



T148622

รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบโต้ตอบผู้ใช้อัตโนมัติสำหรับช่วยเหลือการปล่อยซอฟต์แวร์

Automatic interactive system for supporting software release

นายอริยุต รัชตเจริญชัยกุล

เลขหมู่.....  
เลขทะเบียน 148622  
วัน,เดือน,ปี. = 6 พ.ย. 2560

b. 12872349  
i. ....

ภาควิชา วิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2559

ชื่อโครงการสหกิจศึกษา ระบบโต้ตอบผู้ใช้อัตโนมัติสำหรับช่วยเหลือการปล่อยซอฟต์แวร์

ชื่อ-สกุล นักศึกษา นายอริยุต รัชตเจริญชัยกุล

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมคอมพิวเตอร์

ชื่อ-สกุล อาจารย์นิเทศ อาจารย์จิระศักดิ์ สิทธิกร และ อาจารย์บัณฑิต พัสยา

ชื่อ-สกุล ผู้นิเทศงาน นายอดิภัทร ลาภรัตนากุล

ชื่อสถานประกอบการ บริษัท รอยเตอร์ ซอฟต์แวร์ (ประเทศไทย) จำกัด

## บทคัดย่อ

เนื่องจากการพัฒนาซอฟต์แวร์ของทางบริษัท รอยเตอร์ ซอฟต์แวร์ (ประเทศไทย) จะมีการทำงานในรูปแบบของการแบ่งเป็นทีมเพื่อให้มีความสะดวกในการทำงาน ซึ่งแต่ละทีมจะทำหน้าที่ผู้ดูแลในส่วนต่างๆ ของซอฟต์แวร์ หนึ่งในทีมเหล่านั้นคือทีม eikon desktop ที่ทำหน้าที่ดูแลซอฟต์แวร์ที่ชื่อว่า eikon ในส่วนของ desktop ซึ่ง eikon toolbar เป็นส่วนหนึ่งของซอฟต์แวร์ eikon โดยทีมพัฒนาของ eikon toolbar ได้มีการนำหลักการพัฒนาซอฟต์แวร์ในรูปแบบของ Agile มาประยุกต์กับการพัฒนาซอฟต์แวร์ของทางบริษัททำให้มีการกำหนดรอบของการปล่อยซอฟต์แวร์ให้ผู้ใช้งานทุกๆ 2 อาทิตย์ซึ่งการปล่อยซอฟต์แวร์ในแต่ละครั้งจะมีการเลือกคนในทีม 1 คนเพื่อทำการควบคุมการปล่อยซอฟต์แวร์ในครั้งนั้น โดยขั้นตอนการควบคุมการปล่อยซอฟต์แวร์นั้นในแต่ละครั้งจะมีลักษณะใกล้เคียงกันทำให้ทางทีมพัฒนาเล็งเห็นถึงความไม่จำเป็นที่จะต้องให้คนในทีมมาทำการควบคุมการปล่อยซอฟต์แวร์เนื่องจากจะทำให้เสียเวลาสำหรับการพัฒนาซอฟต์แวร์ไปทั้งยังมีโอกาสทำให้เกิดข้อผิดพลาดในการปล่อยซอฟต์แวร์ที่อาจจะส่งผลให้เกิดความล่าช้าของการปล่อยซอฟต์แวร์ในครั้งนั้น จากปัญหาข้างต้น โครงการนี้จึงได้ถูกพัฒนาขึ้นเพื่อทำการช่วยเหลือคนที่ได้รับเลือกให้ควบคุมการปล่อยซอฟต์แวร์เพื่อให้สามารถทำการพัฒนาซอฟต์แวร์ต่อไปได้โดยไม่ต้องคอยควบคุมการปล่อยซอฟต์แวร์ตลอดเวลาและช่วยป้องกันความผิดพลาดที่อาจจะเกิดขึ้นจากคนที่ทำการควบคุมการปล่อยซอฟต์แวร์

**Co-operative Title:** Automatic interactive system for supporting software release

**Student Intern Name:** Mr.Athiyut Ratchatajaroenchaikul

**Faculty:** Engineering

**Department:** Computer Engineering

**Advisor Name:** Jirasak Sittigorn and Bundit Pasaya

**Mentor Name:** Mr.Adipat Larprattanakul

**Company:** Reuters Software (Thailand) Ltd.

## ABTRACT

In the Reuters software (Thailand) Ltd. The developer work as a team and each team have a responsibility for some part of a software. The eikon toolbar is the one of the software. The team that working with eikon toolbar using the Agile method to deliver the software. With Agile the team set delivery time to 2 weeks per release. So, The team needs to choose one of them called "Release Master" to control each release that means the team will lose someone for each release and sometimes him/her may make some mistake during control the release. So, This project is developed to assist the release master in controlling the release and protecting the release form him/her mistake.

## กิตติกรรมประกาศ

โครงการระบบโต้ตอบผู้ใช้อัตโนมัติสำหรับช่วยเหลือการปล่อยซอฟต์แวร์ สำเร็จลุล่วงได้ด้วยดีด้วยความช่วยเหลือทั้งทางตรงและทางอ้อมจาก ดร.จิระศักดิ์ สิทธิกรและอ.บัณฑิต พัสยา ที่ให้คำปรึกษาแนะนำ ทั้งในด้านของการจัดทำโครงการรวมถึงการปฏิบัติตัวระหว่างเข้าร่วมโครงการสหกิจศึกษา อาจารย์และบุคลากรด้านต่างๆของภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้ความรู้และคำแนะนำ บริษัท รอยเตอร์ซอฟต์แวร์ (ประเทศไทย) จำกัด สำหรับให้สถานที่ในการทำโครงการและซอฟต์แวร์ที่เกี่ยวข้องกับการสร้างระบบโต้ตอบผู้ใช้อัตโนมัติสำหรับช่วยเหลือการปล่อยซอฟต์แวร์ นายอดิภัทร ลากรัตนากุล ที่ให้คำปรึกษาในระหว่างการจัดทำระบบโต้ตอบผู้ใช้อัตโนมัติสำหรับช่วยเหลือการปล่อยซอฟต์แวร์ ในท้ายที่สุดขอขอบคุณบิดามารดาและครอบครัวที่ได้เลี้ยงดูสั่งสอนพร้อมทั้งสนับสนุนและให้กำลังใจตลอดมา

นายอริยุด รัชตเจริญชัยกุล



## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	V
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 วิธีการดำเนินการโครงการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 ทฤษฎีที่เกี่ยวข้อง.....	3
2.2 เทคโนโลยีที่เกี่ยวข้อง.....	6
บทที่ 3 วิธีดำเนินการ.....	11
3.1 ศึกษารายละเอียดของขั้นตอนการปล่อยซอฟต์แวร์.....	11
3.2 ออกแบบโครงสร้างโดยรวมของระบบ.....	12
3.3 พัฒนาระบบในส่วนของการโต้ตอบผู้ใช้อัตโนมัติ.....	20
3.4 พัฒนาระบบในส่วนของการช่วยเหลือการปล่อยซอฟต์แวร์.....	24
บทที่ 4 ผลการดำเนินการ.....	30
4.1 ส่วนของการโต้ตอบผู้ใช้อัตโนมัติ.....	30
4.2 ส่วนของการช่วยเหลือการปล่อยซอฟต์แวร์.....	31

## สารบัญ(ต่อ)

	หน้า
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	32
5.1 สรุปผล.....	32
5.2 ข้อเสนอแนะ.....	32
เอกสารอ้างอิง.....	33



## สารบัญภาพ

ภาพที่	หน้า
3.1 รูปภาพแสดง Use Case Diagram ของระบบ.....	12
3.2 รูปภาพแสดงโครงสร้างของระบบ.....	14
3.3 แผนภาพกิจกรรมของการโต้ตอบกับระบบ.....	15
3.4 แผนภาพกิจกรรมแสดงการสั่งงาน Jenkins ผ่านระบบ.....	16
3.5 แผนภาพกิจกรรมแสดงการปล่อยซอฟต์แวร์สามารถสั่งให้เตรียมการสำหรับ ปล่อยซอฟต์แวร์แบบทีละขั้นตอน.....	17
3.6 แผนภาพกิจกรรมแสดงการปล่อยซอฟต์แวร์สามารถสั่งให้เตรียมการสำหรับ ปล่อยซอฟต์แวร์แบบอัตโนมัติ.....	18
3.7 แผนภาพกิจกรรมแสดงการตรวจสอบสถานะการเตรียมการปล่อยซอฟต์แวร์.....	19
3.8 รูปภาพแสดงระบบในส่วนของการโต้ตอบผู้ใช้อัตโนมัติ.....	20
3.9 รูปภาพแสดงช่องทางในการติดต่อกับผู้ใช้.....	20
3.10 รูปภาพแสดงการปรับแต่งช่องทางในการติดต่อกับผู้ใช้.....	21
3.11 รูปภาพแสดงบริการของกัตัวเชื่อมต่อ.....	21
3.12 รูปภาพแสดงข้อมูลที่ได้รับส่งผ่านตัวเชื่อมต่อ.....	22
3.13 รูปภาพแสดงตัวอย่างการทำงานของ bot.....	23
3.14 รูปภาพแสดงตัวอย่างหน้าแสดงผลของ LUIS.....	23
3.15 รูปภาพแสดงผลลัพธ์ของการทำนาย.....	24
3.16 รูปภาพแสดงระบบในส่วนของการช่วยเหลือการปล่อยซอฟต์แวร์.....	24
3.17 รูปภาพแสดงตัวอย่างของ Google Sheets.....	25
3.18 รูปภาพแสดงตัวอย่างของข้อมูลที่อ่านผ่าน Google Sheets API.....	25
3.19 รูปภาพแสดงตัวอย่างหน้าแสดงผลของ Jenkins.....	26
3.20 รูปภาพแสดงตัวอย่างการแสดงผลการเรียกดูข้อมูล Jenkins API.....	26
3.21 รูปภาพแสดงตัวอย่างหน้าแสดงผลของ Microsoft Outlook.....	27
3.22 รูปภาพแสดงตัวอย่างแสดงของการติดต่อผ่าน Microsoft Outlook API.....	27
3.23 รูปภาพแสดงตัวอย่างหน้าแสดงผลของ GitLab.....	28
3.24 รูปภาพแสดงตัวอย่างแสดงของการติดต่อผ่าน GitLab API.....	28
3.25 รูปภาพแสดงตัวอย่างการทำงานของ SignalR.....	29
4.1 รูปภาพแสดงการโต้ตอบกับผู้ใช้ของระบบ.....	30
4.2 รูปภาพแสดงการโต้ตอบกับผู้ใช้ของระบบเมื่อเกิดความผิดพลาด.....	31

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญ

ปัจจุบันในกระบวนการของการพัฒนาซอฟต์แวร์มีรูปแบบที่หลากหลายหนึ่งในนั้นคือ กระบวนการพัฒนาซอฟต์แวร์ที่นำหลักการของ Agile มาปรับใช้โดยหนึ่งในหลักการนั้นจะกล่าวถึงการปล่อยตัวซอฟต์แวร์ที่ใช้งานได้ให้สม่ำเสมอแทนการปล่อยซอฟต์แวร์แบบครั้งเดียว เพื่อให้ลูกค้าได้ทดลองใช้งานตัวซอฟต์แวร์ในบางส่วนก่อนทำให้นักพัฒนาสามารถปรับแก้ตัวซอฟต์แวร์ให้ได้ตามที่ลูกค้าต้องการมากที่สุดซึ่งการปล่อยซอฟต์แวร์จะมีการกำหนดรอบตามความเหมาะสมของทีมพัฒนาที่นำไปประยุกต์ใช้

ในส่วนของการพัฒนาซอฟต์แวร์ของทางบริษัท รอยเตอร์ ซอฟต์แวร์ (ประเทศไทย) จะมีการทำงานในรูปแบบของการแบ่งเป็นทีมเพื่อให้มีความสะดวกในการทำงาน ซึ่งแต่ละทีมจะทำหน้าที่ผู้ดูแลในส่วนต่างๆของซอฟต์แวร์ หนึ่งในทีมเหล่านั้นคือทีม eikon desktop ที่ทำหน้าที่ดูแลซอฟต์แวร์ที่ชื่อว่า eikon ในส่วนของ desktop ซึ่งซอฟต์แวร์ที่มีชื่อว่า eikon toolbar เป็นส่วนหนึ่งของซอฟต์แวร์ eikon โดยทีมพัฒนาของ eikon desktop ได้มีการนำในหลักการของ Agile มาประยุกต์กับการพัฒนาซอฟต์แวร์ของทางบริษัทซึ่งจากที่กล่าวมาข้างต้นทำให้ทางทีมของนักพัฒนา eikon desktop ได้มีการกำหนดรอบของการปล่อยซอฟต์แวร์ให้ผู้ใช้งานทุกๆ 2 อาทิตย์ โดยขั้นตอนในการปล่อยซอฟต์แวร์ในแต่ละครั้งทางทีมนักพัฒนาจะมีการเลือกคนในทีม 1 คนเพื่อทำการควบคุมการปล่อยซอฟต์แวร์ในครั้งนั้นตามขั้นตอนที่มีการกำหนดไว้ โดยขั้นตอนการควบคุมการปล่อยซอฟต์แวร์นั้นในแต่ละครั้งจะมีลักษณะใกล้เคียงกันทำให้ทางทีมพัฒนาเล็งเห็นถึงความไม่จำเป็นที่จะต้องให้คนในทีมมาทำการควบคุมการปล่อยซอฟต์แวร์เนื่องจากจะทำให้เสียเวลาสำหรับการพัฒนาซอฟต์แวร์ไปทั้งยังมีโอกาสทำให้เกิดข้อผิดพลาดในการปล่อยซอฟต์แวร์ที่อาจจะส่งผลให้เกิดความล่าช้าของการปล่อยซอฟต์แวร์ในครั้งนั้น

เนื่องจากในปัจจุบันที่การใช้โปรแกรมประยุกต์ใช้การช่วยในการตอบคำถามหรือทำงานใดๆที่มีความซ้ำกันเพื่อให้ประหยัดเวลาของผู้ที่ต้องทำการดูแลในส่วนนี้ถูกนำมาใช้อย่างแพร่หลาย รวมถึงมีเครื่องมือในการพัฒนาที่ช่วยให้นักพัฒนาสามารถสร้างโปรแกรมประยุกต์เพื่อโต้ตอบกับผู้ใช้งานได้สะดวกมากยิ่งขึ้นตัวอย่างของเครื่องมือดังกล่าวเช่น Microsoft Bot Framework, Bots for Messenger เป็นต้นนักพัฒนาจึงมีทางเลือกหลายในการพัฒนา

จากที่กล่าวมาทั้งหมด โครงการงานนี้จึงได้ถูกพัฒนาขึ้นเพื่อทำการช่วยเหลือคนที่ได้รับเลือกให้ควบคุมการปล่อยซอฟต์แวร์ให้สามารถทำการพัฒนาซอฟต์แวร์ต่อได้โดยไม่ต้องคอยควบคุมการปล่อยซอฟต์แวร์ตลอดเวลาและช่วยป้องกันความผิดพลาดที่อาจจะเกิดขึ้นจากคนที่ทำการควบคุมการปล่อยซอฟต์แวร์โดยผู้ใช้งานสามารถสื่อสารกับระบบผ่านการส่งข้อความด้วยภาษาที่ใช้สื่อสารกันทั่วไปโดยไม่ต้องมีความรู้ในส่วนของการคำสั่งในการทำงาน

## 1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อพัฒนาระบบสำหรับการช่วยเหลือการปล่อยซอฟต์แวร์
- 2) เพื่อป้องกันความผิดพลาดที่อาจจะเกิดขึ้นหากนักพัฒนาที่ไม่มีความเข้าใจในแต่ละขั้นตอนได้รับเลือกเป็นผู้ควบคุมการปล่อยซอฟต์แวร์

## 1.3 ขอบเขตของโครงการ

- 1) ระบบสามารถช่วยเหลือการปล่อยซอฟต์แวร์ได้เฉพาะซอฟต์แวร์ที่ถูกโปรแกรมไว้แล้วเท่านั้น
- 2) ส่วนของการโต้ตอบอัตโนมัติจะรองรับเฉพาะภาษาอังกฤษเท่านั้น
- 3) ส่วนของการโต้ตอบอัตโนมัติจะรองรับเฉพาะการโต้ตอบผ่านตัวอักษรเท่านั้น

## 1.4 วิธีการดำเนินโครงการ

- 1) ศึกษารายละเอียดของขั้นตอนการปล่อยซอฟต์แวร์
- 2) ออกแบบโครงสร้างโดยรวมของระบบ
- 3) พัฒนาระบบในส่วนของการโต้ตอบผู้ใช้อัตโนมัติ
- 4) พัฒนาระบบในส่วนของการช่วยเหลือการปล่อยซอฟต์แวร์

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ระบบสามารถช่วยเหลือการทำงานของนักพัฒนาได้
- 2) ระบบสามารถป้องกันการความผิดพลาดที่อาจจะเกิดขึ้นได้

## บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

### 2.1 ทฤษฎีที่เกี่ยวข้อง

#### 2.1.1 Software Development

Software Development เป็นการเรียกรวมของกระบวนการในการพัฒนาซอฟต์แวร์ นับตั้งแต่การระบุความต้องการ การวางแผน การออกแบบ การดำเนินการเพื่อสร้างซอฟต์แวร์ การทดสอบซอฟต์แวร์ การทำเอกสารประกอบการพัฒนาซอฟต์แวร์ การนำซอฟต์แวร์ให้ลูกค้าได้ใช้งาน การดูแลปรับปรุงซอฟต์แวร์ และอื่นๆ เพื่อให้ได้มาซึ่งซอฟต์แวร์ที่สามารถใช้งานได้โดยรายละเอียดของขั้นตอนหลักๆ มีดังต่อไปนี้

##### 2.1.1.1 การระบุความต้องการ

การระบุความต้องการสามารถระบุได้จากหลายแหล่งอาจจะมาจากลูกค้า โดยตรงจากการวิเคราะห์ตลาด เป็นต้นการระบุความต้องการมีความสำคัญมากเนื่องจากเป็นขั้นตอนเพื่อรวบรวมข้อมูลสำหรับการวิเคราะห์และการออกแบบซอฟต์แวร์ในขั้นต่อไป

##### 2.1.1.2 การวางแผน

การวางแผนเป็นขั้นตอนสำหรับวางแผนกิจกรรมต่างๆที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์เพื่อตอบสนองความต้องการให้ได้ดีที่สุด การวางแผนจะต้องคำนึงถึงองค์ประกอบทุกๆ ด้านไม่ว่าจะเป็นความสามารถของนักพัฒนาแต่ละคน ความคลุมเครือของความต้องการของลูกค้า เป็นต้น อาจจะมีการจัดทำเอกสารเพื่อให้มีความชัดเจนมากขึ้น

##### 2.1.1.3 การออกแบบระบบ

การออกแบบระบบจะเป็นขั้นตอนที่เกี่ยวข้องกับการออกแบบซอฟต์แวร์ รวมถึงส่วนองค์ประกอบอื่นที่เกี่ยวข้องด้วยเช่น ภาษาที่ใช้ในการเขียนโปรแกรม ระบบปฏิบัติการที่จะนำซอฟต์แวร์ไปทำงาน อุปกรณ์ที่เกี่ยวข้อง เป็นต้น ในขั้นตอนนี้จะมีการจัดทำเอกสารการออกแบบซอฟต์แวร์ขึ้นมาด้วย

##### 2.1.1.4 การดำเนินการเพื่อสร้าง การทดสอบซอฟต์แวร์และการจัดทำเอกสาร

การดำเนินการเพื่อสร้างซอฟต์แวร์เป็นขั้นตอนของการพัฒนาซอฟต์แวร์ โดยจะมีการทำการทดสอบซอฟต์แวร์ ซึ่งเป็นขั้นตอนที่สำคัญเนื่องจากเป็นขั้นตอนที่ทำให้ยืนยันได้ว่าซอฟต์แวร์ที่ให้บริการพัฒนาขึ้นมานั้นสามารถทำงานได้ถูกต้องตามที่ได้ออกแบบไว้ ในส่วนของขั้นตอนการจัดทำเอกสารเป็นขั้นตอนเพื่อจัดทำเอกสารของการพัฒนาที่จะถูกใช้ในอนาคตสำหรับการปรับปรุงหรือเพิ่มเติมซอฟต์แวร์

#### 2.1.1.5 การนำซอฟต์แวร์ให้ลูกค้าได้ใช้งานและการดูแลปรับปรุงซอฟต์แวร์

การนำซอฟต์แวร์ให้ลูกค้าใช้งานเป็นขั้นตอนที่ลูกค้าได้ใช้งานซอฟต์แวร์ตามความต้องการที่ระบุไว้ในขั้นตอนแรก ซึ่งขั้นตอนนี้อาจจะมีการติดตั้งตัวซอฟต์แวร์หรือการปรับแต่งให้เข้ากับลูกค้าและอาจจำเป็นต้องมีการสนับสนุนเพื่อช่วยเหลือการติดตั้งหรือปรับแต่ง การดูแลปรับปรุงซอฟต์แวร์จะเป็นขั้นตอนเพื่อรับมือการผิดพลาดของซอฟต์แวร์รวมถึงการเพิ่มเติมความสามารถของซอฟต์แวร์

#### 2.1.2 Agile Software Development

Agile Software Development คือแนวคิดที่เกิดจากการที่พบจากการพัฒนาซอฟต์แวร์ในรูปแบบเดิมนั้นมีข้อเสียมากมาย เช่น ความพึงพอใจของลูกค้า เวลาในการพัฒนา ต้นทุน เป็นต้น ดังนั้นแนวคิดของ Agile จะเน้นความพึงพอใจของลูกค้าเป็นหลัก โดย Agile จะมีหลักการทั้งหมด 12 ข้อดังนี้

- 1) เน้นความพึงพอใจของลูกค้าโดยการส่งมอบซอฟต์แวร์ที่ใช้งานได้อย่างต่อเนื่อง
- 2) ยอมรับการเปลี่ยนแปลงของความต้องการของลูกค้า
- 3) ส่งมอบซอฟต์แวร์ที่ใช้งานได้บ่อยๆ
- 4) ลูกค้าและนักพัฒนาต้องทำงานร่วมกัน
- 5) ทีมพัฒนามีอำนาจในการตัดสินใจ
- 6) การติดต่อสื่อสารควรเป็นแบบพบหน้า
- 7) วิธีที่ใช้วัดความก้าวหน้าของงานจะวัดให้ส่วนของซอฟต์แวร์
- 8) พัฒนาแบบค่อยเป็นค่อยไปไม่พัฒนาครั้งละมากๆ
- 9) มีความสนใจในเทคโนโลยีใหม่ๆ
- 10) ทำทุกอย่างบนพื้นฐานของความเรียบง่ายไม่ซับซ้อน
- 11) ทีมมีความรับผิดชอบในงานของตัวเอง
- 12) ทีมมีการแลกเปลี่ยนความคิดเห็นเกี่ยวกับการเพิ่มประสิทธิภาพ

จากหลักการทั้งหมดที่กล่าวมาแสดงให้เห็นถึงการเน้นความพึงพอใจของลูกค้าและการติดต่อสื่อสารกันทั้งในส่วนของภายในทีมพัฒนาและส่วนของลูกค้า ในเรื่องของการนำหลักการของ Agile มาใช้จะเป็นไปตามที่แต่ละทีมในการเลือกและปรับใช้ให้เข้ากับสภาพแวดล้อมของทีมตนเอง ซึ่งการประยุกต์นำหลักการของ Agile ไปใช้นั้นจะทำให้มีลักษณะเด่นของการพัฒนาซอฟต์แวร์ดังนี้

- 1) แบ่งการทำงานออกเป็นรอบๆ โดยแต่ละรอบจะมีการทำขั้นตอนตั้งแต่การวางแผนถึงการทดสอบซอฟต์แวร์
- 2) เน้นการติดต่อสื่อสารโดยตรงทั้งภายในทีมและกับลูกค้า
- 3) ภายในทีมพัฒนาจะมีการพูดคุยเกี่ยวกับงานที่ตัวเองทำในวันที่ผ่านมาและกำลังจะทำในวันนี้
- 4) เน้นคุณภาพของงานการใช้เทคโนโลยีเพื่อเพิ่มประสิทธิภาพในการทำงาน

### 2.1.3 Machine learning

Machine learning เป็นศาสตร์ที่กล่าวการทำให้คอมพิวเตอร์สามารถเรียนรู้ได้เองโดยทำให้คอมพิวเตอร์ที่สามารถทำการทำนายข้อมูลได้ปราศจากการโปรแกรมไว้ล่วงหน้าโดยมีใช้หลักการทางสถิติเข้ามาช่วย การจำแนกประเภทของ Machine learning ตามขอบเขตของปัญหาจะแบ่งได้ 3 ประเภทหลักๆคือ supervised learning unsupervised learning และ reinforcement learning ซึ่งแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

#### 2.1.3.1 Supervised learning

รูปแบบของ supervised learning จะมีลักษณะเด่นคือจำเป็นต้องมีการนำข้อมูลตัวอย่างใส่ให้กับโปรแกรมก่อนในตัวอย่างของข้อมูลมีการบอกถึงตัวข้อมูลและการบอกค่าข้อมูลนั้นคือ สิ่งใด เพื่อให้โปรแกรมเรียนรู้จากตัวอย่างข้อมูลและใช้ในการทำนายข้อมูลใหม่ๆที่เข้ามาในโปรแกรม

#### 2.1.3.2 Unsupervised learning

รูปแบบของ unsupervised learning จะเป็นรูปแบบที่ตรงข้ามกับ supervised learning คือไม่มีการให้ตัวอย่างข้อมูลกับโปรแกรมดังนั้นการทำงานจะเป็นรูปแบบของการหาความสัมพันธ์ระหว่างข้อมูลที่ได้รับมาในส่วนของความแตกต่างของตัวข้อมูล ความเหมือนของตัวข้อมูล ตัวอย่างการใช้งานในส่วนใหญ่จะเป็นการใช้เพื่อจัดกลุ่มของข้อมูล

#### 2.1.3.3 Reinforcement learning

รูปแบบของ reinforcement learning จะเป็นรูปแบบที่เน้นการเรียนรู้จากสภาพแวดล้อมและเลือกที่จะได้ตอบอย่างไรตัวอย่างเช่น การเล่นเกม OX ที่โปรแกรมต้องคอยหาวิธีการที่จะชนะโดยอ้างอิงจากการเล่นของผู้เล่นในแต่ละรอบ

### 2.1.4 Natural language understanding

Natural language understanding จะกล่าวถึงการทำให้คอมพิวเตอร์สามารถเข้าใจถึงภาษาของมนุษย์มีขั้นตอนคือ การวิเคราะห์ทางองค์ประกอบ การวิเคราะห์ทางไวยากรณ์ การวิเคราะห์ทางความหมาย การวิเคราะห์ความหมายโดยรวม และการวิเคราะห์ในทางปฏิบัติซึ่งรายละเอียดของแต่ละขั้นตอนมีดังต่อไปนี้

#### 2.1.4.1 การวิเคราะห์ทางองค์ประกอบ

การวิเคราะห์ทางองค์ประกอบเป็นการวิเคราะห์เพื่อแยกย่อยคำออกเป็นหน่วยย่อยที่สุดเพื่อเตรียมพร้อมสำหรับขั้นตอนต่อไป

#### 2.1.4.2 การวิเคราะห์ทางไวยากรณ์

การวิเคราะห์ทางไวยากรณ์เป็นการวิเคราะห์เพื่อตรวจสอบความถูกต้องของไวยากรณ์โดยการระบุหน้าที่ของแต่ละคำเพื่อใช้แสดงความสัมพันธ์ของคำแต่ละคำ

#### 2.1.4.3 การวิเคราะห์ทางความหมาย

การวิเคราะห์ทางความหมายเป็นการวิเคราะห์เพื่อเชื่อมโยงคำและความหมายเข้าด้วยกัน แต่ในขั้นตอนนี้จะไม่มีคำคำนึงถึงคำหรือความหมายของคำโดยรอบ

#### 2.1.4.4 การวิเคราะห์ความหมายโดยรวม

การวิเคราะห์ความหมายโดยรวมเป็นการวิเคราะห์ที่มีการคำนึงถึงความหมายของประโยคก่อนหน้าด้วยเพื่อให้ได้ความหมายที่ถูกต้องของประโยค

#### 2.1.4.5 การวิเคราะห์ในทางปฏิบัติ

การวิเคราะห์ในทางปฏิบัติเป็นการวิเคราะห์โดยจะทำการตีความเพื่อหาความหมายใหม่เพื่อให้ได้ความหมายที่แท้จริงซึ่งจะมีการนำความรู้ของโลกจริงเข้ามาวิเคราะห์ด้วย

### 2.2 เทคโนโลยีที่เกี่ยวข้อง

#### 2.2.1 Bot Framework

Bot framework เป็นเครื่องมือสำหรับการสร้างซอฟต์แวร์เพื่อใช้ในการโต้ตอบกับผู้ใช้ โดยทางนักพัฒนาในสามารถเตรียมคำถามหรือคำตอบสำหรับโต้ตอบได้เอง โดยมีส่วนสำหรับช่วยเหลือในการพัฒนาซอฟต์แวร์คือ ชุดพัฒนาสำหรับสร้างซอฟต์แวร์ ส่วนของการเชื่อมต่อ ช่องทางสำหรับติดต่อกับซอฟต์แวร์ ตัวช่วยการปรับแต่ง และคลังของbot นอกจากนี้ที่กล่าวมาผู้พัฒนาสามารถทดสอบการทำงานของซอฟต์แวร์ที่ใช้ bot framework ในการพัฒนาขึ้นมาด้วยซอฟต์แวร์ในการจำลองการรับส่งข้อมูลที่ทาง microsoft ให้สามารถนำมาใช้ได้โดยไม่เสียเงิน การโต้ตอบระหว่างซอฟต์แวร์กับผู้ใช้จากมาเป็นได้ทั้งข้อความและรูปภาพซึ่งการโต้ตอบอาจจะเป็นได้ในรูปแบบทั้งแบบการโต้ตอบอิสระ และการโต้ตอบโดยการให้ซอฟต์แวร์แนะนำหรือสร้างตัวเลือกให้ผู้ใช้งานเลือกตอบการตามทีซอฟต์แวร์รองรับ การพัฒนาซอฟต์แวร์โดยใช้ bot framework สามารถพัฒนาได้บนภาษา c# และภาษา javascript ที่จะมีชุดพัฒนารองระบบการพัฒนาเพื่อความสะดวกในการพัฒนาหรือถ้านักพัฒนาต้องการใช้ภาษาที่นอกเหนือจากที่กล่าวมาผ่านการติดต่อผ่าน REST API ของทาง framework รายละเอียดของส่วนประกอบของ bot framework มีดังต่อไปนี้

##### 2.2.1.1 ช่องทางสำหรับติดต่อกับซอฟต์แวร์

ช่องทางสำหรับติดต่อนั้นทาง bot framework มีช่องทางหลากหลายให้นักพัฒนาเลือกนำมาใช้เพื่อเป็นตัวกลางในการรับส่งข้อมูลระหว่างผู้ใช้และซอฟต์แวร์ โดยช่องทางต่างๆจะเป็นช่องทางติดต่อผ่านทางซอฟต์แวร์อื่นๆเช่น อีเมล, Slack messages, Facebook messages, Telegram messages เป็นต้น

##### 2.2.1.2 ตัวเชื่อมต่อ

ส่วนนี้เป็นส่วนในการเชื่อมการรับส่งข้อมูลจากส่วนของช่องทางสำหรับติดต่อกับซอฟต์แวร์เข้ากับซอฟต์แวร์เพื่อให้ซอฟต์แวร์สามารถติดต่อสื่อสารกับผู้ใช้ได้นอกจากนี้ส่วนของการเชื่อมต่อจะทำหน้าที่ให้การจัดการกับข้อมูลที่เกี่ยวข้องกับการติดต่อสื่อสารเช่น ชื่อผู้ส่งข้อความ ชื่อผู้รับข้อความ เวลาที่ส่งข้อความ เป็นต้น

### 2.2.1.3 ชุดพัฒนาสำหรับสร้างซอฟต์แวร์

เป็นส่วนที่นักพัฒนานำมาใช้เพื่อพัฒนาซอฟต์แวร์ที่สามารถติดต่อกับผู้ใช้ผ่านทางตัวเชื่อมต่อได้รวมถึงเพิ่มความสะดวกให้แก่ักพัฒนาให้การสร้างการโต้ตอบกับผู้ใช้ในรูปแบบของการโต้ตอบแบบอิสระหรือการโต้ตอบที่มีการแนะนำโดยตัวซอฟต์แวร์

### 2.2.1.4 ตัวช่วยการปรับแต่ง

เป็นที่ทำให้นักพัฒนาได้รับความสะดวกสบายในการนำซอฟต์แวร์เชื่อมต่อกับช่องทางต่างๆ โดยนักพัฒนาสามารถปรับแต่งการเชื่อมต่อได้เองรวมถึงนักพัฒนาสามารถทดสอบการเชื่อมต่อของซอฟต์แวร์ได้

### 2.2.1.5 คลัง bot

เป็นส่วนที่นักพัฒนาสามารถทดลองติดต่อกับซอฟต์แวร์ของนักพัฒนาอื่นๆ หรือหากหากต้องการนักพัฒนาสามารถนำซอฟต์แวร์เข้าไปในคลังเพื่อให้นักพัฒนาหรือบุคคลทั่วไปสามารถทดสอบการทำงานของซอฟต์แวร์ได้

## 2.2.2 LUIS : Language Understanding Intelligent Service

LUIS เป็นบริการจาก Microsoft ที่ให้บริการเกี่ยวกับการช่วยเหลือให้นักพัฒนาสามารถทำการเพิ่มส่วนของ language understanding ในซอฟต์แวร์ของทางผู้พัฒนาได้สะดวกสบายขึ้น LUIS เป็นเทคโนโลยีที่ถูกพัฒนาขึ้นบนพื้นฐานของ machine learning และ language understanding จาก Microsoft research ละเป็นส่วนหนึ่งของโครงการ Microsoft cognitive services ในปัจจุบัน LUIS มีการสนับสนุนภาษาดังนี้คือ อังกฤษ ฝรั่งเศส อิตาลี สเปนและจีน ในส่วนของการปรับแต่งผู้ใช้สามารถปรับแต่งส่วนที่สำคัญได้ดังนี้

### 2.2.2.1 ความตั้งใจ

ผู้ใช้สามารถสร้างความตั้งใจและกำหนดชื่อให้กับความตั้งใจนั้นโดยเมื่อมีการรับข้อมูลเข้ามาจะมีการประมวลผลและให้คะแนนกับทุกความตั้งใจเพื่อแสดงถึงความเป็นไปได้ที่ข้อมูลจะแสดงถึงความตั้งใจนั้น ผู้ใช้จำเป็นต้องทำการสอนด้วยการใส่ข้อมูลตัวอย่างและทำการบอกถึงความตั้งใจของข้อมูลนั้นก่อนเพื่อให้เกิดการเรียนรู้และเพิ่มความแม่นยำในการทำนายข้อมูลใหม่ๆต่อไป

### 2.2.2.2 เอกลักษณ์

ผู้ใช้สามารถระบุเอกลักษณ์ของคำในประโยคลงไปได้ โดยผู้ใช้สามารถนำเอกลักษณ์ไปใช้เพื่อรวมกับความตั้งใจในการเพิ่มความแม่นยำในการทำนายรวมเป็นการระบุว่าข้อมูลนั้นเป็นผู้ใช้ให้ความสนใจ ในการระบุเอกลักษณ์ผู้ใช้งานสามารถระบุโดยการระบุไปที่คำโดยตรงหรือผู้ใช้สามารถนำเอกลักษณ์ที่ทาง Microsoft เปิดให้นำมาใช้งานมาใช้งานก็ได้ เอกลักษณ์สามารถแบ่งได้เป็น 3 แบบคือ เอกลักษณ์แบบที่มีลำดับชั้นซึ่งเป็นเอกลักษณ์ที่สามารถแบ่งเป็นเอกลักษณ์ย่อยได้อีก เอกลักษณ์แบบประกอบซึ่งเป็นเอกลักษณ์ที่เกิดจากการนำเอกลักษณ์จำนวนมากกว่า 1 เอกลักษณ์มารวมกัน เอกลักษณ์ปกติเป็นเอกลักษณ์ที่ไม่เป็นไปตามทั้ง 2 แบบข้างต้น

### 2.2.2.3 ลักษณะของวลี

เป็นการเพิ่มลักษณะของวลีเข้าไปเพื่อช่วยการทำงานของ LUIS ในการตรวจจับเอกลักษณ์ได้ดีขึ้นในการเพิ่มลักษณะของวลีจะต้องทำการเพิ่มตามรูปแบบที่ทาง LUIS กำหนดไว้

### 2.2.2.4 คุณสมบัติของวลี

เป็นการเพิ่มคุณสมบัติให้วลีเพิ่มช่วยการทำงานของ LUIS โดยการใส่ข้อมูลของวลีที่มีคุณสมบัติเหมือนหรือใกล้เคียงกันซึ่งทำให้ LUIS ตรวจจับได้แม่นยำขึ้น

## 2.2.3 ASP.NET SignalR

ASP.NET SignalR เป็นไลบรารีสำหรับการพัฒนาเพื่อการติดต่อสื่อสารของเซิร์ฟเวอร์และส่วนของผู้ใช้งาน โดยเน้นการรับส่งข้อมูลที่รวดเร็วทั้งยังให้ในส่วนเซิร์ฟเวอร์สามารถส่งข้อมูลไปที่ส่วนของผู้ใช้ได้โดยไม่ต้องมีการร้องขอข้อมูลจากทางผู้ใช้งานเพียงแต่มีการเชื่อมต่อกันระหว่างเซิร์ฟเวอร์และผู้ใช้งาน โดย SignalR มีความสามารถเพิ่มเติมคือ การจัดการสำหรับการเชื่อมต่อ การเชื่อมต่อให้ลักษณะของกลุ่มและการยืนยันหรือการให้สิทธิ์ในการเชื่อมต่อ ในการติดต่อทาง SignalR ไลบรารีมีการนำหลักการ Remote Procedure Call (RPC) มาใช้ในการติดต่อสื่อสารซึ่งหลักการดังกล่าวทำให้ทั้งฝั่งของเซิร์ฟเวอร์และฝั่งของเบราว์เซอร์สามารถเรียกให้อีกฝั่งทำงานได้ โดย SignalR รองรับการรับส่งข้อมูลผ่านช่องทางดังต่อไปนี้

### 2.2.3.1 Web sockets

Web sockets เป็นเทคโนโลยีเพื่อการติดต่อสื่อสารระหว่างเบราว์เซอร์ของผู้ใช้กับส่วนของเซิร์ฟเวอร์โดยจะทำการเปิดเซสชันทิ้งไว้เพื่อให้ทั้งสองฝั่งสามารถสื่อสารกันได้ทันทีเมื่อต้องการโดยที่ฝั่งของเบราว์เซอร์ของผู้ใช้ไม่ต้องทำการส่งการร้องขอเพื่อตรวจสอบว่าข้อมูลใหม่เข้ามาหรือไม่ตลอดเวลา

### 2.2.3.2 การส่งเหตุการณ์โดยเซิร์ฟเวอร์

ในกรณีที่ทางเบราว์เซอร์ของผู้ใช้ไม่รองรับการทำงานด้วย web sockets ไลบรารีของ SignalR จะทำการเปลี่ยนการเชื่อมต่อของผู้ใช้กับเซิร์ฟเวอร์เป็นการส่งเหตุการณ์โดยเซิร์ฟเวอร์ โดยการส่งข้อมูลในรูปแบบนี้มีความแตกต่างจาก web sockets ในส่วนของที่ทางการส่งเหตุการณ์โดยเซิร์ฟเวอร์จะไม่สามารถรับข้อมูลได้ทำได้เพียงกับส่งข้อมูลจากเซิร์ฟเวอร์ได้หาผู้ใช้เท่านั้น

### 2.2.3.3 Forever frames

ในกรณีที่ทางเบราว์เซอร์ของผู้ใช้ไม่รองรับการทำงานด้วยการส่งเหตุการณ์โดยเซิร์ฟเวอร์ไลบรารีจะทำการเปลี่ยนวิธีการติดต่อสื่อสารมาเป็นรูปแบบของ forever frames ซึ่งจะมีการสร้าง iframe ในส่วนของผู้ใช้โดย iframe ดังกล่าวจะทำหน้าที่ในการรับส่งข้อมูลตลอดเวลา

#### 2.2.3.4 Long polling

ในกรณีที่ทางเบราว์เซอร์ของผู้ใช้ไม่รองรับการทำงานด้วย forever frames โไลบรารีของ SignalR จำการเป็นมาใช้ในการรับส่งข้อมูลในรูปแบบนี้ซึ่งการรับส่งข้อมูลในรูปแบบนี้ทางเบราว์เซอร์จะทำการส่งการร้องขอข้อมูลไปที่เซิร์ฟเวอร์โดยทางเซิร์ฟเวอร์จะทำการตรวจสอบเพื่อหาว่ามีข้อมูลใหม่หรือไม่ในกรณีที่ไม่มีข้อมูลให้ทางเซิร์ฟเวอร์จะตอบกลับการร้องขอนั้นทันที ในกรณีที่เซิร์ฟเวอร์ตรวจไม่พบว่ามีข้อมูลใหม่ทางเซิร์ฟเวอร์จะทำการรอนกว่าจะมีข้อมูลใหม่เข้ามาแล้วจึงทำการตอบกลับการร้องขอนั้น ซึ่งเมื่อเบราว์เซอร์ได้รับการตอบกลับเบราว์เซอร์จะต้องทำการส่งการร้องขอไปที่เซิร์ฟเวอร์ใหม่อีกครั้ง

#### 2.2.4 GitLab

GitLab เป็นซอฟต์แวร์สำหรับการควบคุมการทำงานของ Git โดยคือซอฟต์แวร์สำหรับการทำการควบคุมเวอร์ชันของซอร์สโค้ดซึ่งเป็นซอฟต์แวร์ที่สร้างขึ้นเพื่อจัดการกับซอร์สโค้ดเพื่ออำนวยความสะดวกให้กับพัฒนาสำหรับการทำงานร่วมกันหรือต้องการเก็บเวอร์ชันเก่าของซอร์สโค้ดไว้สำหรับการปกป้องหากซอร์สโค้ดในเวอร์ชันใหม่เกิดความผิดพลาดนักพัฒนายังสามารถทำการย้อนซอร์สโค้ดกับไปยังเวอร์ชันก่อนหน้าได้ ในส่วนของ GitLab จะมีความสามารถเพิ่มเติมอื่นๆสำหรับช่วยเหลือให้นักพัฒนาให้สามารถทำงานได้สะดวกมากยิ่งขึ้นเช่น การแจ้งปัญหาต่างๆกับนักพัฒนาคนอื่น การเปิดหัวข้อสำหรับการพัฒนา การเขียนคลังความรู้ของโครงการนั้น ความสามารถในการรองรับการทำงานด้วยทีมนักพัฒนาระดับองค์กร เป็นต้น

#### 2.2.5 Jenkins

Jenkins เป็นซอฟต์แวร์ที่เกิดขึ้นตามแนวคิดของที่เรียก “Continuous Integration” หรือการรวมโค้ดอย่างต่อเนื่องโดยแนวคิดนี้เกิดจากปัญหาของการพัฒนาซอฟต์แวร์ที่มีนักพัฒนาหลายคนทำให้เกิดความยุ่งยากในการรวมซอร์สโค้ดของแต่ละนักพัฒนาเข้าด้วยกันซึ่งปัญหาที่เกิดจากสาเหตุดังกล่าวมีตัวอย่างเช่น ความผิดพลาดของการรวมซอร์สโค้ดทำให้เกิดปัญหาที่ตัวซอฟต์แวร์เนื่องจากนักพัฒนาแต่ละคนอาจจะทำการพัฒนาในส่วนของนักพัฒนาคนอื่นทำให้อาจจะมีปัญหาหากไม่มีการอัปเดตซอร์สโค้ดของแต่ละคนอย่างต่อเนื่อง เป็นต้นทั้งนี้ Jenkins เป็นซอฟต์แวร์ที่เปิดเผยซอร์สโค้ดทำให้นักพัฒนาสามารถพัฒนาส่วนเสริมต่างๆลงไปเพื่อให้นักพัฒนาคนอื่นๆสามารถนำไปใช้งานเพื่อปรับแต่ง Jenkins ของตนเองให้สามารถทำงานได้ดีขึ้น การติดต่อเพื่อส่งงานหรือปรับแต่งนั้นนักพัฒนาสามารถทำได้โดยผ่านทางเว็บไซต์ REST API หรือ Command line interface ก็ได้ โดยลักษณะเด่นของ Jenkins คือสามารถใช้งานร่วมกับ version control ต่างๆได้ สามารถสั่งให้ทำการทดสอบตามปรับแต่งไว้ รวมถึงการปรับแต่งทางด้านอื่นๆเช่น ด้านความปลอดภัย สิทธิ์ในการเข้าถึงในแต่ละโครงการ เป็นต้น

## 2.2.6 Google sheets

Google sheets เป็นเว็บไซต์สำหรับการจัดการเอกสารลักษณะคล้ายคลึงกับซอฟต์แวร์ Microsoft excel ของทาง Microsoft คือผู้ใช้สามารถจัดการเอกสารได้ผ่านทั้งทางเว็บไซต์หรือผ่านโปรแกรมประยุกต์ก็ได้ ความสามารถหลักของ Google sheets นอกเหนือจากการจัดการเอกสารจะมีในส่วนของการทำงานร่วมกันคนอื่นในการแก้ไขเอกสารในเวลาเดียวกันและสามารถเปิด แก้ไข และบันทึกไฟล์เอกสารในรูปแบบของ Microsoft excel ได้รวมถึงการแปลงไฟล์ไปกลับระหว่าง Microsoft excel และ Google sheets



## บทที่ 3

### วิธีการดำเนินการ

#### 3.1 ศึกษารายละเอียดของขั้นตอนการปล่อยซอฟต์แวร์

การศึกษารายละเอียดของการปล่อยซอฟต์แวร์ทำโดยการศึกษาจากเอกสารที่ทางทีมพัฒนาจัดทำขึ้นมาประกอบการสอบถามรายละเอียดต่างๆจากนักพัฒนาที่เคยได้ทำการควบคุมการปล่อยซอฟต์แวร์โดยขั้นตอนต่างประกอบด้วย

##### 3.1.1 การสร้าง branch ของเวอร์ชัน

ในขั้นตอนนี้ผู้ที่ทำการควบคุมการปล่อยซอฟต์แวร์จะต้องการสร้าง branch สำหรับเวอร์ชันปัจจุบันของซอฟต์แวร์ โดยก่อนหน้านั้นผู้ที่ทำการควบคุมการปล่อยซอฟต์แวร์จะต้องทำการตรวจสอบก่อนว่าทุก commit ล่าสุดต้องถูกรันโดย Jenkins แล้วจึงจะทำการสร้าง branch ได้หาก commit ล่าสุดยังไม่ได้ถูกรันโดย Jenkins ให้ทำการรัน Jenkins ก่อนแล้วจึงสร้าง branch ของเวอร์ชันนั้น

##### 3.1.2 ปรับแต่งและเพิ่มโครงการของเวอร์ชันถัดไปใน Jenkins

ในขั้นตอนนี้ผู้ที่ทำการควบคุมการปล่อยซอฟต์แวร์ทำการสร้างโครงการสำหรับรับรองการพัฒนาในเวอร์ชันถัดไปรวมทั้งการปรับแต่งให้ทำงานได้ถูกต้อง ปรับแต่งโครงการในเวอร์ชันปัจจุบันเพื่อให้ทำงานบน branch ที่สร้างไว้ในขั้นตอนแรกและทำการระงับการทำงานโครงการของทั้ง 2 เวอร์ชันเพื่อไม่ให้มีการทำงานระหว่างการทดสอบ

##### 3.1.3 เพิ่มข้อมูลลงใน Google sheets

ในขั้นตอนนี้ผู้ที่ทำการควบคุมการปล่อยซอฟต์แวร์จะทำการนำข้อมูลเวอร์ชันปัจจุบันและข้อมูลอื่นๆที่เกี่ยวข้องกับการทดสอบเพิ่มลงใน Google sheets เพื่อให้ในนักพัฒนาในทีมเลือกทดสอบและกรอกผลการทดสอบ

##### 3.1.4 ทำการส่งอีเมลให้กับนักพัฒนา

ในขั้นตอนนี้ผู้ที่ทำการควบคุมการปล่อยซอฟต์แวร์จะส่งอีเมลให้กับนักพัฒนาที่ทำการ commit ลงในเวอร์ชันนี้เพื่อให้ นักพัฒนาคนนั้นทำการทดสอบซอฟต์แวร์และรอรับอีเมลผลรวมของการทดสอบว่าผ่านหรือไม่ผ่านจากนักทดสอบของทีม

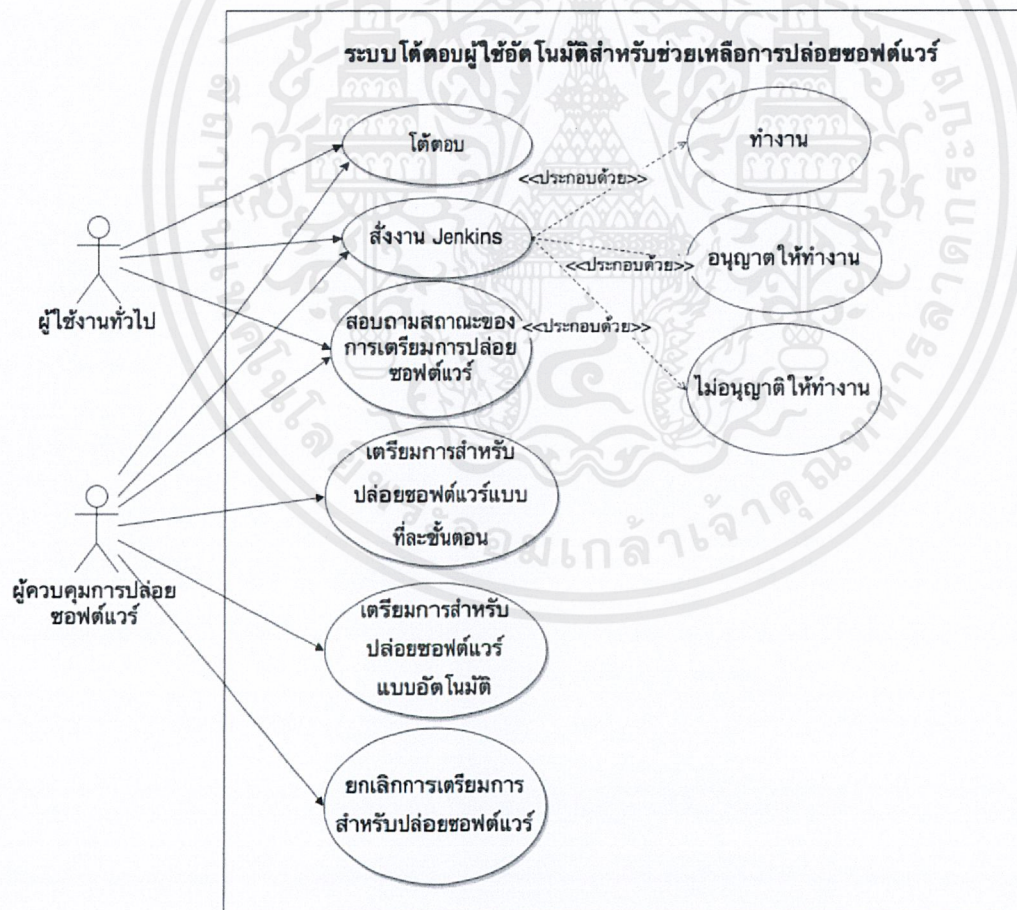
##### 3.1.5 ทำการส่งอีเมลให้กับนักพัฒนา

ในขั้นตอนนี้ผู้ที่ทำการควบคุมการปล่อยซอฟต์แวร์จะส่งอีเมลให้กับนักพัฒนาที่ทำการ commit ลงในเวอร์ชันนี้เพื่อให้ นักพัฒนาคนนั้นทำการทดสอบซอฟต์แวร์และรอรับอีเมลผลรวมของการทดสอบว่าผ่านหรือไม่ผ่านจากนักทดสอบของทีมจากนั้นผู้จัดการของทีมจะทำการตัดสินใจในการปล่อยซอฟต์แวร์ในเวอร์ชันปัจจุบันให้ลูกค้าใช้หรือไม่

### 3.2 ออกแบบโครงสร้างโดยรวมของระบบ

#### 3.2.1 Use Case

- 1) ผู้ใช้งานทั่วไปสามารถโต้ตอบกับระบบได้
- 2) ผู้ใช้งานทั่วไปสามารถสั่งให้ Jenkins ทำงาน อนุญาตหรือไม่อนุญาตให้ทำงานผ่านระบบ
- 3) ผู้ใช้งานทั่วไปสามารถสอบถามระบบเพื่อขอสถานะของการเตรียมการปล่อยซอฟต์แวร์
- 4) ผู้ควบคุมการปล่อยซอฟต์แวร์สามารถสั่งให้เตรียมการสำหรับปล่อยซอฟต์แวร์แบบทีละขั้นตอนได้
- 5) ผู้ควบคุมการปล่อยซอฟต์แวร์สามารถสั่งให้เตรียมการสำหรับปล่อยซอฟต์แวร์แบบอัตโนมัติได้
- 6) ผู้ควบคุมการปล่อยซอฟต์แวร์สามารถสั่งยกเลิกการเตรียมการสำหรับปล่อยซอฟต์แวร์ได้



ภาพที่ 3.1 รูปภาพแสดง Use Case Diagram ของระบบ

## 3.2.2 โครงสร้างของระบบ

### 3.2.2.1 ช่องทางหรือ Channels

เป็นส่วนในการติดต่อกับผู้ใช้โดยมีหลายช่องทางสำหรับติดต่อเช่น อีเมลล์, Facebook messages, Slack messages เป็นต้น นอกจากนี้เป็นช่องทางในการติดต่อกับผู้ใช้แล้วช่องทางต่างๆยังจะทำการนำข้อมูลทั่วไปของผู้ใช้มาด้วยรวมเข้ากับข้อมูลอื่นเช่น ช่องทางที่สื่อสาร เวลาที่ส่งข้อความ ภาษา เป็นต้น

### 3.2.2.2 ตัวเชื่อมต่อหรือ Connector

เป็นส่วนที่รับข้อมูลมาจากช่องทางแล้วทำการตรวจสอบว่าข้อความนั้นควรจะถูกส่งต่อไปที่ใดเพื่อให้สามารถเชื่อมต่อกับหลายๆช่องทางพร้อมกัน

### 3.2.2.3 Bot

เป็นส่วนหลักในการโต้ตอบกับผู้ใช้ โดยในส่วนนี้นักพัฒนาจะต้องการเขียนโปรแกรมเพื่อควบคุมการโต้ตอบกับผู้ใช้ ในส่วนนี้จะรับข้อความรวมถึงข้อมูลที่เกี่ยวข้องจากตัวเชื่อมต่อแล้วทำงานตามที่นักพัฒนาได้ทำการโปรแกรมไว้

### 3.2.2.4 LUIS

เป็นส่วนเพิ่มความสามารถให้ bot ในด้านความสามารถในการเข้าใจภาษามนุษย์ในส่วนนี้ความแม่นยำจะขึ้นอยู่กับการให้ข้อมูลตัวอย่างกับ LUIS โดย bot จะทำการส่งข้อความมาให้ LUIS แล้ว LUIS จะทำการทำนายจากข้อมูลตัวอย่างที่เคยได้รับมารวมกับการปรับแต่งแล้วส่งผลลัพธ์กลับไปให้ bot

### 3.2.2.5 เซิร์ฟเวอร์หรือ Server

เป็นส่วนหลักในการทำงานจริง โดยในส่วนนี้จะรับคำสั่งจากส่วนของ bot แล้วทำการทำงานตามที่ได้รับคำสั่งรวมถึงเก็บข้อมูลที่ใช้ในการทำงาน

### 3.2.2.6 Google sheets

เป็นส่วนที่จะถูกเรียกใช้โดยส่วนของเซิร์ฟเวอร์ที่เพื่อทำการแก้ไขและทำการดึงข้อมูลสำหรับการปล่อยซอฟต์แวร์

### 3.2.2.7 Jenkins

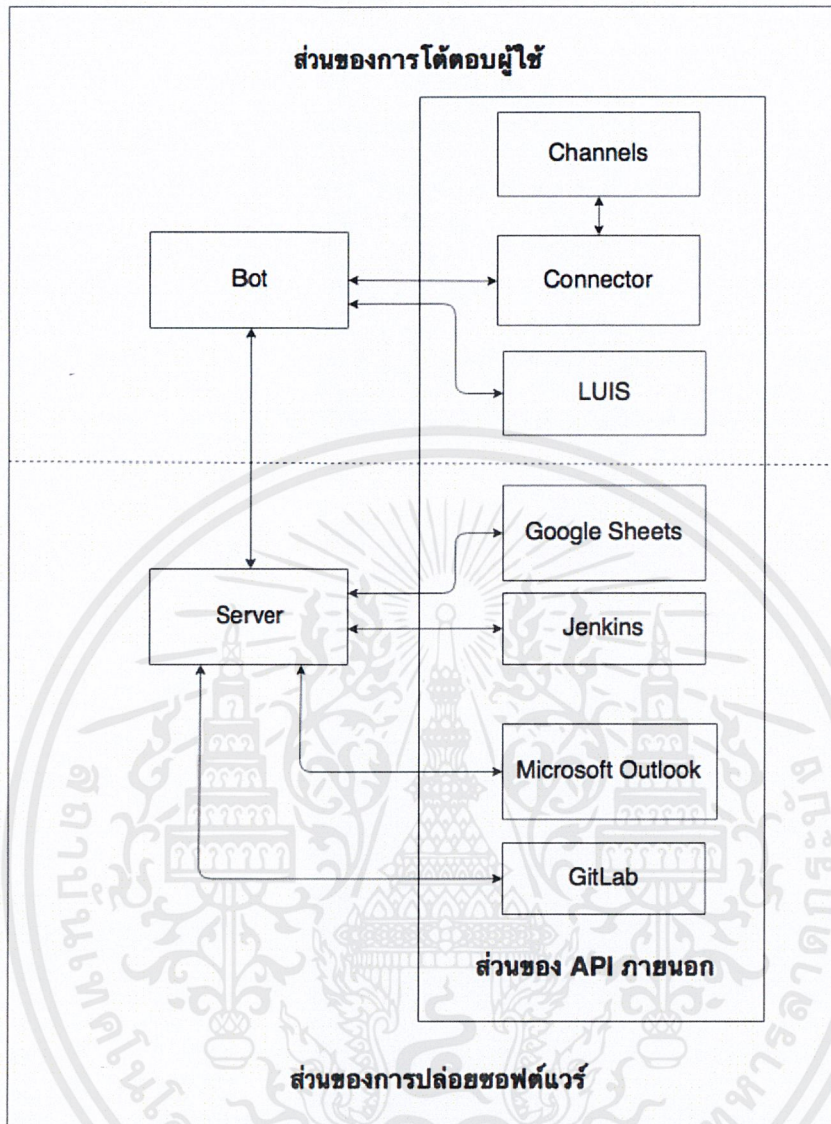
เป็นส่วนที่ช่วยควบคุมการพัฒนาซอฟต์แวร์ โดย Jenkins จะทำการเก็บข้อมูลต่างๆของการพัฒนาซอฟต์แวร์รวมถึงสามารถทำงานตามคำสั่งที่ถูกเขียนไว้ก่อนหน้าเมื่อมีการสั่งให้ Jenkins ได้ทำ events ต่างๆตามที่ได้รับปรับแต่ง

### 3.2.2.8 Microsoft Outlook

เป็นส่วนที่ใช้ในการส่งอีเมลล์ให้กับพัฒนาเพื่อเป็นการแจ้งเตือนให้กับนักพัฒนาในทีมได้รับรู้ถึงการปล่อยเวอร์ชันใหม่และทำการทดสอบก่อนมีการปล่อยเวอร์ชันใหม่

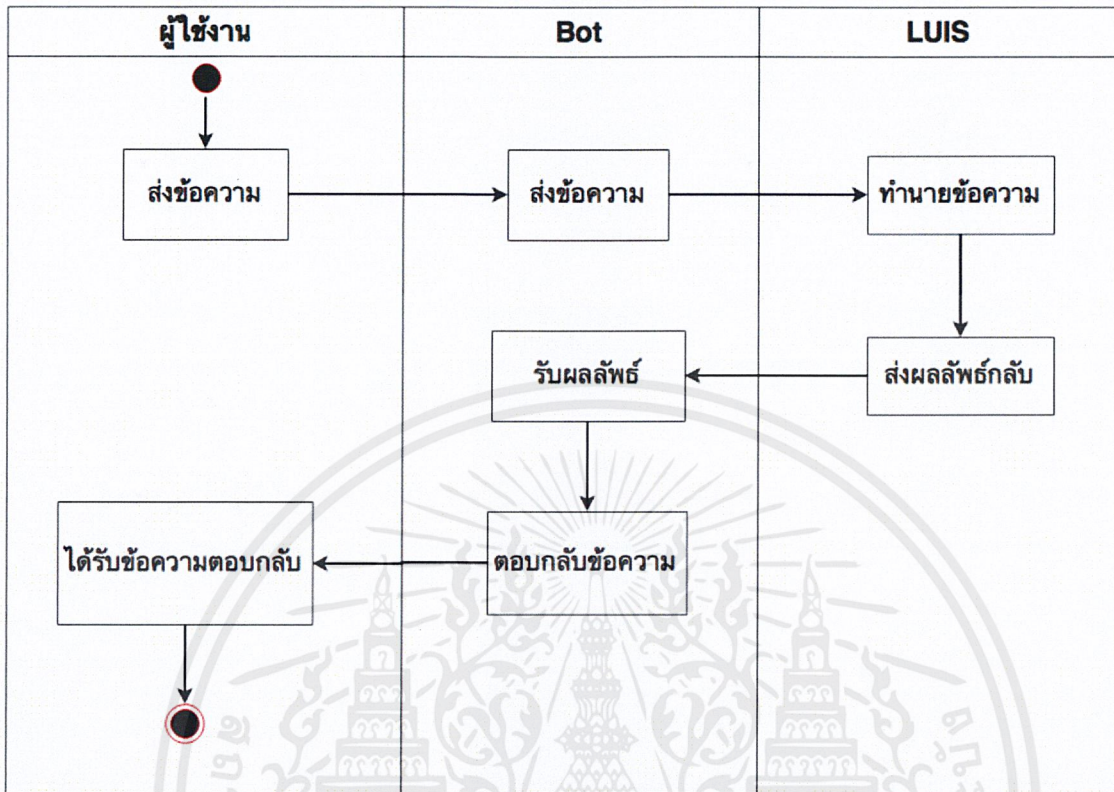
### 3.2.2.9 GitLab

เป็นส่วนที่จัดการซอร์สโค้ดของซอฟต์แวร์ที่จะมีการทำการปล่อยให้ผู้ใช้งานได้ใช้งาน โดยในส่วนนี้เซิร์ฟเวอร์จะทำการดึงข้อมูลและทำการสั่งให้ GitLab ทำงานตามที่ถูกผู้ใช้งาน



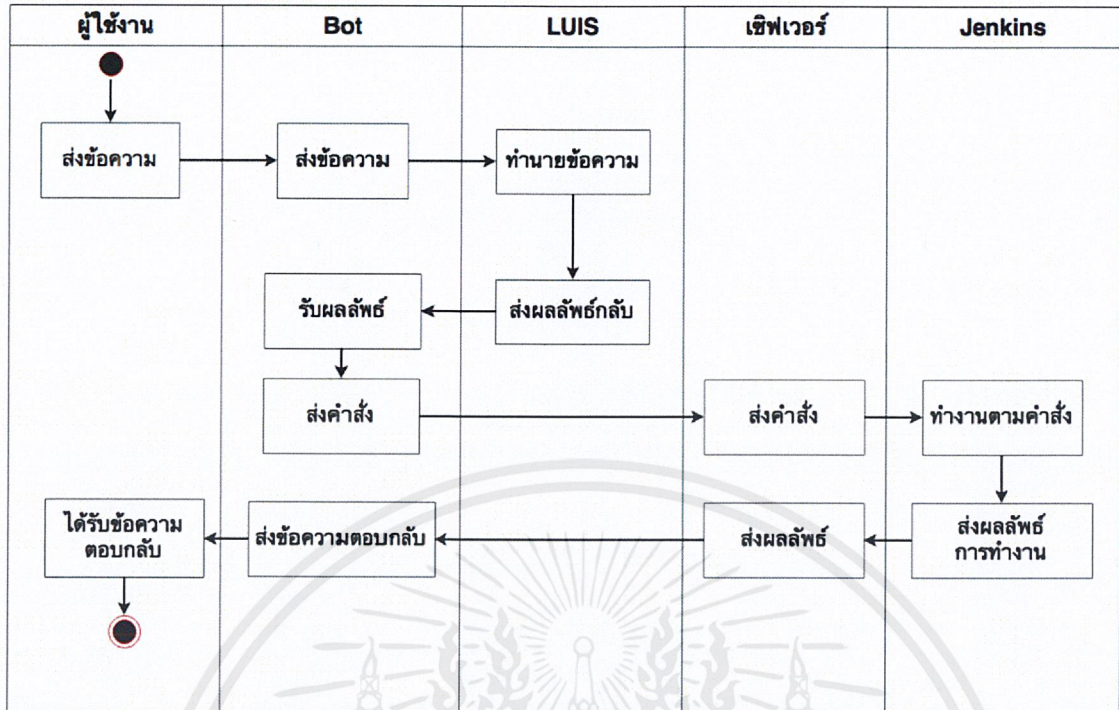
ภาพที่ 3.2 รูปภาพแสดงโครงสร้างของระบบ

### 3.2.3 แผนภาพกิจกรรม



ภาพที่ 3.3 แผนภาพกิจกรรมของการโต้ตอบกับระบบ

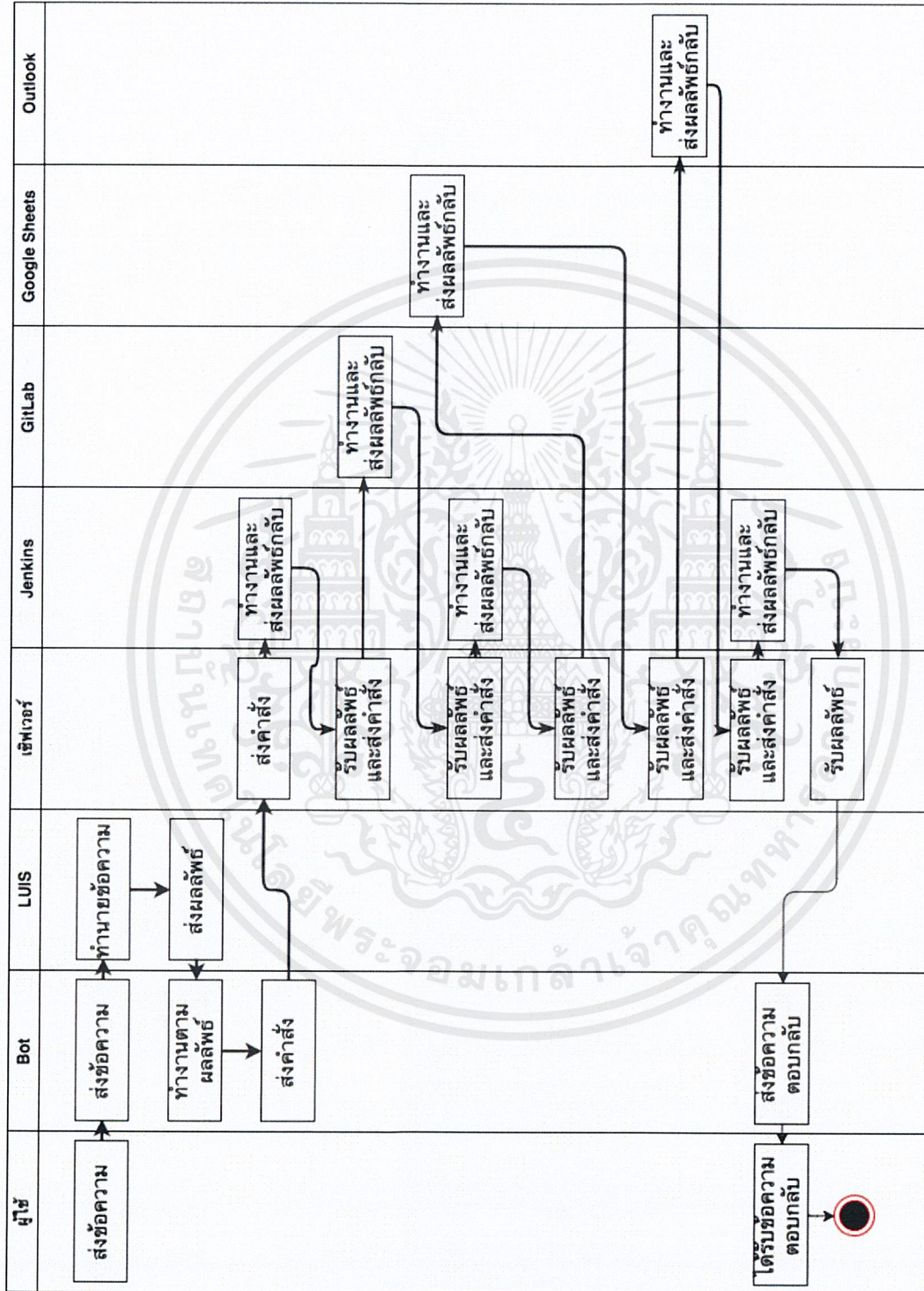
การทำงานในส่วนของการโต้ตอบจะเริ่มจากผู้ใช้งานทำการส่งข้อความเข้าสู่ bot จากนั้น bot จะทำการส่งข้อความต่อไปให้ LUIS เพื่อทำการทำนายความตั้งใจที่ผู้ใช้ต้องการจะสื่อสารและส่งผลลัพธ์กลับไปหา bot เมื่อได้ผลลัพธ์กลับมา bot จะทำสร้างตอบกลับข้อความนั้นโดยการนำข้อความตอบกลับที่ถูกรวบรวมไว้สำหรับแต่ละความตั้งใจของผู้ใช้ เมื่อผู้ใช้ได้รับข้อความตอบกลับจาก bot การทำงานจะสิ้นสุด



ภาพที่ 3.4 แผนภาพกิจกรรมแสดงการสั่งงาน Jenkins ผ่านระบบ

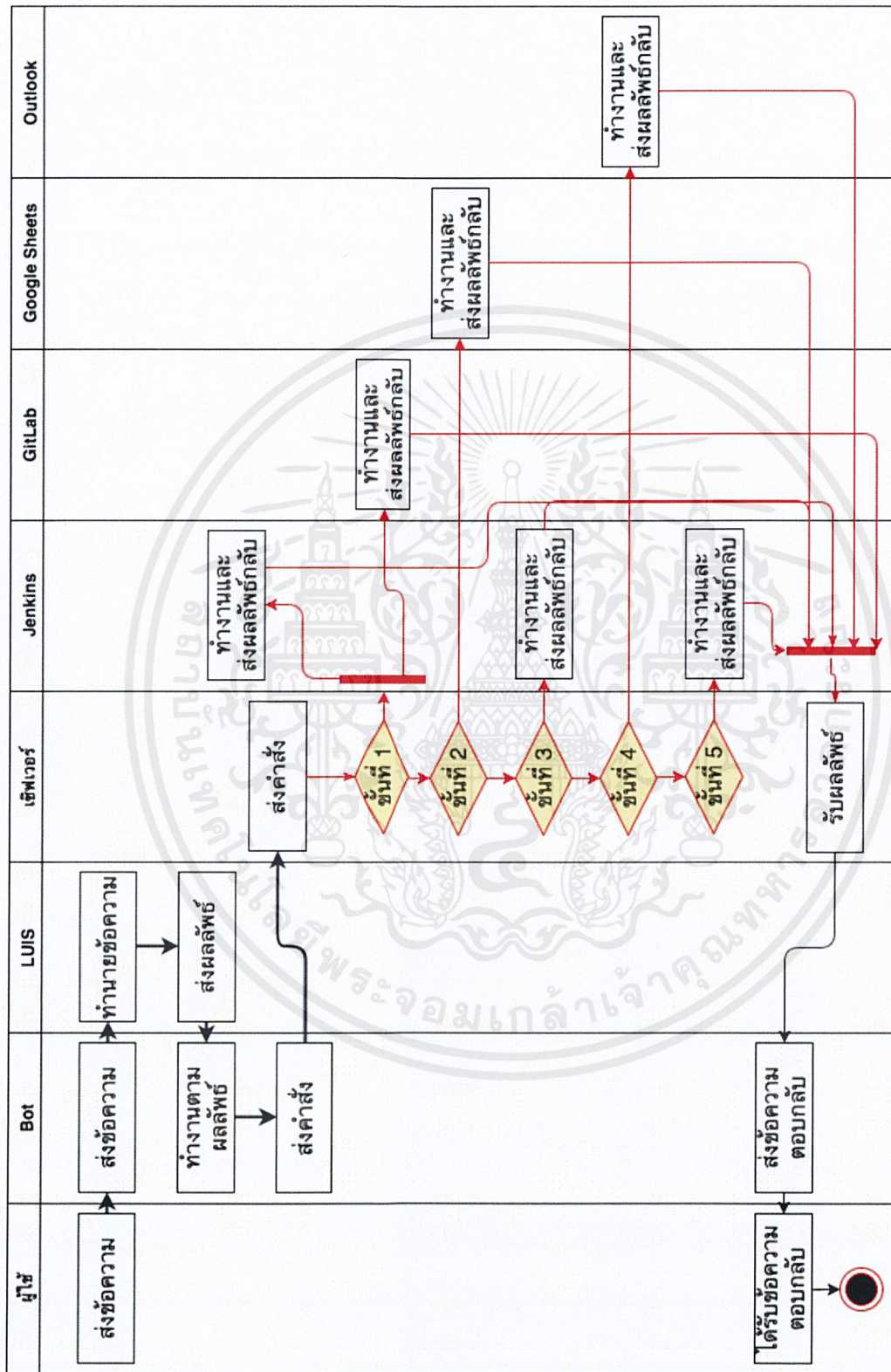
การทำงานในส่วนของการสั่งงาน Jenkins เริ่มต้นด้วยการส่งข้อความจากผู้ใช้งานมาหา bot จากนั้น bot จะส่งข้อความไปให้ LUIS เมื่อได้รับข้อความ LUIS จะทำการทำนายข้อความเพื่อหาความตั้งใจของผู้ใช้จากนั้นจึงทำการส่งผลลัพธ์จากการทำนายกลับไปให้ในส่วน of bot เมื่อได้รับผลลัพธ์มา bot จะทำการเปลี่ยนผลลัพธ์เป็นคำสั่งและทำการส่งคำสั่งให้ส่วน of เซิร์ฟเวอร์ ในส่วน of เซิร์ฟเวอร์ จะทำการแปลงคำสั่งและส่งไปให้ส่วน of Jenkins เมื่อได้รับคำสั่ง Jenkins จะทำงานและส่งผลลัพธ์กลับไปให้เซิร์ฟเวอร์ เซิร์ฟเวอร์ จะทำการส่งต่อผลลัพธ์ไปให้ bot เมื่อได้รับผลลัพธ์การทำงาน bot จะทำการตอบข้อความกลับไปตามผลลัพธ์ของการทำงานที่ได้รับมา เมื่อผู้ใช้ได้รับข้อความตอบกลับจะเป็นการจบการทำงาน

# สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

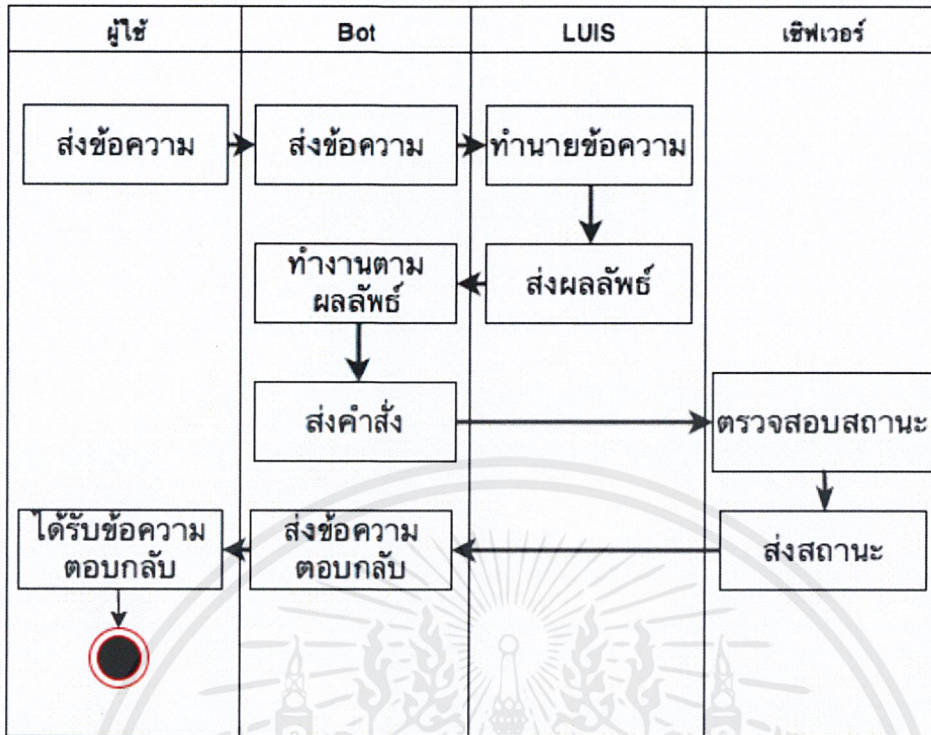


ภาพที่ 3.5 แผนภาพกิจกรรมแสดงการส่งงานเพื่อเตรียมการสำหรับปล่อยซอฟต์แวร์แบบที่ละเอียดขึ้น

148622



ภาพที่ 3.6 แผนภาพกิจกรรมแสดงการส่งงานเพื่อเตรียมการสำหรับปล่อยซอฟต์แวร์แบบที่ละเอียดโน้มนำ

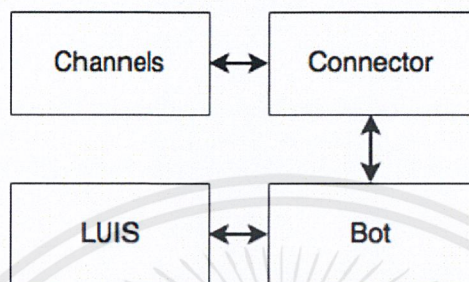


ภาพที่ 3.7 แผนภาพกิจกรรมแสดงการตรวจสอบสถานะการเตรียมการปล่อยซอฟต์แวร์

ภาพที่ 3.5 และภาพที่ 3.6 จะเป็นการทำงานที่คล้ายคลึงกันโดยเริ่มต้นด้วยผู้ใช้ทำการส่งข้อความมาให้ bot และ bot จะทำการส่งต่อให้ LUIS เพื่อทำการทำนายความตั้งใจของผู้ใช้ส่งกลับมาให้ bot และ bot จะทำการส่งคำสั่งไปที่กับฝั่งเซิร์ฟเวอร์ในกรณีที่เป็นภาพที่ 3.5 เซิร์ฟเวอร์จะทำการตรวจสอบและดำเนินการการปล่อยซอฟต์แวร์ต่อไปอีกหนึ่งขั้นตอนและส่งผลลัพธ์กลับไปให้ผู้ใช้ ผู้ใช้จึงจำเป็นต้องทำการส่งข้อความอีกจนกว่าจะจบการเตรียมการปล่อยซอฟต์แวร์ ในกรณีของภาพที่ 3.6 เซิร์ฟเวอร์จะทำการส่งคำสั่งในการทำงานไปเรื่อยๆจนจบการเตรียมการการปล่อยซอฟต์แวร์ ส่วนในภาพที่ 3.7 จะเป็นการส่งข้อความเพื่อถามถึงขั้นตอนปัจจุบันของการเตรียมการการปล่อยซอฟต์แวร์โดยผู้ใช้จะทำการส่งข้อความมาให้ bot ข้อความจะถูกส่งต่อไปยัง LUIS เมื่อทำนายข้อความเสร็จ LUIS จะทำการส่งกลับมาและ bot จะทำการส่งคำสั่งไปที่เซิร์ฟเวอร์เมื่อเซิร์ฟเวอร์ได้รับคำสั่งจะทำการส่งสถานะกลับมาให้ bot จะทำการส่งข้อความกลับไปหาผู้ใช้

### 3.3 พัฒนาระบบในส่วนของการโต้ตอบผู้ใช้อัตโนมัติ

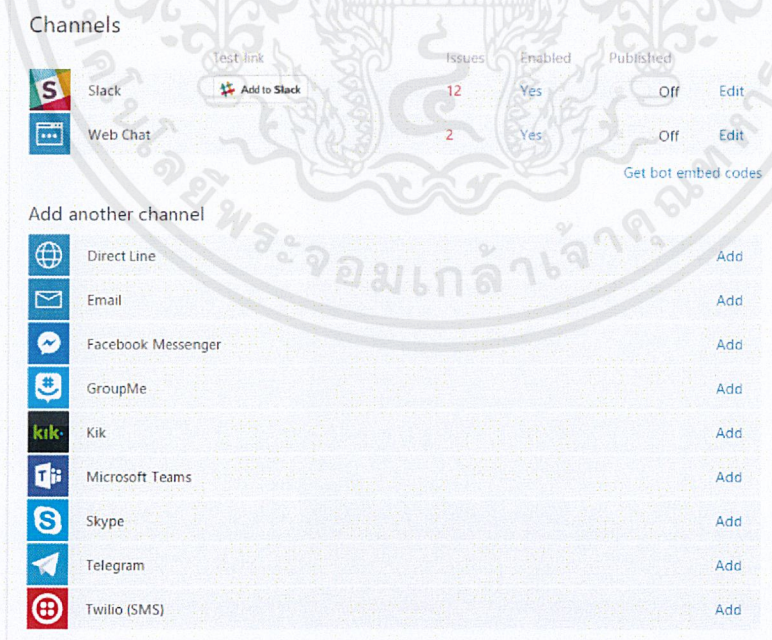
การพัฒนาระบบในส่วนของการโต้ตอบผู้ใช้อัตโนมัติจะเกี่ยวกับการพัฒนาเพื่อให้ระบบสามารถโต้ตอบกับผู้ใช้งานได้ด้วยตนเอง เพื่อรับคำสั่งและทำงานตามคำสั่งนี้รวมถึงอาจจะมีการส่งคำสั่งต่อไปยังส่วนของเซิร์ฟเวอร์



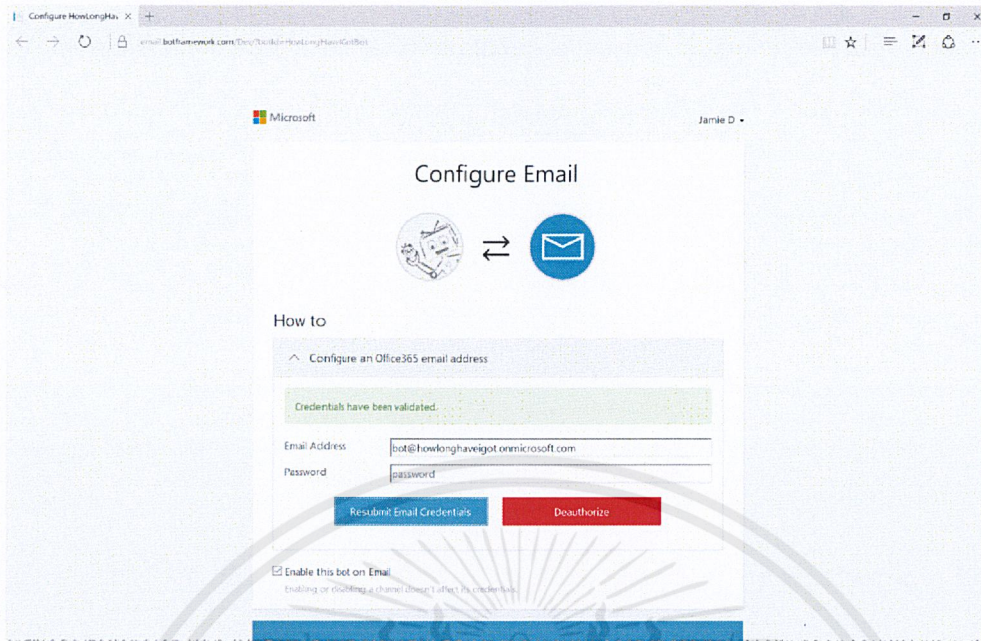
ภาพที่ 3.8 รูปภาพแสดงระบบในส่วนของการโต้ตอบผู้ใช้อัตโนมัติ

#### 3.3.1 ช่องทาง หรือ Channels

ในส่วนของช่องทางเป็นส่วนที่ผู้ใช้จะใช้ในการโต้ตอบกับระบบโดยการพัฒนาระบบในส่วนนี้จะไม่มีการเขียนโปรแกรมขึ้นมาเพื่อทำการควบคุมแต่จะเป็นการปรับแต่งเพื่อในระบบสามารถรับส่งข้อความกับผู้ใช้ได้ในการปรับแต่งสามารถปรับแต่งให้ระบบสามารถทำการโต้ตอบกับผู้ใช้ผ่านหลากหลายช่องทางซึ่งการปรับแต่งของแต่ละช่องทางจะไม่เหมือนกันทั้งหมด



ภาพที่ 3.9 รูปภาพแสดงช่องทางในการติดต่อกับผู้ใช้



ภาพที่ 3.10 รูปภาพแสดงการปรับแต่งช่องทางในการติดต่อกับผู้ใช้  
(ที่มา : <https://msdnshared.blob.core.windows.net/media/2016/05/email-config.png>)

### 3.3.2 ตัวเชื่อมต่อ หรือ Connector

ในส่วนนี้เป็นส่วนที่บริการที่ทาง Microsoft เปิดใช้งานได้ตัวเชื่อมต่อเป็นบริการสำหรับการติดต่อการกลับส่วนของช่องทางเพื่อทำให้ bot สามารถรับข้อความและส่งข้อความกลับไปหาผู้ใช้ได้ โดยของตัวเชื่อมต่อจะเชื่อมต่อกับช่องทางและ bot ในรูปแบบของการส่งข้อมูลแบบ REST API

Attachments		Show/Hide	List Operations	Expand Operations
GET	/v3/attachments/{attachmentId}			GetAttachmentInfo
GET	/v3/attachments/{attachmentId}/views/{viewId}			GetAttachment
Conversations		Show/Hide	List Operations	Expand Operations
POST	/v3/conversations			CreateConversation
POST	/v3/conversations/{conversationId}/activities			SendToConversation
DELETE	/v3/conversations/{conversationId}/activities/{activityId}			DeleteActivity
POST	/v3/conversations/{conversationId}/activities/{activityId}			ReplyToActivity
PUT	/v3/conversations/{conversationId}/activities/{activityId}			UpdateActivity
GET	/v3/conversations/{conversationId}/members			GetConversationMembers
GET	/v3/conversations/{conversationId}/activities/{activityId}/members			GetActivityMembers
POST	/v3/conversations/{conversationId}/attachments			UploadAttachment

ภาพที่ 3.11 รูปภาพแสดงบริการของกตัวเชื่อมต่อ

```

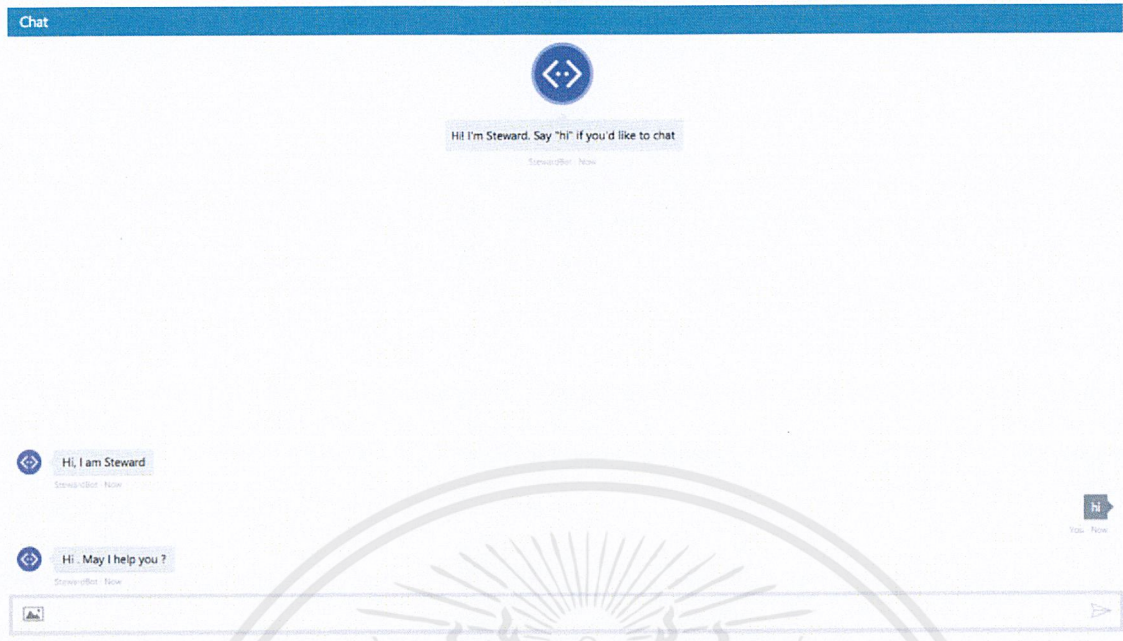
{
  "type": "message",
  "locale": "en-US",
  "channelID": "email",
  "from": {
    "id": "mybot@gmail.com",
    "name": "My bot"
  },
  "recipient": {
    "id": "joe@gmail.com",
    "name": "Joe Doe"
  },
  "conversation": {
    "id": "123123123123",
    "topic": "awesome chat"
  },
  "channelData": {
    "htmlBody": "<html><body style = \"font-fami",
    "subject": "Super awesome message subject",
    "importance": "high"
  }
}

```

ภาพที่ 3.12 รูปภาพแสดงข้อมูลที่รับส่งผ่านตัวเชื่อมต่อ

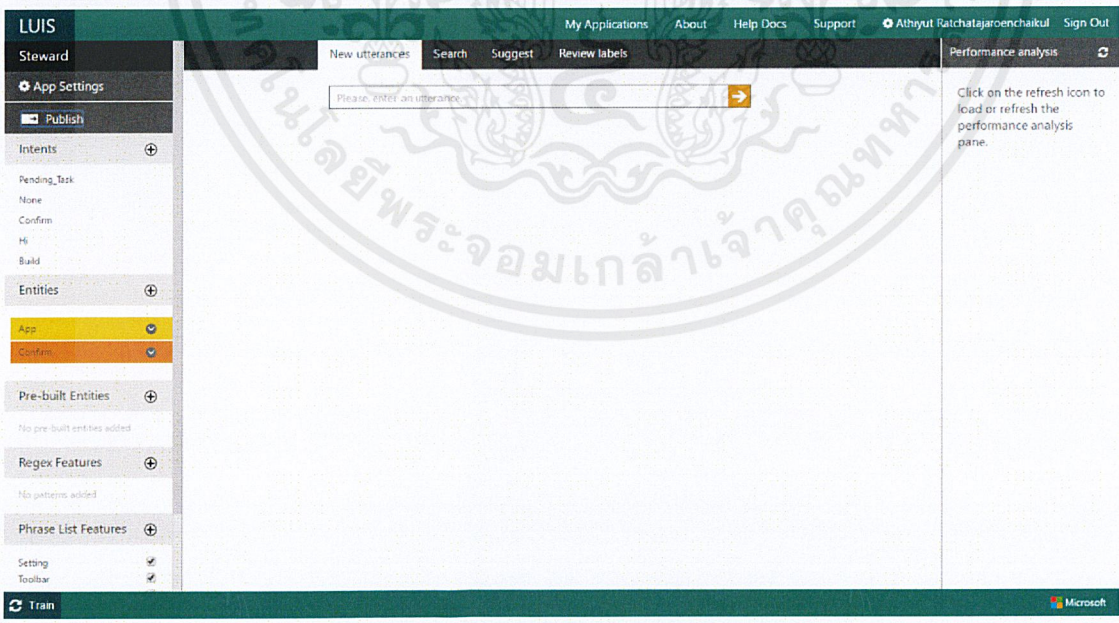
### 3.3.3 Bot

ในส่วนนี้เป็นส่วนหลักของการโต้ตอบกับผู้ใช้โดยในส่วนนี้จะทำการรับข้อความจากผู้ใช้งานผ่านตัวเชื่อมต่อแล้วทำการส่งข้อความนั้นไปให้ในส่วนของ LUIS เพื่อทำนายความตั้งใจของผู้ใช้แล้วจึงทำงานตามที่ผู้ใช้งานต้องการโดยอาจจะมีการส่งคำสั่งไปติดต่อกับส่วนของเซิร์ฟเวอร์เพื่อให้ทำงานกับส่วนอื่นๆที่ถูกติดตั้งไว้ในบริษัทได้ โดยในส่วนนี้ของระบบถูกทำการพัฒนาด้วยภาษา C#



ภาพที่ 3.13 รูปภาพแสดงตัวอย่างการทำงานของ bot  
3.3.4 LUIS

ในส่วนนี้เป็นส่วนของการทำนายข้อความที่ได้รับมาจาก bot โดยผลลัพธ์ของการทำนายจะขึ้นอยู่กับข้อความที่ได้รับมาการเรียนรู้ที่ผ่านมา โดยจะมีการสร้าง ปรับแต่ง และการสอนเพื่อให้ LUIS สามารถทำการทำนายได้ใกล้เคียงหรือตรงตามที่ต้องการซึ่งผลลัพธ์ของการทำนายข้อความจะถูกแปลงออกมาเป็นความตั้งใจของข้อความนั้นและถูกส่งกลับไปในส่วนของ bot



ภาพที่ 3.14 รูปภาพแสดงตัวอย่างหน้าแสดงผลของ LUIS

```

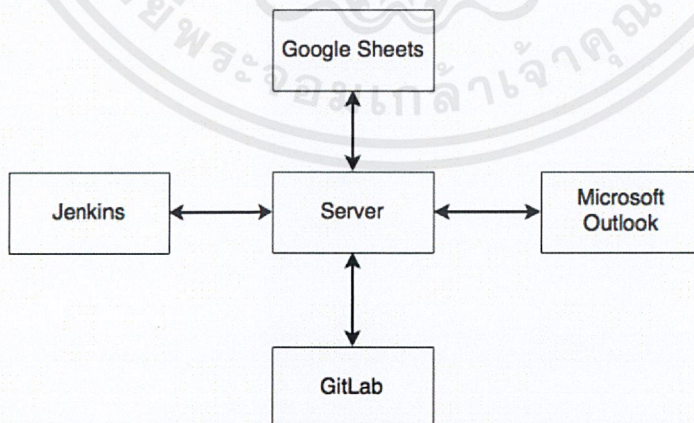
{
  "query": "build toolbar",
  "topScoringIntent": {
    "intent": "Build",
    "score": 0.5950861,
    "actions": [
      {
        "triggered": false,
        "name": "Build",
        "parameters": [
          {
            "name": "App",
            "type": "App",
            "required": true,
            "value": null
          }
        ]
      }
    ]
  },
  "entities": [
    {
      "entity": "toolbar",
      "type": "App::toolbar",
      "startIndex": 6,
      "endIndex": 12,
      "score": 0.9801937
    }
  ],
  "dialog": {
    "prompt": "",
    "parameterName": "App",
    "parameterType": "App",
    "contextId": "23c2722d-93f9-4efa-b1e7-d90ffe905ad4",
    "status": "Question"
  }
}

```

ภาพที่ 3.15 รูปภาพแสดงผลลัพธ์ของการทำนาย

### 3.4 พัฒนาระบบในส่วนของการช่วยเหลือการปล่อยซอฟต์แวร์

การพัฒนาระบบในส่วนของการช่วยเหลือการปล่อยซอฟต์แวร์จะเป็นการพัฒนาเพื่อให้ระบบสามารถทำงานได้จริงโดยในส่วนนี้จะทำการติดต่อเพื่อรับคำสั่งกับส่วนของการโต้ตอบผู้ใช้อัตโนมัติและส่งผลลัพธ์จากการทำงานกลับไป



ภาพที่ 3.16 รูปภาพแสดงระบบในส่วนของการช่วยเหลือการปล่อยซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.1 Google Sheets

Google Sheets เป็นบริการที่ทาง Google เป็นให้ใช้งานโดยไม่เสีย ซึ่งจะสามารถทำการจัดเก็บเอกสารและแก้ไขเอกสารในส่วนนี้จะเกี่ยวข้องกับระบบในส่วนของการให้นักพัฒนาในทีมที่ทำการทดสอบซอฟต์แวร์ก่อนการปล่อยทำการเพิ่มผลการทดสอบในแต่ละส่วนลงไปเพื่อเก็บไว้สำหรับให้นักทดสอบระบบพิจารณาในการปล่อยซอฟต์แวร์ต่อไป

Eikon Toolbar version	Scenario	Eikon Package	Assignee	Test Result (PASS or FAIL)	Notes
4.56.0.14	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	10				
4.55.0.18	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	10				

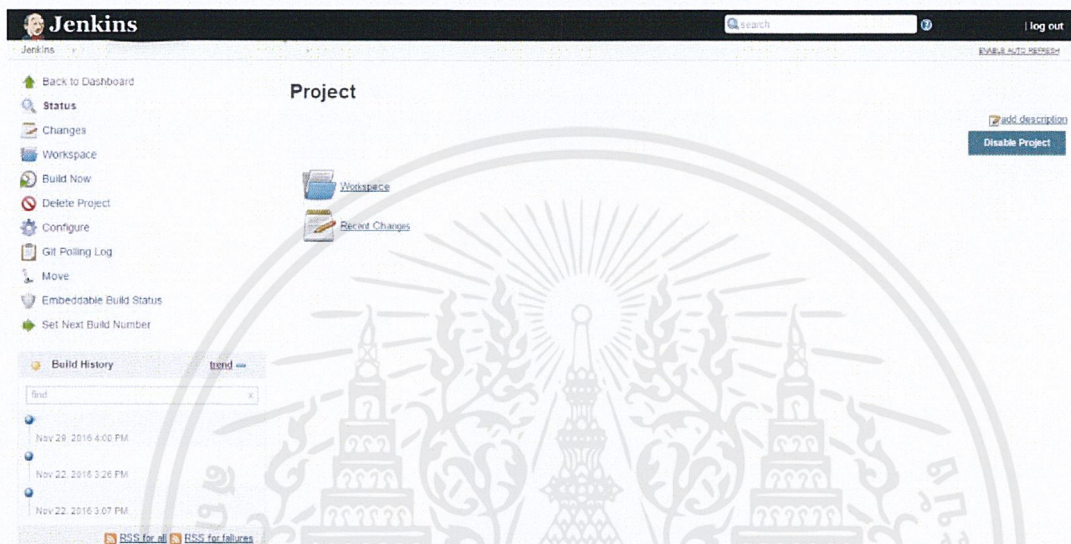
ภาพที่ 3.17 รูปภาพแสดงตัวอย่างของ Google Sheets

```
{  
  "range": "Sheet1!A1:D3",  
  "majorDimension": "COLUMNS",  
  "values": [  
    ["Item", "Wheel", "Door"],  
    ["Cost", "$20.50", "$15"],  
    ["Stocked", "4", "2"],  
    ["Ship Date", "3/1/2016", "3/15/2016"]  
  ],  
}
```

ภาพที่ 3.18 รูปภาพแสดงตัวอย่างของข้อมูลที่อ่านผ่าน Google Sheets API

### 3.4.2 Jenkins

Jenkins จะทำงานในส่วนของการดูแลการทำการรวมของซอร์สโค้ดอย่างต่อเนื่องซึ่ง Jenkins จะมีข้อมูลของแต่ละโครงการไม่ว่าจะเป็น ชื่อโครงการ คำสั่งสำหรับทำงานเมื่อมีการรวมของซอร์สโค้ด ประวัติการรวมของซอร์สโค้ด โดยที่ Jenkins จะเป็นให้ใช้งานผ่านหน้าเว็บไซต์ REST API และ Command Line Interface ในระบบนี้จะทำการติดต่อกับ Jenkins ผ่าน REST API



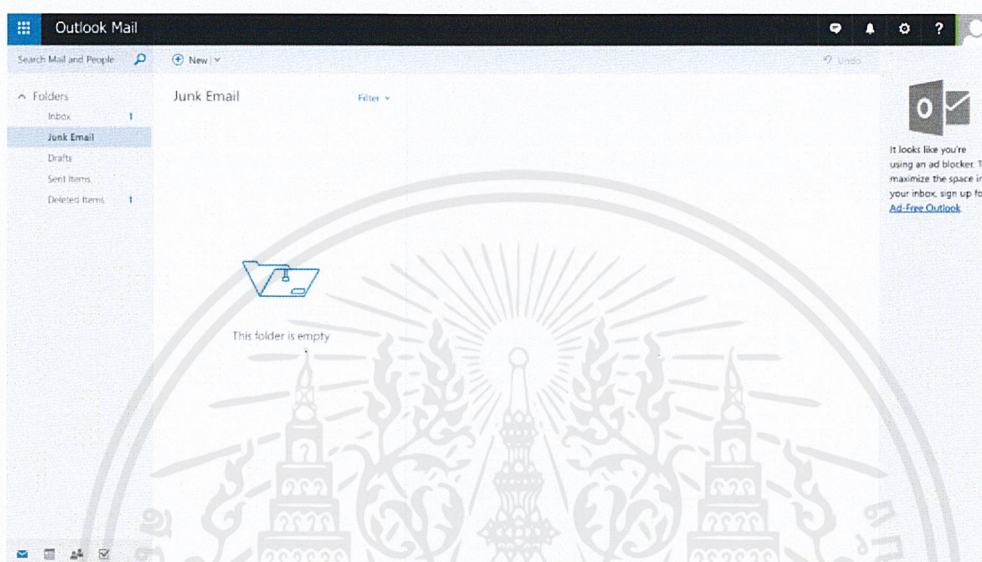
ภาพที่ 3.19 รูปภาพแสดงตัวอย่างหน้าแสดงผลของ Jenkins

```
<hudson webdriver="true">
  <assignedLabel></assignedLabel>
  <mode>NORMAL</mode>
  <nodeDescription>Jenkins Master-Knoten </nodeDescription>
  <nodeName></nodeName>
  <numExecutors>2 </numExecutors>
  <job>
  <job>
  <job>
  <job>
  <overallLoad></overallLoad>
  <primaryView>
  <quietingDown>false </quietingDown>
  <slaveAgentPort>0</slaveAgentPort>
  <unlabeledLoad></unlabeledLoad>
  <useCrumbs>false </useCrumbs>
  <useSecurity>false </useSecurity>
  <view>
</hudson>
```

ภาพที่ 3.20 รูปภาพแสดงตัวอย่างการแสดงผลการเรียกดูข้อมูล Jenkins API  
(ที่มา : <http://www.agile-coding.net/wp-content/uploads/2014/05/screenshot-firebug-jenkins-xml.png>)

### 3.4.3 Microsoft Outlook

Microsoft Outlook เป็นบริการของทาง Microsoft ที่ใช้ในการจัดการกับอีเมล ซึ่งจะถูกรู้จักให้ใช้งานเมื่อระบบต้องการจะทำการส่งอีเมลเพื่อแจ้งถึงการปล่อยซอฟต์แวร์ไปยังนักพัฒนาที่เกี่ยวข้อง โดย Microsoft Outlook เปิดให้สามารถติดต่อเพื่อจัดการอีเมลผ่านเว็บไซต์ หรือ REST API ก็ได้ ในระบบนี้จะติดต่อผ่านทาง REST API



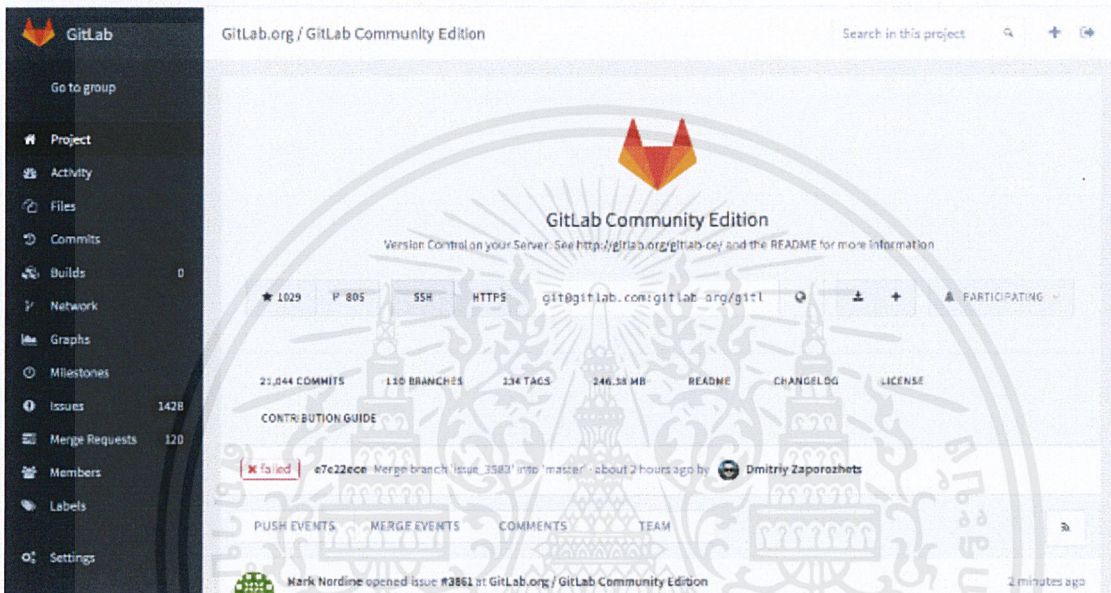
ภาพที่ 3.21 รูปภาพแสดงตัวอย่างหน้าแสดงผลของ Microsoft Outlook

```
[
  {
    "Sender": {
      "EmailAddress": {
        "Name": "Katie Jordan",
        "Address": "katiej@a830edad9050849NDA1.onmicrosoft.com"
      }
    },
    "From": {
      "EmailAddress": {
        "Name": "Katie Jordan",
        "Address": "katiej@a830edad9050849NDA1.onmicrosoft.com"
      }
    },
    "ToRecipients": [
      {
        "EmailAddress": {
          "Name": "Alex D",
          "Address": "alexjd@a830edad9050849NDA1.onmicrosoft.com"
        }
      }
    ]
  }
]
```

ภาพที่ 3.22 รูปภาพแสดงตัวอย่างแสดงของการติดต่อผ่าน Microsoft Outlook API

### 3.4.4 GitLab

GitLab เป็นซอฟต์แวร์สำหรับทำให้ผู้ใช้สามารถใช้งาน Git ได้สะดวกสบายขึ้น โดย GitLab ถูกนำมาใช้ในการเก็บประวัติการ commit โค้ดนักพัฒนา โดย GitLab ในระบบจะทำหน้าที่เก็บประวัติของซอฟต์แวร์ที่จะทำการปล่อยในแต่ละเวอร์ชันรวมถึงเก็บชื่อและอีเมลล์ของนักพัฒนาที่ commit ลงไปในซอฟต์แวร์ ระบบจะทำการติดต่อกับ GitLab ผ่านรูปแบบ REST API เพื่อทำการจัดการ Git ของซอฟต์แวร์



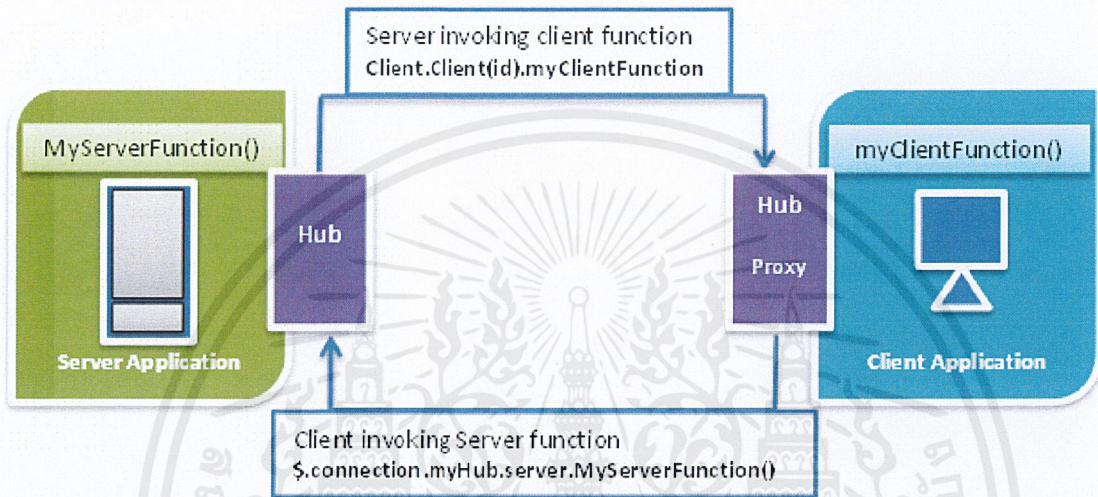
ภาพที่ 3.23 รูปภาพแสดงตัวอย่างหน้าแสดงผลของ GitLab

```
{
  "id": "ed899a2f4b50b4370feeea94676502b42383c746",
  "short_id": "ed899a2f4b5",
  "title": "Replace sanitize with escape once",
  "author_name": "Dmitriy Zaporozhets",
  "author_email": "dzaporozhets@sphereconsultinginc.com",
  "committer_name": "Administrator",
  "committer_email": "admin@example.com",
  "created_at": "2012-09-20T11:50:22+03:00",
  "message": "Replace sanitize with escape once",
  "allow_failure": false
},
```

ภาพที่ 3.24 รูปภาพแสดงตัวอย่างแสดงของการติดต่อผ่าน GitLab API

### 3.4.5 เซิร์ฟเวอร์

เซิร์ฟเวอร์เป็นส่วนสำคัญที่คอยรับคำสั่งจากส่วนไต่ตอบอัตโนมัติแล้วทำการส่งการไปที่ส่วนอื่นๆหากจำเป็น การพัฒนาในส่วนนี้ใช้ภาษา C# ในการพัฒนาโดยมีไลบรารีหลักๆในการพัฒนาคือไลบรารี SignalR ซึ่งเป็นไลบรารีที่ใช้ในการติดต่อสื่อสารกับส่วนของการโต้ตอบผู้ใช้ ส่วนของเซิร์ฟเวอร์จะถูกติดตั้งลงบนเครื่องเซิร์ฟเวอร์ของทางบริษัทเพื่อให้สามารถทำงานกับส่วนอื่นๆที่อยู่ภายในได้



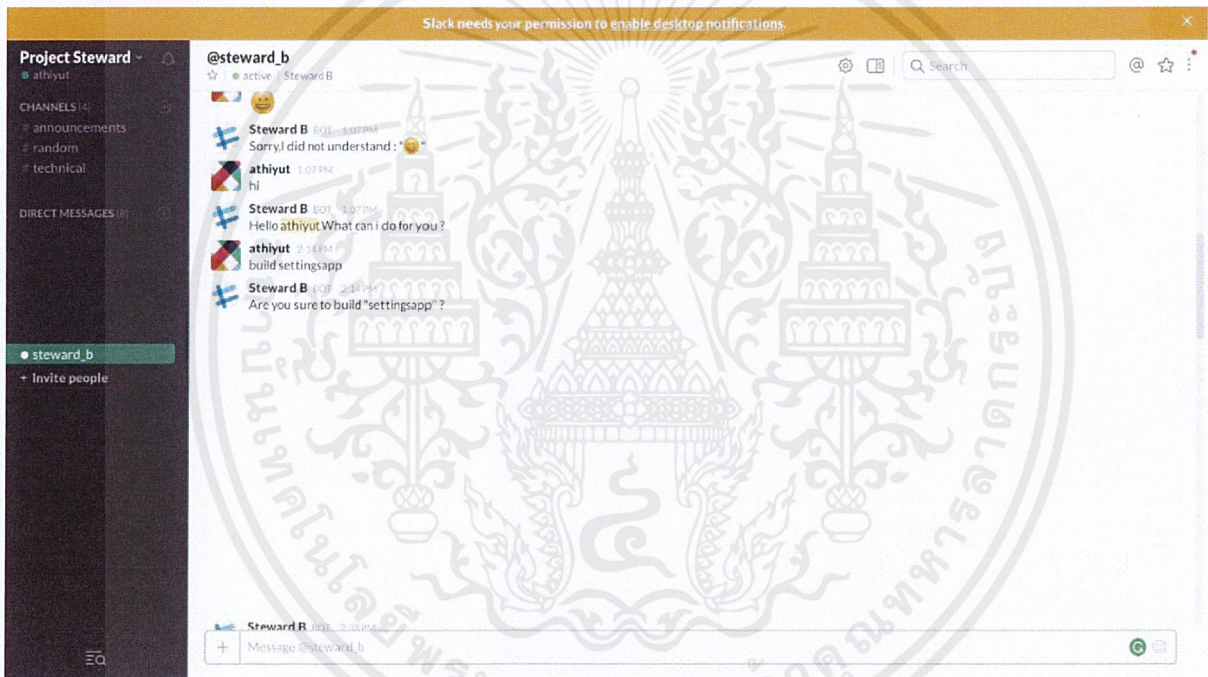
ภาพที่ 3.25 รูปภาพแสดงตัวอย่างการทำงานของ SignalR

(ที่มา : <http://csharpcorner.mindcrackerinc.netdna-cdn.com/UploadFile/abhijmk/what-why-and-how-about-signalr/Images/2.png>)

## บทที่ 4 ผลการดำเนินการ

### 4.1 ส่วนของการโต้ตอบผู้ใช้อัตโนมัติ

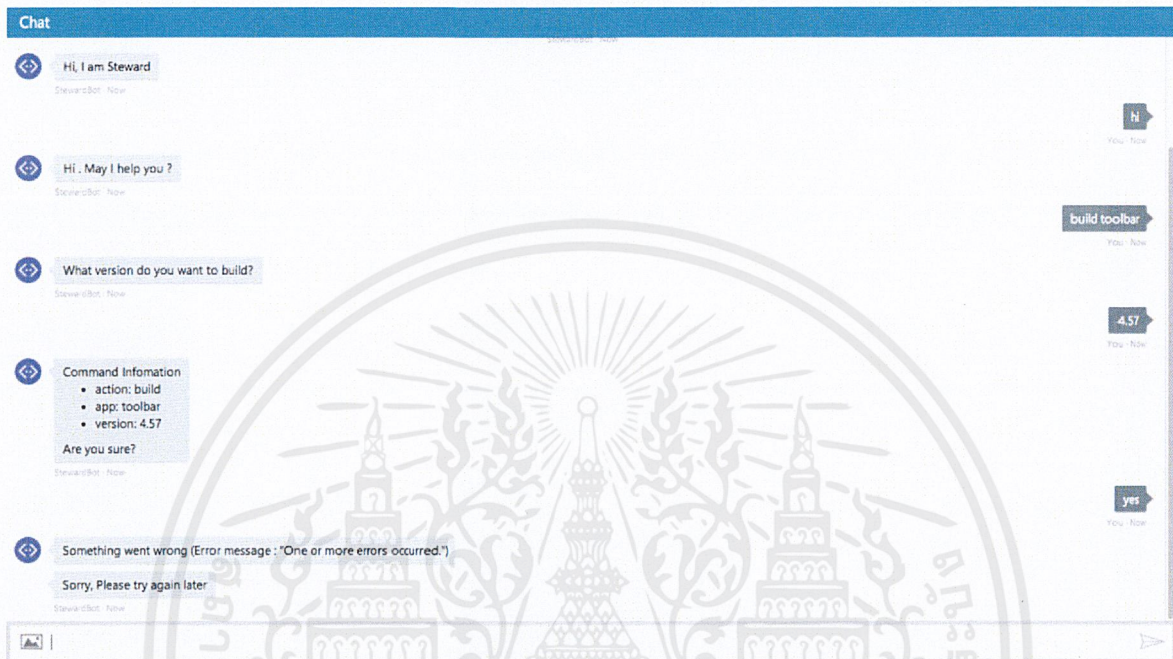
ในส่วนของการโต้ตอบผู้ใช้อัตโนมัติสามารถโต้ตอบผู้ใช้ได้ผ่านช่องทาง Slack messages และ Webchat ตามที่ได้ทำการตั้งค่าไว้ ส่วนของการโต้ตอบผู้ใช้อัตโนมัติสามารถรับส่งข้อความได้ตามที่เตรียมไว้ ความสามารถในการใช้งาน LUIS ในกรณีที่ใช้รับรู้ความตั้งใจอย่างชัดเจน LUIS จะสามารถทำนายได้ผลแม่นยำตามที่คาดหวังไว้ส่วนในกรณีที่ผู้ใช้ไม่ระบุความตั้งใจจะทำให้ได้ผลลัพธ์คือไม่สามารถเข้าใจข้อความในส่วนของการติดต่อกับส่วนของเซิร์ฟเวอร์ได้จะทำการติดต่อเป็นแบบ Web sockets



ภาพที่ 4.1 รูปภาพแสดงการโต้ตอบกับผู้ใช้ของระบบ

## 4.2 ส่วนของการช่วยเหลือการปล่อยซอฟต์แวร์

ในส่วนของเซิร์ฟเวอร์สามารถทำการติดต่อได้กับ bot ได้สามารถรับคำสั่งและตอบกับการทำงาน  
นำได้ถูกต้อง รวมถึงการติดต่อเพื่อสั่งงานหรือดึงข้อมูลจากส่วนของ Jenkins, GitLab, Google Sheets  
และ Microsoft Outlook และสามารถแจ้งไปยัง bot หากเกิดข้อความผิดพลาดในการทำงาน



ภาพที่ 4.2 รูปภาพแสดงการโต้ตอบกับผู้ใช้ของระบบเมื่อเกิดความผิดพลาด

## บทที่ 5

### สรุปผลโครงการและข้อเสนอแนะ

#### 5.1 สรุปผลโครงการ

ระบบโต้ตอบผู้ใช้อัตโนมัติสำหรับช่วยเหลือการปล่อยซอฟต์แวร์สามารถทำงานได้ตามที่คาดหวัง ทั้งในส่วนของการโต้ตอบอัตโนมัติและส่วนของการช่วยเหลือการปล่อยซอฟต์แวร์สามารถติดต่อและทำงานร่วมกับ Jenkins, GitLab, Google Sheets และ Microsoft Outlook เพื่อการทำงานตามที่ใช้ต้องการ สามารถรายงานเมื่อเกิดความผิดพลาดระหว่างการทำงาน ผู้ใช้สามารถสั่งการให้ bot ทำงานของ Jenkins พื้นฐาน ทำการเตรียมการปล่อยซอฟต์แวร์แบบที่ละขั้นตอน ทำการเตรียมการปล่อยซอฟต์แวร์แบบอัตโนมัติ ทำการถามสถานะของการเตรียมการปล่อยซอฟต์แวร์ และทำการยกเลิกการเตรียมการปล่อยซอฟต์แวร์ โดยผู้ใช้งานสามารถใช้ภาษาอังกฤษในการโต้ตอบกับระบบซึ่งไม่จำเป็นต้องเป็นประโยคที่เหมือนเดิม

#### 5.2 ข้อเสนอแนะ

- 1) หากต้องการสร้างซอฟต์แวร์สำหรับโต้ตอบอัตโนมัติอาจจะเลือกเครื่องมืออื่นๆนอกเหนือจาก Bot framework เช่น Bot for messenger, Telegram Bots
- 2) ส่วนของการทำให้ซอฟต์แวร์สามารถเข้าใจภาษานอกเหนือจาก LUIS อาจจะเลือกใช้ NLTK ซึ่งเป็นไลบรารีของภาษาไพธอน

## เอกสารอ้างอิง

- Bot Framework Overview | Documentation | Bot Framework. [ออนไลน์] แหล่งที่มา: <https://docs.botframework.com/en-us/>
- Code, test, and deploy together with GitLab open source git repo management software. [ออนไลน์] แหล่งที่มา: <https://about.gitlab.com/>
- Jenkins. [ออนไลน์] แหล่งที่มา: <https://jenkins.io/>
- Learn about Outlook APIs, Outlook connectors, and Outlook Add-ins - Outlook Dev Center. [ออนไลน์] แหล่งที่มา: <https://dev.outlook.com/>
- LUIS: Language Understanding Intelligent Service (beta). [ออนไลน์] แหล่งที่มา: <https://www.luis.ai/>
- Sheets API | Google Developers. [ออนไลน์] แหล่งที่มา: <https://developers.google.com/sheets/>
- Brownlee, J. (2016). A Tour of Machine Learning Algorithms. [ออนไลน์] Machine Learning Mastery. แหล่งที่มา: <http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>
- Introduction to Natural Language Processing - The Mind Project. [ออนไลน์] แหล่งที่มา: [http://www.mind.ilstu.edu/curriculum/protothinker/natural\\_language\\_processing.php](http://www.mind.ilstu.edu/curriculum/protothinker/natural_language_processing.php)