

เว็บแอปพลิเคชัน มั่งลิงโก้

Monklingo

นายจักรพรรค์ ทองลา

โครงงานนี้เป็นส่วนหนึ่งของการศึกษา

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการข้อมูลและนวัตกรรมซอฟต์แวร์

ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

โครงการ : ระบบแสดงเส้นทางเดินบินทาบาด
 Monklingo
 โดย : นายจักรพรรค์ ทองลา
 อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.สุภาวดี หิรัญพงศ์สิน
 ระดับการศึกษา : วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการข้อมูลและนวัตกรรมซอฟต์แวร์
 ปีการศึกษา : 2567

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา

ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการข้อมูลและนวัตกรรมซอฟต์แวร์

คณะกรรมการสอบประเมินความรู้โครงการของคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.สุภาวดี หิรัญพงศ์สิน)

..... กรรมการ

(ชื่อกรรมการ)

..... กรรมการ

(ชื่อกรรมการ)

..... หัวหน้าภาควิชา

(รศ.ชาญชัย ศุภอรรถกร)

วันที่/...../.....

กิตติกรรมประกาศ

การพัฒนาโครงงานระบบแสดงเส้นทางเดินบินขบาท สำเร็จลุล่วงได้ด้วยความรู้และความช่วยเหลือจากหลายท่าน แม้ว่าจะประสบปัญหาในการดำเนินโครงการ ข้าพเจ้าได้รับคำแนะนำและการช่วยเหลือที่ดีเสมอมา ข้าพเจ้าจึงขอขอบพระคุณทุกท่านที่มีส่วนร่วมในการพัฒนาโครงงานนี้

ข้าพเจ้าขอกราบขอบพระคุณ ผศ.ดร.สุภาวดี หิรัญพงศ์สิน อาจารย์ที่ปรึกษาโครงงาน ซึ่งได้ให้คำแนะนำด้านทฤษฎีและแนวทางในการแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการพัฒนา ระบบ อีกทั้งยังคอยตรวจสอบความก้าวหน้าของการทำงานเป็นระยะ ๆ และให้กำลังใจแก่ผู้พัฒนาเสมอ

ขอขอบพระคุณ อาจารย์ประจำสาขาวิทยาการคอมพิวเตอร์ และ เจ้าหน้าที่ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ ที่ให้ความเอื้ออำนวยทั้งด้านอุปกรณ์และสถานที่สำหรับการปฏิบัติงานของผู้พัฒนา

ข้าพเจ้าขอกราบขอบพระคุณ บิดาและมารดา ที่คอยให้กำลังใจ มอบความรักและความห่วงใยเสมอมา รวมถึงการสนับสนุนด้านทุนทรัพย์เพื่อการศึกษา และอุปกรณ์ที่จำเป็นต่อการพัฒนาโครงงาน

ขอขอบคุณ เพื่อน พี่ น้อง สาขาวิทยาการคอมพิวเตอร์ ที่คอยช่วยเหลือ ให้คำปรึกษา และร่วมกันแก้ไขปัญหาต่าง ๆ จนกระทั่งโครงงานนี้สำเร็จลุล่วงด้วยความเคารพและซาบซึ้งใจ

จักรพรรค์ ทองลา

วัน เดือน ปี

โครงการงาน	:	ระบบแสดงเส้นทางเดินบิณฑบาต
		Monklingo
โดย	:	นายจักรพรรค์ ทองลา
อาจารย์ที่ปรึกษา	:	ผู้ช่วยศาสตราจารย์ ดร.สุภาวดี หิรัญพงศ์สิน
ระดับการศึกษา	:	วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการข้อมูลและนวัตกรรมซอฟต์แวร์
ปีการศึกษา	:	2567

บทคัดย่อ

การบิณฑบาตเป็นกิจกรรมสำคัญทางพระพุทธศาสนาและวัฒนธรรมไทยที่เสริมสร้างความสัมพันธ์ระหว่างพระสงฆ์และชุมชน อย่างไรก็ตาม การเปลี่ยนแปลงทางสังคมและความเจริญของเมืองทำให้พื้นที่ดั้งเดิม วิถีชีวิต และความสัมพันธ์ในชุมชนเปลี่ยนแปลงไป ส่งผลให้การเข้าถึงข้อมูลเส้นทางบิณฑบาตของพระสงฆ์ทำได้ยากขึ้น โครงการนี้จึงได้พัฒนา Monklingo เว็บแอปพลิเคชันที่นำเสนอข้อมูลเส้นทางบิณฑบาตอย่างถูกต้องและเป็นปัจจุบัน โดยพัฒนาด้วย Leaflet JavaScript Library ที่มีฟีเจอร์หลักในการแสดงแผนที่แบบโต้ตอบ โดยมีวัตถุประสงค์คือ 1) เพื่อรวบรวมและวิเคราะห์ข้อมูลเส้นทางบิณฑบาตของพระสงฆ์ในพื้นที่ใกล้เคียงมหาวิทยาลัยอุบลราชธานี และ 2) เพื่อแสดงผลเส้นทางบิณฑบาตและข้อมูลกิจกรรมทางศาสนาได้อย่างแม่นยำ ขอบเขตของโครงการนี้มุ่งเน้นเส้นทางบิณฑบาตของพระสงฆ์จากวัดป่าอรัญญวาสี โดยทดลองในภาคเรียนที่ 2 ปีการศึกษา 2567 ผลที่คาดว่าจะได้รับ คือ การให้ข้อมูลเส้นทางบิณฑบาตที่ถูกต้องและทันสมัย การอำนวยความสะดวกแก่ประชาชนในการเข้าร่วมกิจกรรมทางพุทธศาสนา และการสนับสนุนการสืบสานวัฒนธรรมการบิณฑบาต ซึ่งจะช่วยให้ชุมชนมีส่วนร่วมในการอนุรักษ์วัฒนธรรมพุทธศาสนาได้อย่างยั่งยืน

คำสำคัญ : การบิณฑบาต; พระพุทธศาสนา; วัฒนธรรมไทย; เว็บแอปพลิเคชัน; ลีฟเล็ท

Topic : Monklingo

Author : Jarkapat Tongla

Advisor : Assistant Professor Dr.Supawadee Hiranpongsin

Degree : Bachelor of Science (Data Science and Software Innovation)

Academic Year : 2024

Abstract

Pindapata is a significant practice in Buddhism and Thai culture, reinforcing the relationship between Buddhist monks and local communities. However, social changes and urban development have altered traditional environments, lifestyles, and community interactions, making it increasingly challenging to access information about monks' Pindapata routes. This project developed Monklingo, a web application designed to provide accurate and up-to-date information on Pindapata routes. The application was built using the Leaflet JavaScript Library, incorporating interactive map displays. The objectives of this project are: (1) to collect and analyze data on monks' Pindapata routes in the vicinity of Ubon Ratchathani University and (2) to accurately present Pindapata routes and relevant religious activities. The project focuses on the Pindapata routes of monks from Wat Pa Aranyawasi, with the experimental phase conducted during the second semester of the 2024 academic year. The expected outcomes include providing precise and current information on Pindapata routes, enhancing public participation in Buddhist activities, and supporting the preservation of Pindapata traditions. Ultimately, this initiative aims to foster sustainable community engagement in maintaining Buddhist cultural heritage.

Keywords: Pindapata; Buddhism; Thai Culture; Web Application; Leaflet

สารบัญ

กิตติกรรมประกาศ	3
บทคัดย่อ	4
Abstract	5
สารบัญ	6
สารบัญภาพ	6
บทที่ 1	
บทนำ	6
1.1 ที่มาและเหตุผล	6
1.2 วัตถุประสงค์	7
1.3 ขอบเขตของโครงการ	7
1.4 ประโยชน์ที่ได้รับ	7
1.5 เครื่องมือที่ใช้ในการพัฒนา	7
1.5.1 ฮาร์ดแวร์	7
1.5.2 ซอฟต์แวร์	7
1.6 แผนการดำเนินการ	8
บทที่ 2	
ทฤษฎีที่เกี่ยวข้อง	8
2.1 Kafka	8
บทที่ 3	
การออกแบบระบบ	11
3.1 System architecture	11
3.2 System requirement	11
3.3 การออกแบบหน้าจอ	11
3.4 Use case diagram	11
3.5 Class diagram	11
3.6 Sequence diagram	11
3.7 Data model/ Entity model	11
บทที่ 4	
การพัฒนาระบบ	12
3.1 System requirements	12
3.2 System architecture	13

3.3 การออกแบบหน้าจอ	14
3.4 Use case diagram	15
3.5 Class diagram	16
3.6 Sequence diagram	17
3.7 Data model/ Entity model	18
บทที่ 5	
การทดสอบระบบ	19
5.1 การทดสอบฟังก์ชัน...	20
5.2 การทดสอบฟังก์ชัน ...	20
บทที่ 6	
สรุปและข้อเสนอแนะ	20
6.1 สรุปความสามารถของระบบ	20
6.2 ปัญหาและอุปสรรคในการพัฒนา	20
6.3 แนวทางในการพัฒนาต่อ	20
บรรณานุกรม	20
ภาคผนวก ก	
การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม	22
ภาคผนวก ข	
คู่มือการติดตั้งระบบ	24
ภาคผนวก ค	
คู่มือการใช้งานของระบบ	24

สารบัญภาพ

บทที่ 1

บทนำ

1.1 ที่มาและเหตุผล

การบิณฑบาตรเป็นกิจกรรมที่สำคัญในพุทธศาสนาและมีความสัมพันธ์ใกล้ชิดกับชุมชนไทย การที่พระสงฆ์ออกบิณฑบาตรไม่เพียงแต่เป็นการดำเนินกิจกรรมทางศาสนา แต่ยังเป็นการสร้างความสัมพันธ์ที่แน่นแฟ้นระหว่างพระสงฆ์และประชาชนในชุมชน อย่างไรก็ตามด้วยการเปลี่ยนแปลงทางสังคมและความเจริญของเมืองใหญ่การหาข้อมูลเกี่ยวกับเส้นทางเดินบิณฑบาตรของพระสงฆ์ในปัจจุบันอาจไม่ง่ายดายเหมือนในอดีต

ในยุคสมัยที่บ้านเมืองและการดำเนินชีวิตเปลี่ยนแปลงอย่างรวดเร็วการที่ประชาชนจะทราบข้อมูลเส้นทางเดินบิณฑบาตรของพระสงฆ์ในแต่ละวันอาจไม่ใช่เรื่องง่าย โครงการนี้จึงนำเสนอเว็บแอปพลิเคชัน Monklingo ที่สามารถบอกเส้นทางเดินบิณฑบาตรของพระสงฆ์ได้อย่างถูกต้องและเป็นปัจจุบัน เพิ่มช่องทางที่ช่วยให้ผู้ใช้งานสามารถรู้สถานที่ที่พระสงฆ์เดินออกบิณฑบาตร และสามารถติดตามข่าวสารทางศาสนาพุทธได้อีกด้วย ดังนั้น Monklingo ไม่เพียงแต่จะช่วยในการสืบสานวัฒนธรรมและเสริมสร้างความสัมพันธ์ในชุมชนแต่ยังเป็นการสร้างโอกาสในการศึกษาและพัฒนาความต่อเนื่องเกี่ยวกับกิจกรรมทางศาสนาในสังคมไทย

1.2 วัตถุประสงค์

1.2.1 เพื่อรวบรวมและวิเคราะห์ข้อมูลเส้นทางเดินบิณฑบาตรของพระสงฆ์ในพื้นที่ใกล้เคียงมหาวิทยาลัยอุบลราชธานี

1.2.2 เพื่อพัฒนาเว็บแอปพลิเคชัน Monklingo ที่สามารถแสดงผลเส้นทางเดินบิณฑบาตรของพระสงฆ์และกิจกรรมทางศาสนาพุทธได้อย่างถูกต้องและเป็นปัจจุบัน

1.3 ขอบเขตของโครงการ

1. เส้นทางของวัดในพื้นที่ใกล้เคียงมหาวิทยาลัยอุบลราชธานีที่ศึกษา ได้แก่ วัดป่าอรัญญวาสี

2. ระยะเวลาในการทดลอง ภาคเรียนที่ 2 ปีการศึกษา 2567

1.4 ประโยชน์ที่ได้รับ

1. สามารถให้ข้อมูลเส้นทางเดินบิณฑบาตรของพระสงฆ์ได้อย่างถูกต้องและเป็นปัจจุบัน
2. ช่วยให้ประชาชนได้รับข่าวสารและสามารถเข้าร่วมกิจกรรมทางศาสนาพุทธได้สะดวกยิ่งขึ้น
3. สนับสนุนการสืบสานและอนุรักษ์วัฒนธรรมการบิณฑบาตร

1.5 เครื่องมือที่ใช้ในการพัฒนา

1.5.1 ฮาร์ดแวร์

1. เครื่องคอมพิวเตอร์ส่วนบุคคล เพื่อใช้ในการพัฒนาเว็บแอปพลิเคชัน โดยมีคุณสมบัติดังนี้
 - หน่วยประมวลผลกลาง CPU Intel ® Core™ i5 Quad-Core CPU @ 3GHz
(ขั้นต่ำต้องเป็น intel core i5)
 - หน่วยความจำหลัก ขนาด 8 GB
(ขั้นต่ำต้องเป็นขนาด 8 GB)
 - ระบบปฏิบัติการ: macOS Ventura

1.5.2 ซอฟต์แวร์ (Software)

1. Visual Studio Code (ขั้นต่ำ Version 1.90)
2. Django (ขั้นต่ำ Version 4.2.8) เป็นเฟรมเวิร์ค สำหรับพัฒนาเว็บไซต์
3. ภาษาคอมพิวเตอร์ Python (ขั้นต่ำ Version 3.8.6)
4. Google Chrome ใช้ในการแสดงผล
5. Leaflet Map (API) (ขั้นต่ำ Version 1.9.4)

1.5.3 ภาษาโปรแกรมที่ใช้

1. HTML ใช้ในการพัฒนาหน้าเว็บไซต์ (ขั้นต่ำ Version 5)
2. Tailwind CSS ใช้ในการตกแต่งหน้าเว็บไซต์ (ขั้นต่ำ Version 3.0)
3. Python (ขั้นต่ำ Version 3.8.6)
4. JavaScript ใช้ในการเขียน Leaflet Map (ขั้นต่ำ Version ES6)

1.5.4 ฐานข้อมูลที่ใช้

1. PostgreSQL (ขั้นต่ำ Version 14.15)

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงรายละเอียดเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการพัฒนา เว็บแอป การท่องเที่ยวในแบบสไลด์ของคุณ โดยแบ่งเนื้อหาออกเป็น 2 ส่วน ได้แก่ ส่วนที่หนึ่ง เป็นเนื้อหาพื้นฐานเกี่ยวกับทฤษฎีการเขียนโปรแกรมและเทคโนโลยีที่นำมาใช้ในการพัฒนาในหัวข้อที่ 2.1-2.7 ได้แก่ ความรู้พื้นฐานเกี่ยวกับ Visual studio code, Django Framework, PostgreSQL, Tailwind css, JavaScript, Leaflet map และ ngrok และในส่วนที่สองเป็นเนื้อหาเกี่ยวกับเว็บที่เกี่ยวข้องกับโครงการนี้ซึ่งแสดงไว้ในหัวข้อที่ 2.11

2.1 ความรู้เกี่ยวกับ Visual Studio Code

วิชวล สตูดิโอโค้ด (Visual Studio Code) [1] เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ (Microsoft) มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรีที่ต้องการความเป็นมืออาชีพ ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม (platform) รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา Python, JavaScript และ Node.js มี Extension ต่าง ๆ ให้เลือกใช้อย่างมาก การเปิดใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP หรือ Go สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อนต่างจาก Code Editor ตัวอื่นที่ต้องใช้ควบคู่ไปกับ Git Command

ตัวอย่าง Extension ที่นิยมใช้กัน

- Auto Rename Tag เป็น Extension ที่สามารถช่วยแก้ไขชื่อ Tag แล้วตัวปิด Tag จะทำการเปลี่ยนชื่อไปด้วย ทำให้เพิ่มความสะดวกในการพัฒนา
- Bracket Pair Colorizer เป็นตัวที่จำไว้วงเล็บในตัว Code ของเรามีสีที่หลากหลายทำให้ตรวจดูวงเล็บ รวดเร็วขึ้น
- Prettier เป็นตัวที่ใช้ในการ จัด Format Code ของเราซึ่งจะทำให้ Code ของเราเป็นระเบียบตัวง่ายขึ้นและยังรองรับหลายภาษาเช่น HTML, JavaScript, PHP และ CSS

2.2 ความรู้เกี่ยวกับ Django Framework

Django [2] เป็นเว็บเฟรมเวิร์กที่พัฒนาด้วยภาษา Python ที่ออกแบบมาเพื่อช่วยให้การพัฒนาเว็บแอปพลิเคชันเป็นเรื่องง่ายและรวดเร็ว มันถูกออกแบบมาให้ทำงานได้ดีในระดับสูงและมีคุณสมบัติที่หลากหลาย เพื่อสนับสนุนการพัฒนาเว็บแอปพลิเคชันที่มีความซับซ้อน

คุณสมบัติหลักของ Django

1. การพัฒนาอย่างรวดเร็ว (Rapid Development)
 - Django ช่วยลดเวลาในการพัฒนาเนื่องจากมีเครื่องมือและฟังก์ชันที่พร้อมใช้งานจำนวนมาก
2. การออกแบบโดยคำนึงถึงการรักษาความปลอดภัย (Security)
 - Django มีฟีเจอร์ความปลอดภัยในตัว เช่น การป้องกัน SQLInjection XSS CSRF และ

อื่น ๆ

3. ความสามารถในการขยาย (Scalability)
 - Django สามารถรองรับการพัฒนาแอปพลิเคชันขนาดใหญ่และรองรับการขยายตัวในอนาคตได้อย่างดี
4. การจัดการฐานข้อมูล (Database Management)
 - มี ORM (Object-Relational Mapping) ที่ช่วยให้การทำงานกับฐานข้อมูลเป็นเรื่องง่าย
5. แอดมินอินเทอร์เฟซอัตโนมัติ (Automatic Admin Interface)
 - Django มีแอดมินอินเทอร์เฟซที่สร้างขึ้นโดยอัตโนมัติสำหรับการจัดการข้อมูลในฐานข้อมูล
6. การทำงานร่วมกับแอปพลิเคชันอื่นๆ (Versatility)
 - สามารถทำงานร่วมกับแอปพลิเคชันอื่น ๆ และสามารถใช้งานแพ็คเกจของ Python ได้

2.3 ความรู้เกี่ยวกับ PostgreSQL

PostgreSQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์แบบโอเพนซอร์ส (Open Source Relational Database Management System - RDBMS) ที่ได้รับความนิยมสูงเนื่องจากความสามารถที่ครอบคลุมและความเสถียรที่โดดเด่น ได้รับการพัฒนาอย่างต่อเนื่องตั้งแต่ปี 1986 โดยมีพื้นฐานมาจากโครงการ POSTGRES ที่มหาวิทยาลัยแคลิฟอร์เนีย เบิร์กลีย์ (University of California, Berkeley) เป้าหมายหลักของ PostgreSQL คือการเป็นระบบฐานข้อมูลที่สามารถรองรับมาตรฐาน SQL ได้อย่างสมบูรณ์แบบ รวมถึงการเพิ่มความสามารถในการรองรับข้อมูลที่ซับซ้อนขึ้นและรองรับการขยายตัวได้ดี โดยมีคุณสมบัติที่โดดเด่นมากมายที่ทำให้ PostgreSQL เหมาะสำหรับงานระดับองค์กรและระบบขนาดใหญ่

หนึ่งในจุดเด่นของ PostgreSQL คือความสามารถในการรองรับประเภทข้อมูลที่หลากหลาย ไม่เพียงแต่ข้อมูลพื้นฐานอย่างตัวเลขและข้อความ แต่ยังรวมถึงประเภทข้อมูลที่ซับซ้อน เช่น JSON, XML, HSTORE (key-value store), Array และ Custom Data Types ซึ่งช่วยให้ PostgreSQL สามารถรองรับการพัฒนาแอปพลิเคชันที่ต้องการโครงสร้างข้อมูลที่ยืดหยุ่น นอกจากนี้ยังรองรับการทำงานร่วมกับระบบฐานข้อมูลเชิงเอกสารและ NoSQL ได้อย่างดี โดยสามารถใช้ JSONB ซึ่งเป็นประเภทข้อมูลที่ได้รับการปรับแต่งให้มีประสิทธิภาพสูงในการจัดเก็บและเรียกใช้ข้อมูลแบบ JSON

อีกคุณสมบัติสำคัญของ PostgreSQL คือความสามารถในการจัดการธุรกรรม (ACID Compliance) ซึ่งหมายถึงการรับรองว่าการดำเนินการกับฐานข้อมูลจะเป็นไปตามหลัก Atomicity (ความเป็นหน่วยเดียวของธุรกรรม), Consistency (ความถูกต้องของข้อมูล), Isolation (ความเป็นอิสระของแต่ละธุรกรรม) และ Durability (การคงอยู่ของข้อมูลหลังจากการทำธุรกรรมเสร็จสิ้น) คุณสมบัตินี้ทำให้ PostgreSQL เหมาะสำหรับใช้งานที่ต้องการความน่าเชื่อถือสูง เช่น ระบบการเงินและระบบธุรกิจที่ต้องการการบันทึกข้อมูลที่ถูกต้องและมีความสมบูรณ์

นอกจากนี้ PostgreSQL ยังรองรับการทำงานแบบ Concurrency Control ซึ่งช่วยให้สามารถรองรับการทำงานของผู้ใช้หลายคนได้พร้อมกันโดยไม่เกิดปัญหาการล็อกข้อมูล (Deadlock) หรือปัญหาการอ่านข้อมูลที่ไม่ถูกต้อง โดยใช้เทคนิค Multi-Version Concurrency Control (MVCC) ซึ่งช่วยให้ฐานข้อมูลสามารถให้บริการอ่านและเขียนข้อมูลได้อย่างมีประสิทธิภาพโดยไม่ต้องทำการล็อกตารางหรือแถวของข้อมูลทั้งหมด

อีกหนึ่งจุดเด่นที่สำคัญคือความสามารถในการขยายตัวและรองรับการทำงานแบบกระจาย (Scalability & Replication) PostgreSQL สามารถรองรับการทำงานแบบ Master-Slave Replication และ Logical Replication ซึ่งช่วยให้สามารถกระจายโหลดการทำงานไปยังเซิร์ฟเวอร์หลายตัวและช่วยให้สามารถสำรองข้อมูลเพื่อป้องกันการสูญหายของข้อมูล นอกจากนี้ยังสามารถใช้เทคนิค Sharding และ Partitioning เพื่อรองรับการจัดการข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ โดยสามารถแยกข้อมูลออกเป็นส่วน ๆ และกระจายไปยังเซิร์ฟเวอร์หลายตัวได้

PostgreSQL ยังมีระบบ Indexing ที่ทรงพลัง โดยรองรับหลายรูปแบบ เช่น B-Tree, Hash, GiST, GIN, SP-GiST และ BRIN ซึ่งช่วยให้สามารถเพิ่มประสิทธิภาพในการสืบค้นข้อมูลได้ดีขึ้น โดยเฉพาะอย่างยิ่งสำหรับข้อมูลที่มีโครงสร้างซับซ้อนหรือต้องการการสืบค้นที่รวดเร็ว นอกจากนี้ยังรองรับการใช้ Full-Text Search ซึ่งเป็นคุณสมบัติที่ช่วยให้สามารถค้นหาข้อมูลตามข้อความได้อย่างมีประสิทธิภาพ และสามารถใช้ร่วมกับ JSONB ได้ ทำให้ PostgreSQL เป็นตัวเลือกที่ดีสำหรับแอปพลิเคชันที่ต้องการความสามารถในการค้นหาข้อความเชิงลึก

ด้านความปลอดภัย PostgreSQL ให้ความสำคัญกับการป้องกันข้อมูล โดยมีระบบการควบคุมสิทธิ์การเข้าถึง (Access Control) ที่ยืดหยุ่นและปลอดภัย รองรับการใช้รหัสผ่าน การตรวจสอบสิทธิ์ผู้ใช้แบบต่าง ๆ เช่น MD5, SCRAM-SHA-256, LDAP และ Kerberos นอกจากนี้ยังสามารถใช้ Row-Level Security (RLS) ซึ่งช่วยให้สามารถกำหนดสิทธิ์การเข้าถึงข้อมูลในระดับแถวได้ ทำให้ PostgreSQL เหมาะสมสำหรับแอปพลิเคชันที่ต้องการความปลอดภัยสูง เช่น ระบบที่ต้องการควบคุมการเข้าถึงข้อมูลของผู้ใช้แต่ละกลุ่ม

PostgreSQL ยังมีความสามารถในการรองรับการเขียนโปรแกรมและการขยายตัวผ่าน Stored Procedures, Functions และ Triggers ซึ่งสามารถเขียนด้วยภาษา SQL, PL/pgSQL, Python, Perl และภาษาอื่น ๆ ทำให้สามารถสร้างฟังก์ชันที่ซับซ้อนได้โดยตรงภายในฐานข้อมูล นอกจากนี้ยังสามารถรองรับการทำงานร่วมกับ Machine Learning และ Big Data โดยมีการเชื่อมต่อกับเครื่องมือวิเคราะห์ข้อมูลเช่น Apache Spark, Hadoop และ TensorFlow

PostgreSQL ได้รับความนิยมอย่างมากในหมู่นักพัฒนาและองค์กรขนาดใหญ่ เช่น Apple, Microsoft, Uber, Netflix และ Instagram เนื่องจากมีความสามารถสูง สามารถรองรับการใช้งานที่หลากหลาย และมีชุมชนผู้ใช้ที่แข็งแกร่งที่ช่วยพัฒนาและแก้ไขปัญหาต่าง ๆ อย่างต่อเนื่อง อีกทั้งยังได้รับการสนับสนุนจากองค์กรและบริษัทที่ให้บริการโซลูชันทางด้านฐานข้อมูลเช่น EnterpriseDB และ Citus Data ซึ่งช่วยให้ PostgreSQL สามารถใช้ในระดับองค์กรได้อย่างมีประสิทธิภาพ

ด้วยคุณสมบัติเหล่านี้ PostgreSQL จึงเป็นหนึ่งในระบบจัดการฐานข้อมูลที่มีความสามารถสูงและเหมาะสมสำหรับงานที่ต้องการความน่าเชื่อถือสูง รองรับการใช้งานทั้งในแอปพลิเคชันขนาดเล็กไปจนถึงระบบที่ต้องจัดการข้อมูลขนาดใหญ่ในระดับองค์กร นอกจากนี้ยังมีการพัฒนาและอัปเดตเวอร์ชันใหม่อย่างต่อเนื่อง ทำให้สามารถรองรับเทคโนโลยีที่เปลี่ยนแปลงไปและสามารถแข่งขันกับระบบฐานข้อมูลเชิงพาณิชย์ได้อย่างมีประสิทธิภาพ

2.4 ความรู้เกี่ยวกับ Tailwind css

Tailwind CSS คือ CSS Utility Framework [9] ที่ช่วยให้นักพัฒนาสร้าง UI ที่สำคัญได้ด้วยตัวเองอย่างรวดเร็ว และยังสามารถปรับแต่งในรายละเอียดปลีกย่อยได้ง่าย เนื่องจากมาพร้อมกับ class สำเร็จรูปสุดจนแทบจะไม่ต้องเขียนโค้ดที่ซ้ำกันที่ในกรณีที่ต้องการเปลี่ยน UI หลักของเฟรมเวิร์ก เช่น สี ขนาด การจัดวาง หรือปุ่มต่างๆ นั้นทำให้นักพัฒนาแทบไม่ต้องเข้าไปแก้ไขไฟล์ CSS หลักเลย

```

1  <!-- Bootstrap -->
2  <button type="button" class="btn btn-primary">Primary</button>
3
4  <!-- tailwind css -->
5  <button type="button" class="bg-blue-600 text-gray-200 rounded hover:bg-blue-500 px-4 py-
```

ภาพที่ 2.1 ตัวอย่างการใช้งาน Tailwind css เมื่อเปรียบเทียบกับ Bootstrap

(ที่มา: <https://medium.com/@dektaiimage/ในบทความนี้เราจะมาพูดถึง-css-framework-ตัวหนึ่งที่มาแรงเป็นที่นิยมใช้กัน-ที่มีชื่อว่า-tailwind-css-132bf1ed6790>)

2.5 ความรู้เกี่ยวกับ JavaScript

JavaScript [10] เป็นภาษาสคริปต์เชิงวัตถุ ที่เรียกกันว่า “สคริปต์” (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งก่อนที่จะเริ่มเรียนเรื่อง JavaScript นั้น จำเป็นต้องมีความรู้พื้นฐานเรื่อง HTML และ CSS เสียก่อน

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h2>JavaScript in Body</h2>
6
7  <p id="demo"></p>
8
9  <script>
10 document.getElementById("demo").innerHTML = "My First JavaScript";
11 </script>
12
13 </body>
14 </html>
15

```

ภาพที่ 2.2 ตัวอย่างการใช้งาน JavaScript

(ที่มา: <https://medium.com/@worawit422/พื้นฐาน-javascript-เบื้องต้น-8976595b907a>)

2.6 ความรู้เกี่ยวกับ Leaflet map

Leaflet [11] เป็นไลบรารี JavaScript แบบโอเพนซอร์สที่ใช้ในการสร้างแผนที่แบบอินเทอร์แอคทีฟบนเว็บ เบาและยืดหยุ่น มีฟีเจอร์ที่ช่วยให้นักพัฒนาสามารถเพิ่มแผนที่ลงในเว็บแอปพลิเคชันได้ง่าย

จุดเด่นของ Leaflet map

- เบาและมีประสิทธิภาพ ขนาดเล็ก ทำงานได้เร็ว
- รองรับหลายเบราว์เซอร์และอุปกรณ์เคลื่อนที่ ทำงานได้ดีบนเบราว์เซอร์และอุปกรณ์หลากหลาย
- การปรับแต่งและขยายได้ง่าย มีปลั๊กอินให้เลือกใช้มากมายสำหรับการเพิ่มฟีเจอร์ต่างๆ
- การสร้างแผนที่แบบอินเทอร์แอคทีฟ รองรับการซูม เลื่อน และการเพิ่มองค์ประกอบต่างๆ บนแผนที่
- การจัดการ Events รองรับการจัดการเหตุการณ์ต่าง ๆ เช่น การคลิก การซูม และการเลื่อนแผนที่

May 18, 2023 — Leaflet 1.9.4 has been released!

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. Weighing just about 42 KB of JS, it has all the mapping [features](#) most developers ever need.

Leaflet is designed with *simplicity, performance and usability* in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of [plugins](#), has a beautiful, easy to use and [well-documented API](#) and a simple, readable [source code](#) that is a joy to [contribute](#) to.



Here we create a map in the 'map' div, add tiles of our choice, and then add a marker with some text in a popup:

```
var map = L.map('map').setView([51.505, -0.09], 13);

L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contrib'
}).addTo(map);

L.marker([51.5, -0.09]).addTo(map)
  .bindPopup('A pretty CSS popup.<br>Easily customizable.')
  .openPopup();
```

Learn more with the [quick start guide](#), check out [other tutorials](#), or head straight to the [API documentation](#). If you have any questions, take a look at the [FAQ](#) first.

ภาพที่ 2.3 ตัวอย่างการใช้งาน Leaflet map

(ที่มา: <https://leafletjs.com/>)

2.7 ความรู้เกี่ยวกับ ngrok

ngrok [4] เป็นเครื่องมือที่เชื่อมต่อเว็บแอปพลิเคชันกับ Line chatBot ในเครื่อง localhos โดยทำการสร้าง Tunnel หรือสามารถแชร์ URL ของ Tunnel ให้ผู้อื่นสามารถเข้าถึงเว็บแอปพลิเคชันได้ โดยสามารถใช้งานผ่าน url ของ ngrok ซึ่ง url ที่ได้มานั้น จะเปลี่ยนไปทุกครั้งเมื่อมีการปิด และเปิดใช้งานโปรแกรมใหม่

ข้อดีของ ngrok

- ใช้สำหรับพัฒนาระบบที่ต้องการ SSL
- ใช้สำหรับพัฒนาระบบเพื่อเชื่อมต่อ Web Hook
- ใช้สำหรับ Demo งานจากเครื่อง localhost ไม่มีค่าใช้จ่าย

บทที่ 3

การออกแบบระบบ

ในบทนี้จะกล่าวถึงการวิเคราะห์และออกแบบระบบ Monklingo ซึ่งการวิเคราะห์และการออกแบบระบบที่ใช้ในการพัฒนาจะแบ่งออกเป็น 6 ขั้นตอน ดังนี้

- 3.1 โครงสร้างภาพรวมของระบบ (System Achitecture) เป็นการออกแบบภาพรวมและเทคโนโลยีของระบบ
- 3.2 System Requirements คือ ความต้องการหรือหน้าที่หลักของระบบที่ต้องทำ
- 3.3 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface Design)
- 3.4 Use Case Diagram เป็นแผนภาพที่ใช้แสดงให้ทราบว่าระบบทำงานหรือมีหน้าที่ใดบ้าง
- 3.5 Class Diagram เป็นแผนภาพที่ใช้แสดงคลาส (Class) และความสัมพันธ์ระหว่างคลาส
- 3.6 Sequece Diagram เป็นแผนภาพที่ใช้แสดงผลให้เห็นถึงการตอบโต้ข้อมูลระหว่างคลาส เรียงตามลำดับของเวลาที่เกิดเหตุการณ์จากน้อยไปมาก

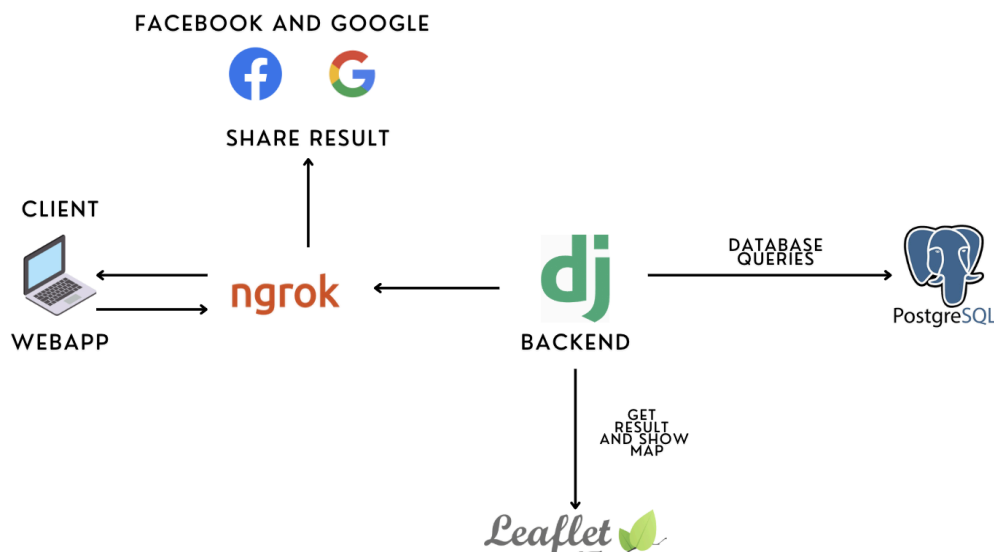
3.1 โครงสร้างภาพรวมของระบบ

3.1.1 โครงสร้างของระบบ

ความหมายของ System Architecture [13] การออกแบบและโครงสร้างของระบบที่ประกอบด้วยส่วนประกอบต่างๆ (components) และความสัมพันธ์ระหว่างส่วนประกอบเหล่านั้น โดยมุ่งเน้นที่การจัดการโครงสร้างเพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพและสอดคล้องกับความต้องการที่กำหนดไว้ การออกแบบสถาปัตยกรรมระบบเป็นสิ่งสำคัญที่ช่วยให้เข้าใจภาพรวมของระบบว่าทำงานอย่างไร และจะสามารถขยายหรือปรับปรุงได้อย่างไรในอนาคต

การออกแบบ System architecture แสดงภาพรวมและเทคโนโลยีของระบบเว็บแอป การท่องเที่ยวในแบบสไตล์ของคุณ ซึ่งมีรายละเอียดดังภาพที่ 3.1

3.1 System architecture



ภาพที่ 3.1 System architecture ระบบเว็บแอป Monklingo

จากภาพที่ 3.1 สามารถอธิบายโครงสร้างและเทคโนโลยี ของระบบได้ดังนี้

1. Client เป็น Web application ของผู้ใช้สำหรับส่งและแสดงผลข้อมูลบนหน้าเว็บ
2. Facebook เป็นแพลตฟอร์มโซเชียลมีเดียที่ให้บริการมากมายในด้านการสื่อสารไว้สำหรับ share results ของผู้ใช้
3. Google เป็นแพลตฟอร์มไว้สำหรับการเชื่อมต่อบัญชีของ Google
4. Ngrok เป็นเครื่องมือที่ทำให้ URL จาก localhost ของระบบเป็น public
5. Backend สร้างด้วย Django ใช้สำหรับจัดการข้อมูลที่ส่งมาจาก Client ผ่าน Ngrok และสามารถจัดการข้อมูล ใน Database
6. Leaflet สำหรับการแสดงแผนที่บนเว็บ ใช้ในการแสดงผลเส้นทางการเดินทางที่คำนวณได้ และสถานที่ท่องเที่ยวบนแผนที่ในเว็บแอปพลิเคชัน
7. Database เป็น PostgreSQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์

จากโครงสร้างของระบบ ที่แสดงดังภาพ 3.1 เริ่มต้นจากผู้ที่ใช้เว็บแอปพลิเคชัน (WEB APP) บนอุปกรณ์ต่าง ๆ เช่น คอมพิวเตอร์ แท็บเล็ต หรือ สมาร์ทโฟน ผู้ใช้ส่งคำร้อง (REQUEST) ไปยัง ngrok ซึ่งทำหน้าที่เป็นตัวกลางเพื่อส่งคำร้องนั้นไปยังเซิร์ฟเวอร์ Django (BACKEND) ที่ทำงานเป็นระบบหลักของแอปพลิเคชัน Django รับคำร้องและดำเนินการประมวลผลโดยการติดต่อกับฐานข้อมูล PostgreSQL เพื่อดึงข้อมูลที่เกี่ยวข้อง เมื่อได้ข้อมูลจากฐานข้อมูล PostgreSQL

แล้ว ระบบจะใช้ข้อมูลเหล่านั้นเพื่อคำนวณเส้นทางที่ดีที่สุดโดยใช้หลัก หลังจากนั้น Django จะนำข้อมูลที่คำนวณได้ไปแสดงบนแผนที่ด้วยการใช้ Leaflet เพื่อแสดงเส้นทางการเดินทาง

3.2 ความต้องการหรือหน้าที่หลักของระบบที่จะต้องทำ (System requirement)

3.2.1 ความต้องการของโปรแกรม (Functional Requirements)

ระบบเว็บแอป แสดงเส้นทางเดินทางเดินทาง แบ่งความสามารถของระบบตามประเภทของผู้ใช้งานดังนี้

ผู้ใช้งานระบบ

- ผู้ใช้งานทั่วไป
 - สมัครสมาชิก
- สมาชิก
 - เข้าสู่ระบบ
 - ออกจากระบบ
 - สามารถเลือกดูเส้นทางเดินทางเดินทางได้
 - ดูข่าวสารพุทธศาสนาได้
 - ดูบทสวดมนต์ได้
 - สามารถแก้ไขข้อมูลส่วนตัวได้
 - แข่งขันอันดับ Ranking การตักบาตร
 - แשרผลงานได้
 - ถ่ายรูปเพื่อยืนยันการตักบาตร
 - แก้ไขรูปการตักบาตร
 - ลบรูปการตักบาตร
 - พิมพ์ตอบโต้กับแอดมินได้
- ผู้ดูแลระบบ
 - โพสต์ข่าวสารพุทธศาสนาได้
 - แก้ไขโพสต์ข่าวสารข้อมูลได้
 - ลบโพสต์ข่าวสารพุทธศาสนาได้
 - เพิ่มบทสวดมนต์ได้
 - ลบบทสวดมนต์ได้
 - แก้ไขบทสวดมนต์ได้
 - ดู dashboard
 - อัปเดตข้อมูลในเว็บแอปพลิเคชัน

- สามารถลบสมาชิกได้
- สามารถดูข้อมูลสมาชิกได้
- สามารถดูรูปการดักบตร
- ลบรูปการดักบตรได้
- พิมพ์ตอบโต้กับสมาชิกได้
- แก้ไขเส้นทางาการเดินบิณบาต
- ลบเส้นทางาการเดินบิณบาต

3.3 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface Design)

ในการออกแบบแบ่งการออกเป็น 3 ส่วนคือ ส่วนผู้ใช้ ส่วนสมาชิก และส่วนผู้ดูแลระบบ

3.3.1 ส่วนของผู้ใช้งาน

โดยประกอบด้วยส่วนการออกแบบหน้าจอตางาน ดังนี้

1. หน้าแรก

Monklingo



ภาพที่ 3.2 หน้าแรก

จากภาพที่ 3.2 หน้าจอแสดงสำหรับหน้าแรก ที่มีปุ่มนำไปสู่หน้าสมัครสมาชิกและหน้าเข้าสู่ระบบ

ลบทิ้ง

2.การออกแบบหน้าลงทะเบียน

เข้าสู่ระบบ

ภาพที่ 3.3 หน้าลงทะเบียน




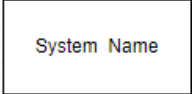
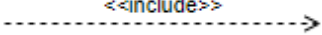
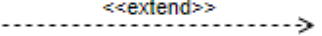
จากภาพที่ 3.3 หน้าจอแสดงสำหรับหน้าลงทะเบียนสมัครสมาชิก หากต้องการลงทะเบียนจะต้องกรอกข้อมูลชื่อผู้ใช้ เบอร์โทรศัพท์ อีเมล รหัสผ่าน และ ยืนยันรหัสผ่าน

3.การออกแบบหน้าสร้างรหัสผ่านใหม่

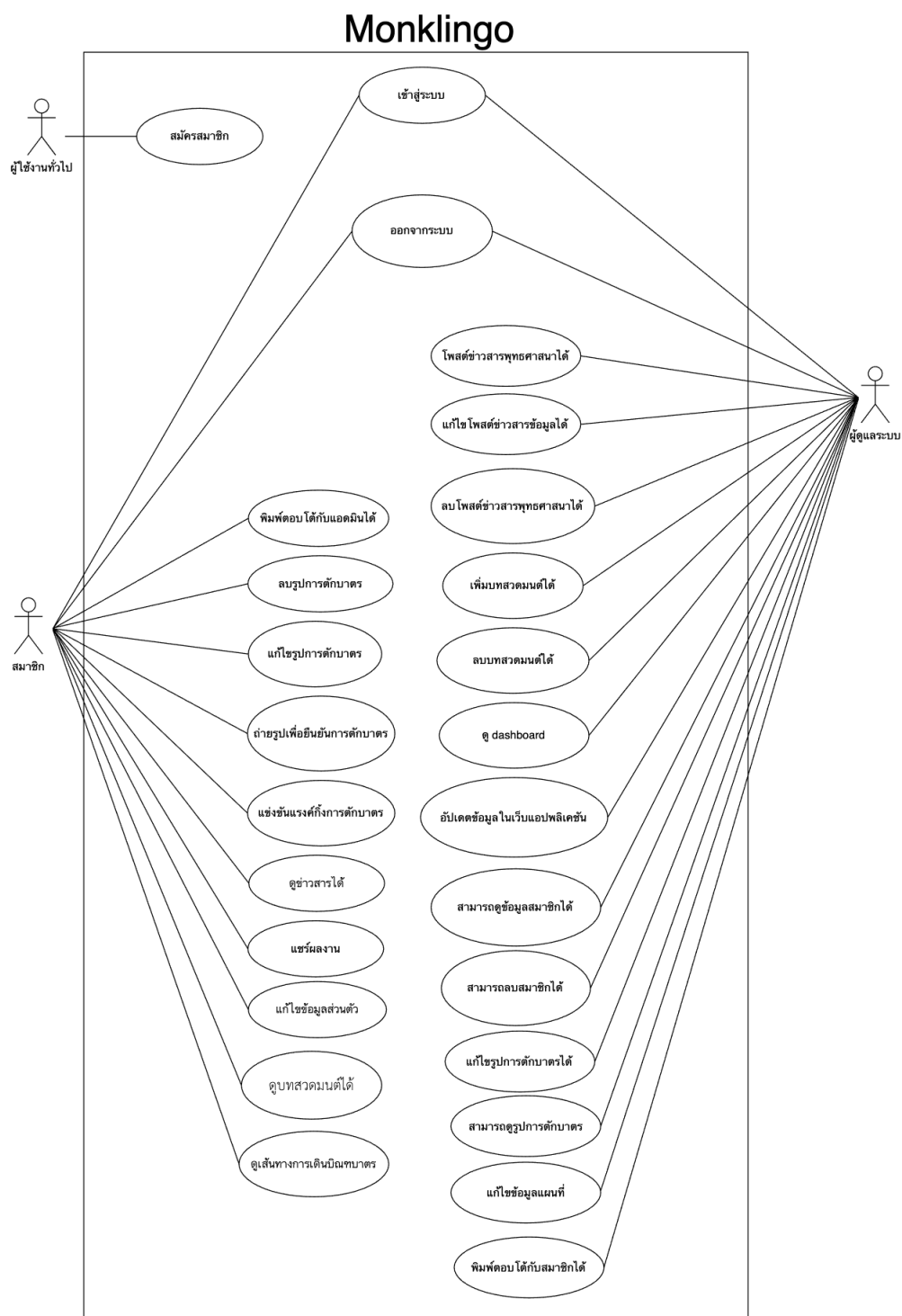
3.4 Use case diagram

Use Case Diagram เป็นแผนผังเพื่อแสดงฟังก์ชันแสดงการทำงานของระบบโดยรวม แสดงส่วนประกอบในระบบและกิจกรรมที่เกิดขึ้นในระบบ สัญลักษณ์ที่ใช้ในการเขียน Use Case Diagram แสดงในตาราง 3.1 และ Use Case Diagram ของเว็บระบบยืมอุปกรณ์ ดังภาพที่ 3.1

ตารางที่ 3.1 สัญลักษณ์ของ Use Case Diagram

สัญลักษณ์	คำอธิบาย
	Use Case คือ ส่วนย่อยของระบบงาน แทนด้วยวงรีและชื่อของ Use Case ภายในวงรี
	Actor คือ บุคคลหรือระบบงานอื่นที่ใช้งานระบบหรือได้รับประโยชน์จากระบบซึ่งอยู่ภายนอกระบบแทนด้วยรูปคนและมีชื่อบทบาทการใช้งานระบบ
	เส้นตรงที่แสดงถึงการใช้งาน Use Case ของผู้กระทำ
	กรอบสี่เหลี่ยม แสดงถึงขอบเขตของระบบโดยแสดงชื่อระบบภายในหรือด้านบนกรอบสี่เหลี่ยม Use Case อยู่ภายในกรอบสี่เหลี่ยม และ Actor อยู่ภายนอกกรอบสี่เหลี่ยม
	ความสัมพันธ์แบบ <<includes>> แสดงว่า Use Case ด้านหางลูกศร จำเป็นต้องมีการของ Use Case ด้านหัวลูกศร ทุกครั้งที่มีการทำงาน
	ความสัมพันธ์แบบ <<extend>> แสดงว่า Use Case หนึ่งดำเนินการตามขั้นตอนของ Use Case อื่นโดยแทนด้วยสัญลักษณ์ลูกศรเส้นประ ซึ่ง Use Case ด้านหางลูกศรเป็นส่วนเสริมในการทำงานของ Use Case ที่อยู่ด้านหัวลูกศร

Use Case Diagram : របប Monklingo



ภาพที่ 3.13 Use Case Diagram : ระบบ Monklingo

ตารางที่ 3.2 อธิบาย Use Case หน้าที่ของระบบในภาพที่ 3.2

Use Case	คำอธิบาย
1. สมัครสมาชิก	ผู้ใช้งานในระบบสามารถสมัครสมาชิกได้
2. เข้าสู่ระบบ	ผู้ดูแลระบบและผู้ใช้งานเป็นสมาชิกสามารถใช้ username และ password ของตัวเอง เพื่อเข้าสู่ระบบได้
3. ออกจากระบบ	ผู้ใช้งานในระบบสามารถกดออกจากระบบได้
4. ค้นหาเส้นทางการเดินทาง	ผู้ใช้งานในระบบค้นหาชื่อวัดเพื่อดูเส้นทางเดินทางของพระวัดนั้นๆได้
5. ดูข่าวสาร	ผู้ใช้งานในระบบสามารถดูข่าวสารของวัดต่างๆได้
6. ดูบทสวดมนต์	ผู้ใช้งานในระบบสามารถดูบทสวดมนต์ต่างๆ
7. แก้ไขข้อมูลส่วนตัว	ผู้ใช้งานในระบบสามารถแก้ไขข้อมูลส่วนตัวได้
8. แข่งขันอันดับแรงค์กัการตักบาตร	ผู้ใช้งานในระบบสามารถแข่งขันการตักบาตรได้ด้วยแต้มที่ได้จากการตักบาตรในแต่ละวัน
9. แร่ผลงาน	ผู้ใช้งานในระบบสามารถแร่ผลงานแรงค์กัตัวเองได้
10. ถ่ายรูปยืนยันการตักบาตร	ผู้ใช้งานในระบบสามารถถ่ายรูปเพื่อยืนยันการตักบาตรของตัวเองได้
11. แก้ไขรูปการตักบาตร	ผู้ใช้งานในระบบสามารถแก้ไขรูปภาพการตักบาตรของตัวเองได้
12. ลบรูปการตักบาตร	ผู้ใช้งานในระบบสามารถลบรูปภาพการตักบาตรของตัวเองได้
13. แชทพูดคุย	ผู้ใช้งานในระบบและผู้ดูแลระบบสามารถแชทพูดคุยกันได้
14. โพสต์ข่าวสาร	ผู้ดูแลระบบสามารถโพสต์ข่าวสารได้
15. แก้ไขโพสต์ข่าวสารข้อมูล	ผู้ดูแลระบบสามารถแก้ไขโพสต์ข่าวสารได้
16. ลบโพสต์ข่าวสารข้อมูล	ผู้ดูแลระบบสามารถลบโพสต์ข่าวสารได้
17. เพิ่มบทสวดมนต์	ผู้ดูแลระบบสามารถเพิ่มบทสวดมนต์ได้
18. ลบบทสวดมนต์	ผู้ดูแลระบบสามารถลบบทสวดมนต์ได้
19. ดู dashboard	ผู้ดูแลระบบสามารถดู Dashboard ได้
20. อัปเดตข้อมูลในเว็บ	ผู้ดูแลระบบสามารถอัปเดตข้อมูลในเว็บแอปพลิเคชันได้
21. ลบสมาชิก	ผู้ดูแลระบบสามารถลบสมาชิกในเว็บแอปพลิเคชันได้
22. ดูข้อมูลสมาชิก	ผู้ดูแลระบบสามารถดูข้อมูลสมาชิกในเว็บแอปพลิเคชันได้

23.ดูรูปการตัดกบตร	ผู้ดูแลระบบสามารถดูรูปการตัดกบตรในเว็บแอปพลิเคชันได้
24.แก้ไขรูปการตัดกบตร	ผู้ดูแลระบบสามารถแก้ไขรูปการตัดกบตรในเว็บแอปพลิเคชันได้

ตารางที่ 3.3 : อธิบาย Use Case สโมสรสมาชิก

Use Case Title : สโมสรสมาชิก	Use case Id : 1
------------------------------	-----------------

Primary Actor : ผู้ใช้งานในระบบ
Stakeholder Actor : -
Main Flow : ผู้ใช้งานในระบบ ทำการกรอกอีเมล ชื่อผู้ใช้งาน รหัสผ่าน ยืนยันรหัสผ่านแล้วสมัครสมาชิก
Exceptional Flow ที่ 1 : กรณีผู้ใช้งานกรอกชื่อผู้ใช้งาน อีเมลที่มีอยู่แล้วในระบบ ระบบจะแสดงข้อความ “สมัครสมาชิกไม่สำเร็จ”
Exceptional Flow ที่ 2 : กรณีผู้ใช้งานไม่ทำการกรอกข้อมูล ระบบจะแสดงข้อความ “กรุณากรอกข้อมูลนั้น ๆ”
Exceptional Flow ที่ 3 : กรณีผู้ใช้งานกรอกชื่อผู้ใช้งาน น้อยกว่า 3 ตัวหรือมากกว่า 20 ตัว ระบบจะแสดงข้อความ “ชื่อผู้ใช้งานต้องมีไม่น้อยกว่า 3 ตัวอักษร และไม่เกิน 20 ตัวอักษร”
Exceptional Flow ที่ 4 : กรณีผู้ใช้งานกรอกรหัสผ่าน น้อยกว่า 8 ระบบจะแสดงข้อความ “รหัสผ่านต้องมีไม่น้อยกว่า 8 ตัวอักษร”

ตารางที่ 3.4 : อธิบาย Use Case เข้าสู่ระบบ

Use Case Title : เข้าสู่ระบบ	Use case Id : 2
Primary Actor : ผู้ใช้งานในระบบ , ผู้ดูแลระบบ	
Stakeholder Actor : -	
Main Flow : เมื่อผู้ใช้งาน ทำการกรอกชื่อผู้ใช้งาน และรหัสผ่านแล้วเข้าสู่ระบบ	
Exceptional Flow ที่ 1 : กรณีผู้ใช้งานไม่กรอกข้อมูล ระบบจะแสดง “กรุณากรอกข้อมูลนั้น ๆ”	
Exceptional Flow ที่ 2 : กรณีผู้ใช้งานกรอกชื่อผู้ใช้งานผิดระบบจะแสดงข้อความ “ไม่มีผู้ใช้งานในระบบหรือรหัสผ่านไม่ถูกต้อง”	
Exceptional Flow ที่ 3 : กรณีผู้ใช้งานกรอกรหัสผ่านผิดระบบจะแสดงข้อความ “ไม่มีผู้ใช้งานในระบบหรือรหัสผ่านไม่ถูกต้อง”	

ตารางที่ 3.4 : อธิบาย Use Case ออกจากระบบ

Use Case Title : ออกจากระบบ	Use case Id : 3
Primary Actor : ผู้ใช้งานในระบบ , ผู้ดูแลระบบ	
Stakeholder Actor : -	

Pre-Condition : ผู้ใช้งานในระบบ , ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน
Main Flow : เมื่อผู้ใช้งาน ทำการกดออกจากกระบบแล้วจะออกจากกระบบ

ตารางที่ 3.4 : อธิบาย Use Case ดูเส้นทางเดินบินबाट

Use Case Title : ดูเส้นทางเดินบินबाट	Use case Id : 4
Primary Actor : ผู้ใช้งานในระบบ	
Stakeholder Actor : -	
Pre-Condition : 1.ผู้ใช้งานในระบบ ต้องทำการเข้าสู่ระบบก่อน 2.ต้องมีข้อมูลวัด	
Main Flow : เมื่อผู้ใช้งาน ทำการกดเลือกชื่อวัด ระบบจะแสดงวัดที่ต้องการ เมื่อกดเลือก ระบบจะแสดงข้อมูลเส้นทางเดินบินबाट ของวัดๆนั้นๆ	

ตารางที่ 3.4 : อธิบาย Use Case ดูข่าวสาร

Use Case Title : ดูข่าวสาร	Use case Id : 5
Primary Actor : ผู้ใช้งานในระบบ	
Stakeholder Actor : -	
Pre-Condition : 1.ผู้ใช้งานในระบบ ต้องทำการเข้าสู่ระบบก่อน 2.ต้องมีข้อมูลข่าวสาร	
Main Flow : เมื่อผู้ใช้งาน ต้องการดูและอ่านข่าวสาร	

ตารางที่ 3.4 : อธิบาย Use Case ดูบทสวดมนต์

Use Case Title : ดูบทสวดมนต์	Use case Id : 6
Primary Actor : ผู้ใช้งานในระบบ	
Stakeholder Actor : -	

Pre-Condition : 1.ผู้ใช้งานในระบบ ต้องทำการเข้าสู่ระบบก่อน 2.ต้องมีข้อมูลทดสอบ
Main Flow : เมื่อผู้ใช้งาน ต้องการดูและอ่านบทสวดมนต์

ตารางที่ 3.4 : อธิบาย Use Case แก้ไขข้อมูลส่วนตัว

Use Case Title : แก้ไขข้อมูลส่วนตัว	Use case Id : 7
Primary Actor : ผู้ใช้งานในระบบ	
Stakeholder Actor : -	
Pre-Condition : ผู้ใช้งานในระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ใช้งานแก้ไขข้อมูลส่วนตัว แล้วกดยืนยันข้อมูลจะถูกบันทึกลงในระบบ	

ตารางที่ 3.4 : อธิบาย Use Case แข่งขันอันดับแรงค์กึ่งการดักบาตร

Use Case Title : แข่งขันอันดับ แรงค์กึ่งการดักบาตร	Use case Id : 8
Primary Actor : ผู้ใช้งานในระบบ	
Stakeholder Actor : -	
Pre-Condition : ผู้ใช้งานในระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ใช้งานดักบาตรและถ่ายรูปส่งในระบบ ระบบจะเพิ่มคะแนนแรงค์ +1 หนึ่งครั้งต่อวัน คะแนนแรงค์กึ่งใช้แข่งขันกับผู้ใช้งานอื่นๆในระบบ และสามารถแชร์ผลงานไปช่องทางโซเชียลได้	

ตารางที่ 3.4 : อธิบาย Use Case แชร์ผลงาน

Use Case Title : แชร์ผลงาน	Use case Id : 9
Primary Actor : ผู้ใช้งานในระบบ	
Stakeholder Actor : -	

Pre-Condition : 1.ผู้ใช้งานในระบบ ต้องทำการเข้าสู่ระบบก่อน 2.ผู้ใช้งานต้องมีอันดับในระบบแรงก์กิ้ง
Main Flow : เมื่อผู้ใช้งานต้องการแชร์ผลงานอันดับแรงก์กิ้ง

ตารางที่ 3.4 : อธิบาย Use Case ถ่ายรูปยืนยันการตัดภาพตร

Use Case Title : ถ่ายรูปยืนยันการตัดภาพตร	Use case Id : 10
Primary Actor : ผู้ใช้งานในระบบ	
Stakeholder Actor : -	
Pre-Condition : ผู้ใช้งานในระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ใช้งานในระบบ ส่งรูปหลักฐานการตัดภาพตรและกดบันทึกข้อมูลระบบจะอัปโหลดรูปภาพเข้าระบบ	

ตารางที่ 3.4 : อธิบาย Use Case แก้ไขรูปการตัดภาพตร

Use Case Title : แก้ไขรูปการตัดภาพตร	Use case Id : 11
Primary Actor : ผู้ใช้งานในระบบ , ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : 1.ผู้ใช้งานในระบบและผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน 2.ผู้ใช้งานต้องมีรูปภาพ	
Main Flow : เมื่อผู้ใช้งานในระบบ ส่งรูปหลักฐานการตัดภาพตรและต้องการแก้ไขสามารถกดปุ่มแก้ไขเพื่อถ่ายรูปใหม่อีกครั้งได้ และเมื่อผู้ดูแลระบบ ต้องการแก้ไขรูปภาพ	
Exceptional Flow ที่ 1 : กรณีผู้ใช้งานกดยืนยันไปแล้วจะไม่สามารถแก้ไขรูปภาพได้	

ตารางที่ 3.4 : อธิบาย Use Case ลบรูปการตัดภาพตร

Use Case Title : ลบรูปการตัดภาพตร	Use case Id : 12
Primary Actor : ผู้ใช้งานในระบบ , ผู้ดูแลระบบ	
Stakeholder Actor : -	

Pre-Condition : 1.ผู้ใช้งานในระบบและผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน 2.ผู้ใช้งานต้องมีรูปภาพ
Main Flow : เมื่อผู้ใช้งานในระบบ ส่งรูปหลักฐานการตัดขาดและลบรูปภาพสามารถกดปุ่มลบเพื่อถ่ายรูปใหม่อีกครั้งได้ และเมื่อผู้ดูแลระบบ ต้องการลบรูปภาพ
Exceptional Flow ที่ 1 : กรณีผู้ใช้งานกดยืนยันไปแล้วจะไม่สามารถลบรูปภาพได้

ตารางที่ 3.4 : อธิบาย Use Case แชทพูดคุย

Use Case Title : แชทพูดคุย	Use case Id : 13
Primary Actor : ผู้ใช้งานในระบบ , ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : ผู้ใช้งานในระบบและผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : ผู้ใช้งานในระบบและผู้ดูแลระบบสามารถแชทพูดคุยตอบโต้กันได้	

ตารางที่ 3.4 : อธิบาย Use Case โฟสต์ข่าวสาร

Use Case Title : โฟสต์ข่าวสาร	Use case Id : 14
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : 1.ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ดูแลระบบ ทำการกรอกข้อมูลครบและกดบันทึกข้อมูล จะทำการสร้างโฟสต์ตามข้อมูลที่ได้กรอกไว้	
Exceptional Flow ที่ 1 : กรณีผู้ดูแลระบบกรอกข้อมูลไม่ครบจะไม่สามารถบันทึกข้อมูลได้ และระบบจะแสดง “กรุณากรอกข้อมูลนั้น ๆ ”	

ตารางที่ 3.4 : อธิบาย Use Case แก้ไขโฟสต์ข่าวสารข้อมูล

Use Case Title : แก้ไขโฟสต์ข่าวสารข้อมูล	Use case Id : 15
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : 1.ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน	

2.ต้องมีข้อมูลโพสต์
Main Flow : เมื่อผู้ดูแลระบบ ทำการกรอกข้อมูลที่ต้องการแก้ไขครบและกดบันทึกข้อมูล จะทำการสร้างโพสต์ตามข้อมูลที่ได้กรอกไว้
Exceptional Flow ที่ 1 : กรณีผู้ดูแลระบบกรอกข้อมูลไม่ครบจะไม่สามารถบันทึกข้อมูลได้ และระบบจะแสดง “กรุณากรอกข้อมูลนั้น ๆ ”

ตารางที่ 3.4 : อธิบาย Use Case ลบโพสต์ข่าวสารข้อมูล

Use Case Title : ลบโพสต์ข่าวสารข้อมูล	Use case Id : 16
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : 1.ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน 2.ต้องมีข้อมูลโพสต์	
Main Flow : เมื่อผู้ดูแลระบบ ต้องการลบโพสต์ของตนเอง	

ตารางที่ 3.4 : อธิบาย Use Case เพิ่มทสวดมนต์

Use Case Title : เพิ่มทสวดมนต์	Use case Id : 17
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : 1.ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ดูแลระบบ ทำการกรอกข้อมูลครบและกดบันทึกข้อมูล จะทำการสร้างทสวดมนต์ตามข้อมูลที่ได้กรอกไว้	
Exceptional Flow ที่ 1 : กรณีผู้ดูแลระบบกรอกข้อมูลไม่ครบจะไม่สามารถบันทึกข้อมูลได้ และระบบจะแสดง “กรุณากรอกข้อมูลนั้น ๆ ”	

ตารางที่ 3.4 : อธิบาย Use Case ลบทสวดมนต์

Use Case Title : ลบทสวดมนต์	Use case Id : 18
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : 1.ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน 2.ต้องมีข้อมูลทสวดมนต์	
Main Flow : เมื่อผู้ดูแลระบบ ต้องการลบทสวดมนต์	

ตารางที่ 3.4 : อธิบาย Use Case ดู dashboard

Use Case Title : ดู dashboard	Use case Id : 19
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ดูแลระบบ ต้องการดู dashboard	

ตารางที่ 3.4 : อธิบาย Use Case อัปเดตข้อมูลในเว็บแอปพลิเคชัน

Use Case Title : อัปเดตข้อมูลในเว็บแอปพลิเคชัน	Use case Id : 20
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ดูแลระบบ ต้องการอัปเดตข้อมูลในเว็บแอปพลิเคชัน	

ตารางที่ 3.4 : อธิบาย Use Case ลบสมาชิก

Use Case Title : ลบสมาชิก	Use case Id : 21
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ดูแลระบบ ลบสมาชิก	

ตารางที่ 3.4 : อธิบาย Use Case ดูข้อมูลสมาชิก

Use Case Title : ดูข้อมูลสมาชิก	Use case Id : 22
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ดูแลระบบ ต้องการดูข้อมูลสมาชิก	

ตารางที่ 3.4 : อธิบาย Use Case ดูรูปการดักบัตร

Use Case Title : ดูรูปการดักบัตร	Use case Id : 23
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : -	
Pre-Condition : ผู้ดูแลระบบ ต้องทำการเข้าสู่ระบบก่อน	
Main Flow : เมื่อผู้ดูแลระบบ ต้องการดูรูปการดักบัตร	

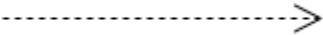

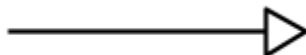

3.5 Class Diagram

Class Diagram คือแผนภาพที่ใช้แสดงคลาสและความสัมพันธ์ในรูปแบบต่างๆ ระหว่างคลาส สัญลักษณ์ที่ใช้ในการเขียน Class Diagram แสดงในตารางที่

3.18

ตารางที่ 3.21: สัญลักษณ์ของ Class Diagram

สัญลักษณ์	คำอธิบาย
-----------	----------

<div>Classname</div> <div>Attribute name</div> <div>Operation name()</div>	<p>Class สัญลักษณ์แทนด้วยสี่เหลี่ยมแบ่งเป็น 3 ส่วน ส่วนบนเป็นชื่อ Class ส่วนกลางเป็น Attribute และส่วนล่างเป็น Operation Name หรือ Method ซึ่งคลาสเป็นสิ่งที่เก็บรวบรวมข้อมูลที่แสดงถึงบุคคล สถานที่ เหตุการณ์หรือสิ่งต่างๆ ที่มีความเกี่ยวข้องกับระบบ Method เป็นการกระทำหรือฟังก์ชันที่คลาสนั้นสามารถทำได้</p>
Method Name()	<p>Method สามารถแบ่งการมองเห็น (Visibility) ได้ 3 ชนิด ได้แก่</p> <ul style="list-style-type: none"> - Public แทนสัญลักษณ์ด้วยเครื่องหมายบวก (+) - Private แทนสัญลักษณ์ด้วยเครื่องหมายลบ (-) - Protected แทนสัญลักษณ์ด้วยเครื่องหมายสี่เหลี่ยม (#)
	<p>Dependency Relationship หมายความว่า คลาสที่อยู่ฝั่งหางลูกศรสามารถเรียกใช้คลาสที่อยู่ฝั่งหัวลูกศร</p>
	<p>Composition Relationship เป็นความสัมพันธ์ระหว่าง object หรือคลาสแบบขึ้นต่อกันและมีความเกี่ยวข้องกันเสมอ</p>
	<p>Generalization Relationship เป็นความสัมพันธ์ระหว่าง object หรือคลาสในลักษณะของการสืบทอดคุณสมบัติจาก Class หนึ่ง (Super class) ไปยังอีก Class หนึ่ง (Sub class)</p>
	<p>Aggregation Relationship เป็นความสัมพันธ์ระหว่าง object หรือคลาสที่เป็นส่วนประกอบกันโดยส่วนที่เป็น Part class จะยังคงอยู่ได้โดยไม่ต้องพึ่งพา Whole class</p>

ภาพที่ 3.14: Class Diagram ระบบ Monklingo

3.5 Class diagram

3.6 Sequence diagram

3.7 Data model/ Entity model

บทที่ 4

การพัฒนาระบบ

3.1 System requirements

3.2 System architecture

3.3 การออกแบบหน้าจอ

3.4 Use case diagram

3.5 Class diagram

3.6 Sequence diagram

3.7 Data model/ Entity model

บทที่ 5

การทดสอบระบบ

ขั้นตอนการทดสอบระบบเพื่อตรวจสอบการทำงานของแต่ละฟังก์ชันกันรวมถึงตรวจสอบความถูกต้องของการทำงาน ทั้งระบบเพื่อให้การพัฒนาโปรแกรมเป็นไปอย่างมีประสิทธิภาพ และทดสอบระบบเพื่อหาข้อผิดพลาดของระบบ

5.1 การทดสอบฟังก์ชัน...

5.2 การทดสอบฟังก์ชัน ...

บทที่ 6

สรุปและข้อเสนอแนะ

6.1 สรุปความสามารถของระบบ

6.2 ปัญหาและอุปสรรคในการพัฒนา

6.3 แนวทางในการพัฒนาต่อ

บรรณานุกรม

Apache Kafka. Available from: <https://kafka.apache.org/>

sanook.com รวมข่าว ดูดวง หว่ย ผลบอล เพลง Joox เกม [Internet]. www.sanook.com. Available from: <https://www.sanook.com/>

ภาคผนวก

ภาคผนวก ก

การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม

การติดตั้งเครื่องมือที่ใช้ในการพัฒนาเว็บแอปพลิเคชันระบบจัดการลำดับเข้าสอบสัมภาษณ์แบบออนไลน์มีโปรแกรมที่จำเป็นในการพัฒนาระบบดังต่อไปนี้

- การติดตั้ง Visual Studio Code
- การติดตั้ง Python
- การติดตั้ง Docker

-

ภาคผนวก ข

คู่มือการติดตั้งระบบ

คู่มือการติดตั้งระบบ ในการติดตั้งเพื่อใช้งานระบบจัดการลำดับเข้าสอบสัมภาษณ์แบบออนไลน์ โดยใช้ Docker สามารถทำได้โดยมีขั้นตอนดังนี้

ภาคผนวก ค

คู่มือการใช้งานของระบบ

คู่มือการใช้งานทั้งหมดของระบบจัดการลำดับเข้าสอบสัมภาษณ์แบบออนไลน์มีขั้นตอนการใช้งาน

สามารถแบ่งออกเป็น 4 ส่วน ดังนี้

1. ส่วนของ Member
2. ส่วนของ Admin

