

ระบบค้นหาสินค้าลดราคาด้วยวิธีการเข้าถึงข้อมูลหน้าเว็บไซต์

Find items for sale with web scrapping

ประหยัด เลวัน และ ธงชัย เทียงธรรม

สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยรามคำแหง

Email: prayatl@gmail.com, mandarinkb@gmail.com

บทคัดย่อ

แอปพลิเคชันค้นหาสินค้าลดราคา จัดทำเพื่อเป็นเครื่องมือช่วยค้นหาสินค้าลดราคาจากเว็บไซต์ต่างๆ ซึ่งใช้วิธีการเข้าถึงข้อมูลหน้าเว็บไซต์ (web scrapping) โดยได้แบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนที่หนึ่ง ระบบเข้าถึงข้อมูลหน้าเว็บไซต์ต่างๆ ซึ่งระบบนี้จะทำการดึงข้อมูลหน้าเว็บไซต์ต่างๆ ที่ต้องการ จากนั้นทำการจัดเก็บข้อมูลลงใน NoSQL Database ที่มีชื่อว่า Elasticsearch ส่วนที่สอง เป็นส่วนแสดงผล โดยแสดงผ่านแอปพลิเคชันค้นหาสินค้าลดราคา บนระบบปฏิบัติการแอนดรอยด์ การพัฒนาระบบค้นหาสินค้าลดราคาด้วยวิธีการเข้าถึงข้อมูลเว็บไซต์ มีเครื่องมือที่ช่วยพัฒนาได้แก่ โปรแกรม Virtual Studio Code โปรแกรม Android Studio โปรแกรม Postman โปรแกรมภาษาจาวา (Java) โปรแกรม Redis โปรแกรม Elasticsearch และโปรแกรม Docker ผลการทดสอบการดำเนินงานของระบบพบว่า ค่าเฉลี่ยผลความครบถ้วนของข้อมูลที่ทำให้การสแครปป์มาจากเว็บไซต์ร้านค้าออนไลน์ขนาดใหญ่ ร้อยละ 99.95 และเมื่อเพิ่มจำนวนบอทจาก 3 ไปเป็น 13 บอทเวลาประมวลผลลดลงถึงประมาณ 10 เท่า (170.24/17.21) ในส่วนแอปพลิเคชันค้นหาสินค้าลดราคา ผลการทดสอบเมนูค้นหาพบว่าการค้นพบสินค้าตามที่ต้องการคิดเป็นค่าเฉลี่ย 92.83 % ผลสรุปด้านประสิทธิภาพของแอปพลิเคชันประกอบด้วยเนื้อหา 10 หัวข้อ โดยได้ให้ผู้ใช้และผู้พัฒนาระบบทดลองใช้งานผลปรากฏว่า ผู้ใช้มีความพึงพอใจอยู่ในระดับ “มาก” ($\bar{X} = 4.20$, $S.D. = 0.59$) และผู้พัฒนาระบบมีความพึงพอใจในมุมมองการออกแบบ

ฟังก์ชันการใช้งานอยู่ในระดับ “มาก” ($\bar{X} = 4.20$, $S.D. = 0.56$) และผลการทดสอบเชิฟเวอร์สามารถรองรับการใช้งาน 270 ผู้ใช้งานต่อวินาที
คำสำคัญ: ระบบปฏิบัติการแอนดรอยด์
แอปพลิเคชัน web scrapping Elasticsearch

บทนำ

ในปัจจุบันอินเทอร์เน็ต (internet) มีบทบาทและมีความสำคัญต่อชีวิตประจำวันมากขึ้น โดยลักษณะการใช้อินเทอร์เน็ตมีจุดประสงค์หลายรูปแบบ ไม่ว่าจะเป็นเพื่อใช้ติดต่อสื่อสาร เพื่อทำงาน เพื่อศึกษาหาความรู้ เพื่อความบันเทิง หรือแม้กระทั่งใช้ค้นหาข้อมูลตามที่ต้องการ เมื่ออินเทอร์เน็ตมีบทบาทสำคัญและผู้คนสามารถใช้อินเทอร์เน็ตได้ง่ายขึ้นผ่านสมาร์ทโฟน ทำให้มีร้านค้าออนไลน์มากขึ้น เมื่อร้านค้าออนไลน์ทำให้มีความสะดวกสบายในการเลือกซื้อสินค้า โดยการเลือกซื้อสินค้าแต่ละครั้ง ผู้สั่งซื้อต้องค้นหาเลือกซื้อสินค้าตามที่ต้องการ อาจจะดูจากราคา โปรโมชั่น คุณภาพสินค้า หรือความน่าเชื่อถือของร้านจากเว็บไซต์ผู้ให้บริการ บางครั้งอาจจะต้องค้นหาจากเว็บไซต์หลายเว็บไซต์ เพื่อเปรียบเทียบราคา ให้ตรงกับความต้องการมากที่สุด

ด้วยเหตุนี้ผู้จัดทำได้พัฒนาแอปพลิเคชันสินค้าลดราคา โดยระบบจะทำการดึงข้อมูลจากหน้าเว็บไซต์และนำสินค้าลดราคามาจัดเก็บไว้ ทำให้ผู้ใช้งานเลือกดูสินค้าที่ลดราคาจากเว็บไซต์ต่างๆ ที่ทางระบบได้กำหนดไว้ ส่วนเครื่องมือและเทคโนโลยีที่เกี่ยวข้องในการพัฒนาระบบมีดังต่อไปนี้

1 การทำ web scrapping วิธีการในการดึงข้อมูลจากหน้าเว็บเพจหรือเว็บไซต์ โดยใช้ภาษา

โปรแกรมมีเป็นเครื่องมือ ปกติการดึงข้อมูลนั้นมี 2 วิธีหลักๆ คือ ดึงจาก API และทำ web scrapping เมื่อได้ข้อมูลจากเว็บไซต์ที่ต้องการ [1] ทำการจัดเก็บข้อมูลลงใน NoSQL ของ Elasticsearch

2 Elasticsearch ที่เก็บข้อมูลที่พัฒนาต่อยอดมาจาก Apache Lucene มีจุดเด่นในเรื่องความสามารถของการค้นหา และสรุปข้อมูลขนาดใหญ่ได้อย่างรวดเร็ว [2] โดยวิธีการคิวรีของ Elasticsearch มีหลายแบบ แบ่งเป็น 2 แบบหลักๆ คือ

1 Full text queries

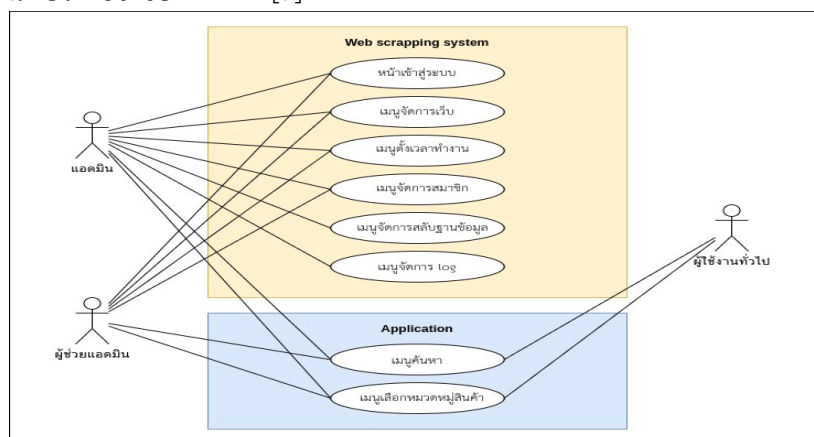
2 Term-level queries แบ่งเป็น 11 คิวรี ได้แก่ exists query , fuzzy query , ids query , prefix query , range query , regexp query , term query , terms query , terms_set query , type query , wildcard query ในระบบได้เลือกการคิวรีแบบ regexp query (regular expression) มาใช้งาน

3 Web API เป็นตัวกลางเชื่อมต่อเพื่อให้แอปพลิเคชันสามารถเชื่อมต่อกับเซิร์ฟเวอร์ โดยข้อมูลอยู่ในรูปแบบของ JSON รับส่งข้อมูลผ่านโปรโตคอล HTTP ทางระบบได้ใช้ HTTPS เป็นรุ่นปลอดภัยของโปรโตคอล HTTP ที่ใช้ SSL/TLS โปรโตคอล สำหรับการเข้ารหัสและพิสูจน์ตัวตน HTTPS ถูกระบุโดย RFC 2818 และใช้พอร์ต 443 เป็นค่าเริ่มต้น แทนพอร์ต 80 ของ HTTP [3]

4 Docker เป็นแพลตฟอร์มซอฟต์แวร์ที่ช่วยสร้าง ทดสอบ และติดตั้งแอปพลิเคชันได้อย่างรวดเร็ว Docker จะบรรจุซอฟต์แวร์ลงไปในหน่วยที่เป็นมาตรฐานเรียกว่า “คอนเทนเนอร์” ซึ่งจะมีทุกสิ่งที่ซอฟต์แวร์ต้องการใช้ในการเรียกใช้งาน รวมทั้ง ไลบรารี เครื่องมือสำหรับระบบ โค้ดและรันไทม์ เมื่อใช้ Docker จะสามารถติดตั้งใช้จริงและปรับขนาดแอปพลิเคชันให้เหมาะสมกับสภาพแวดล้อม [4]

วิธีการดำเนินงาน

ในระบบได้แบ่งการทำงานเป็น 2 ส่วน คือ 1) หน้าเว็บไซต์ที่ใช้ควบคุมบอท 2) แอปพลิเคชันค้นหาสินค้าลดราคา โดยระบบงานหน้าเว็บไซต์ที่ใช้ควบคุมบอท ในขั้นตอนแรกจะทำการเก็บข้อมูลไว้ในฐานข้อมูล ระบบจะมีคุณสมบัติ แสดง เพิ่ม ลบ แก้ไข ข้อมูลในระบบ เมนูหน้าเว็บไซต์ที่ใช้ควบคุมบอทมี 6 เมนู ได้แก่ หน้าเข้าสู่ระบบ เมนูจัดการเว็บ เมนูตั้งเวลาทำงาน เมนูจัดการสมาชิก เมนูจัดการลบฐานข้อมูล เมนูจัดการ log ซึ่งผู้ใช้งานที่สามารถใช้เมนูดังกล่าวได้ มี 2 ผู้ใช้งาน คือ แอดมิน และผู้ช่วยแอดมิน ส่วนแอปพลิเคชันค้นหาสินค้าลดราคา มีรายละเอียด 2 เมนู คือ เมนูค้นหา และเมนูเลือกหมวดหมู่สินค้า ซึ่งผู้ใช้งานทุกคนสามารถเข้าใช้งานได้ ดังรูปที่ 1



รูปที่ 1 Use Case Diagram ระบบค้นหาสินค้าลดราคา

จากนั้นทำการศึกษาโครงสร้างหน้าเว็บไซต์ เมื่อเข้าสู่หน้าเว็บไซต์จะพบหมวดหมู่สินค้าหลายรายการ เมื่อเลือกหมวดหมู่สินค้าเข้าไป จะพบในหมวดหมู่นั้น มีหลายหน้าเว็บไซต์ ในแต่ละเว็บไซต์จะมีสินค้าหลายรายการ จุดมุ่งหมายของระบบดึงข้อมูลหน้าเว็บไซต์ คือต้องการตรวจสอบรายการสินค้าทุกรายการและจัดเก็บสินค้าตามที่ต้องการไว้ จะเห็นว่าทั้ง 3 เว็บไซต์ มีโครงสร้างที่เหมือนกัน ดังรูปที่ 2

จากการศึกษาโครงสร้างหน้าเว็บไซต์ สามารถแบ่งการทำงานออกเป็น 3 ส่วน จุดประสงค์ของการแบ่งงานเป็น 3 ส่วนคือเพื่อให้การทำงานดึงข้อมูลหน้าเว็บไซต์รวดเร็วมากขึ้น โดยจะสร้างบอท (bot) มาทำงานในแต่ละส่วน บอท ในที่นี้หมายถึงโปรแกรมที่ทำงานเพื่อดึงข้อมูลหน้าเว็บไซต์ โดยจะทำตามเงื่อนไขที่ได้กำหนดไว้ภายในโปรแกรม ซึ่งทั้ง 3 บอทนี้ได้แก่ start bot fetch url bot และ web scrapping bot ดังรูปที่ 2

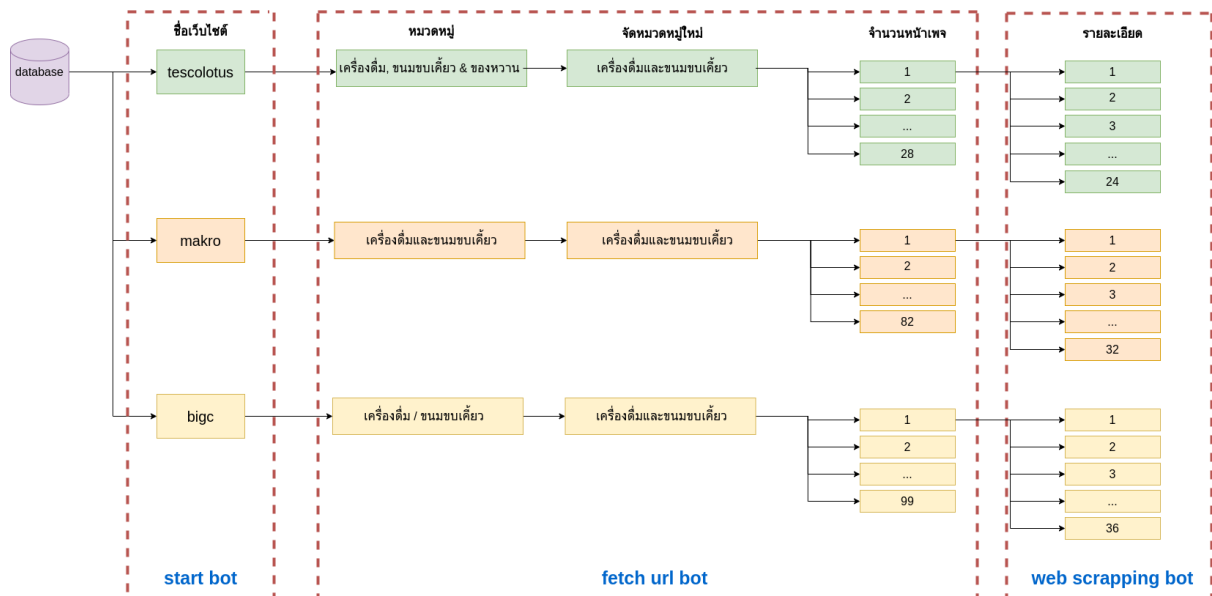
start bot ทำงานตามเวลาที่ได้ตั้งค่าไว้ในระบบ โดยจะเลือกเว็บไซต์ที่ต้องการดึงข้อมูล จากนั้น

ทำการจัดเก็บข้อมูลลง Redis [5] ซึ่งเป็นฐานข้อมูลชั่วคราว เพื่อส่งต่อการทำงานไปให้ fetch url bot

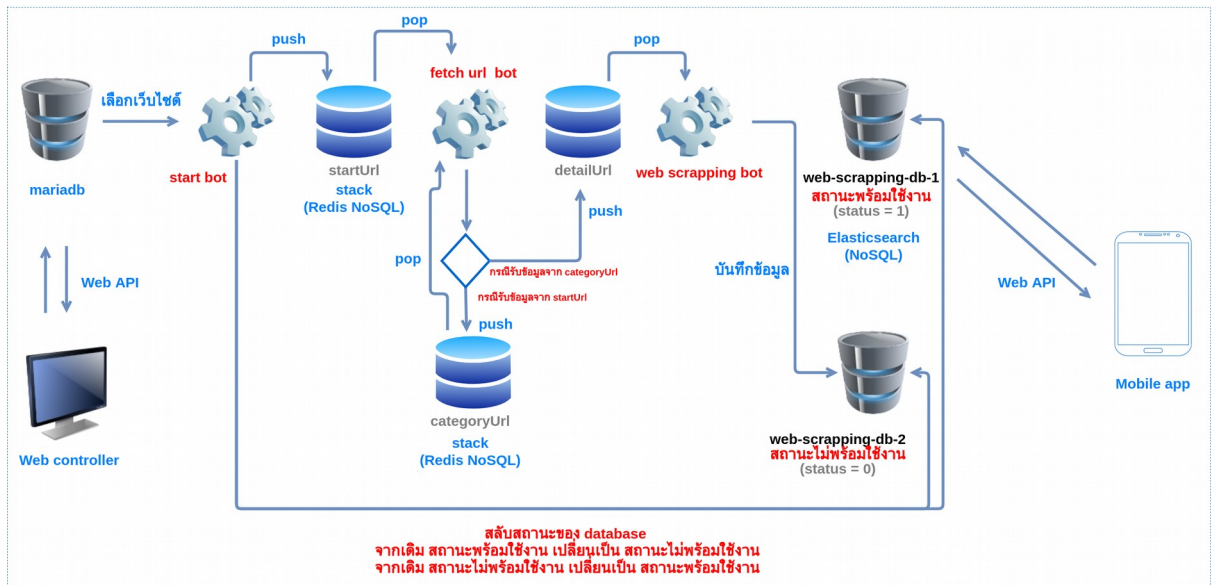
fetch url bot นำข้อมูลที่อยู่ใน Redis มาทำงานตามเงื่อนไขของระบบ และทำการจัดเรียงหมวดหมู่สินค้าขึ้นมาใหม่ จากนั้นทำการจัดเก็บข้อมูลลง Redis เพื่อส่งต่อการทำงานให้ web scrapping bot

web scrapping bot นำข้อมูลที่อยู่ใน Redis มาทำงานโดยจะทำการดึงข้อมูลในหน้าเว็บไซต์ และบันทึกข้อมูลลงฐานข้อมูล ดังรูปที่ 3

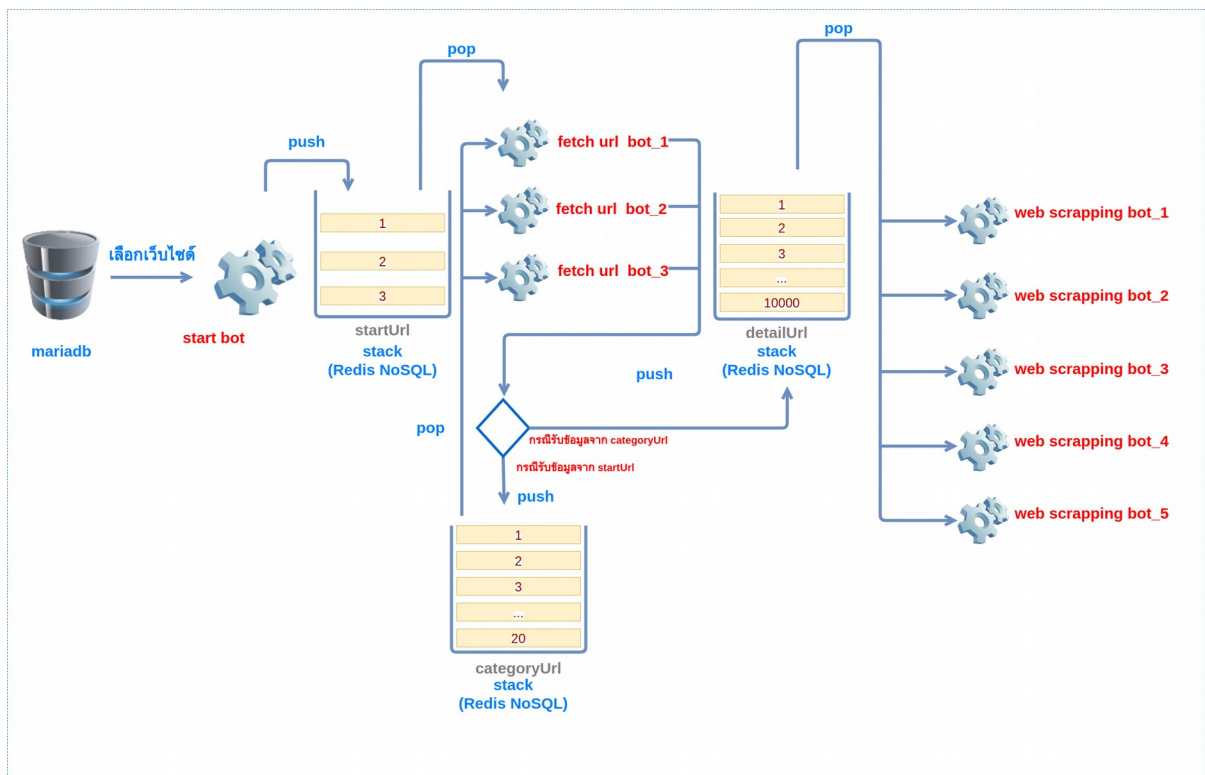
จะเห็นได้ว่าการทำงานดังกล่าวเป็นลักษณะแบบส่งต่อการทำงานเป็นช่วง โดยไม่จำเป็นต้องรอให้บอทตัวใดตัวหนึ่งทำงานจนเสร็จ ถ้ามีข้อมูลใน Redis บอทตัวถัดไปสามารถนำข้อมูลมาทำงานต่อได้เลย และเมื่อข้อมูลที่อยู่ใน Redis ส่วนใดมีข้อมูลจำนวนมาก สามารถเพิ่มจำนวนบอทช่วยกันทำงาน จึงทำให้การทำงานดึงข้อมูลหน้าเว็บไซต์รวดเร็วมากขึ้น ดังรูปที่ 4



รูปที่ 2 โครงสร้างหน้าเว็บไซต์และการแบ่งงานออกเป็น 3 ส่วน



รูปที่ 3 ขั้นตอนการทำงานของบอทเพื่อดึงข้อมูลหน้าเว็บไซต์



รูปที่ 4 แสดงการเพิ่มจำนวนบอท

ผลการศึกษาและอภิปรายผล

ผลการพัฒนา Bot เพื่อดึงข้อมูลหน้าเว็บไซต์

1 ทดสอบความครบถ้วนของข้อมูล ได้ทำการทดสอบดึงข้อมูลหน้าเว็บไซต์ เทสโก้ โลตัส ผลปรากฏว่าความครบถ้วนของข้อมูลอยู่ในระดับ 99.95% ดังตารางที่ 1

2 ทดสอบเพิ่มจำนวนบอท เพื่อดูระยะเวลาการดึงข้อมูลทั้ง 3 เว็บไซต์ คือ เทสโก้ โลตัส แม็คโคร บิ๊กซี ผลปรากฏว่าเมื่อเพิ่มจำนวนบอท เป็น 4 บอท ระยะเวลาการทำงานลดเกินครึ่ง จากนั้นเพิ่มจำนวนบอทไปตามลำดับ ระยะเวลาการทำงานก็ลดลงเรื่อยๆ ดังตารางที่ 2 และรูปที่ 5

ตารางที่ 1 ผลการทดสอบความครบถ้วนของข้อมูล

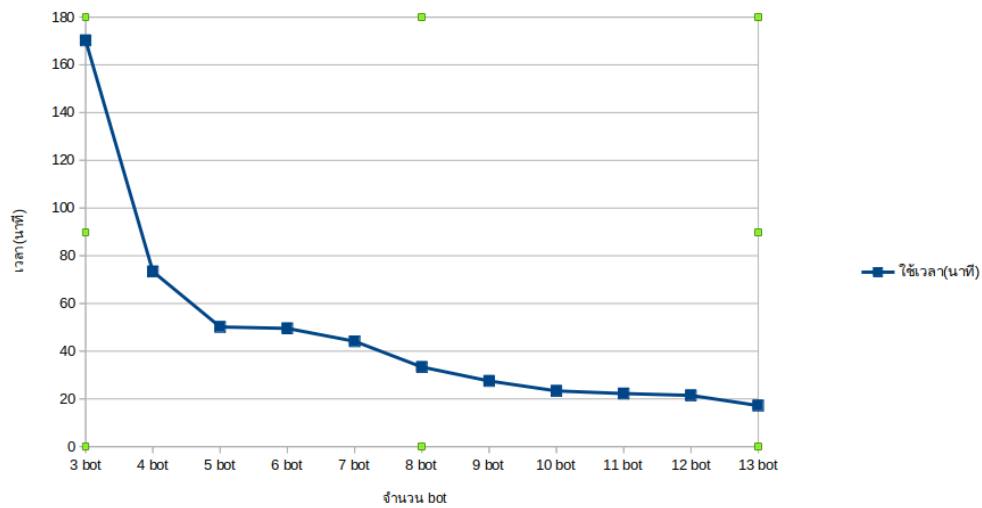
ลำดับ	วันที่	ข้อมูล สินค้า ทั้งหมด	ข้อมูล สินค้าที่ได้	เปอร์เซ็นต์
1	10/05/64	4235	4230	99.88
2	10/05/64	4235	4235	100
3	10/05/64	4235	4235	100
4	10/05/64	4235	4235	100
5	10/05/64	4235	4223	99.71
6	10/05/64	4235	4235	100
7	10/05/64	4235	4234	99.98
8	10/05/64	4235	4235	100
9	10/05/64	4235	4232	99.93
10	10/05/64	4235	4235	100
			ค่าเฉลี่ย	99.95

ตารางที่ 2 ผลการทดสอบเพิ่มจำนวนบอท

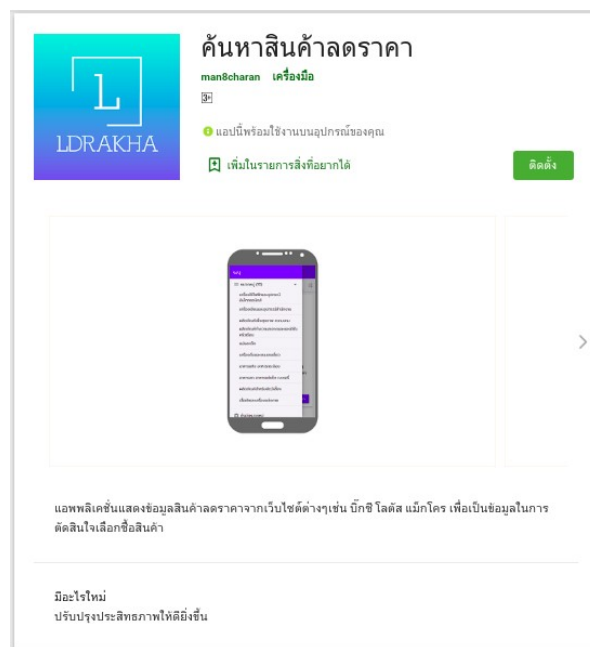
ลำดับ	วันที่	จำนวนบอท	ใช้เวลา (นาที)
1	11/03/64	3 Bot	170.24
2	11/03/64	4 Bot	73.38
3	11/03/64	5 Bot	50.16
4	11/03/64	6 Bot	49.55
5	11/03/64	7 Bot	44.14
6	11/03/64	8 Bot	33.36
7	11/03/64	9 Bot	27.51
8	11/03/64	10 Bot	23.34
9	11/03/64	11 Bot	22.24
10	11/03/64	12 Bot	21.47
11	11/03/64	13 Bot	17.21

ข้อมูลที่ได้แสดงผ่านแอปพลิเคชันค้นหาสินค้าลดราคา

การติดตั้งแอปพลิเคชันค้นหาสินค้าลดราคา สามารถดาวน์โหลดได้จาก Play Store โดยทำการค้นคำว่า “ค้นหาสินค้าลดราคา” จากนั้นกดค้นหา จะพบแอปพลิเคชัน ค้นหาสินค้าลดราคาจากนั้นกดติดตั้ง ดังรูปที่ 6



รูปที่ 5 กราฟแสดงการใช้เวลาทำงานของบอท



รูปที่ 6 หน้าจอแสดงการติดตั้งของแอปพลิเคชัน

แอปพลิเคชันสินค้าลดราคา ในเมนูค้นหาได้ใช้เครื่องมือของ Elasticsearch ช่วยค้นหา โดยเลือกใช้คิวรีแบบ regexp query (Regular Expression) [6] ซึ่งเป็นการกำหนดรูปแบบหรือกลุ่มคำ เพื่อเอาไว้ใช้ค้นหาข้อความต่างๆตามที่ต้องการ

ทดสอบ 100 คำค้น แยกเป็น 10 หมวดหมู่ หมวดหมู่ละ 10 คำค้น ผลลัพธ์ที่ได้

1. ไม่พบข้อมูล 17 คำค้น
2. พบข้อมูล 83 คำค้น คิดเป็นค่าเฉลี่ย 92.83 % ของการค้นหาสินค้าตามที่ต้องการ ดังตารางที่ 3

ตารางที่ 3 ผลการทดสอบค่าค้น
(ข้อมูลวันที่ 23/07/64)

ลำดับ	หมวดหมู่สินค้า	จำนวน เปอร์เซ็นต์ค้นพบ สินค้าตามที่ ต้องการ
1	เครื่องใช้ไฟฟ้าและ อุปกรณ์อิเล็กทรอนิกส์	100.00
2	เครื่องเขียนและ อุปกรณ์สำนักงาน	96.20
3	ผลิตภัณฑ์เพื่อสุขภาพ ความงาม	100.00
4	ผลิตภัณฑ์ทำความสะอาด และของใช้ใน ครัวเรือน	95.41
5	แม่และเด็ก	100.00
6	เครื่องดื่มและขนม ขบเคี้ยว	99.64
7	อาหารแห้ง อาหาร กระป๋อง	99.07
8	อาหารสด อาหารแช่ แข็ง เบเกอรี่	68.03
9	ผลิตภัณฑ์สำหรับสัตว์เลี้ยง	100.00
10	เสื้อผ้าและเครื่องแต่ง กาย	70.00

**ผลการทดสอบเซิร์ฟเวอร์เพื่อดูการรองรับการใช้งาน
จากผู้ใช้งาน**

ปัจจุบันใช้งาน 1 เซิร์ฟเวอร์ (2 cpu ram 4 gb
ssd 80 gb) รองรับการใช้งานได้ 270 ผู้ใช้งานต่อ
วินาที ดังตารางที่ 4

ตารางที่ 4 ผลการทดสอบ load testing

ลำดับ	จำนวน user ต่อ 1 วินาที	ใช้ cpu server (%)	error (%)	error message
1	20	20.03	0.00	
2	40	28.57	0.00	
3	60	39.18	0.00	
4	80	40.82	0.00	
5	100	47.24	0.00	
6	120	50.51	0.00	
7	140	53.81	0.00	
8	160	57.22	0.00	
9	180	64.4	0.00	
10	200	72.28	0.00	
11	270	77.16	0.00	
12	271	79.8	7.01	Connection timed out

สรุปผล

ผลการทดสอบจากผู้ใช้งาน 10 คน โดยแยกเป็น ผู้ใช้งานทั่วไป 5 คน และผู้พัฒนาระบบ 5 คน ผล ปรากฏว่า ผู้ใช้มีความพึงพอใจอยู่ในระดับ “มาก” ($\bar{X} = 4.20$, $S.D. = 0.59$) ส่วนผู้พัฒนาระบบมีความ พึงพอใจในมุมมองการออกแบบและฟังก์ชันการใช้ งานอยู่ในระดับ “มาก” ($\bar{X} = 4.20$, $S.D. = 0.56$)

ตารางที่ 5 ผลจากการประเมินจากผู้ใช้งาน แอปพลิเคชัน

แบบประเมิน	ค่าเฉลี่ย	ค่าเบี่ยงเบน
ผู้ใช้งานทั่วไป	4.20	0.59
ผู้พัฒนาระบบ	4.20	0.56

กิตติกรรมประกาศ

การจัดทำโครงงานฉบับนี้สำเร็จได้ด้วยดีต้อง กราบขอบพระคุณอาจารย์ที่ปรึกษา อาจารย์ ประหยัด เลวัน ที่ได้ให้คำแนะนำ ตรวจสอบโครงงาน เพื่อปรับปรุงแก้ไขข้อบกพร่องต่างๆ และขอ ขอบพระคุณคณาจารย์ทุกท่าน ที่ได้ประสิทธิ์ประสาท วิชาความรู้ เพื่อนำความรู้ที่ได้มาจัดทำโครงงานจน สำเร็จลุล่วงไปด้วยดี ผู้จัดทำจึงขอขอบพระคุณมา ณ โอกาสนี้

เอกสารอ้างอิง

1. STACKPYTHON. (4 ธันวาคม 2563). [ออนไลน์] Web Scraping. สืบค้นจาก <https://stackpython.co/tutorial/web-scraping-python-beautifulsoup-requests> (วันที่สืบค้น 26 กรกฎาคม 2564)

2. AWS. (2564). [ออนไลน์] ElasticSearch. สืบค้นจาก

<https://aws.amazon.com/th/elasticsearch-service/the-elk-stack/what-is-elasticsearch/> (วันที่สืบค้น 26 กรกฎาคม 2564)

3. SSL.com. (13 กรกฎาคม 2563). [ออนไลน์] HTTPS คืออะไร. สืบค้นจาก

<https://www.ssl.com/th/%E0%B8%84%E0%B8%B3%E0%B8%96%E0%B8%B2%E0%B8%A1%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B8%9E%E0%B8%9A%E0%B8%9A%E0%B9%88%E0%B8%AD%E0%B8%A2/%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/> (วันที่สืบค้น 26 กรกฎาคม 2564)

4. AWS. (2564). [ออนไลน์] Docker คืออะไร. สืบค้นจาก

<https://aws.amazon.com/th/docker/> (วันที่สืบค้น 26 กรกฎาคม 2564)

5. Softmelt. (2554). [ออนไลน์] Redis คืออะไร ?. สืบค้นจาก

<https://www.softmelt.com/article.php?id=564> (วันที่สืบค้น 26 กรกฎาคม 2564)

6. Thirawat T. (26 กรกฎาคม 2560). [ออนไลน์] Regular Expressions คืออะไร ?. สืบค้นจาก

https://medium.com/@_trw/regular-expressions-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-2fab4a91ea34 (วันที่สืบค้น 26 กรกฎาคม 2564)