

โครงการการศึกษาความรอบรู้เฉพาะเรื่อง (Senior Project)

เรื่อง

เซทบอทอัจฉริยะ

นายน์ทที เอื้อไพระนันท์
ชั้นมัธยมศึกษาปีที่ 6/6

รายงานนี้เป็นส่วนหนึ่งของกิจกรรมพัฒนาผู้เรียน
โรงเรียนสาธิตแห่งมหาวิทยาลัยเกษตรศาสตร์ ศูนย์วิจัยและพัฒนาการศึกษา
ปีการศึกษา 2565

นัทที เอื้อไพระนันท์ 2565 : โครงการการศึกษาความรอบรู้เฉพาะเรื่อง
เรื่อง แชนบอทอัจฉริยะ

โรงเรียนสาธิตแห่งมหาวิทยาลัยเกษตรศาสตร์ ศูนย์วิจัยและพัฒนาการศึกษา

อาจารย์ที่ปรึกษา : อาจารย์ลัดดาวัลย์ สุวรรณโชติ

ผู้เชี่ยวชาญ : นายโกเศศ เอื้อไพระนันท์

ผู้ปกครอง : นางมนทรัตม์ เอื้อไพระนันท์

บทคัดย่อ

โครงการการศึกษาความรอบรู้เฉพาะเรื่อง เรื่องแชนบอทอัจฉริยะ มีวัตถุประสงค์เพื่อพัฒนาแชนบอทอัจฉริยะและเพื่อศึกษาความคิดเห็นต่อการใช้งานแชนบอทอัจฉริยะ กลุ่มที่ใช้ในการศึกษาคือ นักเรียนชั้นมัธยมศึกษาปีที่ 6 จำนวน 10 คนโดยการเลือกแบบเจาะจง

ผลการศึกษาพบว่า การพัฒนาแชนบอทอัจฉริยะ หลักการเขียนโปรแกรมขั้นสูงแบ่งเป็น 5 อย่างประกอบด้วย 1) ภาษา Python 2) การใช้งาน Replit 3) การใช้งาน Github 4) การใช้งาน Discord.py Library และ 5) การพัฒนาแชนบอท พบว่าผู้เชี่ยวชาญประเมินความคิดเห็นต่อแชนบอทอัจฉริยะอยู่ในระดับดีมาก ผลการประเมินความคิดเห็นของกลุ่มศึกษาต่อการใช้งานแชนบอทอัจฉริยะพบว่าด้านระบบติดตามข่าวประชาสัมพันธ์ ด้านระบบเปิดเพลงเพื่อจรรโลงใจ และด้านระบบติดตามงานและกิจกรรมอยู่ในระดับดีมาก ตามลำดับ

สารบัญ

	หน้า
หลักการและเหตุผล	1
วัตถุประสงค์	2
ประโยชน์ที่คาดว่าจะได้รับ	2
ขอบเขตการศึกษา	2
แนวคิดในการศึกษา	3
ภาษา Python	3
การใช้งาน Replit	4
การใช้งาน Github	4
การใช้งาน Discord.py library	5
การพัฒนาแชทบอท	5
วิธีการศึกษา	6
เครื่องมือที่ใช้ในการศึกษา	7
การเก็บรวบรวมข้อมูล	7
การวิเคราะห์ข้อมูล	8
ผลการศึกษา	9
สะท้อนความคิด	19
บรรณานุกรม	20
ภาคผนวก	21

หลักการและเหตุผล

การใช้เทคโนโลยีการสื่อสารมีความสำคัญและมีบทบาทต่อชีวิตประจำวันของเราเป็นอย่างมาก ไม่ว่าจะเป็นการใช้โทรศัพท์มือถือในการติดต่อ หรือการเข้าสื่อโซเชียลต่างๆ ไม่ว่าจะเป็นเพื่ออ่านบทความ หรือแชร์เรื่องราวต่างๆ ทำให้เทคโนโลยีสื่อสารนั้น มีความสำคัญมากในปัจจุบัน และเป็นตัวเชื่อมโยงที่สำคัญสำหรับนักเรียนหลายๆกลุ่ม (ศาสตราจารย์ ชาติรี, 2566: ออนไลน์)

แพลตฟอร์มซึ่งเป็นหนึ่งในเทคโนโลยีการสื่อสาร และเป็นแอปพลิเคชันซอฟต์แวร์ที่สามารถสื่อสารกับมนุษย์ได้ โดยใช้ปัญญาประดิษฐ์และคำสั่งที่ถูกตั้งค่ามา และสามารถปล่อยให้ใช้อย่างแพร่หลายบนแพลตฟอร์ม Discord ที่เป็นที่ยอมรับของวัยรุ่นในปัจจุบัน (Peter Susic, 2566: ออนไลน์)

ด้วยเหตุนี้ ผู้จัดทำจึงเล็งเห็นว่าสามารถนำเทคโนโลยีดังกล่าวมาใช้ เพื่อประชาสัมพันธ์ข่าวสารต่างๆ หรือเพื่อคอยช่วยติดตามงานให้กับนักเรียน โดยผู้เขียนได้พัฒนาแพลตฟอร์มขึ้นมาเพื่อคอยติดตามข่าวสารใหม่ๆของโรงเรียน และส่งให้ผู้ติดตามเมื่อตรวจพบ ทำให้นักเรียนไม่พลาดข่าวสารนั้นๆอีกต่อไป อีกทั้งยังสามารถช่วยติดตามงานได้อีกด้วย

วัตถุประสงค์

1. เพื่อพัฒนาแชทบอทอัจฉริยะ
2. เพื่อศึกษาความคิดเห็นต่อการใช้งานแชทบอทอัจฉริยะ

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้แชทบอทในการช่วยเหลือด้านการติดตามงานรายวิชาต่างๆ
2. ได้แชทบอทในการช่วยเหลือการติดตามข่าวประชาสัมพันธ์โรงเรียน
3. ได้แชทบอทในการเปิดเพลงจรรโลงใจในกลุ่มของนักเรียน
4. ได้ประสบการณ์ในการเขียนและออกแบบแชทบอทในการนำมาใช้เพื่อให้เกิดประโยชน์ต่อผู้จัดทำต่อไป

ขอบเขตการศึกษา

การศึกษาคั้งนี้ ผู้ศึกษาได้ศึกษาและพัฒนาแชทบอทอัจฉริยะขึ้นมา โดยมีความสามารถดังนี้ คือ

1. ติดตามงานและกิจกรรมต่างๆทั้งภายในโรงเรียน และนอกโรงเรียน
2. ติดตามข่าวประชาสัมพันธ์ของโรงเรียน
3. เปิดเพลงจรรโลงใจกับนักเรียน

โดยใช้แอปพลิเคชัน Discord โดยเขียนด้วยภาษา Python บน VS Code และ host ด้วยเว็บไซต์ Replit อีกทั้งสามารถตรวจสอบความโปร่งใสของบอทได้บน Github ของผู้เขียน โดยมีระยะเวลาการศึกษาตั้งแต่ มกราคม 2563 ถึง มกราคม 2566

กลุ่มตัวอย่างที่ใช้ในการประเมินความคิดเห็นที่ได้จากการใช้งานแชทบอทเป็นนักเรียนระดับชั้นมัธยมศึกษาปีที่ 6 จำนวน 10 คน โดยการสุ่มตัวอย่างแบบเจาะจง

แนวคิดในการศึกษา

เนื่องจากแชทบอทถูกใช้อย่างแพร่หลาย และเป็นที่ยอมรับอย่างมากใน Discord และการที่ Discord ได้สร้างระบบ API ไว้ให้ผู้ใช้สามารถสร้างแชทบอทอย่างง่ายได้ และยังได้สร้างระบบแนะนำแชทบอท อีกทั้งการที่วัยรุ่นส่วนใหญ่ มักสื่อสารผ่าน Discord และใช้เวลาส่วนมากอยู่ในนั้น ผู้ศึกษาจึงได้จัดทำ คณิตคว่ำ และพัฒนาแชทบอทอัจฉริยะขึ้นมาโดยได้แนวคิดในการศึกษาจากเอกสารที่เกี่ยวข้องดังนี้

1. ภาษา Python
2. การใช้งาน Replit
3. การใช้งาน Github
4. การใช้งาน Discord.py library
5. ขั้นตอนการพัฒนาแชทบอท

1. ภาษา Python Coursera (2565: ออนไลน์) กล่าวว่า

ภาษา Python หรือ ไพทอนเป็นภาษาระดับสูง หมายความว่ามนุษย์สามารถทำความเข้าใจได้ดี โดยออกแบบเพื่อมุ่งเน้นให้ผู้เขียนโปรแกรมสามารถอ่านชุดคำสั่งได้โดยง่าย โดยส่วนมากมักจะใช้ในการสร้างเว็บไซต์ และโปรแกรมต่างๆ, การทำงานโดยให้โปรแกรมทำโดยอัตโนมัติ, การวิเคราะห์ข้อมูล และการแสดงแผนภาพข้อมูล

โดยผู้เขียนเลือกใช้เนื่องจากเป็นภาษาที่สามารถเข้าใจได้ง่าย อีกทั้งยังเป็นภาษาระดับสูง หมายความว่าผู้เขียนไม่ต้องมากังวลกับ Memory Leak ต่างๆ โดยหากเป็นภาษา Rust หรือ C จะทำให้ผู้อ่านเข้าใจยาก และต้องจัดการกับการทำงานระดับล่างอีกด้วย รวมถึงภาษาไพทอน มี Library ให้เลือกใช้มากมาย ทำให้สะดวกสบายต่อการเขียนแชทบอทอัจฉริยะเป็นอย่างมาก

2. การใช้งาน Replit

Replit (2565: ออนไลน์) อธิบายไว้ดังนี้

Replit คือ IDE, Editor, Compiler, Interpreter และ REPL ออนไลน์ที่เรียนง่ายแต่มีประสิทธิภาพ สามารถโค้ด คอมไพล์ รัน และโฮสต์ โปรแกรมต่างๆในภาษาการเขียนโปรแกรมมากกว่า 50 ภาษา ออนไลน์ผ่านเว็บไซต์ <https://www.replit.com>

โดยเหตุผลที่ผู้เขียนเลือกใช้เนื่องจาก สามารถโฮสต์โปรแกรมต่างๆได้ฟรี (ต้องใช้ควบคู่กับ <https://uptimerobot.com> เพื่อให้บอทสามารถทำงานต่อเนื่องได้ตลอดวัน) และสามารถนำโครงการไปส่งต่อให้กับผู้สนใจนำไปแก้ไขและใช้ต่อไปได้อีกด้วย

3. การใช้งาน Github

Github คือ เว็บไซต์ที่เป็นศูนย์รวมโค้ดของโครงการและโปรแกรมต่างๆทั่วโลก โดยคำว่า Git หมายถึงการติดตามการเปลี่ยนแปลงในไฟล์ต่างๆ มักใช้งานมากในโปรแกรมที่มีนักพัฒนาเป็นกลุ่ม หรือการควบคุมเวอร์ชันของโปรแกรมนั้นๆ โดยในที่นี้ ผู้เขียนใช้งานเพื่อติดตามปัญหาของโปรแกรม ข้อผิดพลาด การอัปเดตเวอร์ชันใหม่ การเผยแพร่โค้ดของโปรแกรมในสาธารณะ และการควบคุมคุณภาพของโค้ด โดยโค้ดของเซทบอทอัจฉริยะสามารถดูได้ผ่านลิงค์ดังนี้

<https://github.com/Tpmonkey-Nuttee/TpLearn>

The screenshot shows a GitHub repository interface. At the top, there are navigation buttons: 'master', 'Go to file', 'Add file', and 'Code'. Below this is a list of files and folders:

File/Folder	Description	Time
src	Fix embed edit test not p...	3 days ago
.gitignore	Add .gitignore file	23 days ago
README...	Add readme file	3 months ago
db_migr...	Add db migrating tool	23 days ago
install.ba...	Add note	2 months ago
pyprojec...	Add Pillow for Image pro...	last month
requirem...	Added requirements.txt	19 days ago

On the right side, there is an 'About' section with the following information:

- Discord bot for Senior project.
- Readme
- 0 stars
- 1 watching
- 0 forks

Below the 'About' section is a 'Languages' section with a bar chart showing the distribution of code languages:

- Python 99.8%
- Shell 0.2%

4. การใช้งาน Discord.py library

Danny (2565: ออนไลน์) ได้อธิบาย discord.py ไว้ว่า

Discord.py คือ API wrapper สำหรับ Discord ที่ถูกเขียนด้วยภาษาไพทอน โดยมีความทันสมัย, ใช้งานง่าย, คุณสมบัติหลากหลาย พร้อมระบบ Asynchronous

โดยเนื่องจากการทำแชทบอทผ่านแอปพลิเคชัน Discord จำเป็นต้องใช้ Discord API ในการสื่อสารกับเซิร์ฟเวอร์ โดยได้มีโปรเจกต์ Open-Source ในการทำ API wrapper ของ Discord ขึ้นในภาษาไพทอน โดยเป็น API wrapper library ที่มีคนใช้งานมากเป็นอันดับ 2 รองจาก Discord.js โดยสามารถอ่านคู่มือการใช้งานได้ที่ <https://discordpy.readthedocs.io/en/stable/index.html> และโค้ดของ library ได้ที่ <https://github.com/Rapptz/discord.py>

5. การพัฒนาแชทบอท

ผู้เขียนได้ศึกษาตัวอย่างในการเขียนแชทบอทผ่าน Github และตัวอย่างต่างๆ

โดยผู้เขียนได้นำตัวอย่างมาจาก 3 แหล่งคือ

1. <https://github.com/Rapptz/discord.py/tree/master/examples>
2. <https://github.com/Gelbpunkt/IdleRPG>
3. <https://github.com/python-discord/bot>

โดยลิ้งค์ลำดับที่หนึ่งมาจากผู้เขียน Library discord.py ซึ่งเป็นตัวอย่างในการเขียนบอทลิ้งค์ลำดับที่ 2 และ 3 มาจากแชทบอทที่มีชื่อเสียงอยู่แล้ว และสามารถนำมาเป็นตัวอย่างที่ดีในการเขียนโดยเมื่อผู้เขียน

ได้ศึกษาแล้ว และรวมกับประสบการณ์จากการลองผิดลองถูก ผู้เขียนจึงได้พัฒนาแชทบอท

อัจฉริยะที่มีความง่ายต่อการใช้งาน และง่ายต่อการแก้ไขระบบขึ้นมา

วิธีการศึกษา

1. ชั้นศึกษาข้อมูล

1.1 ศึกษาการเขียนภาษา Python จากวิชาเลือกเสรี เขียนโปรแกรม, หนังสือ และออนไลน์

1.2 ศึกษาการใช้งาน Discord.py ที่เป็น API wrapper ของ Discord API

1.3 ศึกษาการใช้ Replit, Github และ Version Control

2. ขั้นตอนการเขียนโปรแกรมและทดลองใช้

2.1 ออกแบบระบบโดยแบ่งออกเป็น 5 ระบบคือ ระบบการติดตั้งบอท ระบบส่วนกลางของบอท ระบบแยกย่อย ระบบสารบัญปก และระบบอัปเดตเวอร์ชันกึ่งอัตโนมัติ

2.2 ติดตั้งระบบทั้งหมดบน Replit เพื่อทำการเปิดใช้งานบอท

2.3 ติดตั้งระบบ Version Control โดยใช้ Github เป็นตัวกลาง

2.4 ทดสอบโปรแกรม ปรับปรุงแก้ไข และทดลองใช้

3. นำแบบประเมินความคิดเห็นให้ผู้เชี่ยวชาญประเมินเพื่อหาประสิทธิภาพของแชทบอทอัจฉริยะ

เครื่องมือที่ใช้ในการศึกษา

เครื่องมือที่ใช้ในการศึกษาประกอบด้วย

1. เครื่องมือที่ใช้ในการสร้างแชทบอทประกอบด้วย VSCode, Replit, Github และ Discord
2. เครื่องมือที่ใช้ในการเก็บข้อมูล คือ แบบสอบถามความคิดเห็นต่อการใช้งานแชทบอทอัจฉริยะ

โดยประเมิน 3 ด้านประกอบด้วย 1) ระบบติดตามงานและกิจกรรม 2) ระบบติดตามข่าวประชาสัมพันธ์ 3) ระบบเปิดเพลงเพื่อจรรโลงใจ

การเก็บรวบรวมข้อมูล

ใช้แบบสอบถามความคิดเห็นต่อการใช้งานแชทบอทอัจฉริยะ โดยสอบถามความคิดเห็นของกลุ่มที่ศึกษาจำนวน 10 คน ใน 3 ด้านคือ 1) ระบบติดตามงานและกิจกรรม 2) ระบบติดตามข่าวประชาสัมพันธ์ 3) ระบบเปิดเพลงเพื่อจรรโลงใจ โดยเก็บข้อมูลมาวิเคราะห์ข้อมูล

การวิเคราะห์ข้อมูล

นำข้อมูลที่ได้จากการสอบถามความเห็นจากกลุ่มที่ศึกษาจำนวน 10 คน มาดำเนินการวิเคราะห์ข้อมูลเพื่อหาค่าร้อยละ ค่าเฉลี่ยและแปรผลค่าเฉลี่ย

การแปลผลค่าเฉลี่ย ตามแนวคิดของเบสท์และคาห์น Best & Kahn (1993)

1. กำหนดน้ำหนักคะแนนดังนี้

ระดับความคิดเห็น ในระดับ “ดีมาก” “ดี” “พอใช้” “น้อย” “น้อยที่สุด”

ให้คะแนน 5 4 3 2 1 คะแนนตามลำดับ

2. การแปลค่าคะแนนเฉลี่ย

4.51 – 5.00 หมายถึง ความคิดเห็นอยู่ในระดับดีมาก

3.51 – 4.50 หมายถึง ความคิดเห็นอยู่ในระดับดี

2.51 – 3.50 หมายถึง ความคิดเห็นอยู่ในระดับพอใช้

1.51 – 2.50 หมายถึง ความคิดเห็นอยู่ในระดับน้อย

1.00 – 1.50 หมายถึง ความคิดเห็นอยู่ในระดับน้อยที่สุด

ผลการศึกษา

โครงการศึกษาความรู้เฉพาะเรื่อง (Senior Project) เรื่อง แชนบอทอัจฉริยะ มีวัตถุประสงค์เพื่อพัฒนาโปรแกรมแชทบอทอัจฉริยะ และเพื่อศึกษาความคิดเห็นต่อการใช้งานแชทบอทอัจฉริยะ ผู้ศึกษาจึงได้ดำเนินการพัฒนาและการทำโปรแกรมแชทบอทอัจฉริยะ และ สํารวจระดับความคิดเห็นต่อการใช้งานแชทบอทอัจฉริยะ โดยมีผลการศึกษาดังนี้

1. ผลจากการพัฒนาแชทบอทอัจฉริยะ

ตามหลักการเขียนโปรแกรมระดับง่าย จะแบ่งโปรแกรมออกเป็น 3 ส่วนได้แก่ ส่วนรับข้อมูล ส่วนประมวลผล และส่วนแสดงผล แต่เนื่องจากโปรแกรมมีความซับซ้อนสูง จึงได้แบ่งออกเป็น 5 ระบบหลักๆ ดังนี้

- 1) การติดตั้งและ Package Management
- 2) ระบบส่วนกลางของบอท
- 3) ระบบแยกย่อย
- 4) ระบบสาธารณูปโภค
- 5) ระบบอัปเดตเวอร์ชันกึ่งอัตโนมัติ

1) การติดตั้งและ Package Management

ผู้เขียนได้พัฒนาไฟล์ 4 ไฟล์ไว้ใช้ในการติดตั้งบอท และใช้งานบอท

โดยแต่ละไฟล์มีหน้าที่ดังนี้

1. `install.bash` ทำหน้าที่ในการติดตั้ง Database ที่บอทใช้ และติดตั้งระบบ poetry เพื่อทำหน้าที่ติดตั้ง Library ที่ใช้และรันบอท
2. `pyproject.toml` ทำหน้าที่บอกระบบ poetry ให้ติดตั้ง Library และรันบอท
3. `requirement.txt` ทำหน้าที่แทนที่หากต้องการรันบอทผ่าน pip แทน poetry
4. `db_migrate.py` ทำหน้าที่ย้ายข้อมูลจาก Database ของ Replit มาที่ Redis Database ในกรณีที่ต้องการรันบนเซิร์ฟเวอร์ปกติ

```

$ install.bash
1  #!/bin/bash
2
3  # Not finished, For moving on to VPS in the future.
4  # Only works on Ubuntu.
5
6  sudo apt update
7  sudo apt upgrade
8
9  # Install redis
10 curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg
11
12 sudo apt-get update
13 sudo apt-get install redis
14
15 # For pyproject.toml file
16 sudo apt install python3-poetry
17 sudo apt install python3-cachecontrol
18
19 # Install poetry
20 poetry install

```

ภาพที่ 2 ตัวอย่างระบบติดตั้ง และ Package Management

2) ระบบส่วนกลางของบอท

ระบบนี้เปรียบเสมือนอวัยวะสำคัญของบอท หากต้องการเปลี่ยนแปลง ต้องทำการเริ่มระบบใหม่ทั้งหมดโดยจะมีทั้งหมด 6 ไฟล์ดังนี้

1. `main.py` ทำหน้าที่เป็นไฟล์เริ่มต้น หากต้องการรันบอทต้องรันไฟล์นี้
2. `config.py` ทำหน้าที่เก็บข้อมูลการตั้งค่าของบอทไว้
3. `db.py` ทำหน้าที่สื่อสารกับ Database
4. `bot.py` ทำหน้าที่เป็นแกนกลางการประมวลผลของบอท
5. `pagination.py` ทำหน้าที่เป็นโครงสร้างระบบเปลี่ยนหน้าข้อความ
6. `webserver.py` ทำหน้าที่สร้าง web server ประสานกับ UptimeRobot เพื่อให้ Replit สามารถทำงานได้ตลอดเวลา

```

306     async def wait_for_message(self, ctx: commands.Context, timeout: int = None) -> discord.Message:
307         """
308         Wait for User message, Time Limit is optional.
309
310         Checks:
311         - Message is the same channel as ran command.
312         - Author needs to be the person who ran command.
313         """
314         def check(m) -> bool:
315             return m.channel == ctx.channel and m.author == ctx.author
316
317         log.debug("waiting for message...")
318         return await self.wait_for("message", check=check, timeout = timeout)
319
320     async def add_reactions(self, message: discord.Message, reactions: list) -> None:
321         """ Add Set/List of Reactions to targeted message. """
322         for reaction in reactions:
323             try:
324                 await message.add_reaction(reaction)
325             except (discord.HTTPException, discord.Forbidden) as e:
326                 log.debug(f"could not add {reaction} reaction with exception: {e}")

```

ภาพที่ 3 ตัวอย่างระบบส่วนกลางของบอท จากไฟล์ `bot.py`

3) ระบบแยกย่อย

จุดประสงค์ของระบบแยกย่อยคือการทำงานที่แบ่งเป็นส่วนๆ โดยหากส่วนไหนมีปัญหา หรือต้องการพัฒนา จะสามารถสั่งให้ระบบส่วนกลางปิดระบบแยกย่อยนั้นๆ ไปชั่วคราว ซึ่งทำให้ไม่ต้องปิดระบบทั้งหมด โดยระบบนี้แบ่งเป็น 17 ไฟล์ดังนี้

1. **admin.py** ทำหน้าที่เป็นระบบแอดมินให้ผู้ดูแลสามารถใช้งานได้
2. **assignment.py** ทำหน้าที่จัดการระบบงานและกิจกรรม โดยเป็นส่วนคำสั่งให้ผู้ใช้สามารถ เพิ่ม ลบ แก้ไข แสดง ข้อมูลของงานต่างๆ ได้
3. **dayloop.py** ทำหน้าที่ติดตามวันเวลาของโลก เพื่อให้สามารถอัปเดตงานต่างๆ ได้ตามเวลา
4. **debug.py** ทำหน้าที่แสดงข้อมูลต่างๆ ของบอท เพื่อนำไปแก้ไขปัญหา หรือ debugging ได้
5. **dm.py** ทำหน้าที่จัดการข้อความส่วนตัวของบอท และส่งให้แอดมิน
6. **dupe_check.py** ทำหน้าที่รับรองว่าบอทจะไม่ทำงานซ้อนกันในกรณีรันบนหลายเครื่อง
7. **errorhand.py** ทำหน้าที่จัดการ ควบคุม และแสดงข้อความหากเกิดข้อผิดพลาดในระบบ
8. **help.py** ทำหน้าที่แสดงข้อมูลของคำสั่งต่างๆ และสอนการใช้งานบอท
9. **kus.py** ทำหน้าที่ติดตามการอัปเดตของเว็บโรงเรียน การตั้งค่าของแต่ละเซิร์ฟเวอร์ และการแจ้งเตือนเมื่อเว็บโรงเรียนมีข่าวสารใหม่ๆ
10. **muleval.py** ทำหน้าที่รันโค้ด Python ให้แอดมิน

11. `music.py` ทำหน้าที่จัดการระบบเพลงส่วนหน้าคือ คำสั่งผู้ใช้งานและตรรกะในการเล่นเพลง
12. `new_patch.py` ทำหน้าที่ตรวจสอบหากมีการอัปเดตของบอท และแสดงการแจ้งเตือน
13. `ping.py` ทำหน้าที่แสดงค่า ping หรือ latency ต่างๆของการเชื่อมต่อของบอทกับ Discord Server และ latency ของระบบ
14. `setup.py` ทำหน้าที่ติดตั้งระบบบอทในการใช้งานครั้งแรก
15. `terminate.py` ทำหน้าที่ลบข้อมูลผู้ใช้ตามที่ผู้ใช้ขอและออกจากเซิร์ฟเวอร์นั้นๆ
16. `utils.py` ทำหน้าที่เบ็ดเตล็ด

```

401 async def handle_request(self, ctx: commands.Context, message: discord.Message) -> None:
402     content = message.content
403     state = self.tasks[ctx.author.id]["details"]["state"]
404
405     # If the content is None, It's likely to be a picture file.
406     if not content.strip():
407         log.debug("empty message, likely to be picture")
408         state = 4
409
410     if state == 1:
411         self.tasks[ctx.author.id]["details"]["headers"]["title"] = content
412     elif state == 2:
413         self.tasks[ctx.author.id]["details"]["headers"]["description"] = content
414     elif state == 3:
415         self.tasks[ctx.author.id]["details"]["date"] = self.format(content)
416     else:
417         image_url = self.tasks[ctx.author.id]["details"]["image"]
418         if len(message.attachments) < 1:
419             if ("http://" in content or "https://" in content) and "." in content:
420                 image_url = content
421                 log.debug(f'{ctx.author.id} added image using url: {image_url}')
422             else:
423                 log.debug(f'{ctx.author.id} added image using file')
424                 image = message.attachments[0]
425                 image_url = await self.bot.get_image_url(image)
426
427         self.tasks[ctx.author.id]["details"]["image"] = image_url

```

ภาพที่ 4 ตัวอย่างระบบแยกย่อย `assignment.py`

4) ระบบสาธารณูปโภค

ระบบสาธารณูปโภค หรือ ระบบหลังบ้าน คอยทำหน้าที่จัดการข้อมูลทั้งหมดที่ระบบแยกย่อยส่งมาให้โดยแบ่งออกเป็น 9 ไฟล์ดังนี้

1. **audio.py** ทำหน้าที่จัดการเรื่องเสียงที่ใช้ในระบบเพลงทั้งหมด หน้าที่ดึงข้อมูลจาก YouTube API ที่จำเป็น และหน้าที่ใส่เอฟเฟกซ์ให้เสียง
2. **checks.py** ทำหน้าที่ตรวจสอบสิทธิ์ของผู้ใช้ก่อนใช้คำสั่งใดๆ
3. **extensions.py** ทำหน้าที่สื่อสารให้กับระบบหลักให้โหนดระบบย่อย
4. **log.py** ทำหน้าที่สร้างและจัดการ log ไฟล์ เพื่อเก็บไว้ในการบันทึกข้อมูลการใช้งาน
5. **manager.py** ทำหน้าที่จัดการช่องสนทนาต่างๆสำหรับระบบงานและกิจกรรม
6. **planner.py** ทำหน้าที่เก็บ จัดการ และควบคุมดูแลข้อมูลทั้งหมดเกี่ยวกับงานต่างๆในระบบงานและกิจกรรม
7. **post_eval.py** ทำหน้าที่รันโค้ด Python ที่ได้จากระบบย่อยในพื้นที่หลัง
8. **spotify.py** ทำหน้าที่สื่อสารกับ Spotify ในการใช้เพลย์ลิสต์หรืออัลบั้ม
9. **time.py** ทำหน้าที่คำนวณเวลา และแสดงเวลาในรูปแบบที่มนุษย์เข้าใจได้

```

23 # Creating and authenticating our Spotify app.
24 client_credentials_manager = SpotifyClientCredentials(cid, secret)
25 spotify = spotipy.Spotify(client_credentials_manager=client_credentials_manager)
26
27 def getTracks(playlistURL: str) -> List[str]:
28     # Getting a playlist.
29     results = spotify.user_playlist_tracks(user="", playlist_id=playlistURL, offset=0)
30     tracklist = []
31     offset = 0
32
33     # Loop until all the tracks are extracted from playlist.
34     # The reason behind is, Spotify API has limit at 100 tracks at a time,
35     # So to work around, we use offset instead
36     while len(results["items"]) != 0:
37         for i in results["items"]:
38             tracklist.append(f'{i["track"]["name"]} {i["track"]["artists"][0]["name"]} ')
39             offset += 100
40         # Get it again, with an offset
41         results = spotify.user_playlist_tracks(user="", playlist_id=playlistURL, offset=offset)
42     return tracklist

```

ภาพที่ 5 ตัวอย่างระบบสาธารณูปโภค spotify.py

5) ระบบอัปเดตเวอร์ชันกึ่งอัตโนมัติ

ระบบทำหน้าที่รับสัญญาณจากระบบหลัก และทำการติดตั้งไฟล์เวอร์ชันใหม่ของบอทที่ถูกอัปเดตบน Github

```

50 async def move_files(path: str) -> None:
51     source_dir = "./updater/TpLearn-master/src"
52     target_dir = path # "./test"
53
54     file_names = os.listdir(source_dir)
55
56     for file_name in file_names:
57         source = os.path.join(source_dir, file_name)
58         dst = os.path.join(target_dir, file_name)
59
60         if os.path.isfile(source):
61             log.info(f"Moving file:\n{source} -> {dst}")
62             shutil.move(source, dst)
63         elif os.path.isdir(source):
64             log.info(f"Moving folder:\n{source} -> {dst}")
65             shutil.copytree(source, dst, dirs_exist_ok=True)
66         else:
67             log.warning(f"Unknown file: {source}")
68
69     try:
70         shutil.rmtree(source_dir)

```

ภาพที่ 6 ตัวอย่างระบบอัปเดตเวอร์ชันกึ่งอัตโนมัติ updater.py

ผลจากการพัฒนาแชทบอทอัจฉริยะ ผู้ศึกษาได้ดำเนินการตามขั้นตอนและหลักการพัฒนาโปรแกรมตามที่ได้ออกแบบและวางแผนไว้ 5 ระบบประกอบด้วย 1) การติดตั้งและ Package management 2) ระบบส่วนกลางของบอท 3) ระบบแยกย่อย 4) ระบบสาธารณูปโภค และ 5) ระบบอัปเดตเวอร์ชันกึ่งอัตโนมัติ เพื่อให้โปรแกรมแชทบอทอัจฉริยะสามารถใช้งานได้จริง มีความเสถียร มีโครงสร้างที่ยืดหยุ่นได้ และได้โปรแกรมที่สมบูรณ์พร้อมใช้งาน

2. ผลการศึกษาความคิดเห็นจากผู้เชี่ยวชาญ

การศึกษาความคิดเห็นของผู้เชี่ยวชาญต่อการใช้งานแพลตฟอร์มอัจฉริยะได้ผลดังนี้

ตารางแสดงจำนวนและร้อยละของความคิดเห็นของผู้เชี่ยวชาญต่อการใช้แพลตฟอร์มอัจฉริยะ

ข้อความ	ระดับความคิดเห็น : จำนวน (ร้อยละ)					ค่าเฉลี่ย	แปลผลระดับ ความคิดเห็น
	ดีมาก	ดี	พอใช้	น้อย	น้อยมาก		
โครงสร้างระบบ	5 (66.7)	4 (33.3)				4.67	ดีมาก
ความเสถียรของแพลตฟอร์ม	5 (33.3)	4 (66.7)				4.33	ดี
ความเร็วของแพลตฟอร์ม	5 (66.7)	4 (33.3)				4.67	ดีมาก
คุณภาพของโค้ด	5 (100)					5.00	ดีมาก
ฟีเจอร์ของแพลตฟอร์ม	5 (100)					5.00	ดีมาก
สรุปโดยรวม						4.73	ดีมาก

จากตารางความคิดเห็นของผู้เชี่ยวชาญต่อแพลตฟอร์มอัจฉริยะ คุณภาพของโค้ด ความเร็วของแพลตฟอร์มมีค่าเฉลี่ย 5.00 อยู่ในระดับดีมาก รองลงมาคือ โครงสร้างระบบ ความเร็วของแพลตฟอร์ม และความเสถียรของแพลตฟอร์ม ตามลำดับ

3. ผลการศึกษาความคิดเห็นของกลุ่มตัวอย่าง

การศึกษาความคิดเห็นของกลุ่มตัวอย่างต่อการใช้งานแชทบอทอัจฉริยะได้ผลดังนี้

ตารางที่ 1 ตารางแสดงจำนวนและร้อยละของความคิดเห็นของกลุ่มตัวอย่างต่อการใช้แชทบอทอัจฉริยะ

ข้อความ	ระดับความคิดเห็น : จำนวน (ร้อยละ)					ค่าเฉลี่ย	แปลผลระดับ ความคิดเห็น
	ดีมาก	ดี	พอใช้	น้อย	น้อยมาก		
1. ระบบติดตามงานและกิจกรรม							
ความง่ายต่อการใช้งาน	5 (80)	4 (10)	3 (10)			4.70	ดีมาก
ประสิทธิภาพของระบบ	5 (60)	4 (30)	3 (10)			4.50	ดี
ความสะดวกสบาย	5 (60)	4 (40)				4.60	ดีมาก
ค่าเฉลี่ยระบบติดตามงานและกิจกรรม						4.60	ดีมาก
2. ระบบติดตามข่าวประชาสัมพันธ์							
ความง่ายต่อการใช้งาน	5 (80)	4 (20)				4.80	ดีมาก
ประสิทธิภาพของระบบ	5 (80)	4 (20)				4.80	ดีมาก
ความสะดวกสบาย	5 (100)					5.00	ดีมาก
ค่าเฉลี่ยระบบติดตามข่าวประชาสัมพันธ์						4.87	ดีมาก
3. ระบบเปิดเพลงเพื่อจรรโลงใจ							
ความง่ายต่อการใช้งาน	5 (80)	4 (20)				4.80	ดีมาก
ประสิทธิภาพของระบบ	5 (60)	4 (30)	3 (10)			4.50	ดี
ความสะดวกสบาย	5 (80)	4 (20)				4.80	ดีมาก
ค่าเฉลี่ยระบบเปิดเพลงเพื่อจรรโลงใจ						4.70	ดีมาก
สรุปโดยรวม						4.72	ดีมาก

จากตารางพบว่า ระบบติดตามข่าวประชาสัมพันธ์มีค่าเฉลี่ยมากที่สุดคือ 4.87 ซึ่งอยู่ในระดับดี

มาก รองลงมาด้วยระบบเปิดเพลงเพื่อจรรโลงใจ และระบบติดตามงานและกิจกรรมตามลำดับ

ตารางที่ 2 ตารางแสดงความคิดเห็นเพิ่มเติม และความถี่

ความคิดเห็นเพิ่มเติม	ความถี่
เป็นบอทที่ดีมาก เปิดเพลงได้หลากหลาย และมีประสบการณ์ในการใช้งานที่ดีมาก	1
เปิดเพลงได้ดี	1
ระบบข่าวประชาสัมพันธ์มีความรวดเร็ว	1

จากตารางที่ 2 พบว่ากลุ่มที่ศึกษามีการให้ความคิดเห็นเพิ่มเติมได้แก่ เป็นบอทที่ดีมาก เปิดเพลงได้หลากหลาย และมีประสบการณ์ในการใช้งานที่ดีมาก เปิดเพลงได้ดี ระบบข่าวประชาสัมพันธ์มีความรวดเร็ว

สะท้อนความคิด

สิ่งที่ได้เรียนรู้จากการศึกษา

ได้เรียนรู้การวางแผน และออกแบบโครงสร้างระบบระดับสูง ที่มีระบบย่อยๆ แยกออกจำนวนมาก และการใช้งาน Version Control ซึ่งเป็นสิ่งที่สำคัญอย่างมาก ในการนำไปใช้ศึกษาต่อและทำงานในอนาคต

ปัญหาหรืออุปสรรค

การใช้เซิร์ฟเวอร์ฟรี มีผลเสียมากมาย โดยเฉพาะด้าน Network เนื่องจากใน 1 เซิร์ฟเวอร์ สามารถมีผู้ใช้ได้มากกว่า 1 คน จึงทำให้เกิดการแย่ง หรือ สเปลม เกิดขึ้น และทำให้บอทโดน Discord API แบน อีกทั้งการทำบอทให้เปิดเพลงได้ ต้องการความรู้ และความสามารถสูง เนื่องจากไม่เป็นไปตามหลักเกณฑ์ปกติของคอมพิวเตอร์คือ Input Process Output เพราะ Output ซึ่งคือเสียงดนตรี สามารถรับค่า Input และ Process อีกครั้งขณะใดก็ได้ ทำให้เกิดการทำงานทับกัน และเกิดข้อผิดพลาดในการประมวลผลมากมาย

แนวทางการแก้ไขปัญหาหรืออุปสรรค

ผู้ศึกษาได้ทำการศึกษาเพิ่มเติม และพัฒนาระบบให้มีความเสถียรระดับหนึ่ง ที่เซิร์ฟเวอร์ฟรีสามารถทำได้ และผู้ศึกษาคาดและหวังอย่างยิ่งว่า จะสามารถพัฒนาแชทบอทให้ดีขึ้นกว่านี้ได้หากมีงบประมาณเข้ามาเกี่ยวข้อง

ข้อเสนอแนะ

ควรปรับปรุงระบบให้สามารถใช้งานง่ายมากกว่านี้ในด้านการเพิ่มงานและกิจกรรม เนื่องจากระบบมีความซับซ้อนสูง ทำให้ผู้ใช้งานเกิดความสับสนได้

บรรณานุกรม

ศาสตราจารย์ ดร. ชชาติรี. 2566. การประยุกต์ใช้เทคโนโลยี สารสนเทศกับการเรียนการสอน. (Online).

[https://docs.google.com/document/d/1nrzFoivvj7Zngm-](https://docs.google.com/document/d/1nrzFoivvj7Zngm-oprcT6ZC0QGFg27pUlsqM9QEXQU/)

[oprcT6ZC0QGFg27pUlsqM9QEXQU/](https://docs.google.com/document/d/1nrzFoivvj7Zngm-oprcT6ZC0QGFg27pUlsqM9QEXQU/), 27 กุมภาพันธ์ 2566

Peter Susic. 2566. **39+ Discord Statistics: Users, Servers, Demographics, Revenue**

(2023). (Online). <https://headphonesaddict.com/discord-users/>, 27 กุมภาพันธ์ 2566

Coursera. 2565. **What is Python Used For? A Beginner's Guide**. (Online).

[https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-](https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python)

[using-python](https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python), 29 มิถุนายน 2565

Replit. 2565. **Our mission**. (Online). <https://replit.com/site/about>, 21 มกราคม 2566

Danny, 2565. **Discord.py Documentation**. (Online).



<https://discordpy.readthedocs.io/en/stable/index.html>, 21 มกราคม 2566

ภาคผนวก

แบบประเมินความคิดเห็นจากผู้เชี่ยวชาญต่อแชทบอทอัจฉริยะ

แชทบอทอัจฉริยะ (TpLearn)

คลิก [ที่นี่](#) เพื่อเชิญบอทให้ Discord Server ของคุณ
(สำหรับผู้เชี่ยวชาญ)

 nuttee.2202@gmail.com (not shared) [Switch accounts](#) 

*Required

ความคิดเห็นต่อแชทบอท *

	ดีมาก	ดี	พอใช้	แย่	แย่มาก
โครงสร้างระบบ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ความเสถียรของบอท	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ความเร็วของบอท	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
คุณภาพของโค้ด	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ฟีเจอร์ของบอท	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

ความคิดเห็นเพิ่มเติม

Your answer

Submit
Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

แบบประเมินความคิดเห็นของกลุ่มตัวอย่างต่อแพลตฟอร์มอัจฉริยะ

แพลตฟอร์มอัจฉริยะ (TpLearn)

คลิกที่นี่ เพื่อเชื่อมแพลตฟอร์มให้ Discord Server ของคุณ

nuttee.2202@gmail.com (not shared) [Switch accounts](#)

*Required

1. ระบบติดตามงานและกิจกรรม *

	ดีมาก	ดี	พอใช้	น้อย	น้อยมาก
ความง่ายต่อการใช้งาน	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ประสิทธิภาพของงาน	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ความสะดวกสบาย	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. ระบบติดตามข่าวประชาสัมพันธ์ *

	ดีมาก	ดี	พอใช้	น้อย	น้อยมาก
ความง่ายต่อการใช้งาน	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ประสิทธิภาพของงาน	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ความสะดวกสบาย	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. ระบบเปิดเพลงเพื่อจรรโลงใจ *

	ดีมาก	ดี	พอใช้	น้อย	น้อยมาก
ความง่ายต่อการใช้งาน	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ประสิทธิภาพของงาน	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ความสะดวกสบาย	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

ข้อเสนอแนะ / ความคิดเห็น

Your answer

Submit
Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

วิธีการใช้งานแชทบอทอัจฉริยะ

ผู้ศึกษาได้ทำวิดีโอสอนการใช้งานแชทบอทอัจฉริยะบน Youtube โดยสามารถดูได้ผ่านทาง

QR Code ด้านล่าง

