



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบสำรวจความพึงพอใจของลูกค้าอย่างง่าย
Simple Customer Satisfaction System

นายรัชญา กรมแสง

สาขาวิชาวิศวกรรมสารสนเทศ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบสำรวจความพึงพอใจของลูกค้าอย่างง่าย
Simple Customer Satisfaction System

นายธนัชญา กรมแสง

สาขาวิชาวิศวกรรมสารสนเทศ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา ระบบสำรวจความพึงพอใจของลูกค้าอย่างง่าย

ชื่อ-สกุล นักศึกษา นายธนัชญา กรมแสง

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา วิศวกรรมสารสนเทศ

ชื่อ-สกุล อาจารย์นิเทศ ผศ. มยุรี เลิศเวชกุล

ชื่อ-สกุล ผู้นิเทศงาน ภวินท์ คงคาสวรรค์

สถานประกอบการ พรอนโต้ทูล

บทคัดย่อ

ปัจจุบันข้อมูลได้เข้ามาเป็นส่วนสำคัญในการดำเนินธุรกิจ ผลตอบสนองของลูกค้านั้นเป็นหนึ่งในสิ่งจำเป็นอันเปรียบเสมือนมาตรวัด ว่าบริการที่เราให้แก่ผู้บริภคผู้น้อยอยู่ในระดับที่ผู้บริภคพึงพอใจหรือไม่ ระบบการทำแบบสอบถามจึงได้เข้ามาเป็นตัวช่วยในการรับผลตอบสนองของลูกค้า แต่ในปัจจุบันนั้นยังมีรูปแบบการทำแบบสอบถามที่ยุ่งยาก และผู้ใช้ต้องเสียเวลาในการทำแบบสอบถามเป็นเวลายาวนาน ส่งผลให้อาจจะไม่ได้รับผลตอบสนองที่ถูกต้องและแม่นยำ ดังนั้นเพื่อทำให้ผลตอบสนองแม่นยำ และผู้บริภคเกิดความพึงพอใจ โดยงานวิจัยฉบับนี้ผู้วิจัยได้พัฒนาเว็บแอปพลิเคชันที่นำรูปแบบของการสอบถามความพึงพอใจของลูกค้าโดยใช้เทคนิค CSAT Score (Customer Satisfaction) และ NPS (Net Promoter Score) ช่วยทำให้ได้รับผลตอบสนองของผู้บริภคได้ดีกว่าการทำสอบถามที่มีความยุ่งยากต่อผู้ใช้ จากการวิจัยพบว่าภายในระยะเวลาในการดำเนินงาน จำนวนผลตอบรับของผู้บริภคผลิตภัณฑ์มีมากขึ้นอย่างเห็นได้ชัด

คำสำคัญ.: การสอบถามความพึงพอใจของลูกค้า, CSAT, NPS

Cooperative Title: Simple Customer Satisfaction System

Student intern name: Thanatchaya Kromsaeng

Faculty: Engineering **Department:** Computer Engineering (information Engineering)

Advisor name: Asst. Prof. Mayuree Lertwatechakul

Mentor name: Phawin Khongkhasawan

Company: Pronto tools

ABSTRACT

At the present time, information has become the most important part of businesses. Customer feedback is the necessary tools for businesses to measure how well the corporate services serve their customers, and what is the suggestion for improvement. Thus, online customer satisfaction surveying system has become an efficient channel to acquire customers feedback, but most of them do not represent the surveying forms to the customer clearly, the complex questions and ineffective interface may confuse the user and the feedback collecting by using such system may not be so effective. Therefore, this may lead to be inaccurate satisfaction survey. Thus, to improve the accuracy of customer satisfaction survey, the researcher developed a web application with ability to measure and collect CSAT (Customer Satisfaction) score and NPS (Net Promoter) score, which will help collect customers feedback more effective than the general methods. The result from the research shows that the number of customer feedback is greatly improved.

Keywords: Customer Satisfaction Survey, CSAT, NPS

กิตติกรรมประกาศ

ข้าพเจ้าได้มีโอกาสเข้าร่วมโครงการสหกิจศึกษาที่ บริษัท พรอนโต้ทูล ตั้งแต่วันที่ 23 พฤษภาคม 2561 จนถึงวันที่ 21 ธันวาคม 2561 ทำให้ข้าพเจ้าได้รับความรู้และประสบการณ์มากมายจากการทำงาน การเข้าร่วมกิจกรรมต่าง ๆ ภายในองค์กร และได้เรียนรู้ถึงวัฒนธรรมองค์กร ซึ่งเป็นประโยชน์ต่อข้าพเจ้าในการนำไปปรับใช้ในอนาคต

ข้าพเจ้าขอขอบพระคุณ บริษัท พรอนโต้ มาร์เก็ตติ้ง และ บริษัท พรอนโต้ทูล ที่ให้โอกาสข้าพเจ้าได้เข้าร่วมปฏิบัติงาน และ ร่วมเป็นส่วนหนึ่งของบริษัท

ข้าพเจ้าขอขอบพระคุณ คุณกานต์ อู่วิรัช ที่ให้ความรู้ ให้คำปรึกษา และคำแนะนำในการทำงาน ข้าพเจ้าจะนำความรู้ และคำแนะนำต่างๆ ไปปรับใช้ในการทำงาน และการใช้ชีวิตประจำวันต่อไป

ข้าพเจ้าขอขอบพระคุณคุณพ่อ คุณแม่ และน้องสาว ที่คอยให้กำลังใจ กำลังทรัพย์ และเป็นที่ยกย่องให้ข้าพเจ้าในยามลำบาก

ข้าพเจ้าขอขอบพระคุณ ผศ. มยุรี เลิศเวชกุล อาจารย์ที่ปรึกษาที่คอยให้คำปรึกษาช่วยเหลือ และให้คำแนะนำ ทำให้รายงานการวิจัยฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

และท้ายที่สุด ข้าพเจ้าขอขอบพระคุณ คณาจารย์คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ประสิทธิ์ประสาทวิชาความรู้ ทำให้ข้าพเจ้าสามารถนำความรู้ที่ได้รับจากการศึกษานั้นมาใช้ในการศึกษาค้นคว้าครั้งนี้

ธนัญญา กรมแสง

สารบัญ

บทที่	หน้า
บทคัดย่อ (ภาษาไทย).....	I
บทคัดย่อ (ภาษาอังกฤษ).....	II
กิตติกรรมประกาศ.....	II
สารบัญ.....	IV
สารบัญรูป.....	VII
สารบัญตาราง.....	X
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 วิธีการดำเนินการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 Customer Satisfaction Score Metric (CSAT).....	4
2.2 Net Promoter Score Metric (NPS).....	5
2.3 Representational State Transfer API (RESTful API).....	6
2.4 รูปแบบข้อมูลประเภท JSON (Javascript Object Notation).....	7
2.5 HTTP (Hypertext Transfer Protocol).....	7
2.6 ภาษา Python.....	8
2.7 Django ORM Framework.....	10

สารบัญ (ต่อ)

บทที่	หน้า
2.8 JavaScript.....	11
2.9 React Framework	14
2.10 Robotframework	16
2.11 Celery (Asynchronous task queue).....	16
2.12 Docker (Software).....	17
2.13 Amazon web services EC2.....	17
2.14 Amazon web services RDS.....	18
2.15 Amazon ElastiCache.....	18
2.16 CircleCI.....	19
2.17 Ansible	20
2.18 Cloudflare.....	21
2.19 PostgreSQL.....	22
2.20 Scrum Framework.....	22
2.21 Agile Methodology.....	23
2.22 Base64 Encryption	24
2.23 Vim (Text Editor).....	24
2.24 Visual Studio Code.....	25
2.25 Amazon web service EFS	25
2.26 Sendgrid	25
บทที่ 3 วิธีการดำเนินการวิจัย	26
3.1 รวบรวมความต้องการต่างๆ.....	27

สารบัญ (ต่อ)

บทที่	หน้า
3.2 วางแผนและออกแบบการพัฒนาโปรแกรม	29
3.3 เริ่มพัฒนาโปรแกรม	43
3.4 ทดสอบการใช้งานของโปรแกรมจริง	43
3.5 นำโปรแกรมไปใช้งานจริง	43
บทที่ 4 ผลการวิจัย	44
4.1 การประเมินเชิงประสิทธิภาพ.....	44
4.2 การประเมินผลเชิงประสิทธิผล.....	49
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	52
5.1 สรุปผลการวิจัย.....	52
5.2 ข้อเสนอแนะ.....	53
เอกสารอ้างอิง.....	54

สารบัญรูป

	หน้า
รูปที่ 2.1 CSAT Satisfaction Scale	4
รูปที่ 2.2 Net Promoter Score with 0-10 Scale	5
รูปที่ 2.3 RESTful API.....	6
รูปที่ 2.4 รูปแบบข้อมูลประเภท JSON	7
รูปที่ 2.5 ตัวอย่างโครงสร้างของ Django Framework.....	10
รูปที่ 2.6 Lifecycle ของ React เวอร์ชันมากกว่า 16.3.....	15
รูปที่ 2.7 Celery Logo.....	16
รูปที่ 2.8 Containers versus Virtual Machines	17
รูปที่ 2.9 Amazon web services EC2.....	18
รูปที่ 2.10 Ansible Logo.....	20
รูปที่ 2.11 Cloudflare Logo	21
รูปที่ 2.12 Postgresql Logo	22
รูปที่ 2.13 Visual Studio Code.....	25
รูปที่ 3.1 ผู้ใช้งานทำการสร้างและทำการส่งแบบสอบถาม.....	27
รูปที่ 3.2 ลูกค้ำของผู้ใช้งานทำการตอบแบบสอบถาม	28
รูปที่ 3.3 หน้าต่างลงทะเบียนใช้งานแอปพลิเคชัน	30
รูปที่ 3.4 หน้าต่างสำหรับเข้าใช้งานแอปพลิเคชัน.....	30
รูปที่ 3.5 หน้า Account Settings.....	31
รูปที่ 3.6 หน้าต่างสำหรับเชิญสมาชิก	32
รูปที่ 3.7 Modal สำหรับเชิญสมาชิก	32
รูปที่ 3.8 Invitation to Company flow.....	33

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.9 Embedded Code.....	33
รูปที่ 3.10 Dashboard.....	34
รูปที่ 3.11 Export Feedback Button.....	35
รูปที่ 3.12 Export Feedback Workflow.....	35
รูปที่ 3.13 NPS Email Setting.....	36
รูปที่ 3.14 NPS Delivery Setting.....	37
รูปที่ 3.15 NPS Delivery Setting.....	37
รูปที่ 3.16 Create NPS Survey Flow.....	38
รูปที่ 3.17 NPS Insight pages.....	39
รูปที่ 3.18 CSAT Insight pages.....	39
รูปที่ 3.19 NPS Survey Flow.....	40
รูปที่ 3.20 Feedback filter.....	40
รูปที่ 3.21 Dropbox Integration Flow.....	41
รูปที่ 3.22 Publish Testimonial Example.....	41
รูปที่ 3.23 Publish Testimonial Code.....	42
รูปที่ 4.1 Dashboard.....	44
รูปที่ 4.2 Create New Survey.....	45
รูปที่ 4.3 Survey Setting.....	45
รูปที่ 4.4 Survey Embed Setting.....	46
รูปที่ 4.5 Generate Embed Code.....	46
รูปที่ 4.6 Send email using embed code.....	47

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.7 Customer rating.....	47
รูปที่ 4.8 Customer Confirmation	48
รูปที่ 4.9 Call to action.....	48
รูปที่ 4.10 Insight Dashboard Graph.....	49
รูปที่ 4.11 Survey Helpdesk Integration	50
รูปที่ 4.12 Helpdesk Customer Integration.....	50



สารบัญตาราง

	หน้า
ตารางที่ 2.1 Built-in types in Python	9
ตารางที่ 2.2 Reserved Words in Python	10
ตารางที่ 2.3 Reserved Words in JavaScript	13
ตารางที่ 2.4 ตัวอย่างไวยากรณ์ของ Robotframework	16
ตารางที่ 2.5 Base64 Encoding Table	24
ตารางที่ 3.1 แผนการวิจัยระหว่างการทำสหกิจศึกษา	26
ตารางที่ 3.2 แผนงานครึ่งแรกสหกิจศึกษา	29
ตารางที่ 3.3 แผนงานครึ่งหลังสหกิจศึกษา	29



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

เนื่องจากบริษัท พรอนโต้ มาร์เก็ตติ้ง (Pronto marketing) เป็นบริษัทที่ให้บริการด้านอินเทอร์เน็ตมาร์เก็ตติ้งสำหรับธุรกิจขนาดกลางและขนาดเล็ก ดังนั้นการได้รับผลตอบรับจากลูกค้าเป็นสิ่งสำคัญ ทำให้รู้ทิศทางของบริษัท

การเข้าร่วมโครงการสหกิจศึกษาครั้งนี้ผู้วิจัยได้เข้าร่วมใน คณะทำงานซิมเปิลแซท (SimpleSAT) ของบริษัท พรอนโต้ทูล (Pronto tools) ซึ่งเป็นส่วนหนึ่งของ บริษัท พรอนโต้ มาร์เก็ตติ้ง ซึ่งมีหน้าที่ในการพัฒนาแอปพลิเคชัน และ สนับสนุนทางเทคนิคให้กับ บริษัท พรอนโต้มาร์เก็ตติ้ง ซึ่งในแต่ละวัน บริษัท

พรอนโต้มาร์เก็ตติ้ง มีลูกค้าจำนวนมาก อาทิเช่น คณะทำงานของบริษัท พรอนโต้ มาร์เก็ตติ้ง, ลูกค้าของบริษัท พรอนโต้ มาร์เก็ตติ้ง และ ลูกค้าของลูกค้า บริษัท พรอนโต้ มาร์เก็ตติ้ง ทำให้ไม่อาจรับรู้ถึงความต้องการและความพึงพอใจของลูกค้าทุกคนได้อย่างทั่วถึง และถึงแม้จะมีแอปพลิเคชันที่ใช้ในการทำแบบสำรวจความพึงพอใจอยู่แล้ว แต่ระบบดังกล่าว มีความซับซ้อน ทำให้ใช้งานยาก

จากปัญหาข้างต้นจึงเกิดการพัฒนาแอปพลิเคชันที่ พัฒนาเพื่อให้เกิดความง่ายในขั้นตอนการทำแบบสำรวจความพึงพอใจ (Satisfaction Survey) โดยผู้ที่ใช้งานสามารถทำการสร้างแบบสำรวจความพึงพอใจ สำหรับวัดความพึงพอใจของลูกค้า และนำมาวิเคราะห์ทิศทางของบริษัท จากข้างต้นทำให้ผู้ใช้งานทำแบบสำรวจความพึงพอใจ และ ใช้แบบสำรวจความพึงพอใจอย่างง่ายตายยิ่งขึ้น

1.2 วัตถุประสงค์ของการวิจัย

- เพื่อให้การทำแบบสำรวจความพึงพอใจดีขึ้นกว่าปัจจุบัน
- เพื่อให้รับรู้ถึง ผลตอบกลับ (Feedback) ของลูกค้าได้ไวขึ้น
- เพื่อให้ลูกค้าได้รับบริการที่ดีขึ้น

1.3 ขอบเขตของการวิจัย

- สามารถสร้างบัญชีในรูปแบบของบริษัทและตัวบุคคลได้
- สามารถสร้างแบบสำรวจความพึงพอใจได้
- แบบสำรวจความพึงพอใจสามารถแสดงผลได้จากหน้าเว็บไซต์
- สามารถได้รับจดหมายอิเล็กทรอนิกส์ทุกครั้งที่ถูกค่าให้ผลตอบกลับ

1.4 วิธีการดำเนินการวิจัย

- รวบรวมความต้องการต่าง ๆ
- วางแผนและออกแบบการพัฒนาโปรแกรม
- เริ่มพัฒนาโปรแกรม
- ทดสอบการใช้งานของโปรแกรม
- นำโปรแกรมไปใช้งานจริง (ภายในบริษัท)
- นำโปรแกรมไปใช้งานจริง (ภายนอกบริษัท)

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการเข้าร่วมโครงการสหกิจศึกษากับ บริษัท พรอนโต้ทูล
จำแนกออกเป็น 2 ส่วนคือ

1.5.1 ประโยชน์ต่อบริษัท

- บริษัทรับรู้ถึงผลตอบกลับจากลูกค้าได้เร็วยิ่งขึ้น
- มีข้อมูลของลูกค้าที่ใช้นำมาวิเคราะห์แผนการตลาดต่อไปในอนาคต
- มีแอปพลิเคชันที่เชื่อมต่อกับระบบหลักของบริษัท พรอนโต้ มาร์เก็ตติ้ง

1.5.2 ประโยชน์ต่อผู้วิจัย

- ได้เรียนรู้วัฒนธรรมขององค์กร และสภาพแวดล้อมในการทำงาน
- ได้เรียนรู้และทำงานจริงในสภาพแวดล้อมบริษัท
- ได้พัฒนาทักษะการสื่อสารภาษาอังกฤษอย่างมีประสิทธิภาพ
- ได้เรียนรู้การรับมือกับปัญหา และการแก้ไขปัญหา
- สามารถเรียงลำดับความสำคัญในการทำงาน
- สามารถทำความเข้าใจ และทักษะต่าง ๆ ไปใช้ในการทำงานภายใต้ภาคหน้าได้
- ได้พัฒนาทักษะในด้านกรเขียนโปรแกรม
- ได้เรียนรู้การทำงานเกี่ยวกับด้าน Infrastructure Operation

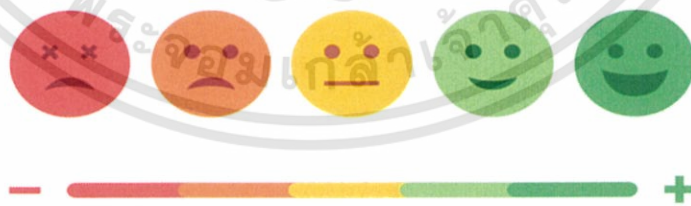
บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื่องจากการพัฒนาเว็บไซต์สำหรับการทำระบบสำรวจความพึงพอใจลูกค้า เป็นการพัฒนาเว็บไซต์ซึ่งจำเป็นจะต้องมีทั้งส่วนหน้า (Frontend) เป็นส่วนที่ลูกค้าจะได้เห็น และส่วนหลัง (Backend) ที่คอยเก็บข้อมูลที่ใช้ในการแสดงผลและคำนวณ และมี โครงสร้าง (Infrastructure) ที่รองรับส่วนหน้าและส่วนหลังให้ทำงานได้เป็นปกติ ดังนั้นผู้วิจัยจึง ต้องศึกษาทฤษฎีต่าง ๆ เพื่อที่จะนำความรู้เหล่านี้มาประกอบรวมกันเพื่อให้ทำงานวิจัยสำเร็จ ลุล่วงไปได้ด้วยดี โดยแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องกับงานวิจัยชิ้นนี้ มีดังต่อไปนี้

2.1 Customer Satisfaction Score Metric (CSAT)

Customer Satisfaction Score เป็นหนึ่งในวิธีการ (Methodology) ในการวัดระดับความพึงพอใจของลูกค้าหลังจากที่สำเร็จบริการแล้ว โดยส่วนใหญ่มักจะมีคำถามที่ถามถึงความพึงพอใจกับบริการที่ได้รับไปหรือไม่ โดยลูกค้าหรือผู้ได้รับบริการจะสามารถตอบคำถามนี้ได้ในรูปแบบของอัตราความพึงพอใจต่อบริการตั้งแต่ 1 – 5 โดยเรียงตามลำดับความพึงพอใจดังรูปที่ 2.1



รูปที่ 2.1 CSAT Satisfaction Scale
(ที่มา: <https://www.niceincontact.com/>)

- 1) ไม่พึงพอใจมาก ๆ (Very unsatisfied)
- 2) ไม่พึงพอใจ (Unsatisfied)
- 3) เป็นกลาง (Neutral)
- 4) พึงพอใจ (Very unsatisfied)
- 5) พึงพอใจมาก ๆ (Very satisfied)

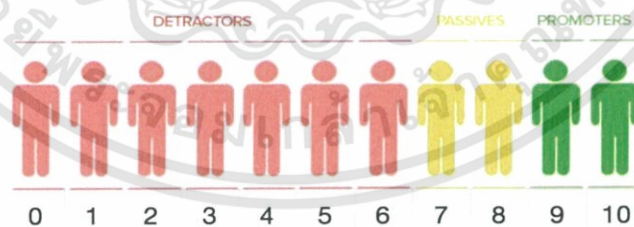
จากหลักการข้างต้นสามารถนำมาสร้างเป็นสูตรสมการสำหรับการคำนวณความพึงพอใจของลูกค้าต่อบริการที่มีได้ดังนี้

$$\frac{\text{Number of satisfied customers}}{\text{Number of survey responses}} \times 100 = \% \text{ of satisfied customers}$$

2.2 Net Promoter Score Metric (NPS)

Net Promoter Score เป็นรูปแบบคำถามที่ใช้ถามเพื่อวัดระดับความพึงพอใจต่อองค์กรและความเต็มใจที่จะแนะนำบริษัทหรือองค์กรต่อคนใกล้ชิด โดยส่วนใหญ่รูปแบบคำถามของ Net Promoter Score จะเป็นคำถามสเกลตั้งแต่ 0-10 ดังรูปที่ 2.2 หรือ 0-100 โดยจะเลือกแบบใดขึ้นอยู่กับความละเอียดของผลตอบรับที่อยากได้ โดยแบ่งบุคคลที่ทำแบบสอบถามเป็น 3 ประเภทได้แก่

- 1) ผู้สนับสนุน (Promoters)
- 2) ผู้เป็นกลาง (Passives)
- 3) ผู้ไม่สนับสนุน (Detractors)



รูปที่ 2.2 Net Promoter Score with 0-10 Scale

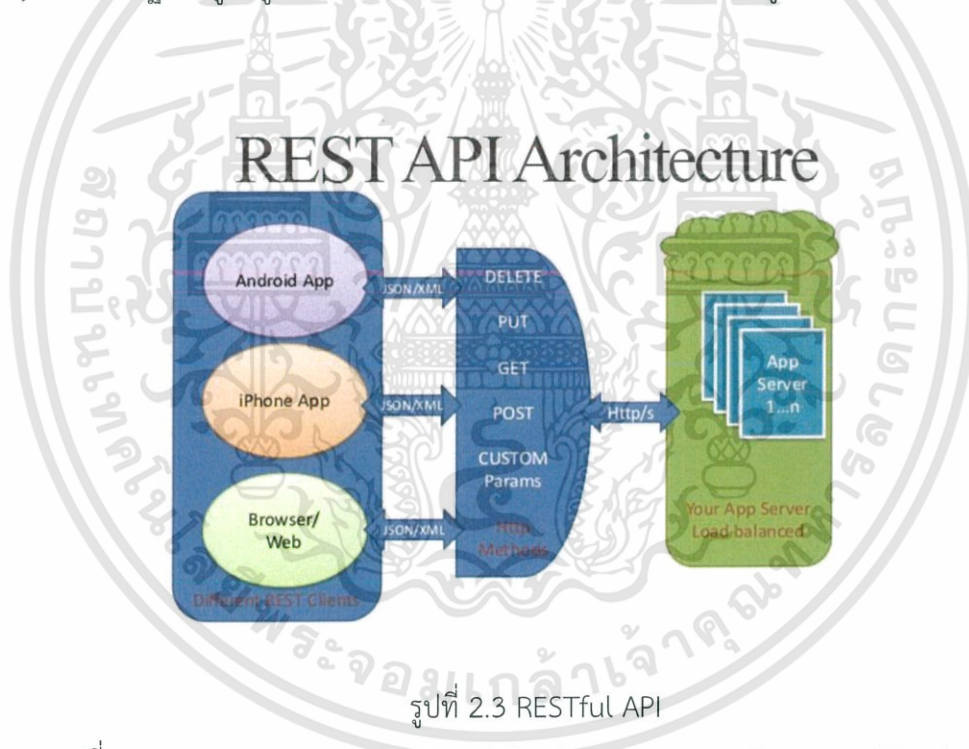
(ที่มา: <https://www.questionpro.com/blog/net-promoter-score-system/>)

จากหลักการข้างต้นสามารถนำมาสร้างเป็นสูตรสมการสำหรับคำนวณหาเปอร์เซ็นต์ความภักดีต่อแบรนด์เนม (Brand name) หรือ องค์กรได้ดังนี้

$$\text{Percent of Promoters} - \text{Percent of Detractors} = \text{Net Promoter Score}$$

2.3 Representational State Transfer API (RESTful API)

RESTful API เป็นสถาปัตยกรรมซอฟต์แวร์ที่กำหนดรูปแบบสำหรับการสร้างและสื่อสารระหว่างเว็บเซอร์วิสโดยจะอาศัยการรับข้อมูลผ่าน URI/URL และส่งข้อมูลกลับไปในรูปแบบ XML, HTML, JSON โดย RESTful API ทำงานบน HTTP protocol มีกระบวนการในการจัดการข้อมูล (Payload) พื้นฐานอยู่ 4 รูปแบบได้แก่ GET, POST, PUT, DELETE ดังรูปที่ 2.3

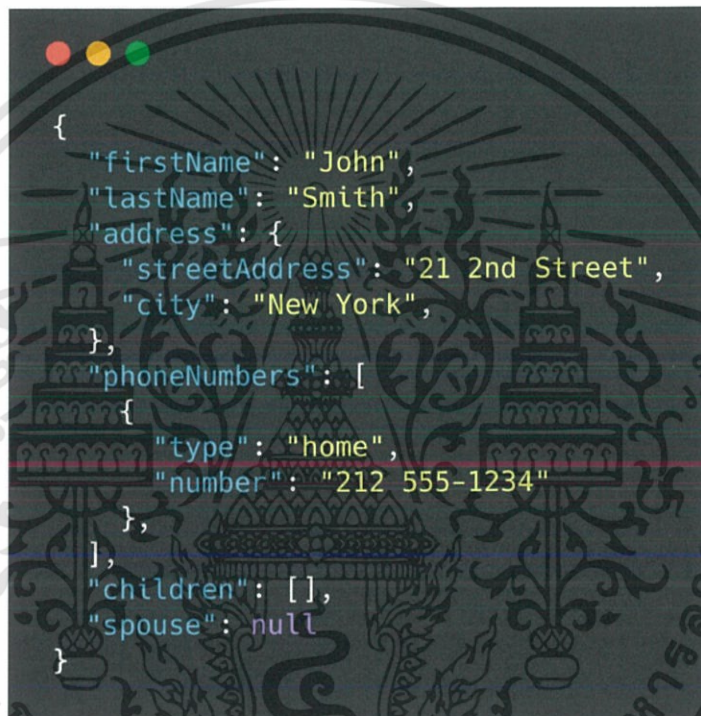


รูปที่ 2.3 RESTful API

(ที่มา: <https://shareurcodes.com/blog/creating a simple rest api in php/>)

2.4 รูปแบบข้อมูลประเภท JSON (Javascript Object Notation)

JSON (Javascript Object Notation) คือรูปแบบของข้อมูลที่ใช้สำหรับแลกเปลี่ยนข้อมูลที่มีขนาดเล็ก สามารถถูกอ่านและสร้างได้ง่ายเพราะมีรูปแบบที่ตายตัว โดยจะทำการส่งข้อมูลในรูปแบบของ อ็อบเจกต์ข้อมูล (Data Object) ซึ่งประกอบด้วย คุณสมบัติ - ค่า (Attribute-Value pair) และ ข้อมูลในรูปแบบ อาร์เรย์ (Array) ปัจจุบันถือเป็นมาตรฐานในการรับส่งข้อมูลผ่าน เอชทีทีพี ตัวอย่างข้อมูลรูปแบบ JSON ดังรูปที่ 2.4

A screenshot of a JSON object displayed in a dark-themed code editor. The JSON object is a root object with several properties: 'firstName' (John), 'lastName' (Smith), 'address' (an object with 'streetAddress' and 'city'), 'phoneNumbers' (an array of objects with 'type' and 'number'), 'children' (an empty array), and 'spouse' (null). The code is color-coded: strings are yellow, numbers are green, and null is white. The background features a faint watermark of a university seal.

```
{
  "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
  ],
  "children": [],
  "spouse": null
}
```

รูปที่ 2.4 รูปแบบข้อมูลประเภท JSON
(ที่มา: <https://carbon.now.sh/>)

2.5 HTTP (Hypertext Transfer Protocol)

เอชทีทีพีเป็นมาตรฐานในการร้องขอและการตอบรับระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่าย ซึ่งเครื่องลูกข่ายคือผู้ใช้ปลายทาง (End-user) และเครื่องแม่ข่ายคือเว็บไซต์ เครื่องลูกข่ายจะสร้างการร้องขอเอชทีทีพีผ่านทางเว็บเบราว์เซอร์ เว็บครอว์เลอร์ หรือเครื่องมืออื่น ๆ ที่จัดว่าเป็น ตัวแทนผู้ใช้ (User agent) ส่วนเครื่องแม่ข่ายที่ตอบรับ ซึ่งเก็บบันทึกหรือสร้าง ทรัพยากร (Resource)

อย่างเช่นไฟล์เอชทีเอ็มแอลหรือรูปภาพ จะเรียกว่า เครื่องให้บริการต้นทาง (Origin server) ในระหว่างตัวแทนผู้ใช้กับเครื่องให้บริการต้นทางอาจมีสื่อกลางหลายชนิด อาทิพร็อกซี เกตเวย์

2.6 ภาษา Python

ภาษา Python เป็นภาษาโปรแกรมมิ่งระดับสูงที่รองรับ multi-paradigm programming language. เช่น Object-oriented programming และ Structured programming โดย ภาษา Python นั้นใช้ช่องว่าง (Whitespace) ในการแบ่งแยกขอบเขตของการทำงาน โดยตัวภาษามีรายละเอียดดังนี้

2.6.1 Statements and control flow

- 1) ภาษา Python ใช้การประกาศตัวแปรโดยไม่ต้องระบุรูปแบบ (Type) โดยตัวภาษาจะรู้ได้เองว่าจะกำหนดให้ตัวแปรนั้นเป็นรูปแบบใด เช่น `name = 'Foo'` เป็นรูปแบบประเภท อักขระ (String) และ `day_in_month = 30` เป็นรูปแบบประเภท ตัวเลข
- 2) if statement ในภาษา Python นั้นใช้ในการกระทำบางอย่างแบบมีเงื่อนไข โดยใช้คู่กับ else และ elif เช่น

```
if day_in_month > 29:
    do_this()
elif day_in_month < 35:
    do_those()
else:
    do_that()
```

- 3) for statement ในภาษา Python นั้นใช้สำหรับวนซ้ำวัตถุทำได้ดังนี้

```
for foo in foobars:
    print(foo)
```

- 4) while statement ในภาษา Python จะกระทำบางอย่างตราบเท่าที่เงื่อนไข เป็นจริงทำได้ดังนี้

```
while true:
    print("Hello")
```

2.6.2 Method

ภาษา Python เรียก function ที่มีความสัมพันธ์กับ Object ว่า method โดยวิธีการสร้าง method ทำได้ดังนี้

```
def hello_world(obj):  
    print(obj)
```

2.6.3 Built-in types in Python

ตารางที่ 2.1 Built-in types in Python

Type	mutable	Description
bool	immutable	Boolean value
bytearray	mutable	Sequence of bytes
bytes	immutable	Sequence of bytes
complex	immutable	Complex number with real and imaginary parts
dict	mutable	Associative array (or dictionary) of key and value pairs; can contain mixed types (keys and values), keys must be a hashable type
ellipsis		An ellipsis placeholder to be used as an index in NumPy arrays
float	immutable	Floating point number, system-defined precision
frozenset	immutable	Unordered set, contains no duplicates; can contain mixed types, if hashable
int	immutable	Integer of unlimited magnitude
list	mutable	List, can contain mixed types
set	mutable	Unordered set, contains no duplicates; can contain mixed types, if hashable
str	immutable	A character string: sequence of Unicode codepoints
tuple	immutable	Can contain mixed types

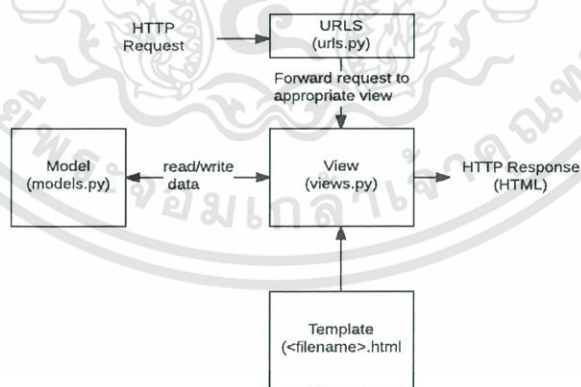
2.6.4 Reserved Words in Python

ตารางที่ 2.2 Reserved Words in Python

False	class	finally	Is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

2.7 Django ORM Framework

Django Framework เป็น web framework ที่ออกแบบตามสถาปัตยกรรม model-view-template (MVT) และ ถูกจัดอยู่ในรูปแบบของ Opinionated frameworks โดย Django นั้นทำให้การสร้างเว็บไซต์ทำได้อย่างรวดเร็ว โดยมีโครงสร้างของการทำงาน และ ตัวอย่างโครงสร้างพื้นฐานดังรูปที่ 2.5 ดังนี้



รูปที่ 2.5 ตัวอย่างโครงสร้างของ Django Framework

(ที่มา <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>)

2.7.1 Django Urls

Urls mapper ใช้สำหรับการ redirect HTTP requests ไปยัง views หรือชุดของคำสั่ง ที่ใช้ในการจัดการข้อมูลที่ส่งเข้ามา โดย Urls mapper นั้นสามารถตรงกับรูปแบบ (Regular expression) หรือ ตรงกับคำสำคัญ (Keyword)

2.7.2 Django View

Views เป็นส่วนที่รับ HTTP requests และส่งคืน HTTP responses. โดย Views นั้นจะ ทำการดึงข้อมูลที่เหมาะสมกับกับคำร้องขอผ่านทางโมเดล (Models) และนำไปแสดงผลบนส่วน แสดงผลที่เรียกว่า Template

2.7.3 Django Models

Models คือ Python objects ที่ใช้เป็นตัวแทนของโครงสร้างแอปพลิเคชัน และมีความสามารถในการจัดการ เพิ่ม, เปลี่ยนแปลง, ลบ และ ขอข้อมูลจากฐานข้อมูล

2.7.4 Django Templates

Templates เป็นส่วนสำหรับกำหนดโครงสร้าง หรือ รูปร่างของไฟล์เช่น HTML page โดย จะมีส่วนรองรับการแสดงผลข้อมูล โดย View นั้นสามารถแสดงผลได้โดยการดึงข้อมูล HTML จาก Template และ แสดงผลข้อมูลจาก Model

2.8 JavaScript

JavaScript เป็นภาษาสคริปต์เชิงวัตถุหรือเรียกว่าอ็อบเจกต์โอเรียนเตด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ตสำหรับผู้เขียนเอกสารด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ทำงานร่วมกับภาษา HTML และภาษาอื่น ๆ ได้ทั้งทางฝั่งลูกข่าย (Client) และทางฝั่งแม่ข่าย (Server) โดยมีลักษณะการทำงานดังนี้

2.8.1 คำสั่งการเขียนสคริปต์ของ JavaScript

เริ่มต้นด้วยการกำหนดขอบเขตของการเขียนภาษา JavaScript ภายใน HTML

```
<script language="JavaScript">  
  
</script>
```

2.8.2 การประกาศตัวแปร (Declaring variables)

ในภาษา JavaScript การประกาศตัวแปร จะทำทั้งหมด 2 ขั้นตอนคือ

- 1) Declaration คือการประกาศชื่อตัวแปรเช่น var foo, let bar, const foobar
- 2) Assignment คือการกำหนดค่าเช่น const foobar = 'Hello';

ในกรณีที่ต้องการกำหนดตัวแปรหลายตัวในบรรทัดเดียวกันให้ใช้เครื่องหมายจุลภาค (,) คั่นระหว่างชื่อตัวแปรและปิดท้ายด้วยเครื่องหมายอัฒภาค (;)

2.8.3 การสร้าง function

การสร้างฟังก์ชัน เป็นการประกาศขอบเขตของการเขียนคำสั่ง โดยการเขียนไวยากรณ์สำหรับสร้างฟังก์ชันนั้นมีดังนี้

- 1) function foo คือชื่อของฟังก์ชัน การกำหนดจะต้องไม่ซ้ำกับฟังก์ชันอื่น
- 2) parameter คือชื่อของพารามิเตอร์อ้างอิง (Argument) ส่งผ่านไปยังฟังก์ชันต้องระบุอยู่ภายในเครื่องหมายวงเล็บเท่านั้นโดยจะมีพารามิเตอร์เพียงตัวเดียวหลายตัวหรือไม่มีเลยกรณีที่มี พารามิเตอร์หลาย ๆ ตัว แต่ละตัวจะต้องเขียนแยกออกจากกันด้วยเครื่องหมายจุลภาค (,)
- 3) statements กลุ่มของโค้ดที่ทำงานภายในฟังก์ชัน

2.8.4 ชนิดข้อมูลของตัวแปร

- 1) Number หมายถึงข้อมูลชนิดตัวเลขประกอบด้วยจำนวนเต็ม (Integer) และเลขจำนวนจริง (Float)

- 2) Logical หมายถึงข้อมูลทางตรรกะมี 2 สถานะคือจริงและเท็จ
- 3) String หมายถึงข้อมูลที่เป็นข้อความซึ่งจะต้องกำหนดไว้ในเครื่องหมายอัญอิง เช่น “Hello world”
- 4) Null หมายถึงข้อมูลที่ไม่มีค่าข้อมูลใด ๆ ซึ่งค่า Null ใช้สำหรับการยกเลิกพื้นที่เก็บค่าของตัวแปรออกจากหน่วยความจำ
- 5) Undefined หมายถึงข้อมูลที่ยังไม่มีการกำหนดค่าใด ๆ

2.8.5 Reserved Words

ตารางที่ 2.3 Reserved Words in JavaScript

abstract	else	instanceof	super
boolean	enum	int	switch
break	export	interface	synchronized
byte	extends	let	this
case	false	long	throw
catch	final	native	throws
char	finally	new	transient
class	float	null	true
const	for	package	try
continue	function	private	typeof
debugger	goto	protected	var
default	if	public	void
delete	implements	return	volatile
do	import	short	while
double	in	static	with

2.9 React Framework

React เป็น JavaScript library สำหรับสร้างส่วนต่อประสานกับผู้ใช้ (User interface) โดย React เปรียบเสมือนส่วนที่มองเห็นสำหรับการสร้างเว็บแอปพลิเคชัน โดย React นั้นจะมี Virtual DOM ที่ทำหน้าที่ในการอัปเดตหน้าตาของเว็บไซต์เฉพาะส่วนที่ถูกเปลี่ยนแปลง

2.9.1 Components

Components คือส่วนที่ใช้อธิบายรูปร่างหน้าตาของเว็บไซต์ โดยจะถูกแบ่งเป็นส่วนย่อย ๆ มาประกอบรวมกัน โดยจะใช้ JSX หรือภาษา JavaScript ที่มี HTML เป็นส่วนในการเขียนอธิบาย รูปร่างหน้าตาแทนการเขียนด้วย JavaScript ปกติ

2.9.2 Props หรือ Properties

Props หรือ Properties เป็นส่วนที่ทำให้สามารถส่งข้อมูลจาก Component ไปยัง Components อื่น ๆ ได้ เช่น

```
class ParentComponent extends React.Component {  
  render() {  
    return <ChildComponent message="Hello World"/>;  
  }  
}  
class ChildComponent extends React.Component {  
  render() {  
    return <p>And then I said, "{this.props.message}"</p>;  
  }  
}
```

ParentComponent ทำการส่ง Props ที่ชื่อว่า message="Hello World" ไปให้กับ ChildComponent เพื่อทำการแสดงผล ผลลัพธ์ที่ได้คือ ChildComponent จะแสดงคำว่า 'And then I said, "Hello World"'

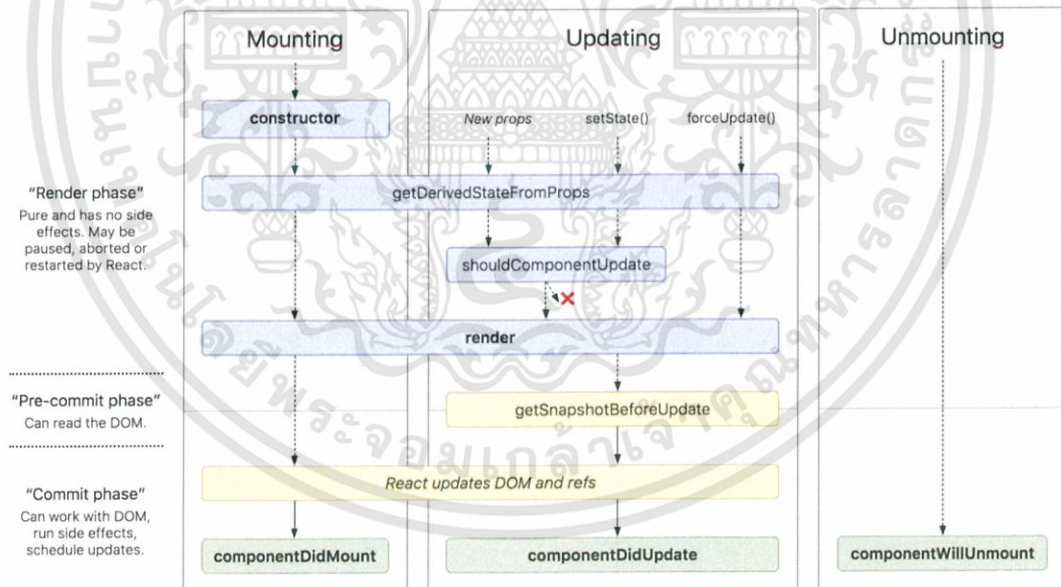
2.9.3 States

States คือส่วนที่ถือครองข้อมูลสำหรับใช้ภายใน Component และสามารถส่งข้อมูล ไปใช้ต่อใน Components อื่น ๆ ได้ผ่าน Props เช่น

```
class ParentComponent extends React.Component {
  state = { color: 'green' };
  render() {
    return (
      <ChildComponent color={this.state.color} />
    );
  }
}
```

2.9.4 Lifecycle methods

Lifecycle methods เปรียบเสมือน Hooks ที่ไว้คอยตรวจสอบการเปลี่ยนแปลงของ Component, State และ Props เมื่อพบว่ามีเปลี่ยนแปลงก็จะทำการ Render Component ใหม่พร้อมกับ State และ Props ที่ได้รับเข้ามาใหม่ ตามดังรูปที่ 2.6



รูปที่ 2.6 Lifecycle ของ React เวอร์ชันสูงกว่า 16.3

(ที่มา https://twitter.com/dan_abramov/status/981712092611989509)

2.10 Robotframework

Robotframework เป็น test automation framework ที่ถูกเขียนอยู่บนภาษา Python และมีไวยากรณ์เป็นของตัวเอง โดย test cases จะเขียนโดยใช้รูปแบบของตาราง สามารถเขียนโดยใช้ plain text, hypertext markup language (HTML), tab-separated values (TSV) หรือ reStructuredText (reST) เช่น

ตารางที่ 2.4 ตัวอย่างไวยากรณ์ของ Robotframework

Test Case	Action	Argument	Argument
Demo	Open Browser	https://www.google.com	ie
	Input Text	id=lst-ib	Hollywood Celebrities
	Click Button	Google Search	

2.11 Celery (Asynchronous task queue)

Celery เป็นส่วนเสริมสำหรับการเขียนโปรแกรมในรูปแบบ asynchronous task queue ทำให้ข้อมูลที่ต้องทำการประมวลผลเป็นระยะเวลาสั้น และไม่ต้องการผลลัพธ์ทันที เช่นการนำข้อมูลจากฐานข้อมูลทั้งหมดส่งอีเมล โดยการใช้ Celery นั้นจะสามารถลดภาระของระบบที่ต้องทำการประมวลผลข้อมูล โดยเปลี่ยนไปใช้ Worker Node สำหรับทำการประมวลผลข้อมูลที่ส่งมาจาก Celery โดยตรงแทน

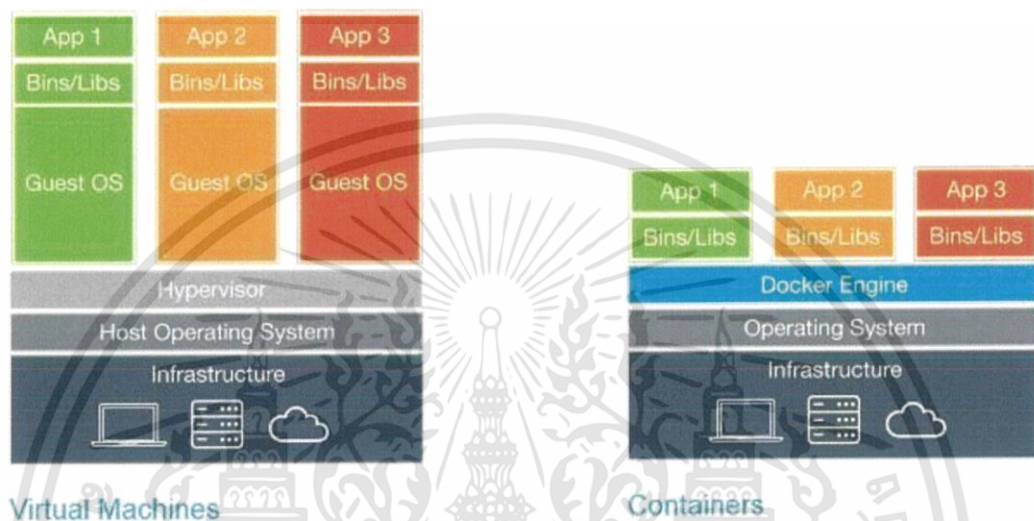


รูปที่ 2.7 Celery Logo

(ที่มา [https://en.wikipedia.org/wiki/Celery_\(software\)](https://en.wikipedia.org/wiki/Celery_(software)))

2.12 Docker (Software)

Docker เป็นโปรแกรมสำหรับการทำการจำลองระดับ Operating system หรืออีกชื่อหนึ่งเรียกว่า Containerization โดยการใช้ Docker นั้นเปรียบเสมือนการสร้าง Environment (สภาพแวดล้อม) เฉพาะให้กับโปรแกรมที่ต้องการ ทำงานได้โดยไม่รบกวนกับ ระบบภายนอก โดยโครงสร้างของ Docker จะแตกต่างจากการใช้ Virtual machines ดังรูปที่ 2.8



รูปที่ 2.8 Containers versus Virtual Machines
(ที่มา <https://www.serverpronto.com>)

2.13 Amazon web services EC2

EC2 หรือ Amazon Elastic Compute Cloud เป็นบริการสำหรับการทำการประมวลผล โดยการเช่าระบบที่ทาง Amazon เป็นผู้ดูแล โดยภายในระบบจะมีส่วนสำคัญเช่น

- 1) Instance หน้าต่างสำหรับการควบคุมสถานะของระบบประมวลผลภายใน
- 2) Network Security ทำหน้าที่จัดการเรื่องการแบ่งแยกของ Network สำหรับแต่ละ Instance
- 3) Load Balancers หน้าต่างสำหรับควบคุมส่วนที่ทำการติดต่อจากภายนอกของ Amazon web services เข้าสู่ภายในระบบเพื่อทำการเชื่อมต่อกับ Instance

หน้าต่างสำหรับควบคุม Amazon web services EC2 ดังรูปที่ 2.9

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
		t2.large	ap-southeast-1a	stopped		None	
		t2.small	ap-southeast-1a	stopped		None	
		t2.micro	ap-southeast-1b	stopped		None	
		t2.medium	ap-southeast-1a	running	2/2 checks ...	None	
		t2.micro	ap-southeast-1a	stopped		None	
		t2.large	ap-southeast-1b	stopped		None	
		t2.small	ap-southeast-1a	running	2/2 checks ...	OK	
		t3.small	ap-southeast-1c	running	2/2 checks ...	None	
		t2.small	ap-southeast-1b	stopped		None	

รูปที่ 2.9 Amazon web services EC2

2.14 Amazon web services RDS

RDS หรือ Amazon Relational Database Service เป็นบริการสำหรับการทำ Scalable relational database ภายใต้ Amazon web services โดยจะมีฐานข้อมูลให้เลือกหลายรูปแบบ เช่น Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database และ SQL Server ภายในระบบของ RDS จะมีความสามารถในเรื่องการทำสำเนา (replicate) ฐานข้อมูลเพื่อให้ฐานข้อมูลนั้นไม่เกิดการสูญหายของข้อมูลสำคัญ และสามารถสั่งการเพิ่มฐานข้อมูลตามการใช้งานได้

2.15 Amazon ElastiCache

Amazon ElastiCache เป็นบริการสำหรับการใช้งาน Redis และ Memcached ซึ่งเป็นโปรแกรมที่มีหน้าที่เก็บข้อมูลภายในเมมโมรี่ ในรูปแบบของ Key-value ทำให้การดึงข้อมูลจากฐานข้อมูลส่วนนี้ทำได้รวดเร็ว และ ช่วยลดการทำงานจากการดึงข้อมูลจาก Disk-based Database เช่น MySQL, PostgreSQL

2.16 CircleCI

CircleCI เป็นบริการสำหรับการทำ CI/CD (continuous integration and continuous delivery) โดย CI/CD นั้นหมายถึงการทำสร้างและการปล่อย (Produce and Deliver) ภายในวงจรสั้น ๆ โดยจะต้องมั่นใจว่า Software นั้นมีเสถียรภาพมากพอที่จะ Deliver ได้ในทุกเวลาสำหรับการทำ CI/CD โดยทั่วไปมีทั้งหมด 6 ขั้นตอน

- 1) Lint (เพื่อทำการตรวจสอบความถูกต้องของไวยากรณ์ภายในโค้ด)
- 2) Unittest (เพื่อทำการตรวจสอบความถูกต้องของการทำงานภายในโปรแกรม)
- 3) Integration Test (เพื่อตรวจสอบความถูกต้องการทำงานร่วมกับระบบอื่น)
- 4) Build Artifact (เพื่อทำการบรรจุโปรแกรมที่ผ่านสามขั้นตอนเบื้องต้น)
- 5) Archive Artifact (เพื่อทำการเก็บโปรแกรมที่ถูกบรรจุแล้ว)
- 6) Deploy (เพื่อนำโปรแกรมที่ถูกบรรจุแล้วไปใช้งานจริง)

การทำขั้นตอนข้างต้นโดยไม่ผ่าน Software ที่ช่วยสำหรับการทำ CI/CD นั้นสามารถทำให้เกิดข้อผิดพลาดได้ และอาจจะเกิดความล่าช้าในการ Deliver โปรแกรม ดังนั้น CircleCI จึงเข้ามาเป็นโปรแกรมที่ช่วยลดข้อผิดพลาดในการ Deliver โปรแกรม โดยสามารถสร้าง Instruction (คำสั่ง) ให้กับ CircleCI ได้ดังนี้

```
jobs:
  build:
    docker:
      - image: buildpack-deps:trusty
    environment:
      FOO: bar
    working_directory: ~/my-app
    branches:
      only:
        - production
        - develop
    steps:
      - run: make lint
      - run: make test
      - run: make build
      - run: make deploy
```

Instruction ข้างต้นนั้นเป็นการสั่งให้ทุกครั้งที่มีการแก้ไขโปรแกรมในระดับของ Production และ Develop ระบบของ CircleCI จะทำการดำเนินการ (Execute) คำสั่งการ Lint, Test, Build และ

Deploy โดยที่ผู้พัฒนาโปรแกรมไม่ต้องเป็นคนที่ทำเอง ช่วยลดความเสี่ยงในการนำโปรแกรมไปใช้งานในชีวิตจริงได้

2.17 Ansible

Ansible เป็นโปรแกรมที่ใช้ในการทำ Application Deployment, , Continuous Delivery และ Software Provisioning โดยรูปแบบการทำงานจะ เป็นการทำให้ IaC (Infrastructure as Code) โดยจะช่วยให้การนำโปรแกรมไปใช้งานทำได้ง่าย ภายใน Ansible จะมีองค์ประกอบดังนี้

- 1) Inventory configuration สำหรับการบริหารเครื่องหรือ IP Address ที่ต้องการให้ ansible สามารถเข้าถึงได้ ตัวอย่างเช่น

```
192.168.6.1
[webservers]
foo.example.com
bar.example.com
```

- 2) Playbooks เป็นการกำหนดหน้าที่ ในการทำงานของโปรแกรม ตัวอย่างเช่น

```
- name: Docker stack deployment
  hosts: "{{ my_hosts }}"
  roles:
    - dev-script-deploy
    - docker-stack-deploy
```

- 3) Roles เป็นการกำหนด Instruction จะถูกนำไปใช้ใน Playbooks

```
- name: Create directory
  file: path={{ project_dir }} state=directory
```



ANSIBLE

รูปที่ 2.10 Ansible Logo

(ที่มา <https://www.ansible.com/>)

2.18 Cloudflare

Cloudflare เป็นบริการ Network ที่ทำหน้าที่เป็นหน้าด่านระหว่างผู้ใช้งานกับเซิร์ฟเวอร์ เพื่อแก้ไขปัญหา Overload request จากผู้ใช้งาน บริการของ cloudflare มีดังนี้

- 1) CDN (content delivery network) คือบริการสำหรับการเรียกใช้งาน assets ของเว็บไซต์ รวมไปถึง HTML pages, javascript file, stylesheet, images และ video ช่วยลดการทำงานของระบบโดยตรงเพราะข้อมูลขนาดใหญ่อยู่ที่ CDN
- 2) ป้องกัน DDoS Attack เมื่อมี cloudflareเป็นหน้าด่านระหว่างผู้ใช้งานกับ เซิร์ฟเวอร์ ทำให้ข้อมูลที่ผิดปกติจะต้องผ่าน cloudflare ก่อนทั้งหมด ทำให้สามารถตรวจสอบและปิดกั้นการโจมตีได้ทันที
- 3) Reverse proxy สำหรับเว็บไซต์ทำหน้าที่เป็นตัวกลางที่สื่อสารระหว่างผู้ใช้งานกับเซิร์ฟเวอร์ ข้อมูลบางอย่างที่มีการเรียกใช้งานซ้ำ ๆ จะถูกทำการ caching ข้อมูลไว้



2.19 PostgreSQL

PostgreSQL เป็นระบบการจัดการฐานข้อมูลเชิงวัตถุ-สัมพันธ์ (object-relational database management system) ที่สามารถใช้ในการเก็บข้อมูลตั้งแต่ระบบขนาดเล็กไปจนถึงการทำ data warehousing โดยความสามารถที่สำคัญของ PostgreSQL ได้แก่

- 1) ACID-compliant การทำ transaction ถึงฐานข้อมูล จะรับประกันการไม่เกิดการซ้ำซ้อนของข้อมูลที่เข้าสู่ฐานข้อมูล
- 2) Transactional ทุกครั้งที่มีการส่งคำสั่งไปสู่ฐานข้อมูลจะต้องเกิดการสำเร็จหรือล้มเหลวของการทำ Transaction จะไม่มีการล้มเหลวครึ่งหนึ่งหรือสำเร็จครึ่งหนึ่ง



รูปที่ 2.12 PostgreSQL Logo
(ที่มา <https://en.wikipedia.org/wiki/PostgreSQL>)

2.20 Scrum Framework

Scrum framework เป็นส่วนหนึ่งของ Agile methodology ใช้ในการทำงานร่วมกันภายในทีมเพื่อจัดการปัญหาที่มีความซับซ้อนสูง Scrum มีองค์ประกอบอยู่ 5 อย่างได้แก่

- 1) Sprint เปรียบเสมือนกรอบเวลาของงานที่มีการวางแผนมาก่อนหน้าแล้วโดยภายในแต่ละ Sprint นั้นจะไม่มีเปลี่ยนแปลงของความต้องการ (Requirement) ปกติแต่ละ Sprint จะใช้เวลาตั้งแต่ 1 อาทิตย์ถึง 1 เดือนแล้วแต่ทีม

- 2) Sprint Planning เป็นช่วงที่ทำการวางแผนสำหรับสิ่งที่จะเกิดขึ้นภายใน Sprint หลังจาก Sprint planning จะต้องรู้ว่าหลังจากจบ Sprint จะได้อะไรบ้าง
- 3) Daily Scrum เป็นเวลาที่ทุกคนในทีมจะพูดถึงสิ่งที่จะเกิดขึ้นภายใน Sprint และสิ่งที่จะทำต่อไปภายใน Sprint
- 4) Sprint Review เป็นช่วงที่ทำการประเมินผลลัพธ์ของ Sprint ให้กับผู้เป็นเจ้าของสินค้า (Product owner) ว่าเป็นไปตามแผนหรือมีสิ่งที่ไม่สามารถทำตามแผนได้หรือไม่ เพื่อนำไปแก้ไขและปรับปรุงภายในอนาคต
- 5) Sprint Retrospective เป็นช่วงที่มองย้อนกลับไปภายใน Sprint และสร้างแผนที่พัฒนาในเรื่องต่างๆ สำหรับ Sprint ถัดไป

2.21 Agile Methodology

เป็นแนวคิดในการทำงาน และไม่จำกัดว่าใช้สำหรับการพัฒนาโปรแกรม โดยความต้องการ (Requirement) และ แนวทางในการแก้ไขเกิดขึ้นจากการทำงานภายในทีม และการทำงานร่วมกับลูกค้าหรือผู้ใช้งาน Agile นั้นจะมีการปรับตัวของความต้องการที่รวดเร็วถ้าเกิดมีการเปลี่ยนแปลงของความต้องการ โดย Agile methodology มีองค์ประกอบที่สำคัญทั้งหมด 12 ข้อ

- 1) ผู้ใช้งานพึงพอใจกับการ Deliver (ส่งออก) ซอฟต์แวร์ที่รวดเร็ว
- 2) สามารถรับการเปลี่ยนแปลงของความต้องการได้
- 3) Deliver ซอฟต์แวร์ภายในระยะเวลาที่สั้น ภายในเวลาน้อยกว่าหนึ่งเดือน
- 4) ฝ่ายธุรกิจและผู้พัฒนาทำงานร่วมกันอยู่เป็นประจำ
- 5) โปรเจกต์ถูกสร้างโดยผู้ที่สามารถฝากความเชื่อใจได้
- 6) การสนทนาตัวต่อตัวเป็นรูปแบบการสื่อสารที่ดีที่สุด
- 7) ซอฟต์แวร์สามารถเป็นมาตรวัดความก้าวหน้าของการทำงาน
- 8) สามารถสร้างและดูแลโปรดค้อย่างสม่ำเสมอ
- 9) ให้ความใส่ใจกับดีไซน์และหลักการทางเทคนิคอยู่เสมอ
- 10) ความเรียบง่ายเป็นสิ่งสำคัญ
- 11) โครงสร้างของทีมที่ดีที่สุดมาจากทีมที่สามารถจัดการตัวเองได้
- 12) คนในทีมไตร่ตรองและพยายามพัฒนาให้การทำงานมีประสิทธิภาพมากขึ้นอยู่เสมอ

2.22 Base64 Encryption

Base64 เป็นรูปแบบการเข้ารหัสของ binary-to-text สามารถแปลงข้อมูลไบนารีเป็น ASCII ทำให้สามารถส่งข้อมูลรูปหรือข้อมูลไบนารีข้ามระบบได้ปลอดภัยกว่าการส่งเป็นแบบ raw source byte โดยการเข้ารหัสคำว่า Man เป็น TWFu ที่อยู่ในรูปแบบ Base64 มีตัวอย่างดังตารางที่ 2.5

ตารางที่ 2.5 Base64 Encoding Table

Source	Text (ASCII)	M						a						n											
	Octets	77 (0x4d)						97 (0x61)						110 (0x6e)											
Bits		0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Base64 encoded	Sextets	19						22						5						46					
	Character	T						W						F						u					
	Octets	84 (0x54)						87 (0x57)						70 (0x46)						117 (0x75)					

2.23 Vim (Text Editor)

Vim เป็น text editor ที่ถูกพัฒนามาจาก vi สำหรับใช้งานภายในระบบ Unix โดยจะมีการทำงานตามคำสั่งที่ป้อนเข้าไปภายใน text user interface (TUI) โดยปกติแล้ว Vim จะมีสิ่งที่ไม่เหมือนกับ Modern text editor เช่น Sublime Text, Visual Studio Code และ Atom ได้แก่ รูปแบบการเคลื่อนไหว (Movement) ภายในโปรแกรมจะทำได้ด้วยการใช้ H, J, K, L สำหรับระบุตำแหน่งภายในโปรแกรมแทนการใช้เมาส์เพื่อไปยังจุดต่างๆ และภายใน Vim จะมีโหมดในการทำงานอยู่ด้วยกันทั้งหมด 3 โหมดได้แก่

- 1) Insert Mode ใช้ในการพิมพ์ข้อความ
- 2) Normal Mode ใช้เคลื่อนไหวและการใส่คำสั่งให้กับระบบ
- 3) Visual Mode ใช้ในการคัดลอกและเน้นข้อความ

2.24 Visual Studio Code

Visual Studio Code คือ source code editor ถูกพัฒนาขึ้นโดยบริษัท ไมโครซอฟต์ เป็นเครื่องมือที่ช่วยให้นักพัฒนาซอฟต์แวร์สามารถพัฒนาโปรแกรมคอมพิวเตอร์ เว็บไซต์ เว็บ-แอปพลิเคชัน และ เว็บเซอร์วิส ระบบที่รองรับการทำงานมีหลากหลายเช่น Window, Linux และ macOS และ เว็บเบราว์เซอร์ ปัจจุบัน Visual Studio Code รองรับมากกว่า 41 ภาษาโปรแกรม



รูปที่ 2.13 Visual Studio Code

(ที่มา https://en.wikipedia.org/wiki/Visual_Studio_Code)

2.25 Amazon web service EFS

EFS หรือ Elastic File System เป็นบริการสำหรับการเข้าพื้นที่เก็บข้อมูลในรูปแบบของ Disk storage สำหรับ Linux-based system เพื่อใช้สำหรับการทำ NFS (Network File System) ระหว่างเครื่องแต่ละเครื่องในระบบของ Amazon web services

2.26 Sendgrid

Sendgrid เป็นบริการด้านการส่ง Email ให้กับผู้ใช้บริการ โดยโครงสร้างของ Sendgrid นั้นมีรูปแบบประเภท SaaS (Software as a service) โดยทาง Sendgrid นั้นเปิดให้บริการ SDK (Software Development Kit) สำหรับการส่ง Email ผ่าน Sendgrid โดย SDK ของ Sendgrid นั้นรองรับทั้งหมด 7 ภาษาโปรแกรมมีได้แก่ C#, Go, Java, Nodejs, PHP, Python และ Ruby

บทที่ 3

วิธีการดำเนินการวิจัย

งานวิจัยฉบับนี้เป็นงานวิจัยเชิงพัฒนา ทั้งในด้านโครงสร้างของระบบ (Infrastructure) และ แอปพลิเคชัน (Application) โดยการพัฒนาระบบในปัจจุบันนั้นจำเป็นต้องมีความรวดเร็ว และ ความสามารถในการเปลี่ยนแปลงกลยุทธ์ในการวางแผนการดำเนินงานได้อย่างฉับไว ผู้วิจัยจึงเลือก ใช้รูปแบบการพัฒนาโปรแกรมแบบ Scrum (อ้างอิงหัวข้อที่ 2.20) และการจัดการงานวิจัย โดยรวม ในรูปแบบของ Agile (อ้างอิงหัวข้อที่ 2.21) ระหว่างการพัฒนาโปรแกรมจะมีการนัดประชุม เพื่อสรุป ผลความคืบหน้าของเดือนที่ผ่านมา และมีการประชุมเพื่อวางแผนงานสำหรับเดือนถัดไป โดยการวิจัย จะเริ่มตั้งแต่การเก็บรวบรวมความต้องการต่าง ๆ นำมาวางแผนและออกแบบ การพัฒนาโปรแกรม การทดสอบการใช้งาน และการนำโปรแกรมไปใช้งานจริงโดยมีแผนงานทั้งหมด ดังตารางที่ 3.1

ตารางที่ 3.1 แผนการวิจัยระหว่างการทำสหกิจศึกษา

งาน	ระยะเวลา				ส.ค.				ก.ย.				ต.ค.				พ.ย.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
ประชุมงาน																				
ศึกษาความต้องการ																				
วางแผนและออกแบบ																				
พัฒนาโปรแกรม																				
ทดสอบการใช้งาน																				
นำไปใช้งานจริง																				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 รวบรวมความต้องการต่าง ๆ

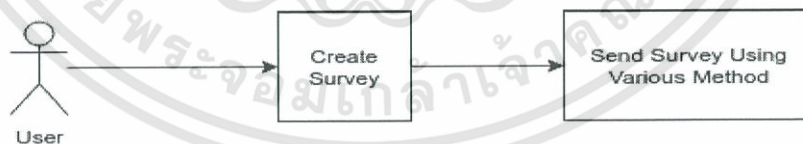
ผู้วิจัยได้ศึกษาค้นคว้าเกี่ยวกับความต้องการต่าง ๆ ที่จำเป็นต่อการพัฒนางานวิจัยฉบับนี้ และได้แบ่งประเภทของความต้องการในการพัฒนาโปรแกรมออกเป็น 2 ส่วน ได้แก่ ความต้องการสำหรับการพัฒนาระบบ และความต้องการของผู้ใช้งาน

3.1.1 ความต้องการสำหรับการพัฒนาระบบ

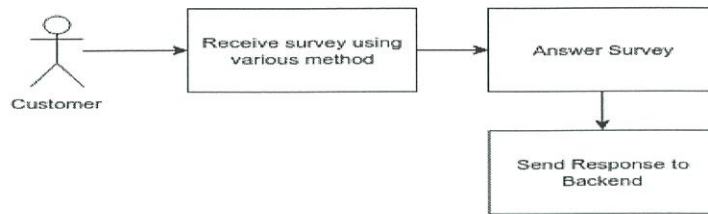
ก่อนที่จะเริ่มทำการพัฒนาโปรแกรมผู้วิจัยได้ศึกษาปัจจัยและข้อจำกัดต่าง ๆ ที่มีผลต่อ การพัฒนาโปรแกรมสำหรับทำแบบสอบถาม ด้วยการสอบถามจากพนักงาน ซึ่งเป็นที่ปรึกษาในการพัฒนาโปรแกรกดังกล่าวของบริษัท เพื่อให้การพัฒนาเป็นไปตามข้อกำหนดและแบบแผนที่วางไว้ ของบริษัทโดยสิ่งที่ต้องเตรียมก่อนพัฒนามีดังนี้

- 1) ศึกษาทำความเข้าใจ Technology Stack ที่ใช้ในการพัฒนาตัวโปรแกรม
- 2) ศึกษาทำความเข้าใจรูปแบบของการทำแบบสอบถาม CSAT, NPS
- 3) เตรียมเครื่องเซิร์ฟเวอร์สำหรับการทำงานของโปรแกรม

นอกจากนั้นผู้วิจัยได้ทำความเข้าใจเกี่ยวกับการทำงานและทฤษฎีที่เกี่ยวข้องกับการทำแบบสอบถาม ต่าง ๆ โดยสรุปได้ใจความดังนี้ ผู้ใช้งานจะต้องสามารถสร้างแบบสอบถามสำหรับให้ลูกค้าของผู้ใช้งาน มาทำแบบสอบถามได้ ดังรูปที่ 3.1 และลูกค้าของผู้ใช้งานสามารถตอบแบบสอบถามได้จากหลาย ๆ ทางเช่น Email, Embeded Links, Helpdesk เช่น Autotask, Zendesk และ Connectwise ดังรูปที่ 3.2



รูปที่ 3.1 ผู้ใช้งานทำการสร้างและทำการส่งแบบสอบถาม



รูปที่ 3.2 ลูกค้าของผู้ใช้งานทำการตอบแบบสอบถาม

3.1.2 ความต้องการของผู้ใช้งาน

ผู้วิจัยได้ทำการศึกษาค้นคว้าเกี่ยวกับความต้องการของผู้ใช้งาน และขอบเขตในการพัฒนา โปรแกรมด้วยการสอบถาม Project owner ซึ่งเป็นผู้เชี่ยวชาญทางด้านมาร์เก็ตติ้ง และเจ้าของบริษัท ที่ใช้ระบบแบบสอบถามเกี่ยวกับ CSAT และ NPS หลังจากทำการวิเคราะห์จากการรวบรวมข้อมูลทั้งหมด ทำให้กำหนดขอบเขตในการพัฒนาความสามารถของโปรแกรมแบบสอบถามได้ทั้งหมด 4 ส่วนดังนี้

- 1) โปรแกรมแบบสอบถามจะต้องสามารถสร้างแบบสอบถามแบบแบ่งประเภทได้
 - แบบสอบถามประเภท NPS
 - แบบสอบถามประเภท CSAT
 - แบบสอบถามประเภท RATER
- 2) แบบสอบถามสามารถเชื่อมต่อกับระบบภายนอกได้
 - Connectwise
 - Autotask
 - Zendesk
- 3) โปรแกรมแบบสอบถามสามารถส่งข้อมูลหาลูกค้าของผู้ใช้ได้
 - Recurring Email
 - Embedded Link
 - Send using Integration
- 4) โปรแกรมสามารถแสดงผลและมี Insight ให้กับผู้ใช้
 - Metric in Dashboard and Insight Page

3.2 วางแผนและออกแบบการพัฒนาโปรแกรม

จากการเก็บรวบรวมข้อมูลความต้องการของผู้ใช้ตามหัวข้อที่ 3.1.2 ผู้วิจัยได้ทำการวางแผนและออกแบบการทำงานของระบบโดยจะแบ่งเป็น 8 sprints โดย 1 sprint ใช้เวลา 2 อาทิตย์ ตามหลักการพัฒนาซอฟต์แวร์ของ Agile ดังตารางที่ 3.2 และ ตารางที่ 3.3

ตารางที่ 3.2 แผนงานครั้งแรกสหกิจศึกษา

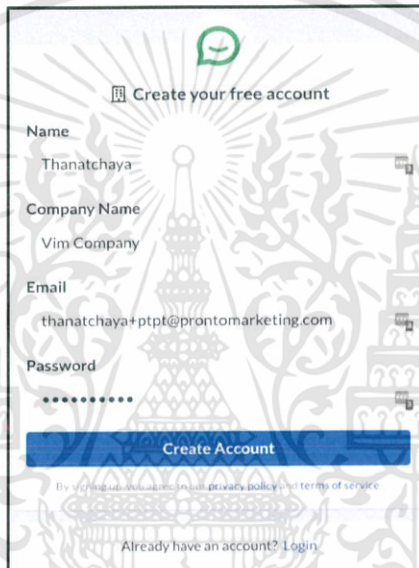
Sprint 1	Sprint 2	Sprint 3	Sprint 4
Signup / Login	Create CSAT Survey	Create NPS Survey	NPS Recurring Email
Account Settings	Dashboard	Dashboard	Insight Pages
Invite to Company	Export Feedback		Survey Presentation

ตารางที่ 3.3 แผนงานครั้งหลังสหกิจศึกษา

Sprint 5	Sprint 6	Sprint 7	Sprint 8
Feedback Filter	Dropbox Integration	Testimonial Publish	Production Deploy
Customer Filter	Staging Deploy		
	Notification		

3.2.1 Signup / Login

การทำเว็บแอปพลิเคชันนั้นมีความซับซ้อนของการเก็บข้อมูลภายในระบบ จึงจำเป็นที่จะต้อง มีระบบที่มากอยแบ่งแยกข้อมูลสำหรับผู้ใช้งานแต่ละคน ในที่นี้แอปพลิเคชันนั้นก็ได้แบ่งระบบของ การลงทะเบียนระบบ ออกเป็น 2 ระดับได้แก่ Account และ Company โดย Account และ Company นั้นมีความสัมพันธ์กันแบบ One to Many โดยแต่ละ Company จะมีได้หลาย Account แต่ 1 Account นั้นสามารถอยู่ได้เพียง Company เดียวเท่านั้น จากหลักการข้างต้นนั้นทำให้สามารถ ทำการสร้างองค์กรขึ้นภายในเว็บแอปพลิเคชันได้ ดังรูปที่ 3.3 และ ดังรูปที่ 3.4



Create your free account

Name
Thanatchaya

Company Name
Vim Company

Email
thanatchaya+ptpt@prontomarketing.com

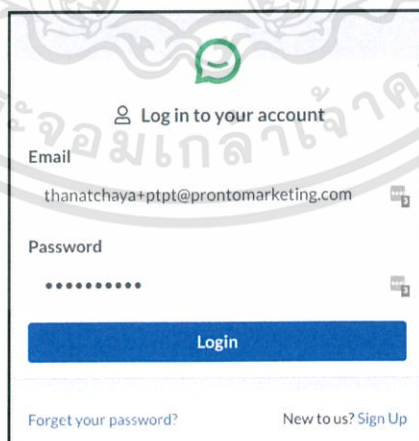
Password
.....

Create Account

By signing up, you agree to our [privacy policy](#) or [terms of service](#)

[Already have an account? Login](#)

รูปที่ 3.3 หน้าต่างลงทะเบียนใช้งานแอปพลิเคชัน



Log in to your account

Email
thanatchaya+ptpt@prontomarketing.com

Password
.....

Login

[Forget your password?](#) [New to us? Sign Up](#)

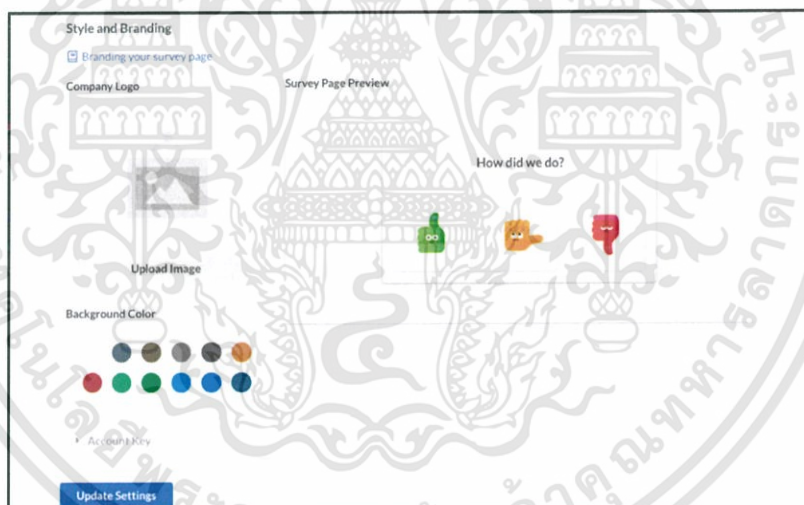
รูปที่ 3.4 หน้าต่างสำหรับเข้าใช้งานแอปพลิเคชัน

โดยเบื้องหลังการทำงานของระบบนั้นรหัสผ่านใช้การเข้ารหัสแบบ Base64 (อ้างถึงหัวข้อที่ 2.22) ก่อนที่จะทำการนำข้อมูลทั้งหมดของ Account บันทึกลงฐานข้อมูล

3.2.2 Account Settings

ส่วนของ Account Settings จะสามารถเปลี่ยนแปลงข้อมูลระดับ Company ได้แก่ การ แก้ไขโลโก้ที่แสดงบนแบบสอบถามและหน้าเว็บแอปพลิเคชัน สามารถแก้ไขสีพื้นหลังที่แสดงบน แบบสอบถามได้ สามารถดู Preview ของแบบสอบถามก่อนที่จะนำไปใช้งานจริงได้ตามรูปแบบ WYSIWYG (What you see is what you get) โดย Logo ทั้งหมดจะถูกเก็บอยู่บน EFS (Amazon Elastic File Sharing) ซึ่งเป็นรูปแบบหนึ่งของ NFS (Network File System) ทำให้ Assets และ Logo ทั้งหมดสามารถใช้ร่วมกันได้ภายในวงเน็ตเวิร์ค หน้า Account Settings ดังรูปที่

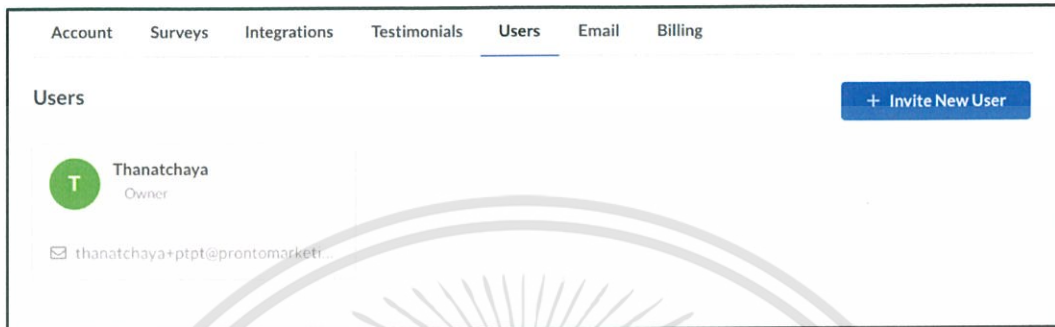
3.5



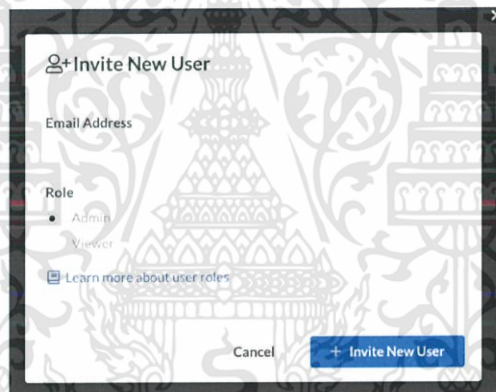
รูปที่ 3.5 หน้า Account Settings

3.2.3 Invite to Company

เนื่องจากฐานลูกค้าของแอปพลิเคชันนั้นเป็น MSP (Managed Service Provider) จึงทำให้ต้องออกแบบระบบการเก็บข้อมูลของบัญชีในรูปแบบของ Company โดยสามารถเชิญเข้าร่วม Company ได้ดังรูปที่ 3.6 และรูปที่ 3.7

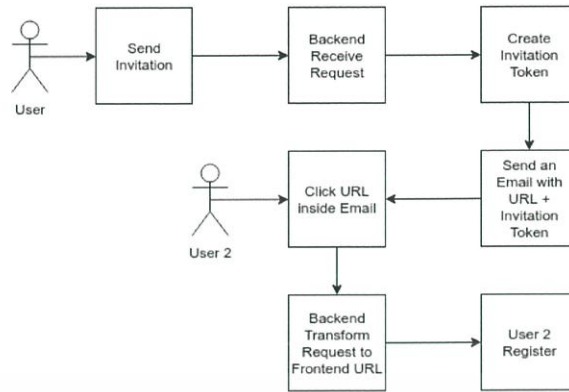


รูปที่ 3.6 หน้าต่างสำหรับเชิญสมาชิก



รูปที่ 3.7 Modal สำหรับเชิญสมาชิก

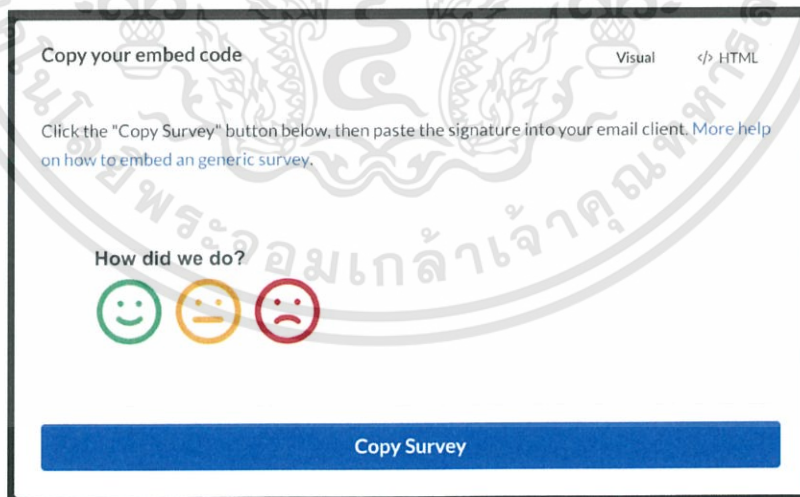
โดยหลักการทำงานหลังจากที่เชิญผู้ใช้งานเข้ามาใน Company นั้นจะมีการส่งคำร้องขอในรูปแบบของ JSON (อ้างถึงหัวข้อที่ 2.4) ส่งไปสู่ Backend หลังจากนั้นระบบจะทำการบันทึก Invitation ในรูปแบบของ Token แล้วนำ Token ไปสร้างเป็น Email เพื่อทำการส่งให้กับผู้ใช้งานที่ถูกเชิญเข้าสู่ Company โดยบริการส่ง Email ที่ใช้คือ Sendgrid (อ้างถึงหัวข้อที่ 2.26) หลังจากนั้นผู้ใช้งานที่ถูกเชิญก็ทำการตอบรับ Email โดยการคลิกที่ URL Token ที่ได้รับหลังจากนั้นคำร้องขอจะถูกส่งไปสู Backend อีกครั้งหนึ่งเพื่อทำการเทียบ Invitation Token ถ้าสำเร็จจะถูกเปลี่ยนทิศทางไปที่ Frontend เพื่อทำการกรอกข้อมูลที่จำเป็นก่อนที่จะทำการสร้างบัญชีที่ผูกกับ Company นั้น ๆ โดยขั้นตอนทั้งหมดเป็นไปตามดังรูปที่ 3.8



รูปที่ 3.8 Invitation to Company flow

3.2.4 Create CSAT Survey

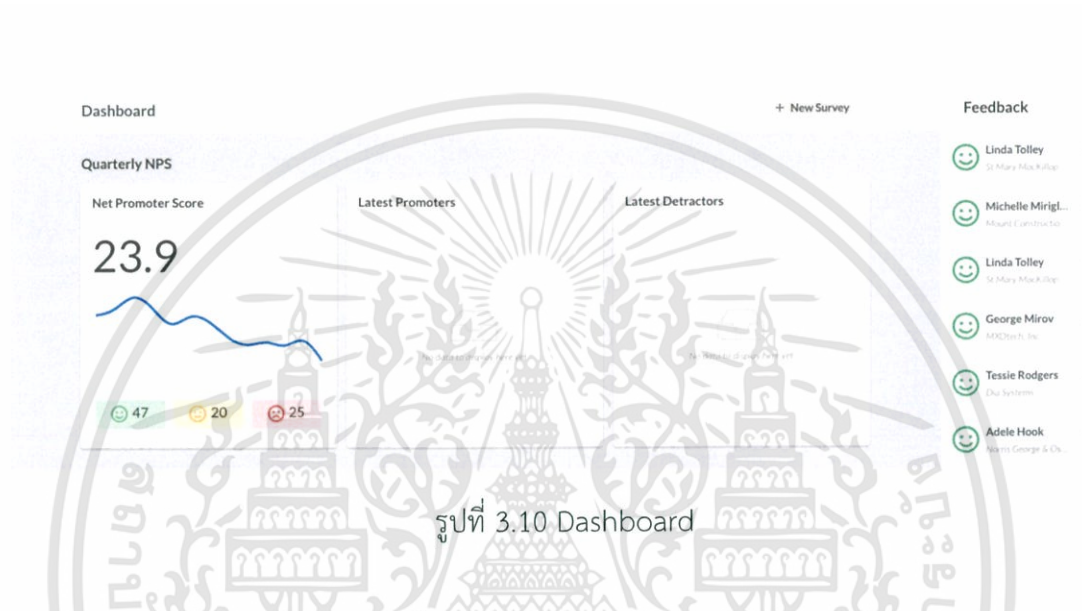
CSAT Survey จัดเป็น 1 ในรูปแบบของแบบสอบถามความพึงพอใจที่เปิดให้ใช้งาน โดย CSAT Survey นั้นจะมีลักษณะที่ไม่เหมือนแบบสอบถามความพึงพอใจรูปแบบอื่นโดย CSAT (อ้างถึงหัวข้อ 2.1) นั้นจะมีลักษณะเป็นการถามความพึงพอใจหลังจากจบการให้บริการแล้วส่วนใหญ่ จะถูกใช้คู่กับบริการ Helpdesk อื่น ๆ โดยจะมีการเปิด Ticket สำหรับแต่ละปัญหาหลังจากจบ การให้บริการแล้วจะมีการส่ง Email หรือสอบถามเพิ่มเติมเกี่ยวกับบริการที่ได้รับ โดยภายในระบบ จะมี Embedded Code สำหรับนำไปใช้ใน Helpdesk ต่าง ๆ ได้โดยการ Copy Survey และนำไป ใส่ระบบส่ง Email ของแต่ละ Helpdesk หรือนำไปส่ง Email เองก็ได้ด้วยวิธีเดียวกันดังรูปที่ 3.9



รูปที่ 3.9 Embedded Code

3.2.5 Dashboard

Dashboard เป็นส่วนสำคัญที่ทำการแสดงข้อมูลเบื้องต้นหลังจากทำการเข้าสู่ระบบ โดยหน้า Dashboard นั้นจะแบ่งเป็นสองส่วน ส่วนแรก ทำการแสดงผลของแต่ละแบบสอบถามในรอบ 3 เดือนและส่วนที่สอง จะทำการแสดง Feedback Stream สำหรับโซลุ่มลำดับการตอบแบบสอบถาม โดยเรียงลำดับจาก Timestamp ในการตอบคำถาม (ล่าสุดไปเก่า) โดย Dashboard จะมีหน้าตา ดังรูปที่ 3.10

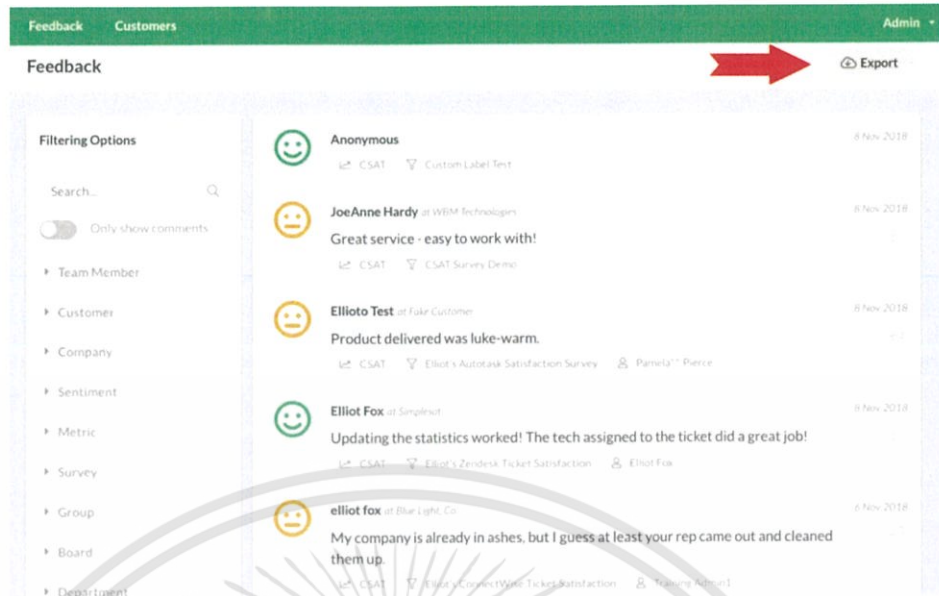


รูปที่ 3.10 Dashboard

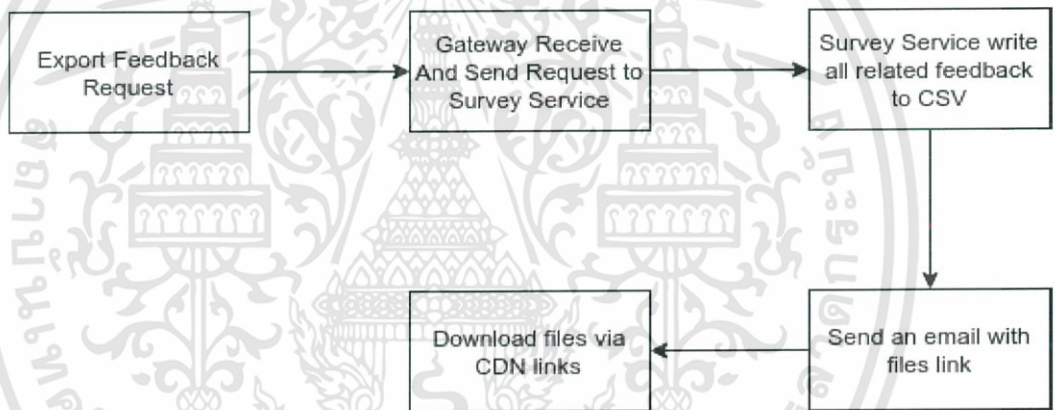
โดย Sentiment ทางด้านซ้ายมีนั้นคำนวณจากฐานข้อมูลทำงานแบบ Periodic Task โดยจะคำนวณทุก ชั่วโมงเพื่อลดการทำงานของระบบทางด้านขวามือคือ Feedback Stream ที่ทำการดึงข้อมูลจาก Elasticsearch ซึ่งข้อมูลจะอยู่ในรูปแบบฐานข้อมูลเอกสาร (Document Database) ทำให้ข้อมูลในส่วนนี้สามารถแสดงได้แบบ Real-time หลังจากเกิดการตอบแบบสอบถาม

3.2.6 Export feedback

ข้อมูลที่ถูกสร้างจากการตอบแบบสอบถาม สามารถส่งออก (Export) และนำไปใช้ใน Helpdesk อื่น ๆ หรือนำข้อมูลไปทำการคำนวณนอกเหนือจากการคำนวณพื้นฐานที่ระบบมีให้ โดย รูปแบบของ เอกสารที่ส่งออกจะเป็นเอกสารประเภท CSV (Comma-separated values) Files โดยสามารถ เข้าถึงการส่งออกได้ผ่าน Feedback Tab จากนั้นจะพบกับ Export Button ดังรูปที่ 3.11 และมีขั้นตอนการทำงานดังรูปที่ 3.12



รูปที่ 3.11 Export Feedback Button

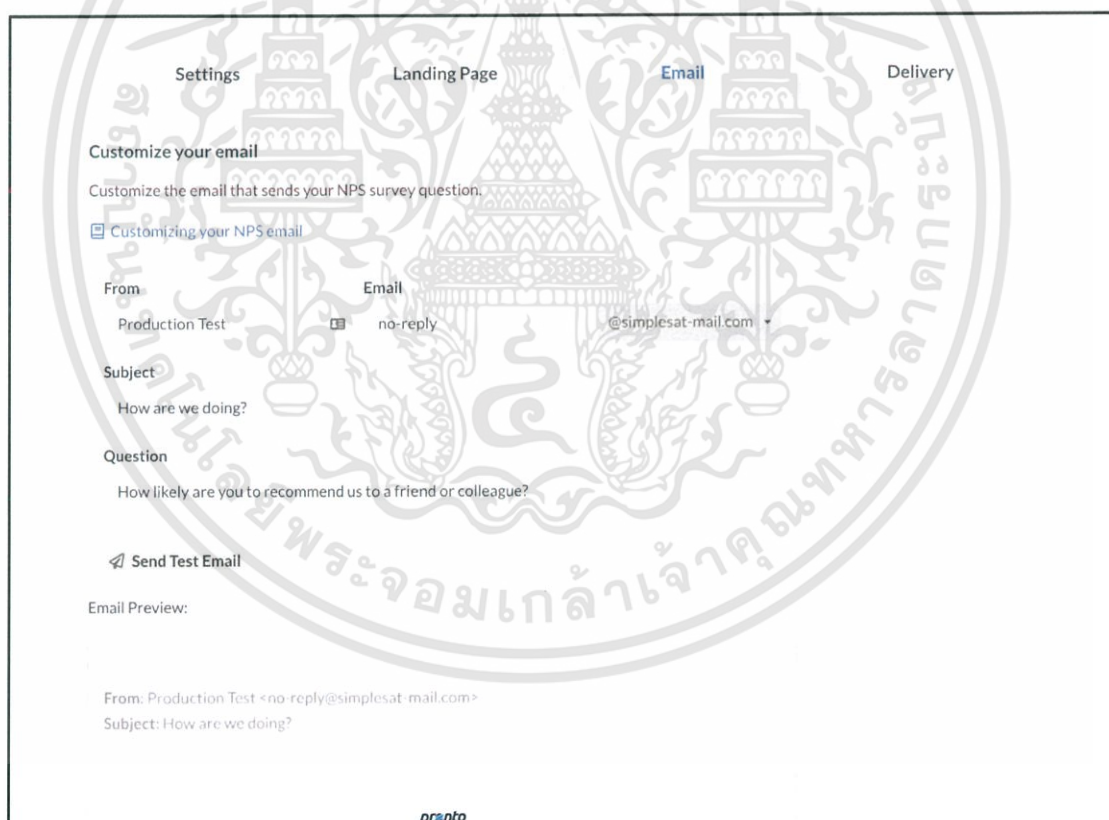


รูปที่ 3.12 Export Feedback Workflow

ขั้นตอนในการ Export Feedback นั้นเริ่มจากการที่กดปุ่ม Export Feedback หลังจากนั้น คำขอ (Request) จะถูกส่งเข้าไปยัง Gateway Service ซึ่งทำหน้าที่เป็นหน้าด่าน ที่ทำการตรวจสอบ Account พร้อมทั้งถอดรหัสข้อมูลสำคัญ หลังจากนั้นข้อมูลจะถูกส่งต่อไปให้กับ Survey Service เพื่อทำการนำข้อมูลจากฐานข้อมูลของ Account นั้น ๆ มาทำการบันทึกลงบน เอกสารประเภท CSV หลังจากบันทึกสำเร็จแล้วจะได้รับที่อยู่สำหรับเข้าถึงเอกสาร โดยผู้ใช้งาน ที่ทำการ Export เอกสารนั้นจะได้รับ Email ที่แนบลิงก์สำหรับดาวน์โหลดไฟล์ผ่านทาง Email Client

3.2.7 Create NPS Survey

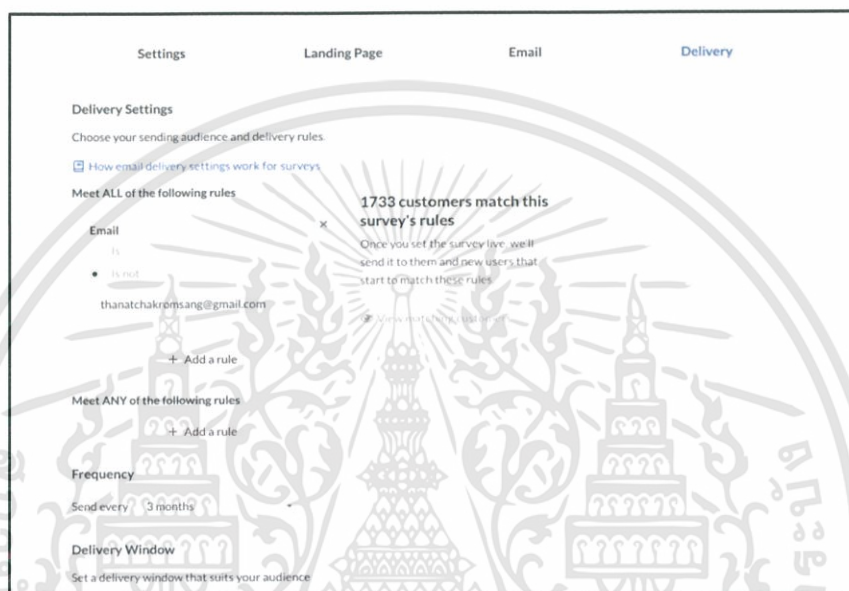
Create NPS Survey นั้นเป็นหนึ่งในรูปแบบของแบบสอบถามความพึงพอใจที่เปิดให้บริการ โดยปกติแล้วการทำ NPS Survey (อ้างอิงหัวข้อ 2.2) จะแตกต่างจาก CSAT Survey ตรงที่ NPS Survey นั้นจะเป็นการสอบถามความพึงพอใจโดยรวมเกี่ยวกับบริษัทที่ให้ทำแบบสอบถามด้วยระบบ การให้คะแนนมาตราการประเมินตั้งแต่ 0-10 แล้วตามด้วยการถามความคิดเห็นของผู้ทำแบบสอบถามว่าทำไมจึงให้คะแนนเช่นนั้น โดยระบบดังกล่าวข้างต้นนั้นจะทำการเป็นรายไตรมาสสำหรับ แต่ละผู้ทำแบบสอบถาม โดยภายในระบบจะใช้การส่ง Email และทำการตรวจสอบว่าถ้าส่งไปแล้ว ไนไตรมาสนี้จะไม่ถูกส่ง Email ถึงอีก การสร้าง NPS Survey นั้นจะมีข้อแตกต่างในการ Configuration เมื่อเทียบกับ CSAT Survey อยู่ด้วยกันสองข้อคือ NPS Survey สามารถกำหนดผู้ส่ง Email ได้ และสามารถกำหนดกลุ่มเป้าหมายลูกค้าที่ต้องการจะส่งแบบสอบถามความพึงพอใจ ดังรูปที่ 3.13, 3.14, 3.15 และ รูปที่ 3.16 สำหรับขั้นตอนการทำงานของระบบ



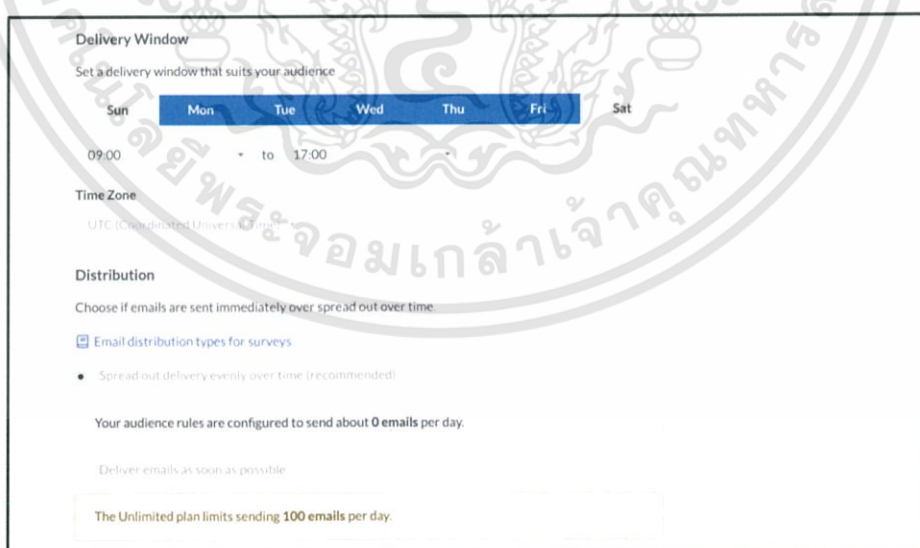
รูปที่ 3.13 NPS Email Setting

สำหรับหน้า NPS Email Setting จะเป็นส่วนสำหรับแก้ไขส่วนต่าง ๆ สำหรับการส่ง Email ได้แก่ หัวข้อ, คำถามที่ต้องการถามภายใน Email และส่งจากใคร โดยในส่วนนี้จะมี Email ใช้สำหรับส่ง แบบสำรวจอยู่ด้วยกัน 2 ประเภทดังนี้

- 1) ส่งในนามของ Email ที่เว็บไซต์ตั้งไว้เป็นค่าเริ่มต้น (ไม่ต้องทำการยืนยัน)
- 2) ส่งในนามของ Domain หรือเว็บไซต์ของตนเอง (ต้องทำการยืนยัน)



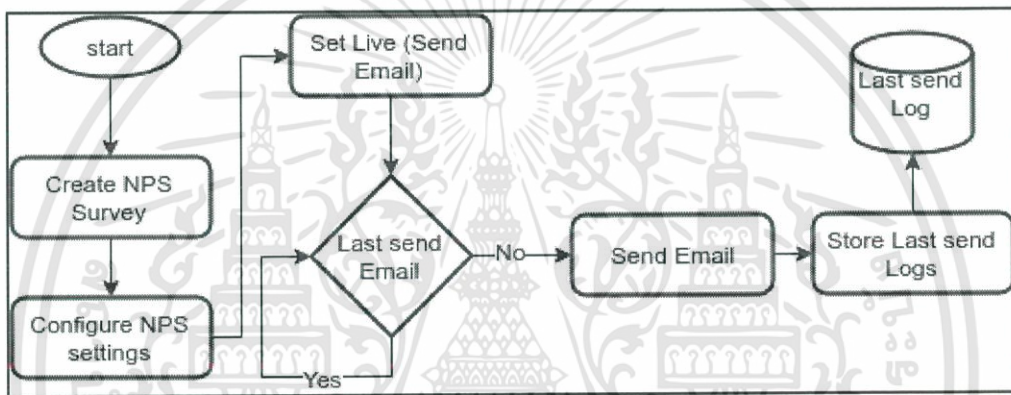
รูปที่ 3.14 NPS Delivery Setting



รูปที่ 3.15 NPS Delivery Setting

หน้า NPS Delivery Setting เป็นส่วนสำหรับเลือกกลุ่มผู้รับ Email จากรายชื่อ Customer ภายในระบบ โดยขั้นตอนการเลือกมีทั้งหมด 5 ขั้นตอนดังนี้

- 1) Audience rules (ประเภทของ Customer สามารถเลือกโดยใช้กฎ AND และ OR)
- 2) Sending Frequency (ความถี่ในการส่งสำหรับหนึ่ง Customer)
- 3) Delivery Window (วันและเวลาในการส่ง)
- 4) Timezone (ปกติเป็น UTC)
- 5) Distribution (รูปแบบการจัดส่ง Email)



รูปที่ 3.16 Create NPS Survey Flow

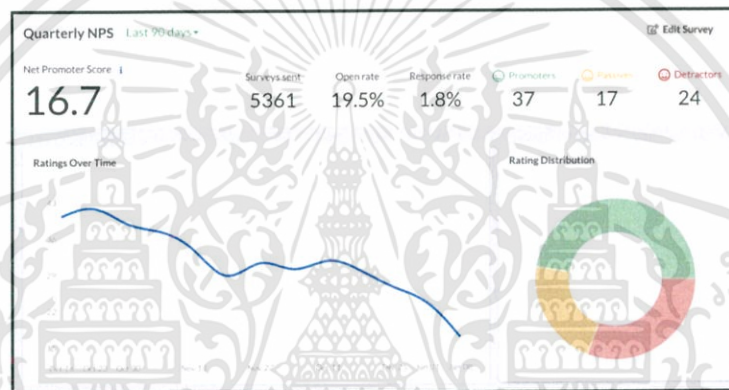
ขั้นตอนการทำงานของ Create NPS Survey เริ่มจากการสร้าง NPS Survey จากนั้นก็ทำการปรับแต่งระบบสำหรับการส่งอีเมลเมื่อทำขั้นตอนข้างต้นสำเร็จแล้ว ทำการส่ง อีเมล ถ้าเกิดลูกค้าที่อยู่ในรายชื่อของ Audience เคยถูกส่ง Email จากแบบสอบถามนี้แล้วจะไม่ได้รับ Email ซ้ำในช่วงเวลาตามความถี่ที่กำหนดไว้ แต่ถ้าไม่เคยได้รับ Email ก็จะถูกส่ง Email แล้วทำการบันทึกลงฐานข้อมูลเพื่อไม่ให้เกิดการส่งเอกสารซ้ำ โดยจะทำซ้ำ ๆ จนครบจำนวนของลูกค้าภายใน Audience Rule ที่กำหนด

3.2.8 Insight Pages

Insight Pages เป็นส่วนที่ใช้ในการแสดงข้อมูลสำคัญทั้งหมดของแต่ละแบบสอบถามในรูปแบบของ NPS และ CSAT ได้ดังนี้

1) NPS Survey Insight

- Survey sent rate
- Open rate
- Response rate
- Sentiment (Promoter, Passive, Detractors)



รูปที่ 3.17 NPS Insight pages

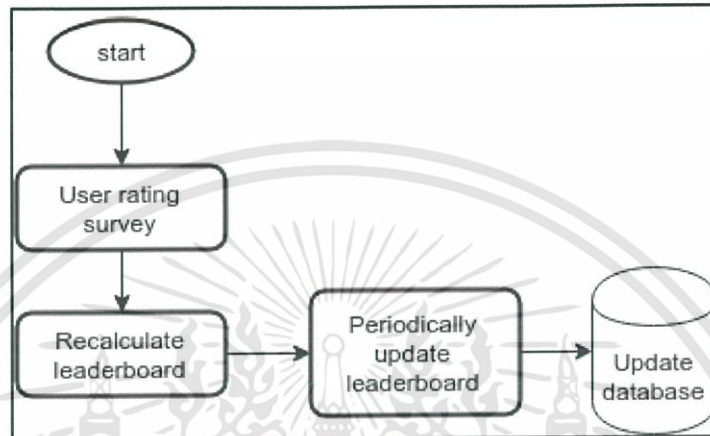
2) CSAT Survey Insight

- Open rate
- Response rate
- Sentiment (Positive, Neutral, Negative)



รูปที่ 3.18 CSAT Insight pages

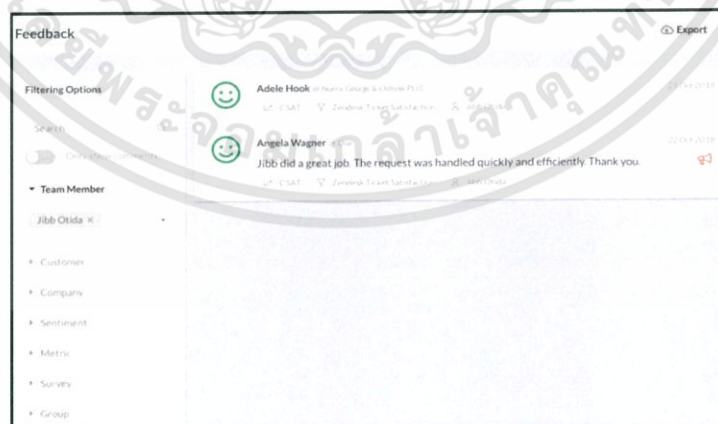
ข้อแตกต่างระหว่างแบบสอบถาม NPS และ แบบสอบถาม CSAT เริ่มต้นจากการส่งแบบ-
 สอบถามไปหาลูกค้าแบบสอบถาม NPS นั้นใช้การส่ง Email ทำให้สามารถเก็บจำนวนการส่ง Email
 ได้แต่ แบบสอบถาม CSAT นั้นไม่สามารถทำได้ โดย Insight pages ของทั้งแบบสอบถาม NPS และ
 CSAT มีขั้นตอนการทำงานเบื้องต้นดังรูปที่ 3.19



รูปที่ 3.19 NPS Survey Flow

3.2.9 Feedback filter

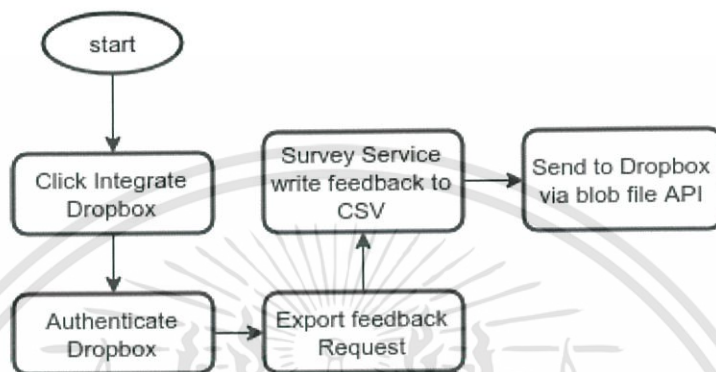
Feedback filter ถูกออกแบบโดยคำนึงถึงการใช้ความสามารถของ search engine ในการค้นหาข้อมูลรูปแบบ document เพื่อความรวดเร็วในการค้นหาข้อมูลผู้วิจัยจึงเลือกใช้ elasticsearch สำหรับการทำ search engine ในโปรแกรม การเก็บข้อมูลจะเป็นรูปแบบ key value store และค้นหาข้อมูลจากค่า key ที่สนใจ ตัวอย่างการทำงานดังรูปที่ 3.20



รูปที่ 3.20 Feedback filter

3.2.10 Dropbox Integration

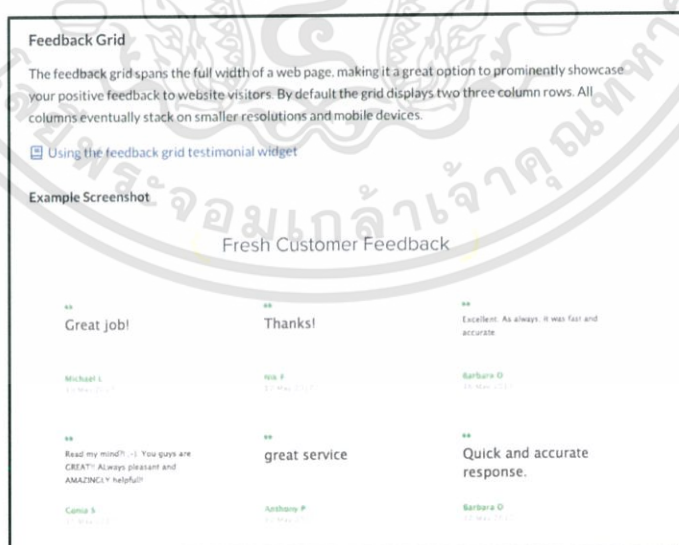
เมื่อผู้ใช้งานต้องการ export ข้อมูลของลูกค้าที่เกิดขึ้นในแต่ละวันโดยที่ไม่ต้องมาทำการ export เองผ่านปุ่ม export feedback สามารถใช้ dropbox integration ในการทำ daily export ไปสู่ dropbox ได้โดยจะมีหลักการทำงานขั้นตอนทั้งหมดดังรูปที่ 3.21



รูปที่ 3.21 Dropbox Integration Flow

3.2.11 Testimonial Publish

ผู้ใช้งานสามารถนำ Testimonial ที่เกิดจากการตอบแบบสอบถามของลูกค้า ไปไว้บนเว็บไซต์ของตัวเองได้ ทำได้โดยการนำชุดโปรแกรมที่ระบบสร้างให้ไปใส่ไว้บนเว็บไซต์ตัวเองภายใต้ <script></script> ดังรูปที่ 3.22 และรูปที่ 3.23



รูปที่ 3.22 Publish Testimonial Example

Code

Place this code wherever you want the plugin to appear on your page.

```
<div id="simplesat-full-width-widget"></div>
<script>
  (function(document) {
    var addWidgetScriptTag = function() {
      var lastScript =
Array.prototype.slice.call(document.getElementsByTagName('script')).slice(-1)[0]
      var script = document.createElement('script')
      script.src = 'https://s3-eu-west-1.amazonaws.com/simplesat-cdn/fullWidth.js'
      lastScript.parentNode.insertBefore(script, lastScript.nextSibling)
      script.addEventListener('load', function() {
        Widget.config({
          companyKey: 'MTAwMA==',
          primaryColor: '#57DD84',
          showTestimonialDate: true,
          maxFeedback: 6
        })
        Widget.render()
      })
    }
    document.addEventListener('DOMContentLoaded', addWidgetScriptTag)
  })(document)
```

รูปที่ 3.23 Publish Testimonial Code

3.2.12 Production Deployment

การนำโปรแกรมไปใช้งานจริงถือเป็นหนึ่งในส่วนสำคัญของการทำซอฟต์แวร์โดยแผนการนำโปรแกรมไปใช้งานในระดับ production ผู้วิจัยได้วางแผนไว้ได้แก่

- 1) High availability services
- 2) Monitoring
- 3) Blue-green deployment
- 4) Fallback plan
- 5) Automatic CI/CD deployment flow

หัวข้อที่กล่าวมาข้างต้นเป็นขั้นเริ่มต้นสำหรับการนำซอฟต์แวร์ขึ้นระดับ production สำหรับโปรแกรมที่ผู้วิจัยพัฒนาได้มีการวางแผนสำหรับการทำแต่ละขั้นตอนเริ่มจากการศึกษาจากระบบต่าง ๆ พบว่าการใช้ Cloud service provider เช่น Amazon web services ช่วยลดเวลา และค่าใช้จ่ายสำหรับการวางโครงสร้างระบบได้ โดยตัวอย่างระบบปัจจุบัน

3.3 เริ่มพัฒนาโปรแกรม

ผู้วิจัยได้เริ่มพัฒนาโปรแกรมตามแผนการที่วางไว้ในหัวข้อที่ 3.2 โดยแบ่งการพัฒนาทั้งหมดเป็น 8 sprints เมื่อครบระยะเวลาในการพัฒนา 1 sprint จะมีการทำสำรวจผลของแผนที่ได้ตั้งเป้าไว้กับปริมาณงานที่ได้ทำไปจริงว่าสัมพันธ์กันหรือไม่ และอาจจะทำให้การพัฒนาโปรแกรมในแต่ละ sprint มีการเปลี่ยนแปลงแผนตามความเหมาะสม โดยแต่ละ sprint มีรูปแบบการพัฒนาโปรแกรมตาม feature ทำให้ภายใน 1 sprint จะต้องพัฒนาโปรแกรมด้วยภาษา Javascript, Python และ Shell script ต่าง ๆ

3.4 ทดสอบการใช้งานของโปรแกรมจริง

ผู้วิจัยได้ทดสอบการใช้งานของโปรแกรมทุกครั้งที่พัฒนาโปรแกรมขึ้นในแต่ละขั้นตอนเมื่อทำการพัฒนาเสร็จสมบูรณ์ด้วยการทำ Unittest สำหรับชุดโปรแกรมที่ไม่ได้เชื่อมต่อกับ API และทำ Integration Test และ Automation Test สำหรับชุดโปรแกรมที่เชื่อมต่อกับ API ผ่านขั้นตอนการทำ CI/CD (continuous integration and continuous delivery) สำหรับทุกครั้งที่เกิดการเปลี่ยนแปลงโปรแกรมในระดับชั้น Staging และ Production

3.5 นำโปรแกรมไปใช้งานจริง

หลังจากขั้นตอนการทดสอบจึงสามารถนำโปรแกรมไปใช้งานจริง โดยการนำไปใช้งานจริงหมายถึงการนำชุดโปรแกรมทั้งหมดขึ้นสู่ production โดยการขึ้นสู่ production ได้มีการนำโปรแกรมส่วน backend ไปทำการติดตั้งไว้บน Amazon web services EC2 (อ้างถึงหัวข้อ 2.13) โดยขั้นตอนนี้จำเป็นจะต้องใช้ container orchestration tools เช่น docker swarm (อ้างถึงหัวข้อ 2.12) สำหรับดูแลโปรแกรมส่วน backend ที่เป็น microservice ทั้งหมดจากนั้นได้มีการติดต่อขอใบรับรองความปลอดภัย SSL certificate สำหรับใช้งาน HTTPS protocol อย่างถูกกฎหมาย

นอกจากนั้นสำหรับโปรแกรมส่วน frontend ได้ทำการติดตั้งไว้บน netlify ซึ่งเป็นบริการสำหรับ publish static website สำหรับให้ผู้ใช้งานเข้าถึงเว็บไซต์ได้ ส่วนของโปรแกรมส่วน frontend ก็ได้ทำการขอ SSL certificate สำหรับใช้งาน HTTPS protocol เช่นกัน

บทที่ 4

ผลการวิจัย

ในส่วนของผลการวิจัยผู้วิจัยจะแบ่งผลการวิจัยออกเป็น 2 ส่วนคือ ประสิทธิภาพ และ ประสิทธิภาพในการประเมินประสิทธิภาพของการวิจัยฉบับนี้ว่าสำเร็จหรือไม่ ผู้วิจัยจะประเมินจากการนำวัตถุประสงค์ของการวิจัยที่กำหนดไว้ในหัวข้อ 1.2 มาใช้เป็นหัวข้อในการประเมิน ส่วนในด้านการประเมินประสิทธิภาพของการวิจัย ผู้วิจัยจะประเมินจะประเมินจากผลการทำงาน ของโปรแกรมว่าสามารถทำงานได้หรือไม่ โดยมีรายละเอียดของผลการวิจัยดังนี้

4.1 การประเมินเชิงประสิทธิภาพ

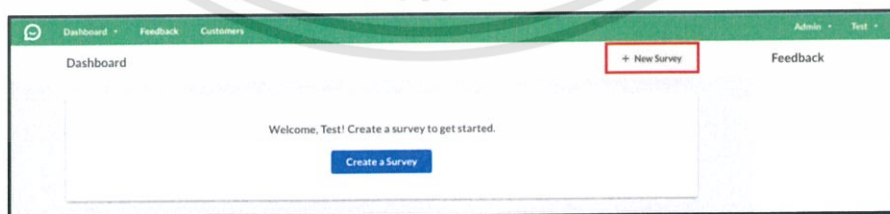
ผู้วิจัยได้วัดประสิทธิภาพของการพัฒนาโปรแกรมว่ามีประสิทธิภาพหรือไม่ จากการนำ วัตถุประสงค์ของการวิจัยที่กำหนดไว้ในหัวข้อที่ 1.2 ได้ดังนี้

4.1.1 เพื่อให้การทำแบบสำรวจความพึงพอใจง่ายขึ้นกว่าปัจจุบัน

ผู้วิจัยได้ทำการประเมินประสิทธิภาพการทำงานของระบบแล้วพบว่าเมื่อเทียบกับระบบอื่น ๆ ในปัจจุบันเช่น Survey Monkey และ Typeform ระบบที่ผู้วิจัยทำการพัฒนาผู้ใช้งานสามารถเข้าถึงได้ง่ายกว่า 2 ระบบข้างต้นที่ยกตัวอย่าง ทั้งในเรื่อง On boarding และ User experience ทั้งนี้ผู้วิจัยจะทำการยกตัวอย่างการใช้งานเบื้องต้นดังรูป 4.1 ถึงรูป 4.9

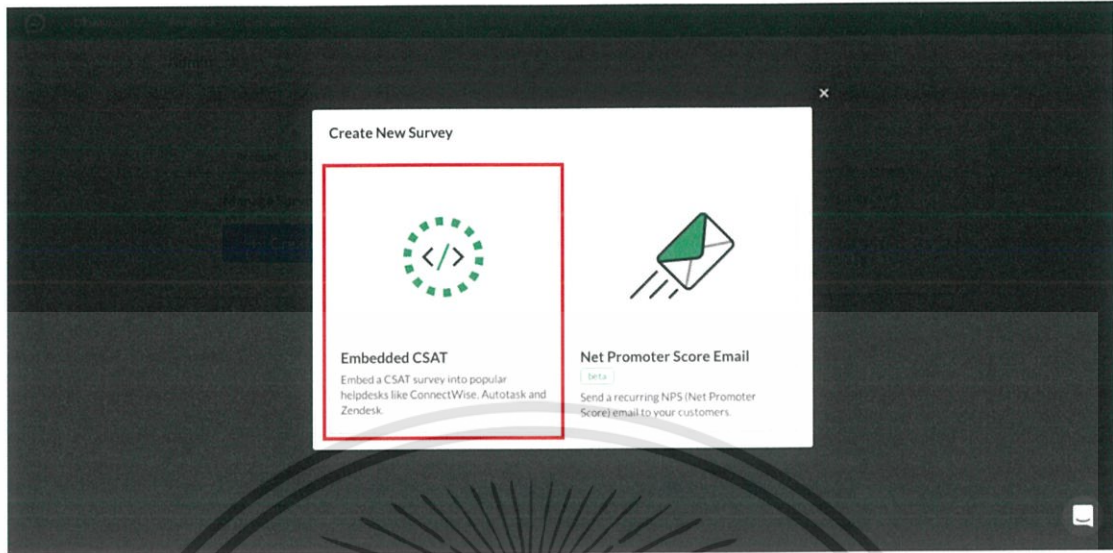
ขั้นตอนการสร้างแบบสอบถามและการตอบแบบสอบถาม

- เข้าสู่หน้า Dashboard หลังจากทำการเข้าสู่ระบบ



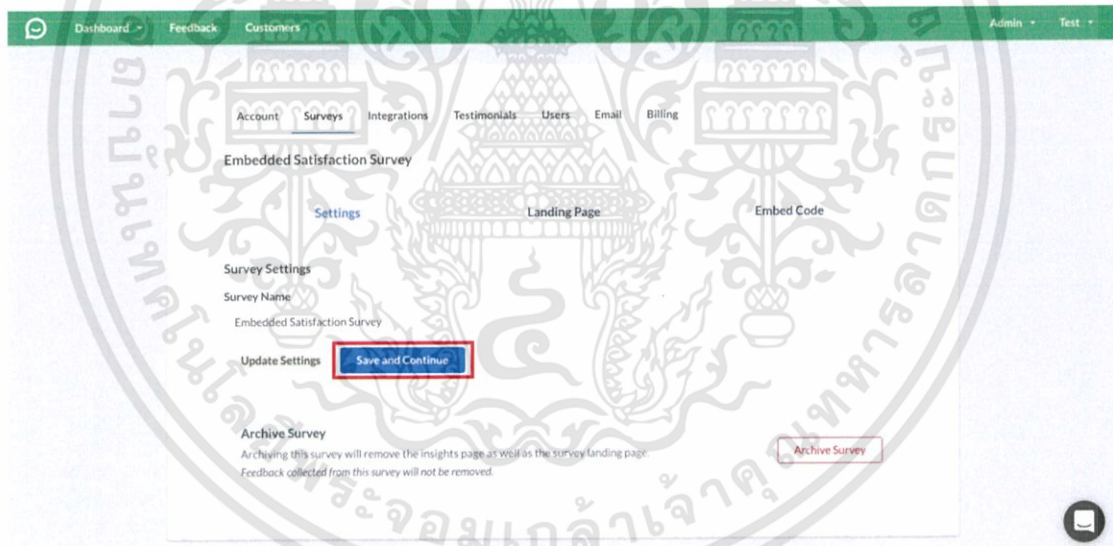
รูปที่ 4.1 Dashboard

- สร้างแบบสอบถามใหม่โดยเลือกรูปแบบ embedded CSAT



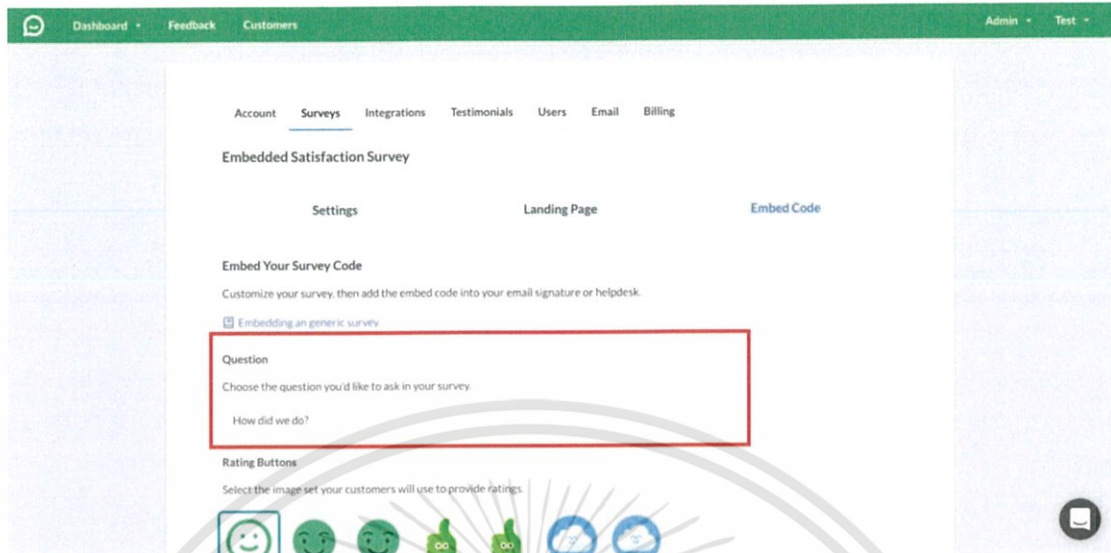
รูปที่ 4.2 Create New Survey

- ตั้งชื่อแบบสอบถาม



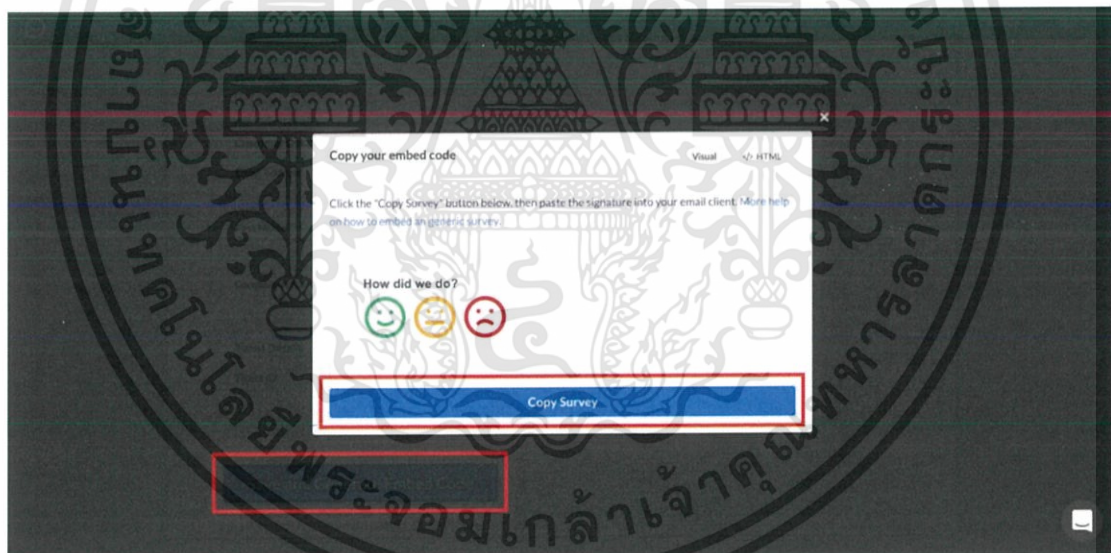
รูปที่ 4.3 Survey Setting

- ตั้งชื่อหัวข้อที่จะถามลูกค้าของแบบสอบถาม



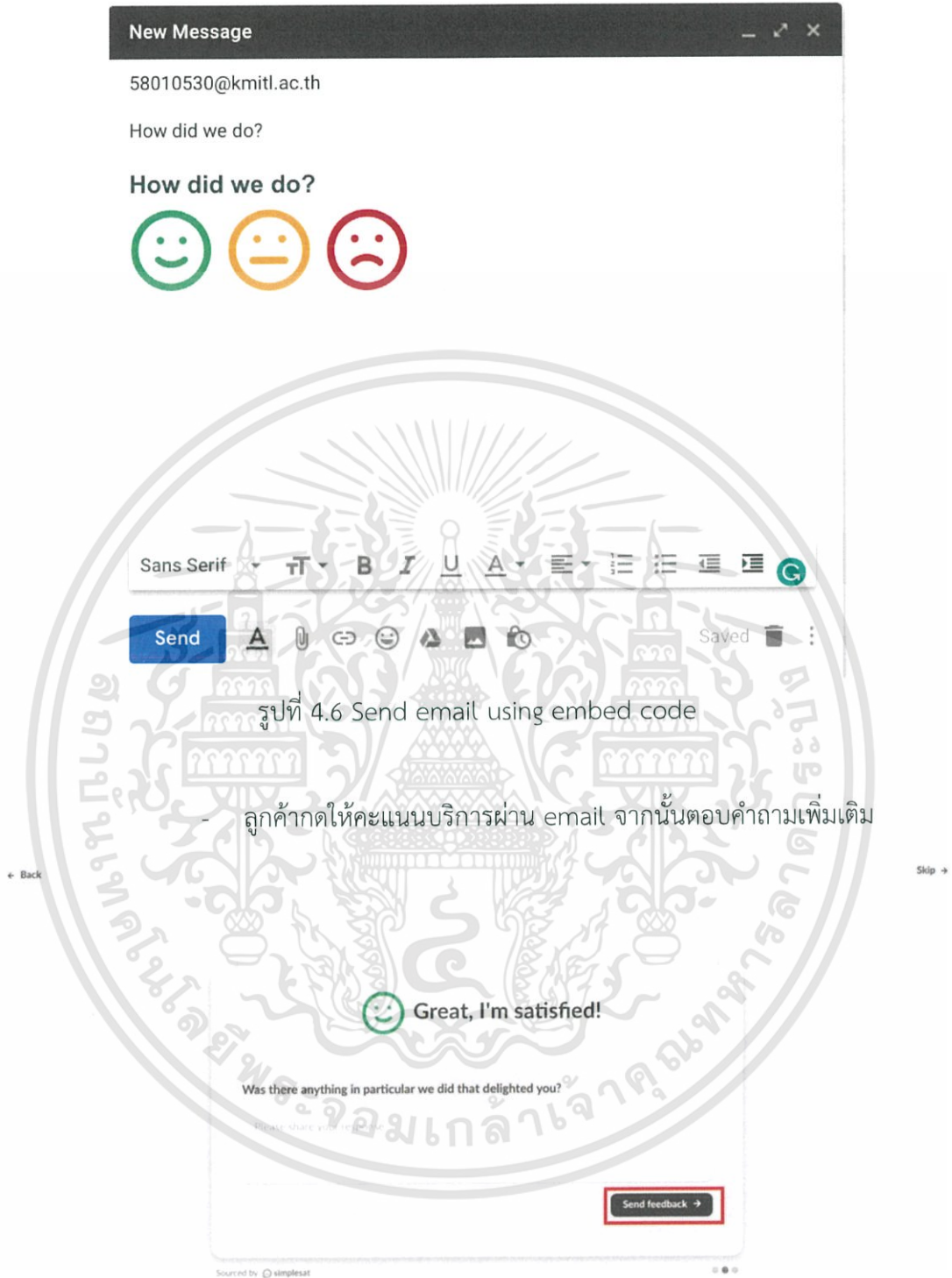
รูปที่ 4.4 Survey Embed Setting

- สร้างและทำการคัดลอก embed code



รูปที่ 4.5 Generate Embed Code

- คัดลอก embed code ใส่ email เพื่อทำการส่งให้กับลูกค้า



รูปที่ 4.6 Send email using embed code

- ลูกค้ากดให้คะแนนบริการผ่าน email จากนั้นตอบคำถามเพิ่มเติม

รูปที่ 4.7 Customer rating

- ลูกค้ากรอกชื่อและอีเมลเพื่อเก็บลงฐานข้อมูลของแต่ละ account

← Back

Confirm →

Let us know who you are

Thanatchaya

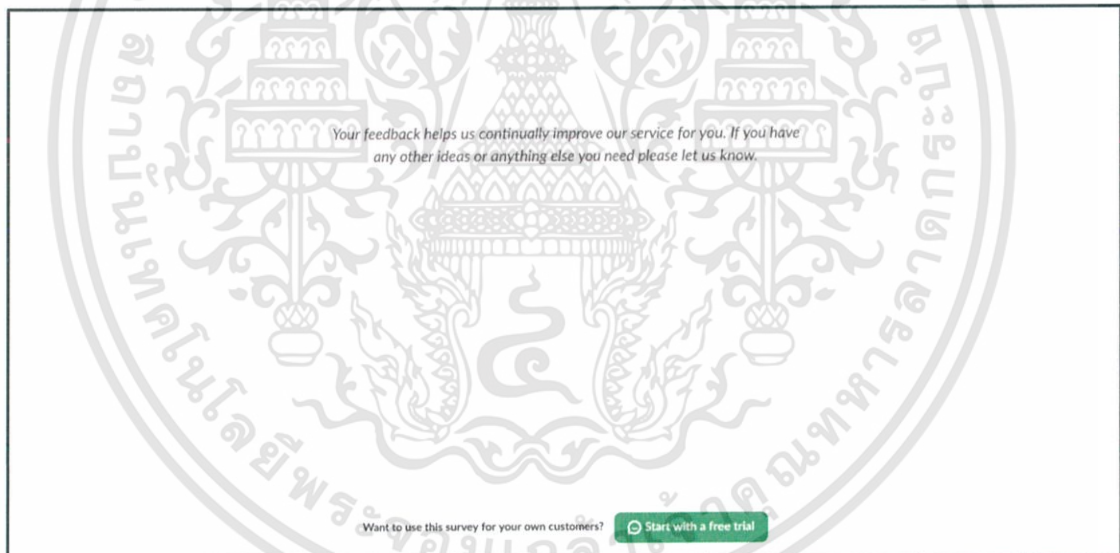
banana@apple.com

Confirm →

Sourced by simplesat

รูปที่ 4.8 Customer Confirmation

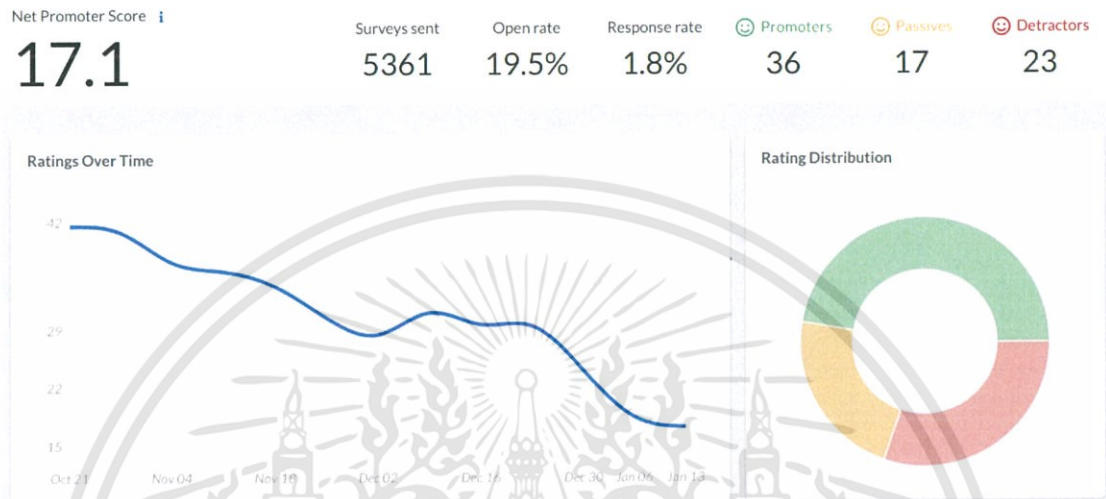
- จบกระบวนการทำงานทั้งหมดเข้าสู่หน้า call to action



รูปที่ 4.9 Call to action

4.1.2 เพื่อให้รับรู้ถึง Feedback ของลูกค้าได้ไวขึ้น

จากผลสำรวจผู้ใช้งานระบบพบว่าผู้ใช้งานสามารถรับรู้ทิศทางของบริการตัวเองว่าไป ในทิศทาง บวกหรือลบได้ผ่าน Insight dashboard ได้ทันทีหลังจากที่เกิดการทำแบบ สอบถามโดยสามารถดูทิศทางของผลสำรวจได้จากกราฟ



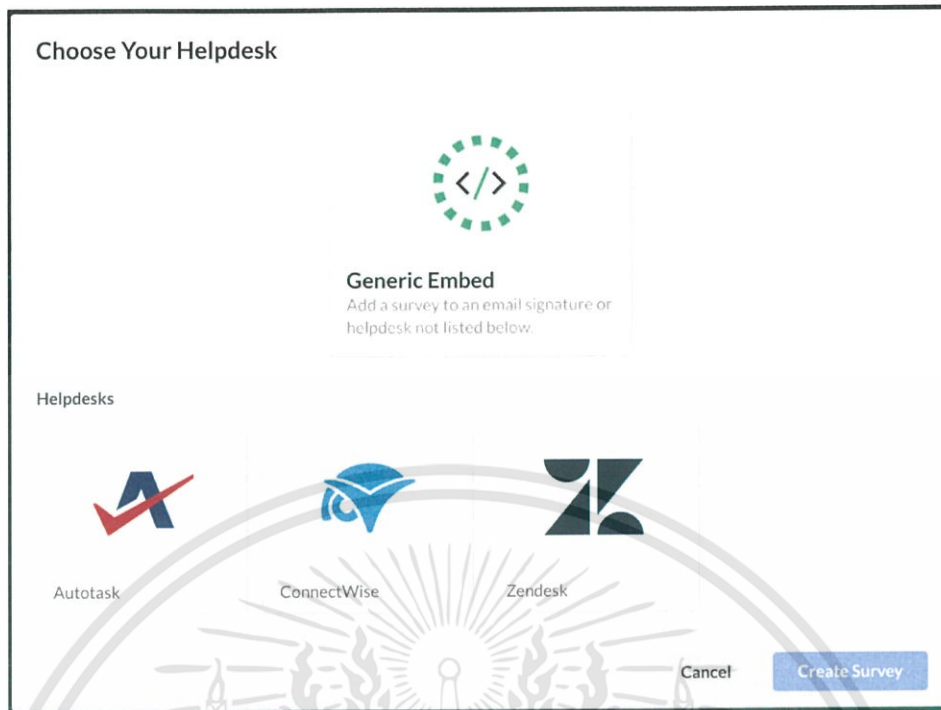
รูปที่ 4.10 Insight Dashboard Graph

4.1.3 เพื่อให้ลูกค้าได้รับบริการที่ดีขึ้น

ผู้วิจัยได้ทำการสอบถามบริษัทที่เป็นผู้ใช้งานระบบเป็นจำนวน 2 บริษัท พบว่าบริษัทดังกล่าวข้างต้นได้นำข้อมูลที่ได้รับ จากการให้ลูกค้าทำแบบสอบถามนำมาปรับปรุง บริการของตนเองพบว่า ทำให้ทิศทางของบริการมีแนวโน้มบวก ซึ่งส่งผลดีต่อทั้งบริษัทและผู้ใช้บริการ

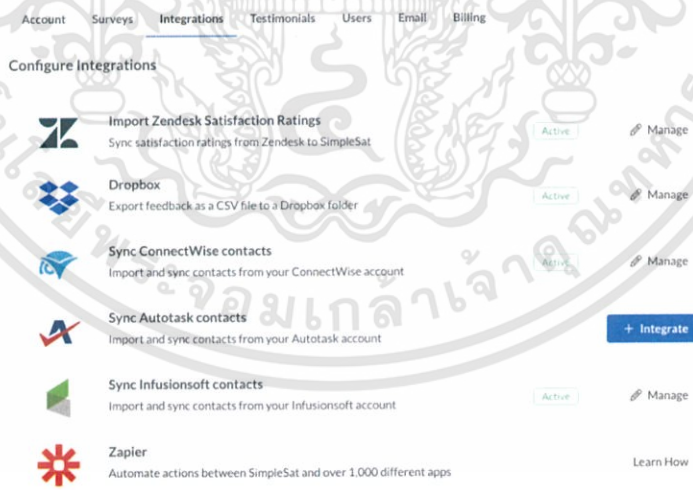
4.2 การประเมินผลเชิงประสิทธิผล

ผู้วิจัยได้วัดประสิทธิผลของการพัฒนาโปรแกรมว่ามีประสิทธิผลหรือไม่ จากการทดสอบการ ใช้งานของโปรแกรม และพบว่าโปรแกรมสามารถใช้สำหรับทำแบบสอบถามความพึงพอใจได้ผ่านทาง embed links, automatic sending email, helpdesk integration ได้ตามความต้องการของผู้ใช้ แต่เนื่องจาก helpdesk ในโลกความเป็นจริงนั้นมีจำนวนมาก บาง helpdesk ไม่มี API สำหรับให้ผู้วิจัยทำการเชื่อมต่อ เข้ากับระบบของผู้วิจัย จึงไม่สามารถทำ helpdesk integration ได้ทุก helpdesk ดังนั้นผู้วิจัยจึงได้ทำการเชื่อมระบบเข้ากับ helpdesk บางส่วนเท่านั้นดังรูปที่ 4.11 และรูปที่ 4.12



รูปที่ 4.11 Survey Helpdesk Integration

ภายในระบบได้ทำการเชื่อมต่อกับ helpdesk ทั้งหมด 3 service สำหรับการส่งแบบสอบถามอัตโนมัติ แต่ถ้าหากใช้บริการ helpdesk อื่นๆ จำเป็นจะต้องใช้ generic embed survey



รูปที่ 4.12 Helpdesk Customer Integration

การจะใช้งานระบบที่ผู้วิจัยทำขึ้นมา นั้นจำเป็นต้องมีรายชื่อลูกค้าของผู้ใช้บริการถึงจะทำให้ใช้
บริการได้คุ้มค่าที่สุด สำหรับลูกค้าที่ใช้งานระบบอื่นมาก่อนแล้ว ก็สามารถมาใช้งานคู่กับระบบที่
ผู้วิจัยพัฒนาได้ โดยการเชื่อมต่อกับ helpdesk ของตนเองทำให้ได้รายชื่อสำหรับการส่ง email
หรือทำการวิเคราะห์ข้อมูลของลูกค้าก็ย่อมได้



บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ระบบสำรวจความพึงพอใจของลูกค้า (Simple Customer Satisfaction Platform) เป็นระบบที่จัดทำขึ้นมาเพื่ออำนวยความสะดวกให้กับ MSP (Managed Service Provider) หรือผู้บริหารธุรกิจขนาดเล็กและขนาดกลางทำให้ทราบและเข้าใจความพึงพอใจของลูกค้าที่มีต่อบริการที่มีจากการที่ได้พัฒนาและทดสอบการใช้งานของลูกค้า ซึ่งได้ผลลัพธ์ดังนี้

- สามารถเชิญชวนผู้ใช้งานได้หลายคนภายในหนึ่งบัญชี
- สามารถมีบทบาทภายในหนึ่งบัญชีได้หลายตำแหน่งเช่น Owner, Admin, Viewer เพื่อจำกัดสิทธิการเข้าถึงข้อมูลสำคัญ
- สามารถทำการนำเข้าข้อมูลลูกค้าจากระบบต่าง ๆ รวมถึงการนำเข้าข้อมูลลูกค้าจาก CSV (Comma-separated values file)
- สามารถทำการส่งแบบสอบถามให้กับลูกค้าที่อยู่ภายใน และ ภายนอกระบบ
- สามารถเชื่อมต่อกับระบบภายนอกเพื่อทำการส่งแบบสอบถามให้กับลูกค้า
- สามารถวิเคราะห์ข้อมูลที่ได้รับจากการทำแบบสอบถามของลูกค้า
- สามารถทำการนำออกข้อมูลลูกค้าในรูปแบบ CSV เพื่อทำไปทำการวิเคราะห์ข้อมูล

จากผลลัพธ์ทั้งหมด สามารถสรุปได้ว่า การพัฒนาระบบสำรวจความพึงพอใจของลูกค้า สามารถตอบสนองและอำนวยความสะดวกให้กับ MSP (Managed Service Provider) หรือ ผู้บริหารธุรกิจขนาดเล็กและขนาดกลางได้เป็นอย่างดี

5.2 ข้อเสนอแนะ

จากการพัฒนาระบบสำรวจความพึงพอใจของลูกค้าซึ่งมีผู้ใช้งานหลักเป็นธุรกิจขนาดกลางและขนาดเล็ก ทำให้ทราบถึงปัญหาและข้อจำกัดบางประการจึงมีข้อเสนอแนะดังต่อไปนี้

- ควรมีพื้นฐานในเรื่องการออกแบบโครงสร้างของ Cloud Architecture
- ควรจะมีพื้นฐานภาษาอังกฤษ เนื่องจากทรัพยากรที่เกี่ยวข้อง มีข้อมูลภาษาไทยน้อย
- ควรมีความรู้พื้นฐานภาษา Javascript และ ReactJS
- ควรมีความรู้พื้นฐานภาษา Python และ Django framework



เอกสารอ้างอิง

[1] Django: The Web framework for perfectionists with deadlines, Available :

<https://www.djangoproject.com/>

(วันที่ค้นข้อมูล: 24 ตุลาคม 2561)

[2] React – A JavaScript library for building user interfaces, Available :

<https://reactjs.org/>

(วันที่ค้นข้อมูล: 21 ตุลาคม 2561)

[3] What Is Customer Satisfaction Score, Available :

<https://blog.hubspot.com/service/customer-satisfaction-score>

(วันที่ค้นข้อมูล: 9 กันยายน 2561)

[4] What Is Net Promoter Score, Available :

<https://www.medallia.com/net-promoter-score/>

(วันที่ค้นข้อมูล: 9 กันยายน 2561)