟังก์ชัน constructor และฟังก์ชัน onInit เพื่อร้องขอข้อมูลยาจากเว็บเซอร์วิสที่คลาส SearchProvider ผ่านทางฟังก์ชัน getDrugById เพื่อร้องขอข้อมูลรายละเอียดยา และผ่านทางฟังก์ชัน getDimgById เพื่อร้องขอ ข้อมูลรูปภาพ และฟังก์ชัน initBookMarks เพื่อตรวจสอบรายการที่ถูกบุกมาร์กไว้ และแสดง หน้ารายละเอียดยาแก่ผู้ใช้งาน

3.4.4 Sequence Diagram ของการดูรายการบุกมาร์ก



รูปที่ 3.7: Sequence Diagram ของการดูรายการบุกมาร์ก

จากรูปที่ 3.7 Sequence Diagram ของการดูรายละเอียดยา สามารถอธิบายได้ดังนี้ เมื่อผู้ใช้งานเปิดแอปพลิเคชันค้นหารายเพื่อคุณ ระบบจะเริ่มต้นการทำที่คลาส Login ด้วย

ฟังก์ชัน ionViewDidLoad และ constructor เพื่อสร้างหน้าแรกและแสดงหน้าแรกให้ผู้ใช้งาน หลังจากนั้นระบบจะเรียกใช้งานฟังก์ชัน Login และในขณะเดียวกันก็จะเรียกใช้งานคลาส ConfigProvider ฟังก์ชัน Constructor เพื่อเรียกใช้งานฟังก์ชัน init สำหรับกำหนดค่าเริ่มต้นของໂທເຄນ ໂດຍຈະມີເງື່ອນໄຂໃນตรวจสอบເຄື່ອງຜູ້ໃຊ້ງານມີໂທເຄນຫຼືວ່າມີໆ ຄ້າຫາກໄມ້ມີໂທເຄນຈະເຮັດໃຫ້ຜູ້ໃຊ້ງານຟັງກິນ tokenSign ແລະ ຄ້າຫາກມີໂທເຄນຈະເຮັດໃຫ້ຜູ້ໃຊ້ງານຟັງກິນ tokenVerify ລັງຈາກນັ້ນຈະຄືນ apiToken ທີ່ເປັນຜລັບພົງກລັບມາທີ່ຄລາສ Login ແລະ ດຳເນີນການຕ່ອໄປດ້ວຍຝັ້ງ-ໝັ້ນ goToTabs ສໍາຮັບເຮັດໃຫ້ຄລາສ Tabs ຝັ້ງທີ່ຈະສໍາຮັບເຮັດໃຫ້ຜູ້ໃຊ້ງານ ເນື່ອຜູ້ໃຊ້ງານເລືອກຫຼັກໄລບຣາຣີ ຈະເຮັດໃຫ້ຄລາສ Library ຝັ້ງທີ່ຈະສໍາຮັບເຮັດໃຫ້ຄລາສ Bookmarks ພັ້ນທີ່ຈະສໍາຮັບເຮັດໃຫ້ຄລາສ SearchProvider ຝັ້ງທີ່ຈະສໍາຮັບເຮັດໃຫ້ຄລາສ Detail ຝັ້ງທີ່ຈະສໍາຮັບເຮັດໃຫ້ຄລາສ DrugDetail ພຽມກັບສ່ວນຫຼາຍເລຂຍາໄປກັບຝັ້ງທີ່ຈະສໍາຮັບເຮັດໃຫ້ຄລາສ SearchProvider ຜ່ານທາງຝັ້ງທີ່ຈະສໍາຮັບເຮັດໃຫ້ຄລາສ getDrugById ເພື່ອຮັບຂອ້າມຸນລາຍຈາກເວັບເຂອງວິວທີ່ຄລາສ SearchProvider ຜ່ານທາງຝັ້ງທີ່ຈະສໍາຮັບເຮັດໃຫ້ຄລາສ getDimgById ເພື່ອຮັບຂອ້າມຸນຮູ້ປາພ ແລະ ພັ້ນທີ່ຈະສໍາຮັບເຮັດໃຫ້ຄລາສ initBookMarks ເພື່ອຕວບສອບຮາຍການທີ່ຄູກບຸກມາຮັກໄວ້ ແລະ ແສດງໜ້າຮາຍລະເອີຍດຍາແກ່ຜູ້ໃຊ້ງານ

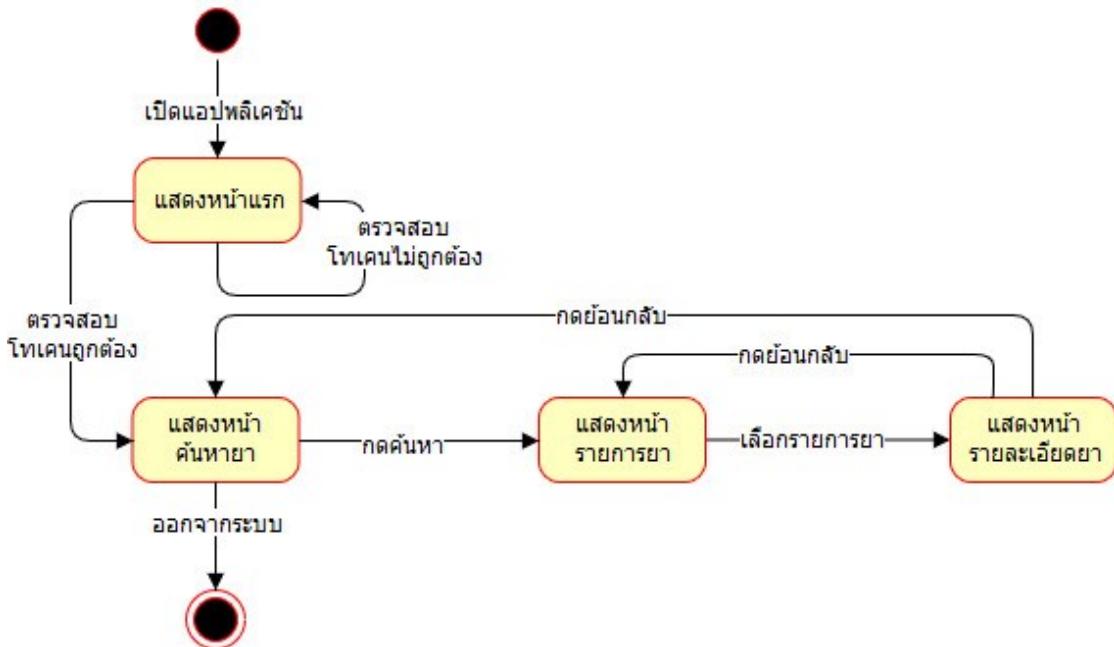
3.5 State Diagram

State Diagram เป็นภาพແນທີແສດງອົບເຈັກຕໍ່ແຕ່ລະຕົວ ໂດຍສານະຮວມຂອງຮະບບເກີດຈາກສານະຍ່ອຍຂອງອົບເຈັກຕໍ່ແຕ່ລະຕົວມັກນເປັນກລໄກທີ່ທຳໃຫ້ຮະບບມີການເປີ່ຍນສານະຄືການສ່າງ message ໃນທາງ Object orientation ຄືການເຮັດໃຫ້ຝຶກໜ້າຂອງອົບເຈັກຕໍ່ນັ້ນເອງ ຈຶ່ງສ່ວນປະກອບແລະສັນລັກຜົນທີ່ໃໝ່ໃນການເຂີນ State Diagram ແສດງໃນຕາரາງທີ່ 3.4

ຕາරາງທີ່ 3.4: ສັນລັກຜົນຂອງ State Diagram

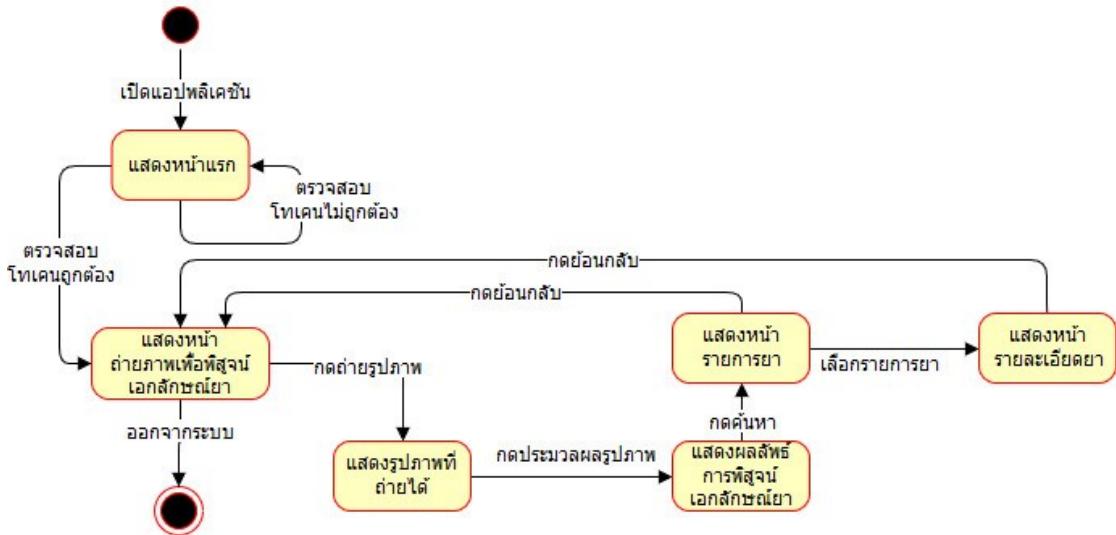
ສັນລັກຜົນ	ການໃຊ້ຈານ
	Initial state ດີວ່າສານະເຮີ່ມຕົ້ນແສດງຄື່ງອົບເຈັກຕໍ່ທີ່ເກີດຂຶ້ນ
	Initial state ດີວ່າສານະເຮີ່ມຕົ້ນແສດງຄື່ງອົບເຈັກຕໍ່ທີ່ເກີດຂຶ້ນ
	State ດີວ່າແສດງສານະອົບເຈັກຕໍ່
	Event ດີວ່າເຫດກາລົນທີ່ເກີດຂຶ້ນທຳໃຫ້ເກີດການເປີ່ຍນສານະ ໂດຍມີເງື່ອນໄຂ ຈຶ່ງອົບເຈັກຕໍ່ຈະເປີ່ຍນສານະເມື່ອເງື່ອນໄຂດັ່ງກ່າວເປັນຈິງ
	Transition ດີວ່າການເປີ່ຍນສານະແສດງຄື່ງການເປີ່ຍນສານະຂອງອົບເຈັກຕໍ່ຈາກສານະໜຶ່ງໄປຢັງສານະອື່ນ

State Diagram ໃຊ້ສໍາຮັບອົບໄຍກການທຳການຂອງແອປພລິເຄີ້ນຄັ້ນຫາຍາເພື່ອຄຸນຈຶ່ງປະກອບໄປດ້ວຍ ສ່ວນການຄັ້ນຫາຍາແລະການແສດງຮາຍລະເອີ້ດຍາ ແລະສ່ວນການຄ່າຍຽຸປເພື່ອການພຶສູຈົນໆເອກລັກຜົນຍາ ໂດຍມີຮາຍລະເອີ້ດັ່ງຕ່ອໄປນີ້



รูปที่ 3.8: State Diagram ส่วนการค้นหายาและการแสดงรายละเอียดยา

จากรูปที่ 3.8 เป็น State Diagram ของส่วนการค้นหายาและการแสดงรายละเอียดยา สามารถอธิบายได้ดังต่อไปนี้ ผู้ใช้งานเปิดแอปพลิเคชันขึ้นมาระบบจะอยู่ในสถานะแสดงหน้าแรก เมื่อระบบตรวจสอบความถูกต้องของโทเคนสำเร็จ ระบบจะอยู่ในสถานะแสดงค้นหายา เมื่อผู้ใช้งานกดค้นหายาระบบจะอยู่ในสถานะแสดงหน้ารายการ เมื่อผู้ใช้งานเลือกรายการยาเพื่อย้ายสถานะของระบบไปที่สถานะแสดงหน้ารายละเอียดยา จากนั้นผู้ใช้งานสามารถกดย้อนกลับ (รายการยา) เพื่อย้ายสถานะของระบบไปที่สถานะแสดงหน้ารายการและผู้ใช้งานสามารถกดย้อนกลับ (ค้นหายา) เพื่อย้ายสถานะของระบบไปที่สถานะแสดงหน้าค้นหา



รูปที่ 3.9: State Diagram ส่วนการถ่ายรูปเพื่อการพิสูจน์เอกสารลักษณ์ฯ

จากรูปที่ 3.9 เป็น State Diagram ของส่วนการถ่ายรูปเพื่อการพิสูจน์เอกสารลักษณ์ฯ สามารถอธิบายรายละเอียดได้ดังต่อไปนี้ ผู้ใช้งานเปิดแอปพลิเคชันค้นหาฯเพื่อคุณขึ้นมาระบบจะอยู่ในสถานะแสดงหน้าแรก เมื่อระบบตรวจสอบความถูกต้องของโทเคนสำเร็จ สถานะของระบบจะถูกย้ายไปที่สถานะแสดงหน้าถ่ายภาพเพื่อพิสูจน์เอกสารลักษณ์ฯ เมื่อผู้ใช้งานกดถ่ายรูปภาพสถานะของระบบจะย้ายไปที่สถานะแสดงรูปภาพที่ถ่ายได้ เมื่อผู้ใช้งานกดประมวลผลรูปภาพระบบจะย้ายไปที่สถานะแสดงผลลัพธ์การพิสูจน์เอกสารลักษณ์ฯ เมื่อผู้ใช้งานกดคืนหาฯสถานะของระบบจะย้ายไปที่สถานะหน้ารายการฯ ผู้ใช้งานสามารถกดย้อนกลับเพื่อย้ายสถานะระบบไปที่สถานะแสดงหน้าถ่ายภาพเพื่อพิสูจน์เอกสารลักษณ์ฯ และผู้ใช้งานสามารถเลือกรายการฯเพื่อย้ายสถานะระบบไปที่สถานะแสดงหน้าถ่ายภาพเพื่อพิสูจน์เอกสารลักษณ์ฯ และเมื่อผู้ใช้กดย้อนกลับเพื่อย้ายสถานะระบบไปที่สถานะแสดงหน้าถ่ายภาพเพื่อพิสูจน์เอกสารลักษณ์ฯ

3.6 การประมวลผลภาพยาเม็ดเพื่อการพิสูจน์เอกสารลักษณ์ยาเม็ด

ส่วนการประมวลผลภาพยาเม็ดเพื่อการพิสูจน์เอกสารลักษณ์ยาเม็ดพัฒนาด้วยภาษา Javascript มีรายละเอียดดังต่อไปนี้

- ทดลองการจำแนกรูปทรงของยาเม็ดด้วยการเรียนรู้ของเครื่อง (Machines Learning)
- การหาขนาดด้านยาวของเม็ดยาโดยใช้วัตตุอ้างอิงที่รู้ขนาด
- การหาลักษณะสีของยาเม็ด

3.6.1 ทดลองการจำแนกรูปทรงของยาเม็ดด้วยการเรียนรู้ของเครื่อง (Machines Learning)

การจำแนกรูปทรงด้วยการเรียนรู้ของเครื่องโดยใช้อัลกอริทึมทั้งหมด 3 อัลกอริทึม ได้แก่ Support Vector Machines (SVM), K-nearest neighbor (KNN) และ Random Forest Classifier (RFC) เพื่อหาความถูกต้องที่มากที่สุดของแบบจำลองที่นำมาใช้งานจริง และข้อมูลรูปทรงเม็ดยาที่ใช้ในการเตรียมการฝึกและทดสอบแบบจำลองทั้งสามแบบจำลองเป็นข้อมูลชุดเดียวกัน

1. การเตรียมข้อมูล

รูปทรงเม็ดยาทั้งหมด 8 รูปทรง มีสีเหลี่ยม สามเหลี่ยม วงกลม วงวี แคปซูล หัวท้ายมน หกเหลี่ยมและแปดเหลี่ยม แต่เนื่องจากรูปทรงวงวี แคปซูลและหัวท้ายมน มีรูปทรงที่คล้ายกันจึงได้รวมรูปทรงทั้งสามรูปทรงเข้าด้วยกัน เป็นรูปทรงแคปซูล ดังนั้นรูปทรงเม็ดยาที่ใช้ในการฝึกและทดสอบแบบจำลองจะมีทั้งหมด 6 รูปทรง คือ รูปทรงสีเหลี่ยม จำนวนรูปภาพที่ใช้สำหรับฝึกทั้งหมด 44 รูปภาพ และจำนวนรูปภาพที่ใช้สำหรับทดสอบ 20 รูปภาพ รูปทรงสามเหลี่ยม จำนวนรูปภาพที่ใช้สำหรับฝึกทั้งหมด 45 รูปภาพ และจำนวนรูปภาพที่ใช้สำหรับทดสอบ 20 รูปภาพ รูปทรงวงกลม จำนวนรูปภาพที่ใช้สำหรับฝึกทั้งหมด 29 รูปภาพ และจำนวนรูปภาพที่ใช้สำหรับทดสอบ 20 รูปภาพ รูปทรงแคปซูลจำนวนรูปภาพที่ใช้สำหรับฝึกทั้งหมด 38 รูปภาพ และจำนวนรูปภาพที่ใช้สำหรับทดสอบ 20 รูปภาพ รูปทรงหกเหลี่ยม จำนวนรูปภาพที่ใช้สำหรับฝึกทั้งหมด 59 รูปภาพ และจำนวนรูปภาพที่ใช้

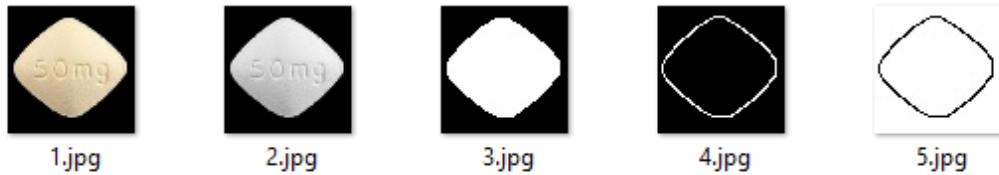
สำหรับทดสอบ 20 รูปภาพ และรูปทรงแปดเหลี่ยม จำนวนรูปภาพที่ใช้สำหรับฝึกหั้งหมวด 43 รูปภาพ และ จำนวนรูปภาพที่ใช้สำหรับทดสอบ 20 รูปภาพ มีรายละเอียดแสดงดังตารางที่ 3.5

ตารางที่ 3.5: ตารางแสดงจำนวนของข้อมูลรูปทรงยาเม็ดที่ใช้ฝึกและทดสอบ

รูปทรง	จำนวนรูปที่ใช้ฝึก	จำนวนรูปที่ใช้ทดสอบ
สี่เหลี่ยม	44	20
สามเหลี่ยม	45	20
วงกลม	29	20
แคบซูล	38	20
หกเหลี่ยม	59	20
แปดเหลี่ยม	43	20



รูปที่ 3.10: ตัวอย่างข้อมูลสำหรับการฝึกแบบจำลองรูปทรงสี่เหลี่ยม



รูปที่ 3.11: แสดงการขั้นตอนการทำความสะอาดข้อมูลรูปสำหรับใช้ฝึกและสอน

จากรูปที่ 3.10 รูปภาพตัวอย่างข้อมูลสำหรับการฝึกแบบจำลองรูปทรงสี่เหลี่ยม รูปทรงของยารูปทรงสี่เหลี่ยมมีหลายรูปแบบ ได้แก่ รูปทรงสี่เหลี่ยมขนมเปียกปูน รูปทรงสี่เหลี่ยมจัตุรัส รูปทรงสี่เหลี่ยมผืนผ้า และรูปทรงสี่เหลี่ยมขนมเปียกปูน จากรูปทรงทั้งหมด คือ รูปทรงสี่เหลี่ยม จากรูปที่ 3.11 แสดงลำดับการทำความสะอาดข้อมูลรูปภาพก่อนจะใช้ฝึกและสอนรูปแบบจำลอง SVM kNN และ RFC โดยรูปที่ 1.jpg คือรูปภาพต้นฉบับขนาด 64×64 พิกเซล รูปที่ 2.jpg คือรูปภาพสีเทาที่ถูกแปลงจาก BGR รูปที่ 3.jpg คือรูปที่ถูกปรับ Threshold รูปที่ 4.jpg คือรูปที่ผ่านการตรวจหา Canny Edge และรูปภาพสุดท้ายรูปที่ 5.jpg คือรูปที่ผ่านการลับบิท ตามลำดับ

2. การฝึกและทดสอบแบบจำลอง

การกำหนดค่าพารามิเตอร์ของแบบจำลอง SVM คือ กำหนดชนิดของ kernelType เป็น RBF กำหนดค่า c เป็น 12.5 และ กำหนดค่า gamma เป็น 0.50625 แสดงดังรูปที่ 3.12

```

1 const svm = new cv.SVM({
2     kernelType: cv.ml.SVM.RBF,
3     c: 12.5,
4     gamma: 0.50625
5 });

```

รูปที่ 3.12: การกำหนดค่าพารามิเตอร์ของแบบจำลอง SVM

จากตารางที่ 3.6 จะเห็นได้ว่ารูปทรงที่ใช้ฝึกและทดสอบมี 6 รูปทรง รูปทรงสี่เหลี่ยมจำแนกถูกต้อง 20 รูปภาพ คิดเป็นร้อยละ 100 รูปทรงสามเหลี่ยมจำแนกถูกต้อง 19

รูปภาพ คิดเป็นร้อยละ 95 รูปทรงกลมจำแนกถูกต้อง 19 รูปภาพ คิดเป็นร้อยละ 95 รูปทรงแคปซูลจำแนกถูกต้อง 20 รูปภาพ คิดเป็นร้อยละ 100 รูปทรงหกเหลี่ยมจำแนกถูกต้อง 18 รูปภาพ คิดเป็นร้อยละ 90 รูปทรงแปดเหลี่ยมจำแนกถูกต้อง 17 รูปภาพ คิดเป็นร้อยละ 85

ดังนั้นการฝึกและทดสอบแบบจำลอง SVM มีความถูกต้องโดยเฉลี่ยร้อยละ 94.17

ตารางที่ 3.6: แสดงความถูกต้องของแบบจำลอง SVM

รูปทรง	จำแนกถูกต้อง	คิดเป็นร้อยละ
สี่เหลี่ยม	20	100%
สามเหลี่ยม	19	95%
วงกลม	19	95%
แคปซูล	20	100%
หกเหลี่ยม	18	90%
แปดเหลี่ยม	17	85%

การกำหนดค่าพารามิเตอร์ของแบบจำลอง KNN ดังนี้ train คือ ชุดข้อมูลสำหรับการใช้ฝึกสอนแบบจำลอง cv2.ml.ROW_SAMPLE คือ การกำหนดรูปแบบของชุดข้อมูลเป็นแบบแคลา train_labels คือ คลาสของชุดข้อมูลสำหรับการใช้ฝึกสอนแบบจำลอง และ k คือ จำนวนของคลาสที่ใกล้เคียงสูงสุด แสดงดังรูปที่ 3.13

```

1 knn = cv2.ml.KNearest_create()
2 knn.train(train, cv2.ml.ROW_SAMPLE, train_labels)
3 result = knn.findNearest(test, k=6)

```

รูปที่ 3.13: การกำหนดค่าพารามิเตอร์ของแบบจำลอง KNN

จากตารางที่ 3.7 การฝึกและทดสอบแบบจำลอง KNN โดยจะเริ่มจากค่า $k = 1$ ถึง $k = 6$ จะเห็นได้ว่า K ที่ 1 มีค่าความถูกต้องของแบบจำลองร้อยละ 79.17 K ที่ 2 มีค่าความถูกต้องของแบบจำลองร้อยละ 76.67 K ที่ 3 มีค่าความถูกต้องของแบบจำลองร้อย

ละ 74.17 K ที่ 4 มีค่าความถูกต้องของแบบจำลองร้อยละ 70.83 K ที่ 5 มีค่าความถูกต้องของแบบจำลองร้อยละ 72.50 และ K ที่ 6 มีค่าความถูกต้องของแบบจำลองร้อยละ 71.67

ดังนั้นการฝึกและทดสอบแบบจำลอง kNN มีความถูกต้องโดยเฉลี่ยร้อยละ 74.17 และ K ที่ 1 มีค่าความถูกต้องสูงที่สุดของแบบจำลองร้อยละ 79.17

ตารางที่ 3.7: แสดงความถูกต้องของแบบจำลอง kNN

ค่า k	คิดเป็นร้อยละ
K = 1	79.17%
K = 2	76.67%
K = 3	74.17%
K = 4	70.83%
K = 5	72.50%
K = 6	71.67%

การกำหนดค่าพารามิเตอร์ของแบบจำลอง KNN โดยการกำหนดพารามิเตอร์ n_estimators คือ จำนวนของต้นไม้ตัดสินใจในการสร้างแบบจำลอง train คือ ชุดข้อมูลสำหรับการใช้ฝึกสอนแบบจำลอง และ train_labels คือ คลาสของชุดข้อมูลสำหรับการใช้ฝึกสอนแบบจำลอง แสดงดังรูปที่ 3.14

```
1 RFC = RandomForestClassifier(n_estimators=100)
2 RFC.fit(train, train_labels)
```

รูปที่ 3.14: การกำหนดค่าพารามิเตอร์ของแบบจำลอง RFC

จากตารางที่ 3.8 การฝึกและทดสอบแบบจำลอง RFC โดยมีการปรับค่า N_estimators ตั้งแต่ 10 ถึง 100 จะเห็นได้ว่า N_estimators เท่ากับ 10 มีค่าความถูกต้องของแบบจำลองร้อยละ 53.33 N_estimators เท่ากับ 30 มีค่าความถูกต้องของแบบจำลองร้อยละ 67.50 N_estimators เท่ากับ 50 มีค่าความถูกต้องของแบบจำลองร้อยละ

66.67 N_estimators เท่ากับ 70 มีค่าความถูกต้องของแบบจำลองร้อยละ 65.00
 N_estimators เท่ากับ 90 มีค่าความถูกต้องของแบบจำลองร้อยละ 69.17 และ N_estimators
 เท่ากับ 100 มีค่าความถูกต้องของแบบจำลองร้อยละ 66.67
 ดังนั้นการฝึกและทดสอบแบบจำลอง RFC มีความถูกต้องโดยเฉลี่ยร้อยละ 64.72 และ
 N_estimators เท่ากับ 90 มีค่าความถูกต้องสูงที่สุดของแบบจำลองร้อยละ 69.17

ตารางที่ 3.8: แสดงความถูกต้องของแบบจำลอง RFC

ค่า N_estimators	คิดเป็นร้อยละ
10	53.33%
30	67.50%
50	66.67%
70	65.00%
90	69.17%
100	66.67%

3. สรุปผลการทำนาย

จากตารางที่ 3.6 3.7 และ 3.8 จะเห็นได้ว่าแบบจำลอง SVM มีค่าความถูกต้องร้อยละ 94.17 ซึ่งมีค่าความถูกต้องมากกว่าแบบจำลอง kNN มีค่าความถูกต้องร้อยละ 79.17 และแบบจำลอง RFC มีค่าความถูกต้องร้อยละ 69.17 ตามลำดับ

ดังนั้นผู้พัฒนาจึงเลือกใช้แบบจำลอง SVM ที่มีความถูกต้องในการจำแนกรูปทรงเม็ดยาสูบ
 ที่สุด และนำมาใช้เป็นแบบจำลองในโครงงานแอปพลิเคชันค้นหายาเพื่อคุณ

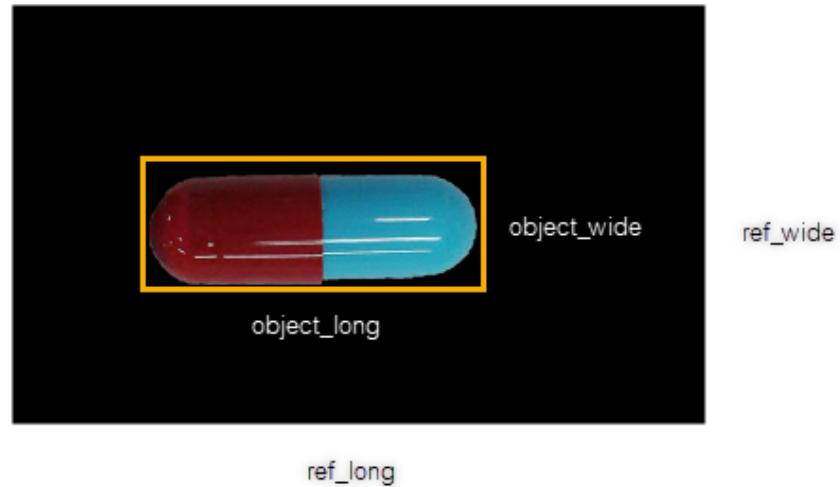
ตารางที่ 3.9: สรุปผลการทดลอง

แบบจำลอง	ค่าความถูกต้อง
Support vector machines	94.17%
k-Nearest Neighbors	79.17%
Random Forest Classifier	69.17 %

3.6.2 การหาขนาดด้านยาวของเม็ดยาโดยใช้วัสดุอ้างอิงที่รู้ขนาด

การถ่ายรูปภาพเพื่อหาขนาดด้านยาวของเม็ดยาจำเป็นต้องใช้วัสดุอ้างอิงสีดำที่มีขนาดเท่ากับ 11.3×7.3 เซนติเมตร หรือเท่ากับการใช้กระดาษ A4 พับครึ่ง 3 ครั้ง จากรูปที่ 3.152 พื้นที่สีดำคือวัตถุอ้างอิงที่ทราบขนาด และกรอบสีเหลืองคือ Contour ที่ตรวจพบ หน่วยในภาพ เป็นหน่วยพิกเซล ดังนั้นสามารถใช้การเทียบบัญญติตรายางศ์เพื่อหาขนาดด้านยาวได้ “1 พิกเซล เท่ากับกี่เซนติเมตร”

ถ้าสมมติให้ขนาดของวัสดุอ้างอิงเท่ากับ 146.9×94.9 พิกเซล และกรอบสีเหลืองมีขนาด เท่ากับ 40×23 พิกเซล และวัสดุอ้างอิงขนาดเท่ากับ 11.3×7.3 เซนติเมตร เมื่อ 146.9 พิกเซล เท่ากับ 11.3 เซนติเมตร และ 1 พิกเซล เท่ากับ $(11.3 / 146.9) = 0.0769$ เซนติเมตร ดังนั้น 40 พิกเซล เท่ากับ $(40 \times 0.0769) = 3.0769$ เซนติเมตร



รูปที่ 3.15: แสดงการวัดขนาดวัตถุด้วยการใช้วัตถุอ้างอิง

3.6.3 การหาลักษณะสีของยาเม็ด

จากรูปที่ 3.15 สามารถตรวจพบ Contour ครอบสีเหลี่ยมสีเหลือง และการหาลักษณะสีของยาเม็ด จะได้จากการแบ่งพื้นของกรอบออกเป็น 4 ส่วนเท่ากัน และดึงค่าสี BGR ที่จุดพิกเซลตรงกลางของแต่ละส่วนนำไปเปรียบเทียบกับตารางที่ 3.6.5 สีที่มีอยู่ในฐานข้อมูลพิสูจน์เอกสารลักษณ์ยาเพื่อหาค่าต่าง

ตารางที่ 3.10: ตารางแสดงรหัสสีจากฐานข้อมูลพิสูจน์เอกสารกับรหัสสีตามนี้

ลำดับ	ชื่อสี	รหัสสี
1	ขาว	#FFFFFF
2	แดง	#ea0000
3	ส้ม	#fe700e
4	เหลือง	#ffe400
5	ครีม	#efeddb
6	เขียว	#a8e26a
7	ฟ้า	#4a9fe0
8	น้ำเงิน	#120992
9	ชมพู	#fdb1ef
10	ม่วง	#c977db
11	น้ำตาล	#c2924a
12	เทา	#cccccc
13	ดำ	#000000

บทที่ 4

การสร้างระบบ

การสร้างระบบงานเป็นส่วนหนึ่งของการพัฒนาแอปพลิเคชันค้นหาเพื่อคุณและเว็บ-เซอร์วิส ซึ่งจะอธิบายการทำงานของแต่ละส่วน ดังนี้

- การค้นหาแบบทั่วไป
- การค้นหาแบบขั้นสูง
- การถ่ายรูปภาพ
- การเรียกใช้งานเว็บเซอร์วิส
- การติดต่อฐานข้อมูลของเว็บเซอร์วิส
- การสร้างและการตรวจสอบ JSON Web Token
- การถอดรหัสรูปภาพ BASE64
- การ routing ของเว็บเซอร์วิส
- การประมวลผลภาพ

4.1 การค้นหาแบบทั่วไป

เมื่อผู้ใช้งานกรอกข้อความในแบบค้นหาทั่วไป ระบบจะร้องขอการค้นหากับเว็บเซอร์วิสและส่งรายการการค้นหากลับมาแสดงที่หน้ารายการค้นหา แสดงดังภาพที่ 4.1

```

1  onInput() {
2      this.offsetStart = 1;
3      this.offsetEnd = 20;
4      this.content.resize();
5      if (this.searchText) {
6          this.searchProvider.ganeralSearch(this.
7              searchText.trim(), this.offsetStart,
8              this.offsetEnd).then(data => {
9                  this.drugs = data['results'];
10                 this.drugsLength = data['length'];
11                 if (data['length'] == 0) {
12                     this.presentToast();
13                 }
14             }).catch((error) => {
15                 let alert = this.alertCtrl.create
16                     ({
17                         title: Error,
18                         buttons: ['close']
19                     });
20                 alert.present();
21             }
22         }
23     }
24 }
```

รูปที่ 4.1: การค้นหาแบบทั่วไป

จากภาพที่ 4.1 อธิบายการทำงานของการค้นหาแบบทั่วไป ได้ดังนี้

- บรรทัดที่ 2-3 กำหนดค่าของตัวแปร สำหรับกำหนดช่วงของการค้นหา
- บรรทัดที่ 4 เป็นคำสั่งเพื่อปรับขนาดของหน้า content
- บรรทัดที่ 5 เป็นเงื่อนไข ถ้าหากตัวแปรว่างหรือไม่เก็บข้อความใดๆ จะไม่มีการเรียกใช้งานเซอร์วิส
- บรรทัดที่ 6 เป็นการใช้งานคลาส SearchProvider พังก์ชัน ganeralSearch เพื่อร้องขอการค้นหากับเว็บเซอร์วิส
- บรรทัดที่ 7 เก็บรายการของการค้นหาเป็นอาร์เรย์ในตัวแปร

- บรรทัดที่ 8 เก็บความยาวของรายการการค้นหาไว้ในตัวแปร
- บรรทัดที่ 9-10 เชื่อมต่อ API เพื่อตรวจสอบความยาวของรายการยา ถ้าหากเท่ากับ 0 แสดงว่าไม่พบเจอรายการ และจะแสดงข้อความขึ้นมาบนหน้าจอ
- บรรทัดที่ 13-17 ถ้าหากการร้องขอรายการค้นหาเกิดข้อผิดพลาด จะแสดงข้อความขึ้นมาบนหน้าจอ

4.2 การค้นหาแบบขั้นสูง

เมื่อผู้ใช้งานกรอกรายละเอียดยาจากหน้าค้นหา ระบบจะร้องขอการค้นหากับเว็บเซอร์วิส และส่งรายการการค้นหากลับมาแสดงที่หน้ารายการค้นหา แสดงตัวอย่างที่ 4.2

```

1 advanceSearch() {
2     let params = this.advanceModel();
3     let loading = this.loadingCtrl.create({
4         content: 'Searhing...'
5     });
6     loading.present();
7     this.content.resize();
8     this.offsetStart = 1;
9     this.offsetEnd = 20;
10    this.searchProvider.advanceSearch(params, this
11        .offsetStart, this.offsetEnd).then(data => {
12        this.drugs = data['results'];
13        this.drugsLength = data['length'];
14        loading.dismiss();
15        if (data['length'] == 0) {
16            this.presentToast();
17        }
18    }).catch((error) => {
19        loading.dismiss();
20        let alert = this.alertCtrl.create({
21            title: Error
22        });
23        alert.present();
24    });
}

```

รูปที่ 4.2: การค้นหาแบบขั้นสูง

จากภาพที่ 4.2 อธิบายการทำงานของการค้นหาแบบขั้นสูง ได้ดังนี้

- บรรทัดที่ 2 กำหนดตัวแปรสำหรับเก็บข้อมูลโมเดลของ
- บรรทัดที่ 3-6 แสดงการ loading
- บรรทัดที่ 7 เป็นคำสั่งเพื่อปรับขนาดของหน้า content
- บรรทัดที่ 8-9 กำหนดค่าของตัวแปร สำหรับกำหนดช่วงของการค้นหาที่ต้องการ
- บรรทัดที่ 10 เป็นการใช้งานคลาส SearchProvider ฟังก์ชัน advanceSearch เพื่อร้องขอการค้นหากับเครื่องเซอร์วิส
- บรรทัดที่ 11 เก็บรายการของการค้นหายาเป็นอาเรย์ในตัวแปร
- บรรทัดที่ 12 เก็บความยาวของการค้นหายาในตัวแปร
- บรรทัดที่ 13 ปิดการแสดงการ loading
- บรรทัดที่ 14-15 เชื่อมต่อการตรวจสอบความยาวของรายการยา ถ้าหากเท่ากับ 0 แสดงว่าไม่พบเจอรายการ และจะแสดงข้อความขึ้นมาบนหน้าจอ
- บรรทัดที่ 17-21 ถ้าหากการร้องขอรายการค้นหาเกิดข้อผิดพลาด จะแสดงข้อความขึ้นมาบนหน้าจอ

4.3 การถ่ายรูปภาพ

ในหน้าการถ่ายรูปภาพเพื่อพิสูจน์เอกสารก็จะมี เมื่อผู้ใช้งานกดเปิดกล้อง ระบบจะทำการเรียกใช้งานไลบรารี Camera ของ Cordova เพื่อทำเรียกใช้งานกล้องของอุปกรณ์ แสดงดังภาพที่ 4.3

```

1 options: CameraOptions = {
2     quality: 70,
3     destinationType: this.camera.DestinationType.
4         DATA_URL,
5     encodingType: this.camera.EncodingType.JPEG,
6     mediaType: this.camera.MediaType.PICTURE
7 }
8 takePicture() {
9     this.camera.getPicture(this.options).then( (
10        imageData) => {
11        this.imagebase64 = 'data:image/jpeg;base64,' +
12        imageData;
13    }, (err) => {
14        console.log(err);
15    });
16 }

```

รูปที่ 4.3: การถ่ายรูปภาพ

จากภาพที่ 4.3 อธิบายการทำงานของการถ่ายรูปภาพ ได้ดังนี้

- บรรทัดที่ 1-6 กำหนด options ของการถ่ายรูป คุณภาพของรูป ชนิดของรูปภาพ
- บรรทัดที่ 7-13 เป็นฟังก์ชันการถ่ายรูปภาพ เมื่อการถ่ายรูปสำเร็จ รูปภาพจะถูกเก็บในรูปแบบ base64 ในตัวแปรชื่อ imagebase64
- บรรทัดที่ 8 เปิดการทำงานของกล้องถ่ายรูปบนอุปกรณ์
- บรรทัดที่ 9 กำหนดตัวแปร imagebase64 สำหรับเก็บรูปภาพในรูปแบบ base64 ที่ได้จากการถ่ายรูป

4.4 การเรียกใช้งานเว็บเซอร์วิส

การเรียกใช้งานเว็บเซอร์วิสด้วยการใช้ไลบรารี HTTP ของ Cordova แสดงดังภาพที่ 4.4

```

1 setOption() {
2     let headers = new Headers;
3     headers.set('api-key', this.apiToken);
4     headers.append('Content-type', 'application/json');
5     this.opt = new RequestOptions({
6         headers: headers
7     });
8     generalSearch(input: string, offsetStart: any,
9         offsetEnd: any) {
10        if (offsetStart == -1 || offsetEnd == -1) {
11            offsetStart = 1, offsetEnd = 10;
12        }
13        return new Promise((resolve, reject) => {
14            this.http.get(`.${this.apiEndPoint}/drugs
15                /search?input=${input}&offsetStart=${offsetStart}
16                &offsetEnd=${offsetEnd}`,
17                this.opt).map(res => res.json()).subscribe(data => {
18                    resolve(data);
19                }, error => {
20                    reject(error);
21                });
22        });
23    }

```

รูปที่ 4.4: การเรียกใช้งานเว็บเซอร์วิส

จากภาพที่ 4.4 อธิบายการทำงานของการใช้งานเว็บเซอร์วิส ได้ดังนี้

- บรรทัดที่ 1-7 เป็นฟังก์ชันการกำหนดค่าเริ่มต้นของ Header โดยจะกำหนดรูปแบบและส่ง api-token
- บรรทัดที่ 8-19 เป็นฟังก์ชันสำหรับติดต่อกับเว็บเซอร์วิส โดยยกตัวอย่างการค้นหาแบบทั่วไป
- บรรทัดที่ 12-17 เป็นการเรียกใช้งาน HTTP แบบ GET จะร้องขอไปที่อยู่ (URL) ของเว็บเซอร์วิส

4.5 การติดต่อฐานข้อมูลของเว็บเซอร์วิส

การติดต่อกับฐานข้อมูลการพิสูจน์เอกสารณัมยาเม็ดหรือแคปชูลของคณะเภสัชศาสตร์ มหาวิทยาลัยอุบลราชธานี ใช้ไลบรารีของ nodejs ชื่อ mysql และดังภาพที่ 4.5

```

1 const mysql = require('mysql');
2 const configDB = require('./config');
3 var pool = mysql.createPool(configDB.info);
4 pool.getConnection(function(err, connection) {
5     if (err) throw err;
6     console.log('Connected', new Date());
7 });

```

รูปที่ 4.5: การติดต่อฐานข้อมูลของเว็บเซอร์วิส

จากภาพที่ 4.5 อธิบายการทำงานของการติดต่อฐานข้อมูลของเว็บเซอร์วิส ได้ดังนี้

- บรรทัดที่ 1 กำหนดตัวแปรชื่อ mysql สำหรับเก็บไลบรารีชื่อ mysql
- บรรทัดที่ 2 กำหนดตัวแปรชื่อ configDB สำหรับเก็บข้อมูลการ key-value เพื่อใช้ติดต่อกับฐานข้อมูล
- บรรทัดที่ 3 กำหนดตัวแปรชื่อ pool สำหรับเก็บข้อมูลการสร้างการเชื่อมต่อกับฐานข้อมูล
- บรรทัดที่ 4-7 เป็นการเรียกใช้งานฟังก์ชันชื่อว่า getConnection เพื่อติดต่อกับฐานข้อมูล

4.6 การสร้างและการตรวจสอบ JSON Web Token

นำ JSON Web Token มาช่วยในการระบุตัวตนก่อนให้ใช้งานกับเว็บเซอร์วิส โดยใช้ไลบรารีของ nodejs ชื่อว่า jsonwebtoken และดังภาพที่ 4.6

```

1 var jwt = require('jsonwebtoken');
2 var fs = require('fs');
3 const uuid = require('uuid/v4');
4 module.exports = {
5     sign: function(req, res, next) {
6         var data = {
7             id: req.query.id,
8             uuid: uuid()
9         };
10        var cert = fs.readFileSync('./auth/keys/
11            private.key');
12        jwt.sign(data, cert, {
13            algorithm: 'RS256',
14            expiresIn: "120d"
15        }, function(err, token) {
16            if (err) {
17                console.log('Token error ',
18                    err.message);
19                return res.status(405).send
20                    ({
21                        err: err
22                    });
23            }
24            req.query.token = token;
25            return res.status(200).send({
26                token: req.query.token
27            });
28        });
29    },
30    verify: function(req, res, next) {
31        var cert = fs.readFileSync('./auth/keys/
32            public.pem');
33        jwt.verify(req.headers['api-key'], cert,
34        {
35            algorithms: ['RS256']
36        }, function(err, payload) {
37            if (err) {
38                return res.status(401).send
39                    ({
40                        err: err
41                    });
42            }
43            req.query.payload = payload;
44            next();
45        });
46    }
47};

```

รูปที่ 4.6: การสร้างและการตรวจสอบ JSON Web Token

จากภาพที่ 4.6 อธิบายการทำงานของการสร้างและการตรวจสอบ JSON Web Token ได้ดังนี้

- บรรทัดที่ 1 กำหนดตัวแปรชื่อ jwt สำหรับเรียกใช้งานไลบรารีชื่อว่า jsonwebtoken
- บรรทัดที่ 2 กำหนดตัวแปรชื่อ js สำหรับเรียกใช้งานไลบรารีชื่อว่า js เพื่อการเขียนไฟล์
- บรรทัดที่ 5-26 เป็นฟังก์ชันการสร้าง jwt โดยจะใช้ fs อ่านไฟล์ชื่อว่า private.pem เพื่อใช้สำหรับการเข้ารหัส กำหนดรูปแบบและอายุการใช้งานของ jwt
- บรรทัดที่ 27-40 เป็นฟังก์ชันการตรวจสอบความถูกต้องของ jwt โดยจะใช้ fs อ่านไฟล์ชื่อว่า public.pem เพื่อใช้สำหรับการถอดรหัสและตรวจสอบ jwt

4.7 การถอดรหัสรูปภาพ BASE64

ในการส่งรูปภาพจากเครื่องผู้ใช้งานมายังเว็บเซอร์วิส จะถูกส่งรูปภาพมาในรูปแบบ base64 แสดงดังรูปภาพที่ 4.7

```

1 var fs = require('fs');
2 const uuid = require('uuid/v4');
3 module.exports = {
4     decode64: function(req, res, next) {
5         var body = req.body;
6         var base64Data = body.imageBase64.
7             replace(/^data:image\/jpeg;base64,/,
8                 "");
9         let filename = `${uuid()}.jpeg`;
10        let filePath = './assets/images/${filename
11            }`;
12        fs.writeFile(filePath, base64Data, 'base64', function(err) {
13            if (err) {
14                console.log('Error ', err.
15                    message);
16                return res.status(500).send(
17                    err.message);
18            }
19            req.query.filename = filename;
20            req.query.filePath = filePath;
21            console.log(req.query.filePath);
22            next();
23        });
24    }
25};

```

รูปที่ 4.7: การถอดรหัสรูปภาพ BASE64

จากภาพที่ 4.7 อธิบายการทำงานของการถอดรหัสรูปภาพ BASE64 ได้ดังนี้

- บรรทัดที่ 1 กำหนดตัวแปรชื่อ js สำหรับเรียกใช้งานไลบรารีชื่อว่า js เพื่อการเขียนไฟล์
- บรรทัดที่ 2 กำหนดตัวแปรชื่อ uuid สำหรับเรียกใช้งานไลบรารีชื่อว่า uuid/v4 เพื่อการสร้างชื่อไฟล์
- บรรทัดที่ 4-19 เป็นฟังก์ชันการถอดรหัสรูปภาพ BASE64
- บรรทัดที่ 5 กำหนดตัวแปรชื่อ body สำหรับเก็บค่าจาก req.body

- บรรทัดที่ 6 กำหนดตัวแปรชื่อ base64Data สำหรับเก็บข้อมูลรูปภาพจาก body
- บรรทัดที่ 7 กำหนดตัวแปรชื่อ filename สำหรับเก็บชื่อไฟล์ภาพ จากการเรียกใช้งานฟังก์ชัน uuid ของไลบรารี uuid/v4
- บรรทัดที่ 9 เรียกใช้งานฟังก์ชัน writeFile จาก fs สำหรับการเขียนไฟล์
- บรรทัดที่ 14-15 เก็บค่า filename กับ filePate ไว้ใน req.query
- บรรทัดที่ 17 เรียกฟังก์ชัน next() เพื่อทำงานฟังก์ชันต่อไป

4.8 การ routing ของเว็บเซอร์วิส

การ routing ของเว็บเซอร์วิสใช้เบราว์เซอร์ของ nodejs ชื่อ express สำหรับเป็นเว็บเซอร์วิส ติดต่อกับฐานข้อมูลและประมวลผลภาพ แสดงดังภาพที่ 4.8

```

1 var app = require('express')();
2 var bodyParser = require('body-parser');
3 var cors = require('cors');
4 var POST = process.env.PORT || 5000;
5 var con = require('./db/connectDB')
6 var jwt = require('./auth/auth');
7 var decode64 = require('./base64/base64');
8 var opencv = require('./openCV/svmTest');
9 app.use(bodyParser.json({
10     limit: '5mb'
11 }));
12 app.use(bodyParser.urlencoded({
13     limit: '5mb',
14     extended: true
15 }));
16 app.use(cors());
17 app.use('/drugs', jwt.verify, con.checkAuth);
18 app.get('/token/sign', jwt.sign);
19 app.get('/token/verify', jwt.verify, jwt.verified);
20 app.get('/drugs/id', con.getDrug);
21 app.get('/drugs/search', con.genaralSearchDrugs);
22 app.get('/drugs/search/advance', con.
23     advanceSearchDrugs);
24 app.get('/drugs/dimg', con.getDimg);
25 app.get('/drugs/bookmarks', con.getBookmarks)
26 app.get('/ddl/color', con.ddlColor);
27 app.get('/ddl/dgroup', con.ddlDgroup);
28 app.get('/ddl/drtype', con.ddlDrtype);
29 app.get('/ddl/dshape', con.ddlDshape);
30 app.get('/ddl/dsize', con.ddlDsize);
31 app.get('/ddl/dstatus', con.ddlDstatus);
32 app.get('/ddl/dtype', con.ddlDtype);
33 app.get('/ddl/shapetype', con.ddlShapetype);
34 app.post('/api/drugs/upload', decode64.decode64,
35     opencv.getSizeAndColors, opencv.predict);
36 var server = app.listen(POST, function() {
37     console.log('express is running on port:' +
38         POST);
39 });

```

รูปที่ 4.8: การ routing ของเว็บเซอร์วิส

จากภาพที่ 4.8 อธิบายการทำงานของการ routing ของเว็บเซอร์วิส ได้ดังนี้

- บรรทัดที่ 1 เรียกใช้งานไลบรารี Express และเก็บไว้ในตัวแปรชื่อ app เป็นไลบรารีหลักในการทำงานของ Web application framework
- บรรทัดที่ 2 เรียกใช้งานไลบรารี bodyparser และเก็บไว้ในตัวแปรชื่อ bodyparser เป็น Middleware ทำหน้าที่เป็นตัวกรอง request
- บรรทัดที่ 3 เรียกใช้งานไลบรารี cors และเก็บไว้ในตัวแปรชื่อ cors เป็นกลไกที่ทำให้เว็บเซอร์วิสสามารถอนุญาตการร้องขอทรัพยากรจากเครื่องผู้ใช้งาน
- บรรทัดที่ 4 กำหนดตัวแปรชื่อ PORT
- บรรทัดที่ 5 เรียกใช้งานการเชื่อมต่อกับฐานข้อมูล เก็บไว้ในตัวแปรชื่อ
- บรรทัดที่ 6 เรียกใช้งานการจัดการ JSON web token เก็บไว้ในตัวแปรชื่อ jwt
- บรรทัดที่ 7 เรียกใช้งานการถอดรหัสรูปภาพ base64 เก็บไว้ในตัวแปรชื่อ decode64
- บรรทัดที่ 8-15 กำหนดค่าของเริ่มต้นของ Middleware
- บรรทัดที่ 16 กำหนดการ routing เพื่อตรวจสอบการร้องขอ ก่อนจะให้เรียกทรัพยากรด้วยการตรวจสอบ JWT ที่ส่งมาด้วยกับ HTTP Request
- บรรทัดที่ 18 กำหนดการ routing สำหรับการจ้องขอเพื่อขอ JWT
- บรรทัดที่ 19 กำหนดการ routing สำหรับการตรวจสอบ JWT
- บรรทัดที่ 20 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการยาแบบละเอียด
- บรรทัดที่ 21 กำหนดการ routing สำหรับการร้องขอทรัพยากรระบบการค้นหายาทั่วไป
- บรรทัดที่ 22 กำหนดการ routing สำหรับการร้องขอทรัพยากรระบบการค้นหายาขั้นสูง
- บรรทัดที่ 23 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการยาที่ผู้ใช้งานบุกมา
- บรรทัดที่ 24 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการยาที่ผู้ใช้งานบุกมา

- บรรทัดที่ 25 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการ Dropdownlist ลักษณะสีของยา
- บรรทัดที่ 26 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการ Dropdownlist ประเภททะเบียนยา
- บรรทัดที่ 27 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการ Dropdownlist รูปแบบผลิตภัณฑ์
- บรรทัดที่ 28 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการ Dropdownlist สัญลักษณ์ที่ปรากฏบนเม็ดยา
- บรรทัดที่ 29 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการ Dropdownlist ขนาดของยา
- บรรทัดที่ 30 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการ Dropdownlist สถานะของยา
- บรรทัดที่ 31 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการ Dropdownlist รูปร่างลักษณะยา
- บรรทัดที่ 32 กำหนดการ routing สำหรับการร้องขอทรัพยากรายการ Dropdownlist ของรูปทรงยา
- บรรทัดที่ 33 กำหนดการ routing สำหรับการร้องขอเพื่อส่งรูปภาพมาที่เว็บเซอร์วิส เพื่อประมวลผลรูปภาพยา
- บรรทัดที่ 34-36 ให้เว็บเซอร์วิสทำงานที่ PORT ที่ถูกกำหนดไว้

4.9 การประมวลผลภาพ

การประมวลภาพรูปภาพใช้ไลบรารีของ nodejs ชื่อ opencv4nodejs แสดงดังภาพที่

4.9

```
1 predict: function(req, res, next) {
2     let testDataPath = `./src/assets/threshold/${req.
3         query.filename}`;
4     let img = cv.imread(testDataPath);
5     img = img.resize(64, 64);
6     try {
7         const desc = computeHOGDescriptorFromImage(img
8             , false);
9         if (!desc) {
10             throw new Error('Computing HOG descriptor
11                 failed for file: ${file}');
12         }
13         let svmPredict = svm.predict(desc);
14         let dshape = [];
15         let predict = {
16             label: lccs[svmPredict],
17             colors: req.query.colors,
18             colorid: req.query.colorid,
19             colordname: req.query.colordname,
20             width: req.query.width,
21             dshape: lccs[svmPredict]}
22         }
23         return res.status(200).send(predict);
24     }
```

รูปที่ 4.9: การประมวลผลภาพ

```

22     catch (err) {
23         return res.status(200).send({
24             isError: true,
25             messages: err
26         });
27     }, getSizeAndColors: function(req, res, next) {
28
29     let testDataPath = `./src/assets/images/${req.
30         query.filename}`;
31     let mat = cv.imread(testDataPath);
32     mat = mat.resize(1024, 1024);
33     const know_width = 11.3
34     const know_height = 7.3
35     let boundingRect = [];
36     let num = 0;
37     const thd = 1500;
38     const threshold = 100;
39     const matGray = mat.bgrToGray();
40     var thresh = matGray.threshold(100, 255, 0);
41     const contours = thresh.findContours(cv.
42         RETR_CCOMP, cv.CHAIN_APPROX_SIMPLE, new cv.
43         Point(0, 0));
44
45     try {
46         for (let i = 0; i < contours.length; i++) {
47             var brect = contours[i].boundingRect();
48             const pt1 = new cv.Point(brect.x, brect.
49                 y);
50             const pt2 = new cv.Point(brect.x + brect.
51                 width, brect.y + brect.height);
52             if (brect.width * brect.height >= thd) {
53                 boundingRect.push(brect);
54                 num++;
55                 mat.drawRectangle(pt2, pt1, blue);
56                 const region = mat.getRegion(brect
57                     );
58             }
59         }
60     }
61     else {
62         mat.drawRectangle(pt2, pt1, red);
63     }
64 }
65
66 var matThresh = mat.threshold(100, 255, 0);

```

รูปที่ 4.10: การประมวลผลภาพ(ต่อ)

```

56     if (num < 2) {
57         return res.send({
58             isError: true,
59             messages: 'contounrs < 2.'
60         });
61     }
62     boundingRect.sort(function(rect1, rect2) {
63         if (rect1.height + rect1.width > rect2.
64             height + rect2.width) {
65             return 1;
66         }
67     });
68     const padding = 5;
69     const newRect = new cv.Rect((boundingRect[0].x
70         - padding) >= 0 ? boundingRect[0].x -
71         padding : 0, (boundingRect[0].y - padding)
72         >= 0 ? boundingRect[0].y - padding : 0, mat
73         .cols - (boundingRect[0].x + boundingRect
74         [0].width + padding * 2) >= 0 ? (
75             boundingRect[0].width + padding * 2) :
76             boundingRect[0].width, mat.rows - (
77                 boundingRect[0].y + boundingRect[0].height
78                 + padding * 2) >= 0 ? (boundingRect[0].
79                 height + padding * 2) : boundingRect[0].
80                 height);
81     const region = thresh.getRegion(newRect);
82     var cols = region.cols;
83     var rows = region.rows;
84     var pad = 0;
85     if (region.cols > region.rows) {
86         pad = region.cols - region.rows;
87         rows += pad;
88     }
89     else {
90         pad = region.rows - region.cols;
91         cols += pad;
92     }
93     var matPadded = new cv.Mat(rows, cols, cv.
94         CV_8UC1, 0);
95     for (let i = 0; i < region.rows; i++) {
96         for (let j = 0; j < region.cols; j++) {
97             let atMat = region.at(i, j);
98             if (region.cols > region.rows) {
99                 matPadded.set(i + (pad / 2),
100                     j, atMat);
101             }
102         }
103     }
104 
```

```

88         else {
89             matPadded.set(i, j + (pad /
90                             2), atMat);
91         }
92     }
93     matPadded = matPadded.resize(64, 64);
94     let matCanny = matPadded.canny(threshold,
95         threshold, 3, true);
96     let matInvert = matCanny.bitwiseNot();
97     cv.imwrite(`./src/assets/threshold/${req.query
98         .filename}`, matInvert);
99     var point = [];
100    var colors = [];
101    var colorid = [];
102    var colordname = [];
103    let hexColors = ['FFFFFF', 'ea0000', 'fe700e',
104        'ffe400', 'efeddb', 'a8e26a', '4a9fe0',
105        '120992', 'fdb1ef', 'c977db', 'c2924a', ''
106        'cccccc', '000000']
107    let nameColors = [];
108    point.push([boundingRect[0].x + (boundingRect
109        [0].width / 4), boundingRect[0].y + (
110            boundingRect[0].height / 4)]);
111    point.push([boundingRect[0].x + (boundingRect
112        [0].width / 4), boundingRect[0].y + (
113            boundingRect[0].height * 3 / 4)]);
114    point.push([boundingRect[0].x + (boundingRect
115        [0].width * 3 / 4), boundingRect[0].y + (
116            boundingRect[0].height / 4)]);
117    point.push([boundingRect[0].x + (boundingRect
118        [0].width * 3 / 4), boundingRect[0].y + (
119            boundingRect[0].height * 3 / 4)]);
120    for (let i = 0; i < 4; i++) {
121        let bgr = matThresh.atRaw(parseInt(point
122            [i][1]), parseInt(point[i][0]));
123        let diffColor = 101;
124        let indexColor = 0;
125        let hexColor = rgbHex(bgr[2], bgr[1],
126            bgr[0]);

```

รูปที่ 4.12: การประมวลผลภาพ(ต่อ)

```

112         for (let j = 0; j < hexColors.length; j
113            ++) {
114             let tmp = cd.compare(hexColor,
115                 hexColors[j]);
116             if (tmp < diffColor) {
117                 diffColor = tmp;
118                 indexColor = j;
119             }
120             if (!colorid.includes(indexColor + 1)) {
121                 colors.push(hexColors[indexColor])
122                     ;
123                 colorid.push(indexColor + 1);
124                 colordname.push(nameColors[
125                     indexColor]);
126             }
127             const Rwidth = boundingRect[1].width <
128                 boundingRect[1].height ? boundingRect[1].
129                     height : boundingRect[1].width;
130             const Owidth = boundingRect[0].width <
131                 boundingRect[0].height ? boundingRect[0].
132                     height : boundingRect[0].width;
133             const width = (know_width / Rwidth) * Owidth;
134             req.query.colors = colors;
135             req.query.colorid = colorid;
136             req.query.colordname = colordname;
137             req.query.boundingRect = boundingRect[0];
138             req.query.width = width;
139             next();
140         }
141     }

```

รูปที่ 4.13: การประมวลผลภาพ(ต่อ)

จากภาพที่ 4.9 อธิบายการทำงานของการประมวลผลภาพ ได้ดังนี้

- บรรทัดที่ 1-26 พังก์ชันการจำแนกรูปทรงของยา
- บรรทัดที่ 27-137 พังก์ชันสำหรับการประมวลผลภาพเพื่อหาขนาดและสีของเม็ดยา
- บรรทัดที่ 2-3 กำหนด path และอ่านไฟล์มาเก็บในตัวแปร img
- บรรทัดที่ 4 ปรับขนาดของรูปภาพเป็นขนาด 64×64 พิกเซล
- บรรทัดที่ 6-9 เรียกพังก์ชันสำหรับการแปรรูปภาพให้เป็นอาเรย์ เพื่อที่จะนำไปจำแนกรูปทรง
- บรรทัดที่ 10 เรียกพังก์ชันเพื่อจำแนกรูปทรงของยาและเก็บที่ตัวแปรชื่อว่า svmPredict
- บรรทัดที่ 11 กำหนดตัวชี้อ dshape เป็น label ของรูปทรงต่างๆ
- บรรทัดที่ 12-19 กำหนดตัวชี้อ predict สำหรับเก็บผลลัพธ์การประมวลผล
- บรรทัดที่ 20 คืนผลลัพธ์กลับไปที่เครื่องผู้ใช้งาน
- บรรทัดที่ 28-29 กำหนด path และอ่านไฟล์มาเก็บในตัวแปร mat
- บรรทัดที่ 31-32 กำหนดขนาดด้านยาวและด้านกว้างของวัตถุอ้างอิง ในตัวแปรชื่อ know_width และ know_height
- บรรทัดที่ 33 กำหนดตัวแปรชื่อว่า boundingRect สำหรับเก็บ Contounr
- บรรทัดที่ 35 กำหนดตัวแปรชื่อ thd เก็บค่า threshold เพื่อค่าขั้นต่ำของขนาด Contounr
- บรรทัดที่ 36 กำหนดตัวแปรชื่อ threshold เก็บค่า threshold ของการปรับภาพให้เป็น Canny Edge
- บรรทัดที่ 37 ปรับภาพเป็นให้ภาพ Grayscale
- บรรทัดที่ 38 ปรับ threshold ของภาพ Grayscale
- บรรทัดที่ 39 นำภาพจากตัวแปร thresh มาหา Contounrs ด้วยฟังก์ชัน findContounrs

- บรรทัดที่ 41-53 ตรวจหา contour ที่เป็นวัตถุอ้างอิงและวัตถุเป้าหมาย (ya) และเก็บในอาร์เรย์ชื่อ boundingRect
- บรรทัดที่ 55-59 ถ้าหากมี contour เพียง 0 – 1 อัน จะไม่สามารถดำเนินการต่อไปได้ และคืนข้อผิดพลาดกลับไปที่เครื่องผู้ใช้งาน
- บรรทัดที่ 61-63 เป็นการเรียงลำดับขนาดพื้นที่ของ contour จากน้อยไปมาก
- บรรทัดที่ 66-89 เป็นการตัดภาพเอาเฉพาะภาระยาเม็ด และปรับภาพให้เป็นสีเหลี่ยมจัตุรัส
- บรรทัดที่ 91-93 นำภาพเม็ดยามาหา Canny Edge และเขียนไฟล์รูปภาพ
- บรรทัดที่ 94 กำหนดตัวแปรชื่อ point สำหรับการเก็บตำแหน่งของสีของภาพเม็ดยา
- บรรทัดที่ 95-97 กำหนดตัวแปรชื่อ colors colored และ colorname สำหรับเก็บข้อมูลของสี
- บรรทัดที่ 98 กำหนดตัวแปรชื่อ hexColors สำหรับเก็บค่าโค้ดสีจากฐานข้อมูล
- บรรทัดที่ 99 กำหนดตัวแปรชื่อ nameColors สำหรับชื่อสีจากฐานข้อมูล
- บรรทัดที่ 100-120 เป็นการดึงค่าสี BGR จากภาระยาเม็ดด้วยตำแหน่งที่ได้อาร์เรย์ชื่อ point และนำค่าสีไปเปลี่ยนเทียบกับอาร์เรย์ชื่อ hexColor เพื่อหาค่าสีที่ใกล้เคียงกันมากที่สุด
- บรรทัดที่ 122-124 เป็นการวัดขนาดด้านยาวของเม็ดยา โดยใช้การอ้างอิงจากวัตถุที่รู้ขนาดอยู่แล้ว
- บรรทัดที่ 125-129 เก็บผลลัพธ์การพิสูจน์เอกสารกษณ์ยาเม็ด ไว้ใน req.query
- บรรทัดที่ 130 เรียกฟังก์ชัน next() เพื่อทำงานฟังก์ชันต่อไป
- บรรทัดที่ 131-135 ถ้าหากการประมวลผลภาพเกิดข้อผิดพลาด จะทำการส่งข้อความบอกข้อผิดพลาดไปที่เครื่องผู้ใช้งาน

บทที่ 5

การทดสอบระบบ

ขั้นตอนการทดสอบระบบงานถือเป็นขั้นตอนหนึ่งที่สำคัญของการสร้างระบบงานเนื่องจาก เป็นการสรุปงานหรือเพื่อคุณภาพของการพัฒนาระบบงานรวมทั้งทำให้ทราบถึงปัญหาที่อาจเกิดขึ้นก่อนที่จะนำระบบงานไปใช้งานจริง เพื่อทำการแก้ไขปัญหาได้ทันท่วงที ทั้งนี้เพื่อให้การพัฒนาระบบงานมีความสมบูรณ์มากที่สุด ทำให้เกิดประโยชน์ต่อผู้ใช้งาน และผู้พัฒนาระบบต่อไป แอปพลิเคชันค้นหายาเพื่อคุณที่ได้พัฒนาขึ้นนี้ ได้มีการทดสอบระบบงานในส่วนต่างๆ ดังนี้

- ผลการทดสอบการค้นหายาแบบทั่วไป
- ผลการทดสอบการค้นหายาแบบขั้นสูง
- ผลการทดสอบเลือกดูรายละเอียดยา
- ผลการทดสอบถ่ายรูปภาพเพื่อการพิสูจน์เอกสารยามีด
- ผลการทดสอบการ rounding ของเว็บเซอร์วิส

5.1 ผลการทดสอบการค้นหาแบบทั่วไป

ตารางที่ 5.1: ผลการทดสอบการค้นหาแบบทั่วไป

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการค้นหาแบบทั่วไป	ไม่กรอกข้อมูล และกดค้นหา	แอปพลิเคชัน แสดง ข้อความ “ไม่พบคำร้องขอ”
	กรอก ข้อมูล ครบ และ กดค้นหา	แอปพลิเคชัน แสดง รายการค้นหาให้กับผู้ใช้งาน
	กรอกข้อมูลด้วยภาษาไทย	แอปพลิเคชัน แสดง รายการค้นหา และแสดงรายการ
	กรอกข้อมูลด้วยภาษาอังกฤษ	แอปพลิเคชัน แสดง รายการค้นหา และแสดงรายการ
	กดปุ่ม Cancel	แอปพลิเคชันจะกลับมาแสดงหน้าค้นหา

5.2 ผลการทดสอบการค้นหาแบบขั้นสูง

ตารางที่ 5.2: ผลการทดสอบการค้นหาแบบขั้นสูง

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการค้นหาแบบขั้นสูง	ไม่กรอกข้อมูลลักษณะยา	แอปพลิเคชัน แสดง ข้อความ “ไม่พบคำร้องขอ”
	กรอกข้อมูลลักษณะยาไม่ครบ	แอปพลิเคชัน แสดง รายการค้นหา และแสดงรายการ
	กรอกข้อมูลลักษณะยาที่ไม่มีในฐานข้อมูล	แอปพลิเคชัน แสดง ข้อความ “ไม่พบคำร้องขอ”
	กรอก ข้อมูล ลักษณะ ยาค รับ ถูกต้อง	แอปพลิเคชัน แสดง รายการค้นหา และแสดงรายการ
	กดปุ่มย้อนกลับ	แอปพลิเคชันจะกลับมาแสดงหน้าค้นหา

5.3 ผลการทดสอบเลือกตู้รายละเอียดยา

ตารางที่ 5.3: ผลการทดสอบเลือกตู้รายละเอียดยา

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
เลือกตู้รายละเอียดยา	กดบันทึกบັນການຮ່ວມຍາ ທີ່ຍັງໄມ້ຖຸກບັນທຶກບັນການຮ່ວມຍາ	ແອປພລິເຄັ້ນ ແສດງ ຂໍ້ອຄວາມ “ບັນທຶກ ບັນທຶກ ບັນທຶກສໍາເລື່ອຈ ແລ້ວ”
	กดບັນທຶກບັນການຮ່ວມຍາ ທີ່ຖຸກບັນທຶກບັນການຮ່ວມຍາແລ້ວ	ແອປພລິເຄັ້ນ ແສດງ ຂໍ້ອຄວາມ “ລົບບັນທຶກບັນການຮ່ວມຍາສໍາເລື່ອຈ ແລ້ວ”
	กดຍ້ອນກັບ	ແອປພລິເຄັ້ນ ຈະ ແສດງ ມຳ ຮ່ວມຍາກັບມີຄົນຫາຍາ
	กดປຸ່ມ Home	ແອປພລິເຄັ້ນ ຈະ ແສດງ ມຳ ຄົນຫາຍາ
	กดທີ່ຮູ່ປາພາພາ	ແອປພລິເຄັ້ນ ຈະ ແສດງຮູ່ປາພ ພາແບບຂໍາຍາໃໝ່ຈິ້ນ

5.4 ผลการทดสอบถ่ายรูปภาพเพื่อการพิสูจน์เอกสารลักษณ์ยาเม็ด

ตารางที่ 5.4: ผลการทดสอบถ่ายรูปภาพเพื่อการพิสูจน์เอกสารลักษณ์ยาเม็ด

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบถ่ายรูปภาพ เพื่อ การพิสูจน์เอกสารลักษณ์ยาเม็ด	กดถ่ายรูปภาพ	แอปพลิเคชัน จะ เปิด กล้อง ถ่ายรูปของเครื่องผู้ใช้งาน
	กดประมวลผลภาพ	แอปพลิเคชันจะแสดงผลลัพธ์ การพิสูจน์เอกสารลักษณ์ยา และ แสดงหน้ารายการค้นหา
	ถ่ายรูปภาพที่ไม่ใช่ยา และ กด ประมวลผลภาพ	แอปพลิเคชัน จะ แสดง ข้อความ “เกิดข้อผิดพลาด ลองถ่ายรูปใหม่”
	ถ่ายรูปภาพยา ที่ ไม่ได้ใช้วัตถุ อ้างอิง	แอปพลิเคชันจะ แสดงผลลัพธ์ และ ไม่สามารถคำนวณขนาด ของยาได้
	ถ่าย รูปภาพ ยา ที่ มี ลักษณะ เหมือนกัน สี ขนาด รูปทรง และเป็นยาต่างชนิดกัน	แอปพลิเคชันจะ แสดงผลลัพธ์ ที่ เหมือนกัน เนื่องจากมีลักษณะทางกายภาพเหมือนกัน

5.5 ผลการทดสอบการ rounting ของเว็บเซอร์วิส

ตารางที่ 5.5: ผลการทดสอบการ rounting ของเว็บเซอร์วิส

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
ร้อง รอ ทรัพยากร จาก เว็บเซอร์วิส	ร้องขอไปที่ /token/sign และ ส่ง ข้อมูล ไอ ดี เครื่อง ไป ด้วย	เว็บเซอร์วิสคืนโตเคนกลับมา ให้
	ร้องขอไปที่ /token/sign และไม่ส่งข้อมูลไอดีเครื่องไป ด้วย	เว็บเซอร์วิสคืนโตเคนกลับมา ให้
	ร้องขอไปที่ /token/verify และส่งโตเคนที่ถูกต้องไป	เว็บเซอร์วิส คืน สถานะ ปกติ ของโตเคนกลับมา
	ร้องขอไปที่ /token/verify และส่งโตเคนที่ไม่ถูกต้องไป	เว็บเซอร์วิสคืนสถานะผิดปกติ ของโตเคนกลับมา
	ร้องขอไปที่ /token/verify และไม่ส่งโตเคนไปด้วย	เว็บเซอร์วิสคืนสถานะผิดปกติ ของโตเ肯กลับมา
	ร้องขอไปที่ /drugs/search/ id พร้อม ส่ง หมายเลขยา และ โตเคนที่ถูก ต้องไปด้วย	เว็บเซอร์วิส คืน ราย ละเอียด ยา กลับมา
	ร้องขอไปที่ /drugs/search/ id พร้อม ส่ง หมายเลขยา และ โตเคนที่ไม่ ถูกต้องไปด้วย	เว็บเซอร์วิส คืน ข้อ ผิด พลาด ของสถานะโตเคนกลับมา
	ร้องขอไปที่ /drugs/search/ id พร้อม ส่ง หมายเลขยา และ โตเ肯ที่ไม่ ถูกต้องไปด้วย	เว็บเซอร์วิส คืน ข้อ ผิด พลาด ของสถานะโตเคนกลับมา
	ร้องขอไปที่ /api/drugs/upload พร้อม ส่ง รูปภาพ และ โตเคนที่ถูกต้องไปด้วย	เว็บเซอร์วิส คืน ผลลัพธ์ การ ประมวลภาพรูปภาพกลับมา
	ร้องขอไปที่ /api/drugs/upload พร้อมส่งไฟล์ที่ไม่ ใช่รูปภาพและ โตเคนที่ถูกต้อง ไปด้วย	เว็บเซอร์วิส คืน ข้อ ผิด พลาด กลับมา

biblatex MathCS.bib

บทที่ 6

สรุปและข้อเสนอแนะ

จากขั้นตอนการศึกษาและเริ่มพัฒนาแอปพลิเคชันค้นหายาเพื่อคุณ โดยผ่านกระบวนการต่างๆ ได้แก่ การวิเคราะห์แบบออกแบบระบบ การพัฒนาโปรแกรมและการทดสอบโปรแกรม จนสิ้นสุดกระบวนการซึ่งสามารถสรุปผลของโครงการได้ดังนี้

- สรุปความสามารถของระบบ
- ปัญหาและอุปสรรคในการพัฒนา
- แนวทางการพัฒนาต่อ

6.1 สรุปความสามารถของระบบ

แอปพลิเคชันค้นหายาเพื่อคุณ ได้ดำเนินการออกแบบสร้างระบบงานและทำการทดสอบระบบงาน สามารถสรุปผลได้ดังนี้ โดยจะแบ่งเป็น 2 ส่วน คือ ส่วนแอปพลิเคชัน และส่วนเว็บ-เซอร์วิส ดังนี้

1. ส่วนแอปพลิเคชัน

- สามารถค้นหายาเม็ดได้ทั้งแบบทั่วไปและแบบขั้นสูง
- สามารถถ่ายรูปภาพเพื่อการพิสูจน์เอกสารยานยาเม็ดได้
- สามารถรายละเอียดของยาได้
- สามารถบันทึกบัญการร้ายการรยาได้

2. ส่วนเว็บเซอร์วิส

- สามารถติดต่อกับฐานข้อมูลการพิสูจน์เอกสารยานยาเม็ดหรือแคปซูล คณะเภสัชศาสตร์ มหาวิทยาลัยอุบราชธานี

- สามารถจำแนกรูปทรงของเม็ดยาหั้ง 6 รูปทรง ได้แก่ สี่เหลี่ยม สามเหลี่ยม วงกลม วงกลม หกเหลี่ยม และแปดเหลี่ยม
- สามารถระบุเอกสารลักษณะสีของยาเม็ดได้ และระบุความขนาดของเม็ดยาได้ ด้วยการ ประมวลผลภาพ
- สามารถป้องกันการร้องขอทรัพยากรจากบุคคลอื่นที่ไม่ใช่แอปพลิเคชันค้นหาเพื่อ คุณ

6.2 ปัญหาและอุปสรรคในการพัฒนา

1. ปัญหา : การติดตั้งเว็บเซอร์วิสที่เครื่องเซิฟเวอร์นอกเนื้องจากการพัฒนา จะเป็นต้องติดตั้ง โปรแกรมเสริมจำนวนมากและเวอร์ชันของโปรแกรมไม่เหมือนกัน ทำให้ไม่สามารถเริ่มการ ทำงานของเว็บเซอร์วิสได้
แนวทางการแก้ไข : การพัฒนาเว็บเซอร์วิสไว้ใน Docker container เพื่อลดปัญหาการ ติดตั้งโปรแกรมเสริมและการไม่เข้ากันของเวอร์ชันของโปรแกรม เพียงแค่เครื่องเซิฟเวอร์ ติดตั้ง Docker
2. ปัญหา : การหาข้อมูลรูปภาพรูปทรงต่างๆ สำหรับฝึกและสอนแบบจำลอง รูปทรงหก เหลี่ยมและรูปทรงแปดเหลี่ยมในฐานข้อมูลการพิสูจน์เอกสารลักษณ์ยาเม็ดหรือแคปซูล คณะ เกสัชศาสตร์ มหาวิทยาลัยอุบราชธานี มีจำนวนน้อย
แนวทางการแก้ไข : ค้นหารูปภาพจากแหล่งที่มาอื่นมาเพิ่ม
3. ปัญหา : รูปภาพที่ได้มาจากการฐานข้อมูลการพิสูจน์เอกสารลักษณ์ยาเม็ดหรือแคปซูล และแหล่ง ที่มาอื่น ยังไม่สามารถนำมาใช้งานได้ต้องทำการทำความสะอาดข้อมูลรูปภาพเพื่อฝึกสอน และทดสอบแบบจำลอง
แนวทางการแก้ไข : เขียนโค้ดภาษา Javascript สำหรับการทำความสะอาดรูปภาพจำ- นวนมาก โดยแยกตามรูปทรงและแบ่งข้อมูลสำหรับฝึกสอนและทดสอบแบบจำลอง

6.3 แนวทางการพัฒนาต่อ

1. การปรับแบบจำลองให้รองรับกับรูปทรงอื่นๆ ที่นอกเนื่องจากรูปทรงที่ผู้พัฒนาใช้ในการสร้างแบบจำลอง เช่น รูปหัวใจ รูปสัตว์ต่างๆ เป็นต้น
2. พัฒนาถ่ายรูปและประมวลผลแบบ Real Time เพื่อการพิสูจน์เอกสารลักษณ์ที่สะท้อนสภาพชีวิต

(1)

(2)

(3)

(4)

(5)

(6)

(7)

(8)

(9)

(10)

(11)

(12)

(13)

(14)

(15)

(16)

(17)

(18)

(19)

(20)

บรรณานุกรม

1. JSON Web Token, “Json web token มาตรฐานใหม่ ใน การ ทำ authentication,” March 2018.
2. Routing, “Routing protocol,” March 2018.
3. Express, “Express - node.js web application framework,” March 2018.
4. Contours, “Opencv: Contours : Getting started,” March 2018.
5. Canny Edge, “Opencv: Canny edge detection,” March 2018.
6. WebView, “Javascript in webview,” March 2018.
7. MySQL, “เกี่ยวกับ mysql,” March 2018.
8. Nginx, “Nginx,” March 2018.
9. Docker, “ทำความรู้จัก docker,” March 2018.
10. Navigating Lifecycle Events, “Navigating lifecycle events,” March 2018.
11. Native Application, “Native application,” April 2018.
12. Web Application, “What is a web application (or ”webapp”)?,” April 2018.
13. Drug Identification Database, “Drug identification database | ฐานข้อมูลการพิสูจน์เอกสารชั้นเยี่ดและแครปซูลในประเทศไทย โดย คณะเภสัชศาสตร์ มหาวิทยาลัยอุบลราชธานี,” April 2018.

14. TCP/IP, “Transmission control protocol/internet protocol,” April 2018.
15. Docker Hub, “Docker hub,” April 2018.
16. Ionic Framework, “Build amazing native apps and progressive web apps with ionic framework and angular,” April 2018.
17. Ubuntu, “The leading operating system for pcs, iot devices, servers and the cloud | ubuntu,” April 2018.
18. Nearest Neighbors, “Nearest neighbors scikit-learn 0.19.1 documentation,” April 2018.
19. RandomForestClassifier, “Randomforestclassifier scikit-learn 0.19.1 documentation,” April 2018.
20. បណ្តុះពិន្ទុក្រសាយ, “បណ្តុះពិន្ទុក្រសាយ,” April 2018.

ภาคผนวก

ภาคผนวก ก

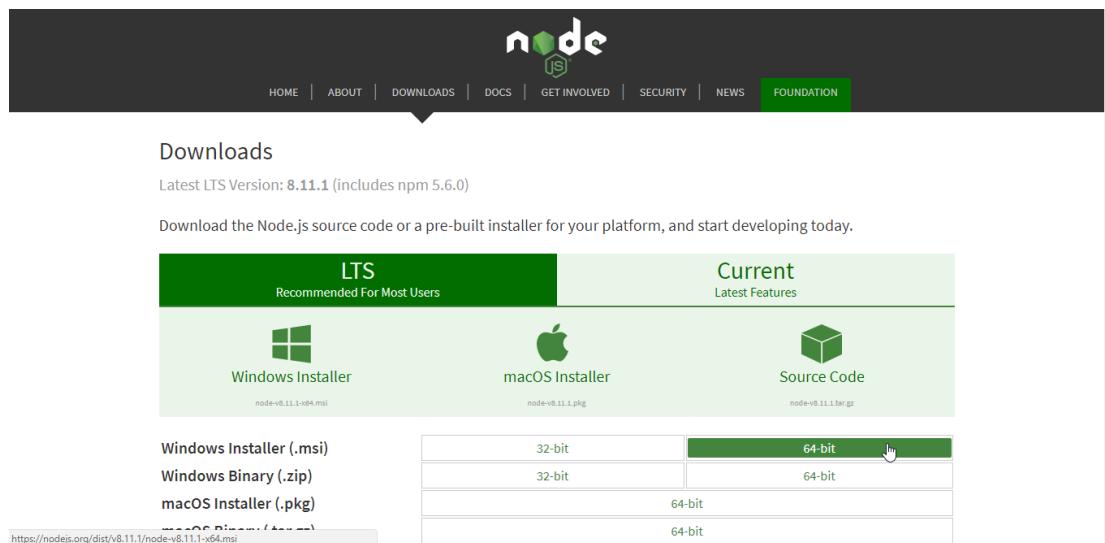
การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม

การติดตั้งเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันค้นหายาเพื่อคุณ มีโปรแกรมที่จำเป็นใน การพัฒนาระบบดังต่อไปนี้

- การติดตั้ง Node.js
- การติดตั้ง Ionic Framework
- การติดตั้ง OpenCV

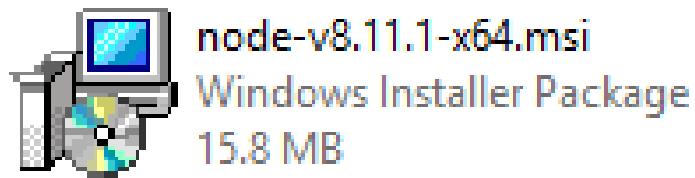
ก.1 การติดตั้ง Node.js

1) สามารถดาวน์โหลด Node.js ได้ที่ <https://nodejs.org/en/download/> ดัง แสดงในภาพที่ ก.1



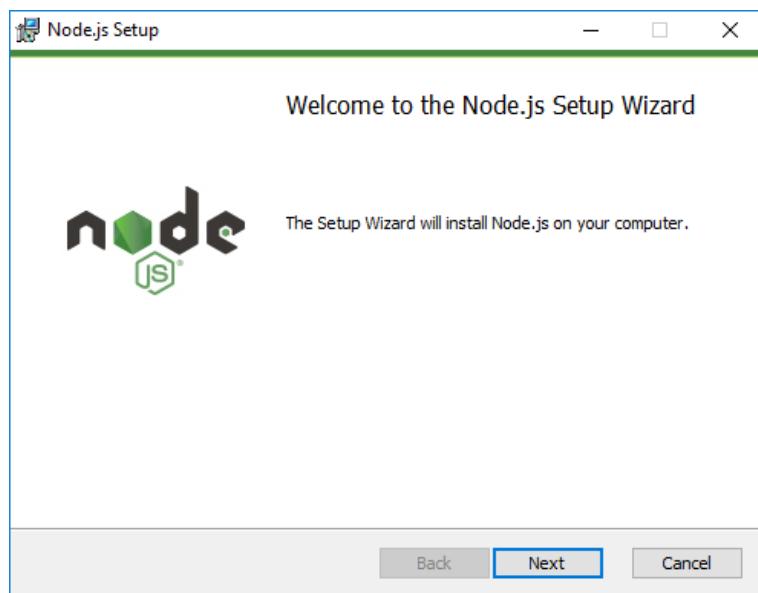
รูปที่ ก.1: หน้าเว็บดาวน์โหลด Node.js

2) เปิดไฟล์ติดตั้ง ชื่อ node-v8.11.1-x64.msi เพื่อติดตั้ง ดังแสดงในภาพที่ ก.2



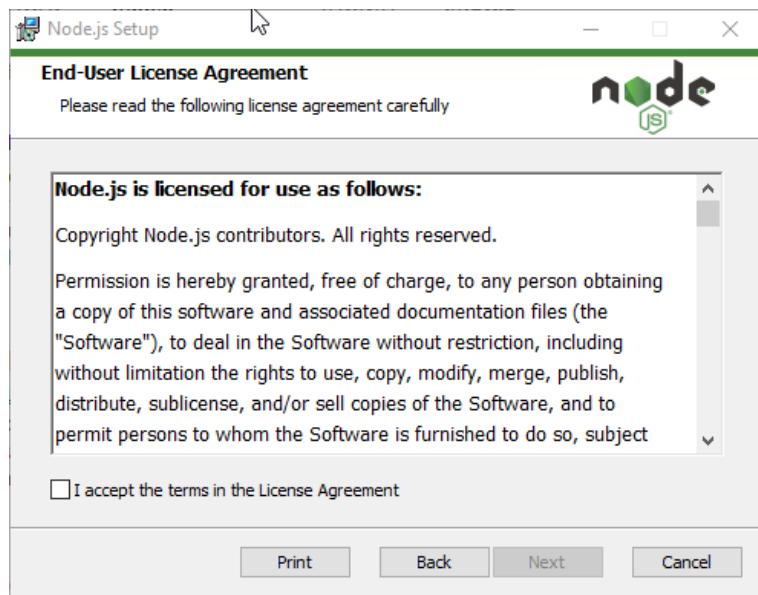
รูปที่ ก.2: ไฟล์ติดตั้งสำหรับติดตั้ง Node.js

3) แสดงหน้าต่างตอนรับของ Node.js ให้กด Next ดังแสดงในภาพที่ ก.3



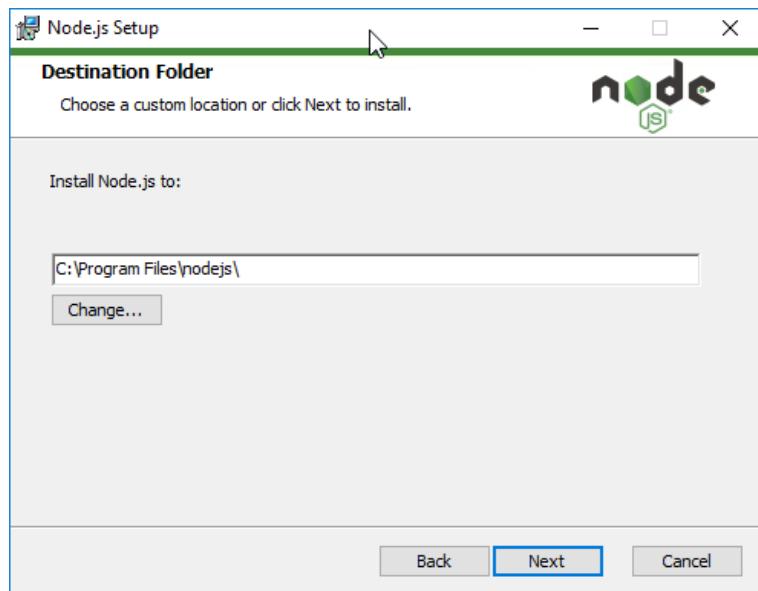
รูปที่ ก.3: หน้าต่างตอนรับของ Node.js

4) แสดงหน้าต่างข้อตกลงในการใช้ Node.js ให้เลือกช่อง I accept the terms in the License Agreement และกด Next ดังแสดงในภาพที่ ก.4



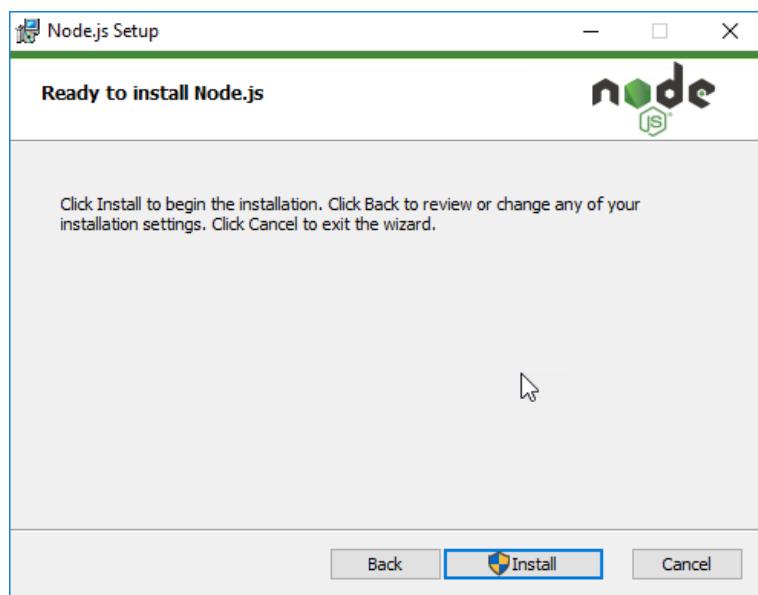
รูปที่ ก.4: หน้าต่างข้อตกลงในการใช้ Node.js

5) แสดงหน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง ดังแสดงในภาพที่ ก.5



รูปที่ ก.5: หน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง Node.js

6) แสดงหน้าต่างสำหรับติดตั้ง Node.js ให้กด Install เพื่อทำงานติดตั้ง ดังแสดงในภาพที่ ก.6



รูปที่ ก.6: หน้าต่างติดตั้ง Node.js

ก.2 การติดตั้ง Ionic Framework

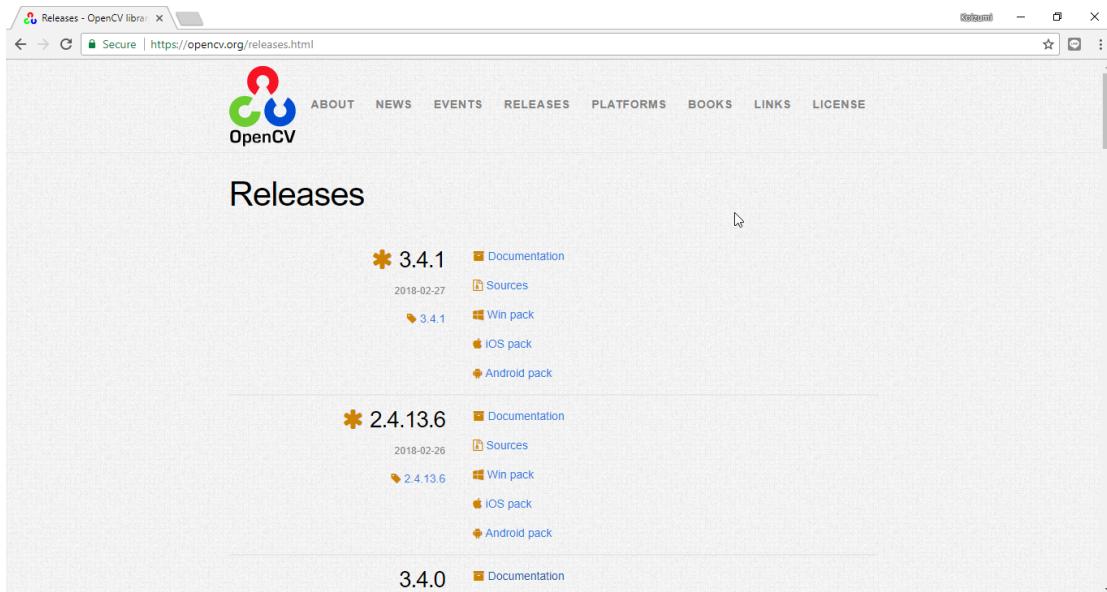
การติดตั้ง Ionic Framework สามารถทำผ่านคำสั่ง command line ได้ ดังแสดงในภาพที่ ก.7

```
sudo npm -install g cordova
sudo npm -install g ionic
```

รูปที่ ก.7: คำสั่งสำหรับติดตั้ง ionic framework

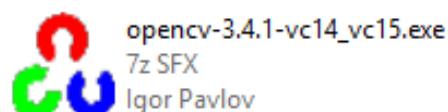
ก.3 การติดตั้ง OpenCV

- 1) สามารถดาวน์โหลด OpenCV ได้ที่ <https://opencv.org/releases.html> ดังแสดงในภาพที่ ก.8



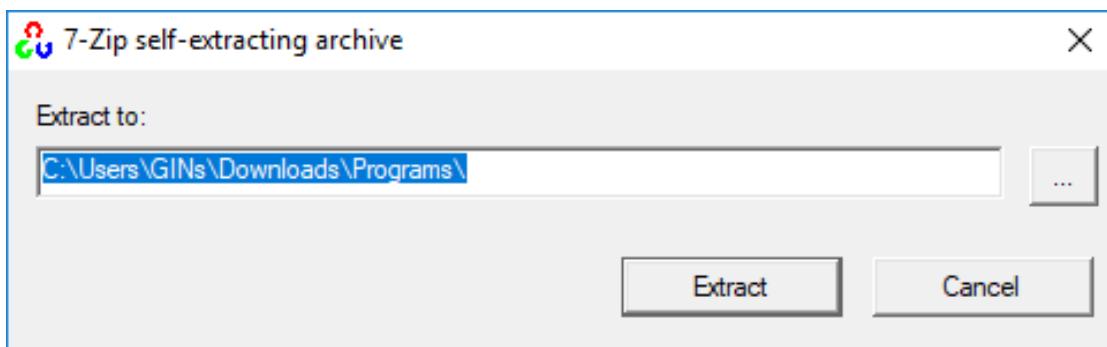
รูปที่ ก.8: หน้าเว็บสำหรับดาวน์โหลด OpenCV

2) เปิดไฟล์สำหรับติดตั้ง OpenCV ดังแสดงในภาพที่ ก.9



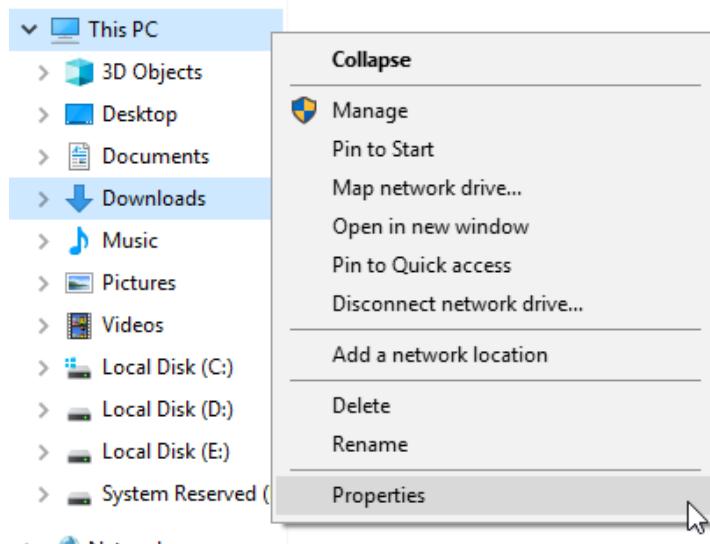
รูปที่ ก.9: ไฟล์ติดตั้ง OpenCV

3) เลือกโปรแกรมที่ต้องการตั้งติดและกด Extract ดังแสดงในภาพที่ ก.10



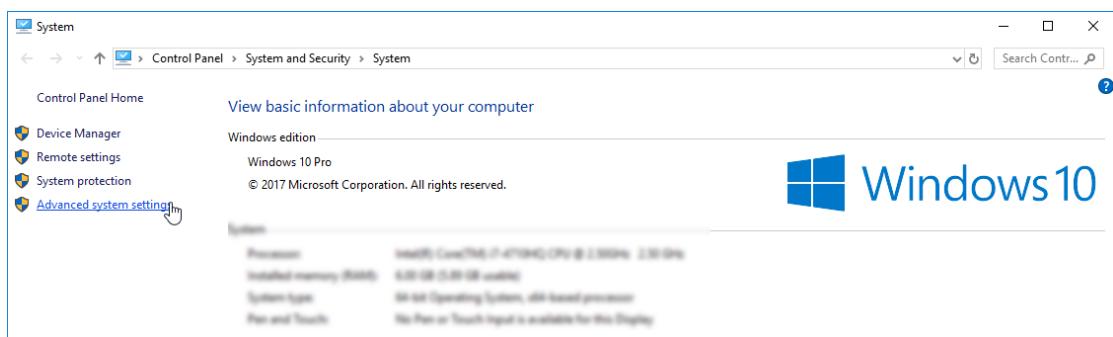
รูปที่ ก.10: เลือกโปรแกรมสำหรับติดตั้ง OpenCV

4) เปิด This PC ขึ้นมาและคลิกขวาที่ This PC เพื่อเข้าไปที่หน้า System Properties ดังแสดงในภาพที่ ก.11



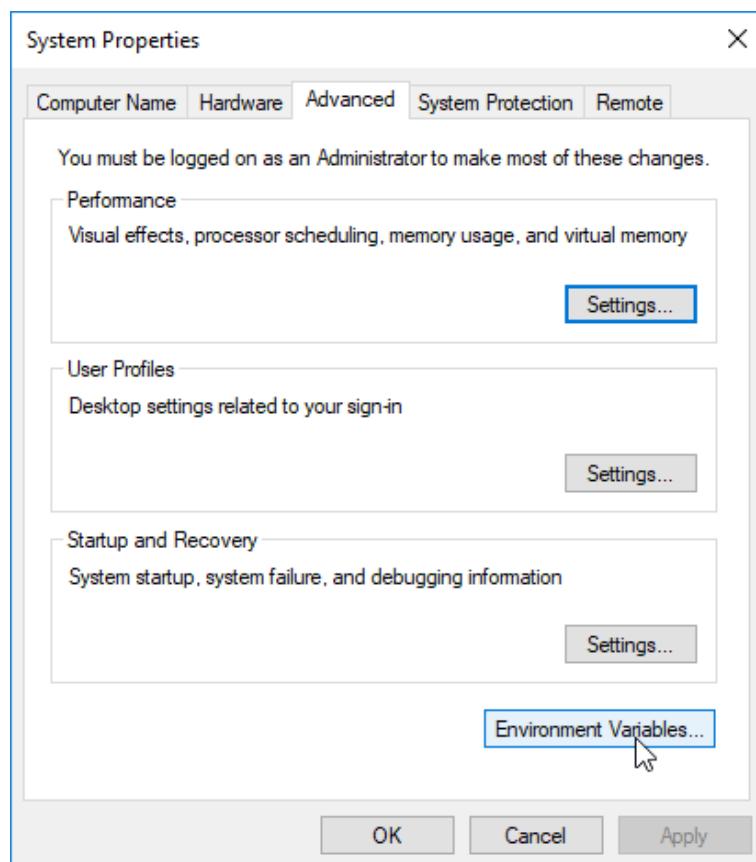
รูปที่ ก.11: เปิด properties ของ This PC

5) เลือก Advanced system setting ดังแสดงในภาพที่ ก.12



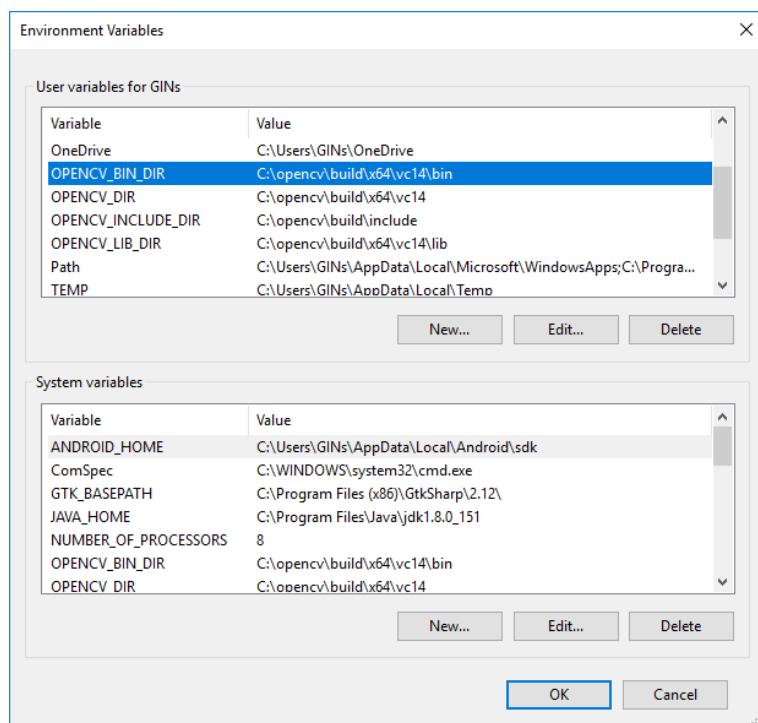
รูปที่ ก.12: หน้าต่างของ System

6) เลือก Enviroment Variables ดังแสดงในภาพที่ ก.13



รูปที่ ก.13: หน้าต่าง System Properties

7) ทำการสร้างตัวแปร OPENCV_BIN_DIR, OPENCV_DIR, OPENCV_INCLUDE_DIR, OPENCV_LIB_DIR และกำหนดค่าของตัวแปร และกด OK ดังแสดงในภาพที่ ก.14



รูปที่ ก.14: หน้าต่าง Environment Variables

ภาคผนวก ข

การติดตั้งแอปพลิเคชันค้นหารายาเพื่อคุณ

การติดตั้งแอปพลิเคชันค้นหารายาเพื่อคุณได้ผ่านบริการของ Google Play สำหรับอุปกรณ์โทรศัพท์สมาร์ทโฟนที่มีระบบปฏิบัติการแอนดรอยด์ แสดงดังภาพที่ ข.1

ดาวน์โหลดได้ที่ : <https://play.google.com/store/apps/details?id=com.ubu.cs.drugiden>



รูปที่ ข.1: หน้าดาวน์โหลดแอปพลิเคชันค้นหารายาเพื่อคุณผ่านบริการ Google Play

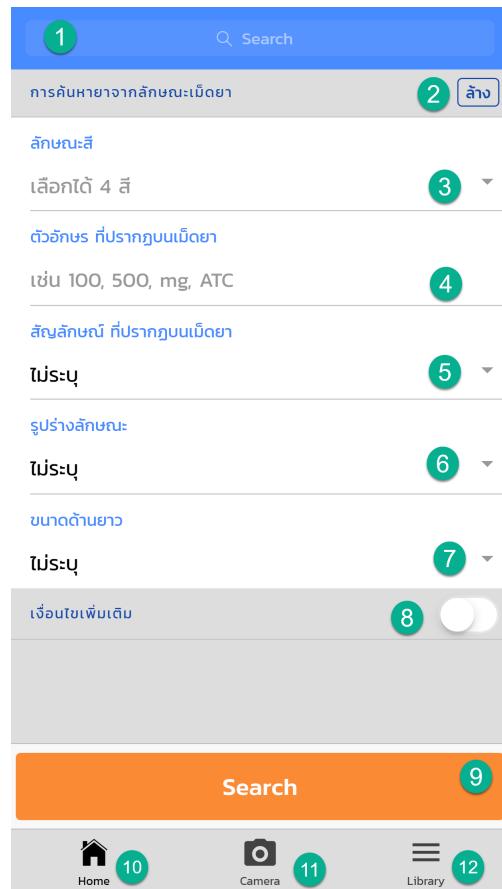
ภาคผนวก ค

คู่มือการใช้งานแอปพลิเคชันค้นหารายาเพื่อคุณ

คู่มือการใช้งานแอปพลิเคชันค้นหารายาเพื่อคุณ มีวิธีการใช้งานดังต่อไปนี้ เมื่อผู้ใช้งานเปิดแอปพลิเคชันจะพบกับหน้าแรก ดังแสดงในภาพที่ ค.1 และหลังจากแสดงหน้าแรก 1.5 วินาทีระบบเปลี่ยนไปแสดงหน้าค้นหา ดังแสดงในภาพที่ ค.2



รูปที่ ค.1: หน้าแรกของแอปพลิเคชันค้นหารายาเพื่อคุณ

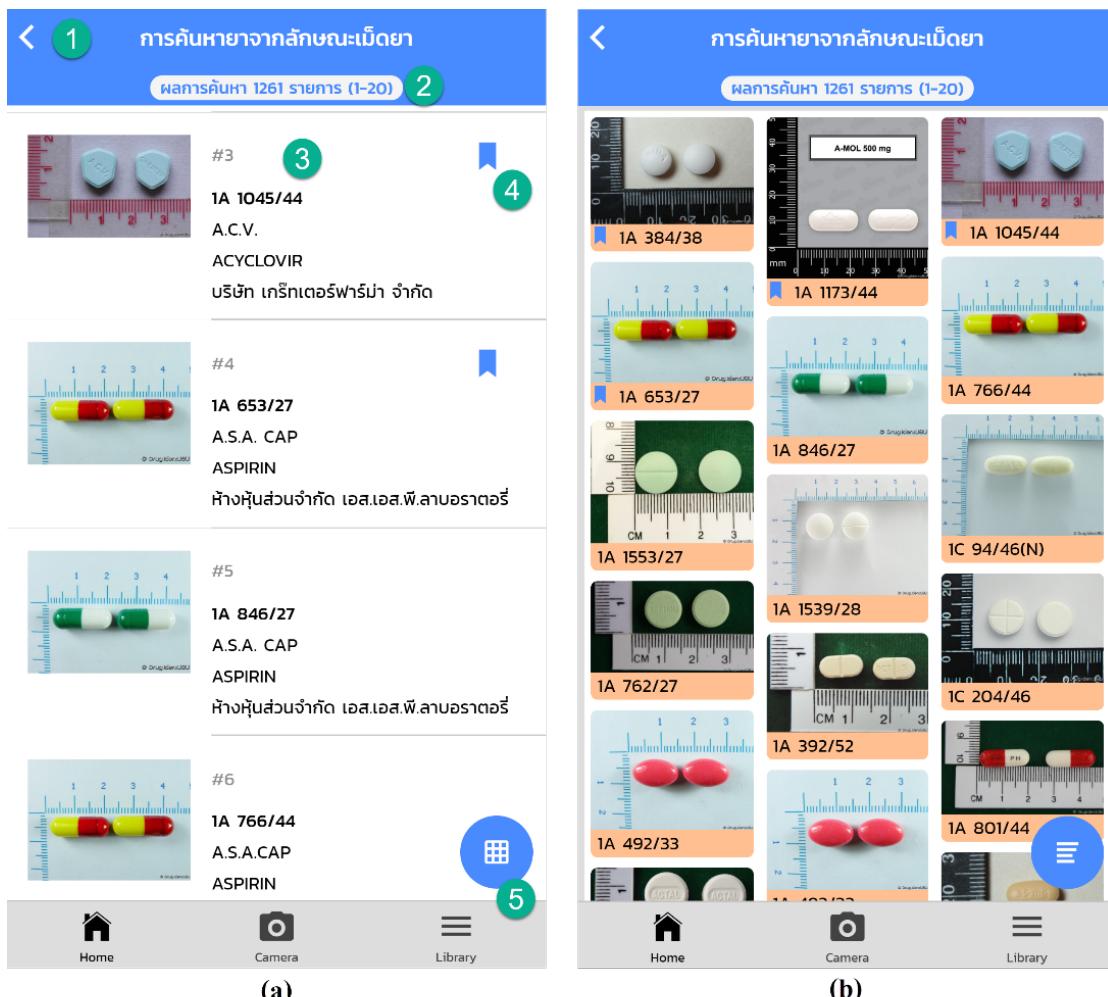


รูปที่ ค.2: หน้าค้นหายาของแอปพลิเคชันค้นหายาเพื่อคุณ

จากการที่ ค.2 หน้าการค้นหายา เป็นหน้าสำหรับค้นหายาโดยใส่ข้อมูลลักษณะของยาลงไปในช่องกรอกข้อมูล และสามารถอธิบายรายละเอียดต่างๆของหน้าค้นหายาได้ดังนี้

- หมายเลข 1 คือ ช่องกรอกคำค้นหาแบบทั่วไป สามารถกรอกข้อมูลคำค้นได้
- หมายเลข 2 คือ สามารถกดเพื่อล้างคำค้นหาในหน้าค้นหายาได้
- หมายเลข 3 คือ สามารถกดเพื่อเลือกลักษณะสีของยาได้มากสุด 4 สี
- หมายเลข 4 คือ ช่องกรอกข้อมูลตัวอักษณะที่ปรากฏบนเม็ดยา
- หมายเลข 5 คือ สามารถกดเพื่อเลือกสัญลักษณ์ที่ปรากฏบนเม็ดยาได้
- หมายเลข 6 คือ สามารถกดเพื่อเลือกรูปร่างลักษณะของเม็ดยาได้

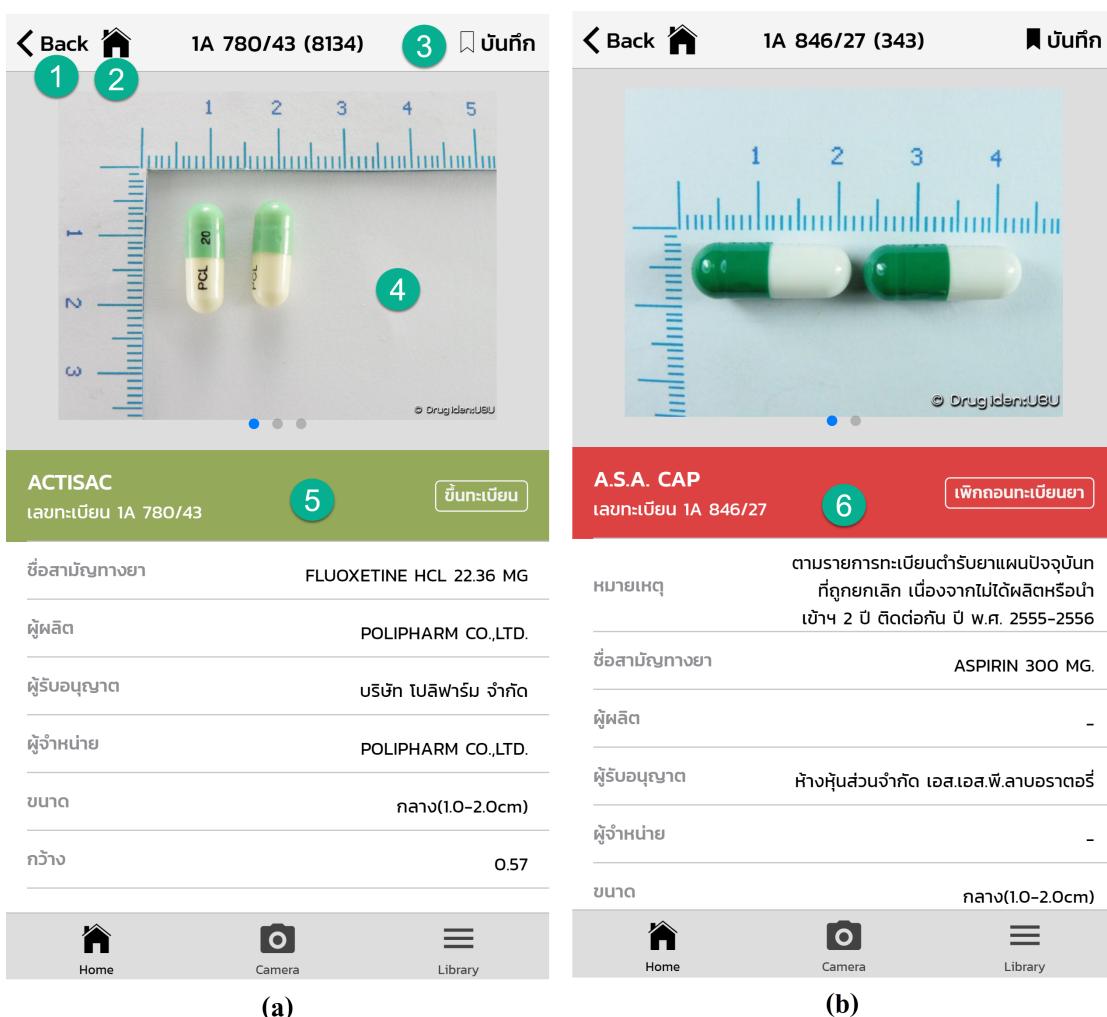
- หมายเลข 7 คือ สามารถกดเพื่อเลือกขนาดด้านยาวของเม็ดยาได้
- หมายเลข 8 คือ สามารถกดเพื่อแสดงเงื่อนไขเติมของการค้นหายาจากลักษณะเม็ดได้
- หมายเลข 9 คือ สามารถกดเพื่อค้นหาจากการนำข้อมูลหมายเลข 3-7 ไปค้นหา และจะแสดงไปยังหน้ารายการค้นหา
- หมายเลข 10 คือ สามารถกดเพื่อไปยังหน้าค้นหา (Home)
- หมายเลข 11 คือ สามารถกดเพื่อไปยังหน้ากล้องถ่ายรูป
- หมายเลข 12 คือ สามารถกดเพื่อไปยังหน้าไลบรารี



รูปที่ ค.3: หน้ารายการการค้นหายา

จากภาพที่ ค.3 หน้ารายการการค้นหายาแบบรายการ เมื่อผู้ใช้งานกดค้นหาหรือกรอกคำค้นหาแบบทั่วไปในช่องค้นหา แอปพลิเคชันจะแสดงหน้ารายการค้นหายา ผู้ใช้งานสามารถเลื่อนขึ้นและเลื่อนลงเพื่อดูรายการค้นหายาได้ ถ้าหากผู้ใช้กดเลือกรายการยา แอปพลิเคชันจะแสดงหน้ารายละเอียดเม็ดยาที่ผู้ใช้งานกดเลือก อธิบายรายละเอียดตั้งนี้

- หมายเลข 1 คือ สามารถกดย้อนกลับไปหน้าค้นหาได้
- หมายเลข 2 คือ แสดงจำนวนรายการที่ค้นหาเจอทั้งหมด
- หมายเลข 3 คือ แสดงเลขลำดับของรายการยา และสามารถแตะเพื่อดูรายละเอียดของรายการยาได้
- หมายเลข 4 คือ เครื่องหมายบุ๊กมาร์ก ถ้าหากรายการยาไม่เครื่องหมายบุ๊กมาร์กติดอยู่หมายความว่า รายการยาถูกบุ๊กมาร์กไว้แล้วในเครื่องผู้ใช้งาน
- หมายเลข 5 คือ สามารถกดเพื่อเปลี่ยนรูปแบบการแสดงรายการจากการเป็นรูปขนาดย่อ (Thumbnails) ได้ ดังภาพที่ ค.3 (b)



รูปที่ ค.4: หน้าแสดงรายละเอียดยา

จากภาพที่ ค.4 หน้าแสดงรายละเอียดยา เมื่อผู้ใช้งานกดเลือกรายการยาจากหน้ารายการยา แอปพลิเคชันจะแสดงหน้ารายละเอียด ของรายการรายละเอียดดังนี้

- หมายเลขอ 1 คือ สามารถกดเพื่อย้อนกลับไปยังหน้าแสดงรายการการค้นหายาได้
- หมายเลขอ 2 คือ สามารถกดเพื่อกลับไปยังหน้าค้นหายาได้
- หมายเลขอ 3 คือ สามารถกดบันทึกบุ๊กマーกได้ ถ้าหากสัญลักษณ์บุ๊กマーกเป็นตั้งภาพที่ ค. 4 (a) หมายถึงรายการยังไม่ได้ถูกบุ๊กマーก

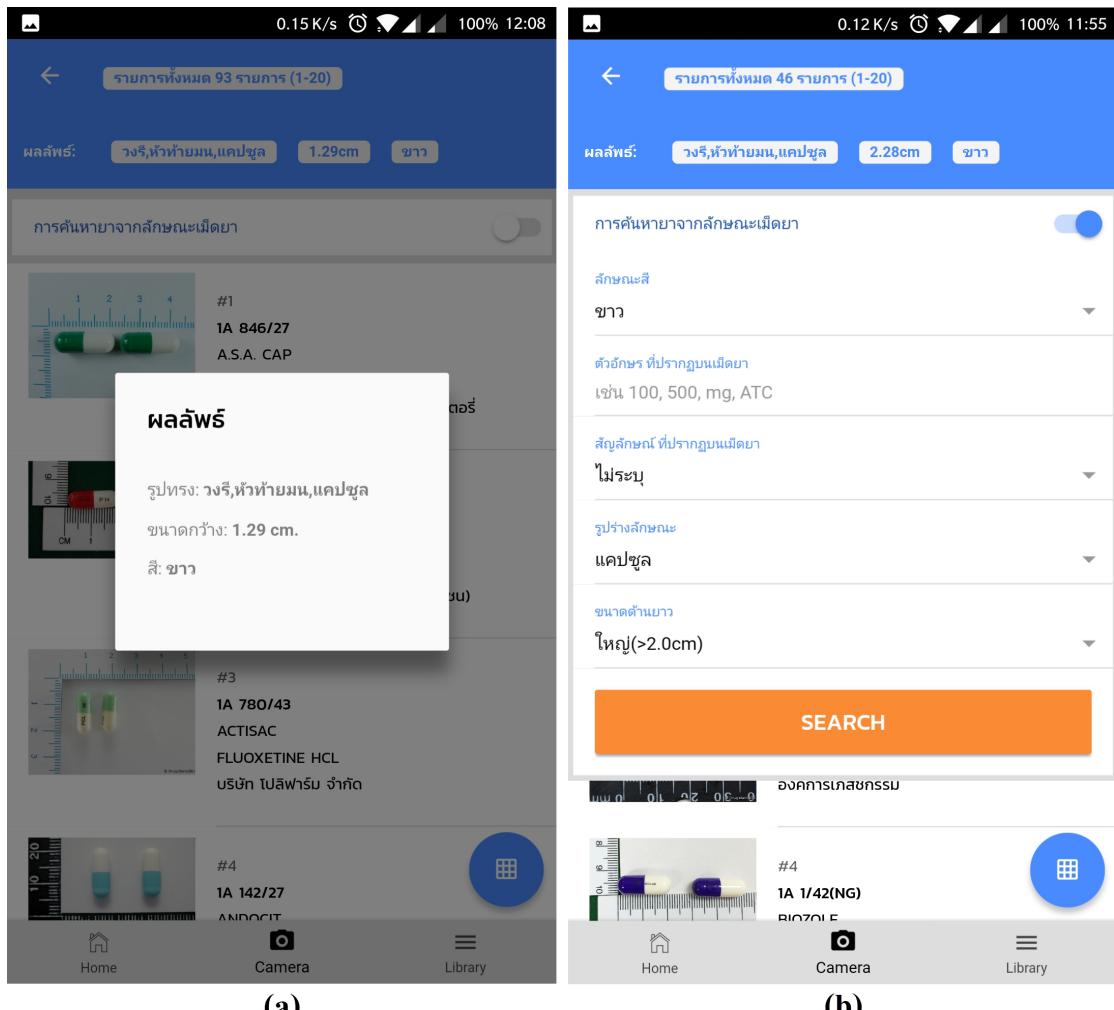
- หมายเลขอ 4 คือ สามารถกดที่รูปภาพเพื่อขยายรูปภาพให้ใหญ่ขึ้นได้
- หมายเลขอ 5 คือ แสดงชื่อสามัญของเม็ดยา เลขทะเบียน และถ้ามีบราเป็นสีเขียวหมายถึง สถานะของยาขึ้นอยู่เที่ยน
- หมายเลขอ 6 คือ แสดงชื่อสามัญของเม็ดยา เลขทะเบียน และถ้ามีบราเป็นสีแดงหมายถึง สถานะของถูกเพิกถอนไปแล้ว และสามารถดูรายละเอียดที่หมายเหตุได้



รูปที่ ค.5: หน้าถ่ายรูปภาพ

จากการที่ ค.5 หน้าถ่ายรูปภาพ เป็นหน้าสำหรับการถ่ายรูปภาพเม็ดยาเพื่อพิสูจน์เอกสาร โดยนำเม็ดยาวางไว้บนวัสดุอ้างอิงสีดำที่มีขนาดด้านยาว 10 เซนติเมตร ด้านกว้าง 7 เซนติเมตร และใช้กล้องถ่ายรูปถ่ายรูปจากมุมสูง เมื่อถ่ายรูปภาพเสร็จแล้ว แอปพลิเคชันจะแสดง ดังภาพที่ ค.5 (a)

เมื่อผู้ใช้งานกดประมวลผล (Process) แอปพลิเคชันจะแสดงผลลัพธ์การพิสูจน์เอกสารลักษณ์ยาเม็ดที่หน้าแสดงผลลัพธ์การพิสูจน์เอกสารลักษณ์ยา ดังภาพที่ ค.5 (b) และยังสามารถแก้ไขผลลัพธ์เพื่อใช้ค้นหาได้ด้วยการกดค้นหาเพื่อดูรายการค้นหาที่แก้ไขไปได้



รูปที่ ค.6: หน้าแสดงผลลัพธ์การพิสูจน์เอกสารลักษณ์ยา

ประวัติผู้เขียน

ชื่อ-สกุล: นายกฤษณะ ชินโคตร

รหัสประจำตัวนักศึกษา: 5711404802

วัดเกิด: 15 กันยายน 2538

ที่อยู่ที่สามารถติดต่อได้: 44/1 ถ.เกษตรวน ต.варิน อ.วารินชำราบ จ.อุบลราชธานี 34190

เบอร์โทรศัพท์: (+66) 93 325 9005

อีเมลล์: kritsana.ch.57@ubu.ac.th

ระดับมัธยมต้น: โรงเรียนเบญจมมหาราช จังหวัดอุบลราชธานี

ระดับมัธยมปลาย: โรงเรียนเบญจมมหาราช จังหวัดอุบลราชธานี

ระดับอุดมศึกษา: ภาควิชาคณิตศาสตร์ สัตติ และคอมพิวเตอร์ สาขาวิทยาการ คอมพิวเตอร์ คณะ
วิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี