



## โครงการปัญหาพิเศษ (Special Problems Proposal)

เรื่อง เว็บแอประบบบริหารการจัดการในชมรม

Club Management System

เสนอต่อ

สาขาวิชาเทคโนโลยีสารสนเทศ คณะศิลปศาสตร์และวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

เพื่อขออนุมัติทำปัญหาพิเศษ

ปริญญา วิทยาศาสตรบัณฑิต

ภาคการศึกษา ต้น

โดย นายชิติพันธุ์ อินแพง

ภายใต้การควบคุมของ อาจารย์ สุริยะ พนิจการ

สาขา เทคโนโลยีสารสนเทศ

ปีการศึกษา 2568

ลายเซ็นต์อาจารย์ที่ปรึกษา.....

รับเรื่องวันที่ ...../...../.....

## คำนำ

### (Introduction)

ชมรมในมหาวิทยาลัยมีข้อมูลจำนวนมากที่ต้องจัดเก็บ เช่น สมาชิก ตารางกิจกรรม รายรับรายจ่าย ครุภัณฑ์ และเอกสารสำคัญต่าง ๆ การจัดการด้วยเอกสารหรือระบบที่ไม่มีประสิทธิภาพอาจนำไปสู่ความซ้ำซ้อน การสื่อสารล่าช้า และข้อมูลสูญหาย

โครงการนี้จึงเสนอการพัฒนาระบบเว็บแอปพลิเคชัน “ระบบบริหารการจัดการในชมรม” เพื่อบริหารข้อมูลภายในชมรมอย่างมีประสิทธิภาพ ครอบคลุมสมาชิก กิจกรรม เอกสารทางการเงินและสิทธิ์การเข้าถึงข้อมูล

## วัตถุประสงค์ของการทำปัญหาพิเศษ (Objectives)

เว็บแอประบบบริหารการจัดการในชุมชน มีวัตถุประสงค์ดังต่อไปนี้

- จัดเก็บและบริหารข้อมูลสมาชิกอย่างเป็นระบบ
- รองรับการลงทะเบียนนักศึกษา และตารางซ้อม/แสดง
- บันทึกและจัดทำรายงานทางการเงินอย่างโปร่งใส
- จัดเก็บเอกสารสำคัญอย่างเป็นหมวดหมู่
- บริหารข้อมูลโครงการและกิจกรรมให้มีประสิทธิภาพ
- บันทึกและติดตามทรัพย์สินครุภัณฑ์
- แจ้งเตือนกิจกรรมล่วงหน้า
- จัดการสิทธิ์ผู้ใช้งานตามบทบาท เช่น Admin, กรรมการ, สมาชิกทั่วไป

## การตรวจเอกสาร (Literature Review)

เว็บแอปสั่งอาหารในโรงอาหารล้วงหน้าของโรงอาหารในมหาวิทยาลัย ต้องใช้ความรู้และโปรแกรมในการพัฒนาระบบ ซึ่งประกอบด้วยโปรแกรมดังต่อไปนี้

- HTML, CSS, JavaScript

ภาษา HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) และ JavaScript เป็นเทคโนโลยีพื้นฐานสำหรับพัฒนาเว็บแอปพลิเคชันที่สามารถแสดงผลบนอุปกรณ์ต่างๆ ได้อย่างเหมาะสม โดย HTML ใช้โครงสร้างข้อมูล, CSS ใช้จัดรูปแบบ และ JavaScript ใช้ควบคุมการโต้ตอบ กับผู้ใช้งาน

( ที่มา: Mozilla Developer Network (MDN). HTML, CSS, and JavaScript Guides. )

## 2. Vue.js

Vue.js เป็น JavaScript Framework ที่เน้นความเรียบง่ายและยืดหยุ่น เหมาะสมสำหรับการสร้างส่วนติดต่อผู้ใช้งานแบบ Component-based โดยสามารถจัดการข้อมูลแบบ Reactive ได้อย่างมีประสิทธิภาพ และมีขนาดเบากว่า Framework อื่น ๆ เหมาะสมสำหรับเว็บแอปที่ต้องการความรวดเร็วในการโหลดและพัฒนา

(ที่มา: <https://vuejs.org/>)

## 3. Node.js และ Express.js

Node.js คือ Runtime Environment สำหรับเขียน JavaScript ฝั่งเซิร์ฟเวอร์ ส่วน Express.js เป็น Framework ที่ช่วยสร้าง API ได้อย่างรวดเร็ว

( ที่มา: nodejs.org, expressjs.com )

## 4. MySQL

MySQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ที่ใช้เก็บข้อมูลผู้ใช้ เมนูร้านค้า คำสั่งซื้อ และการให้คะแนนร้านอาหาร

( ที่มา: mysql.com )

## 5. Firebase Authentication

Firebase เป็น Platform ของ Google ที่มี Authentication Module สำหรับยืนยันตัวตนผู้ใช้งาน เช่น Email/Password หรือ Google Sign-in

( ที่มา: firebase.google.com )

## 6. Git และ GitHub

Git เป็นระบบควบคุมเวอร์ชัน (Version Control) ช่วยจัดการโค้ดและบันทึกการเปลี่ยนแปลง GitHub ใช้สำหรับเก็บ Repository ออนไลน์

( ที่มา: git-scm.com, github.com )

## System Requirement

เพื่อให้ระบบบริหารการจัดการในชุมชนสามารถทำงานได้อย่างมีประสิทธิภาพและตอบสนองความต้องการของผู้ใช้งาน ระบบจึงต้องมีคุณสมบัติที่สอดคล้องทั้งในด้านฟังก์ชันการทำงาน (Functional) และประสิทธิภาพระบบ (Non-Functional) ดังรายละเอียดต่อไปนี้

### 1. Functional Requirement

- ระบบสามารถเพิ่ม แก้ไข ลบ และค้นหาข้อมูลสมาชิกชุมชนได้
- รองรับการลงทะเบียนวงดนตรี และการจัดตารางซ้อม/แสดง
- บันทึกข้อมูลรายรับรายจ่าย พร้อมแนบเอกสารหลักฐาน และออกรายงานการเงินรายเดือน/รายปี เป็นไฟล์ PDF
- จัดเก็บและค้นหาเอกสารสำคัญ พร้อมกำหนดสิทธิ์การเข้าถึงตามบทบาทผู้ใช้
- จัดการข้อมูลโครงการและกิจกรรม เช่น ชื่อกิจกรรม งบประมาณ ผู้รับผิดชอบ และสถานะความคืบหน้า
- บันทึกและติดตามครุภัณฑ์ ทรัพย์สิน พร้อมค้นหาและออกรายงานประจำปี
- แสดงปฏิทินกิจกรรม สามารถเพิ่ม แก้ไข ลบ และให้ผู้ใช้ดูได้
- กำหนดสิทธิ์การเข้าถึงข้อมูลในแต่ละโมดูล ตามบทบาท เช่น ผู้ดูแลระบบ คณะกรรมการ และสมาชิกทั่วไป

### 2. Non-Functional Requirement

- ระบบต้องสามารถรองรับผู้ใช้งานพร้อมกันได้ไม่น้อยกว่า 100 คน โดยไม่เกิดปัญหาความล่าช้า
- รองรับการใช้งานทั้งบนคอมพิวเตอร์ แท็บเล็ต และสมาร์ตโฟน
- ข้อมูลภายในระบบต้องมีความปลอดภัย
- อินเทอร์เฟซต้องใช้งานง่าย สะดวกสำหรับผู้ใช้ทั่วไปที่ไม่ชำนาญด้านเทคโนโลยี

# แผนการทำปัญหาพิเศษ

## (Special Problem plan)

### 2. วัสดุและอุปกรณ์ (Materials and Equipments)

ในการทำเว็บแอปนี้ มีความจำเป็นอย่างยิ่งที่จะต้องนำคอมพิวเตอร์มาใช้ในการทำงาน โดยระบบคอมพิวเตอร์ที่จะนำมาใช้พัฒนาระบบนี้จะประกอบด้วย 2 ส่วนที่สำคัญ คือ

#### 1. ต้านฮาร์ดแวร์ (Hardware) มีรายละเอียด ดังนี้

##### 1.1 คอมพิวเตอร์สำหรับพัฒนาโปรแกรม

- CPU Intel i5/i7 หรือเทียบเท่า ความเร็วตั้งแต่ 2.5 GHz ขึ้นไป
- RAM ตั้งแต่ 8 GB ขึ้นไป
- Hard Disk ตั้งแต่ 256 GB ขึ้นไป

##### 1.2 สมาร์ตโฟนหรือแท็บเล็ต สำหรับทดสอบระบบ

##### 1.3 อุปกรณ์เชื่อมต่อเครือข่ายอินเทอร์เน็ต

#### 2. ต้านซอฟต์แวร์ (Software) มีรายละเอียด ดังนี้

##### 2.1 Visual Studio Code

##### 2.2 Node.js

##### 2.3 MySQL Server

##### 2.4 Git

##### 2.5 Postman

##### 2.6 Web Browser: Google Chrome, Microsoft Edge

#### 3. ไลบรารีและเฟรมเวิร์ก

##### 3.1 Vue.js

##### 3.2 Express.js

##### 3.3 Firebase Authentication SDK

##### 3.4 Bootstrap หรือ Tailwind CSS

#### 4. เครื่องมือจัดทำเอกสาร

##### 1.1 Microsoft Word

##### 1.2 Microsoft Visio หรือ Draw.io

### 3. วิธีการ (Methodology)

ขั้นตอนการดำเนินการ โครงการนี้ได้แบ่งการพัฒนาออกเป็น 6 ขั้นตอนหลัก คือ

#### 1. ศึกษาปัญหาและความต้องการของระบบ

ทำการเก็บรวบรวมข้อมูลจากคณะกรรมการและสมาชิกชุมชน เพื่อวิเคราะห์ปัญหาในการจัดการข้อมูลภายในชุมชน เช่น การจัดเก็บสมาชิก, การบริหารตารางกิจกรรม, การจัดการเอกสาร และทรัพย์สินของชุมชน จากนั้นสรุปเป็น Requirement สำหรับใช้ในการออกแบบระบบ

#### 2. วิเคราะห์ระบบ

วิเคราะห์บทบาทของผู้ใช้งาน เช่น ผู้ดูแลระบบ คณะกรรมการ และสมาชิกทั่วไป เพื่อนำมากำหนดสิทธิ์การเข้าถึงข้อมูลและฟังก์ชันต่าง ๆ จากนั้นจัดทำ Use Case Diagram และ Data Flow Diagram เพื่อแสดงกระบวนการทำงานและการไหลของข้อมูลภายในระบบ พร้อมออกแบบ ER Diagram เพื่อกำหนดโครงสร้างฐานข้อมูลให้เหมาะสมกับการใช้งานในแต่ละส่วน

#### 3. ออกแบบระบบ

ออกแบบหน้าต่างระบบด้วย Wireframe โดยเน้นให้ใช้งานง่ายและตอบโจทย์ผู้ใช้แต่ละกลุ่ม จากนั้นจึงออกแบบโครงสร้างระบบทั้งผึ้งผู้ใช้งานและผึ้งเชิฟเวอร์ รวมถึงออกแบบฐานข้อมูลให้สามารถจัดเก็บและเขียนอย่างข้อมูลในแต่ละโมดูลได้อย่างมีประสิทธิภาพ

#### 4. พัฒนาระบบ

เริ่มต้นพัฒนาระบบโดยใช้ Vue.js สำหรับส่วน Frontend และ Node.js/Express.js สำหรับส่วน Backend เชื่อมต่อกับฐานข้อมูล MySQL และใช้ Firebase สำหรับระบบยืนยันตัวตน พัฒนาแต่ละโมดูลให้ตรงตามความต้องการที่กำหนดไว้ แล้วเตรียมระบบให้พร้อมสำหรับการทดสอบในขั้นตอนถัดไป

#### 5. ทดสอบการทำงาน สรุปผลการทำงาน

ดำเนินการทดสอบการทำงานของระบบในแต่ละส่วนแยก (Unit Test) และการทำงานร่วมกันของระบบทั้งหมด (Integration Test) โดยมีการทดสอบจริงกับกลุ่มผู้ใช้งานเป้าหมาย เช่น กรรมการและสมาชิกชุมชน เพื่อรับ Feedback และนำมาปรับปรุงให้ระบบสมบูรณ์ยิ่งขึ้น

#### 6. จัดทำเอกสารรูปเล่มปัญหาพิเศษ

รวบรวมผลการดำเนินงานตั้งแต่ขั้นตอนการวิเคราะห์ ออกแบบ พัฒนา และทดสอบระบบ นำเสนอในรูปแบบของเอกสารรายงาน พร้อมจัดทำคู่มือการใช้งานระบบสำหรับผู้ดูแลชุมชนและสมาชิกทั่วไป รวมถึงแนวทางการพัฒนาต่อไปในอนาคต

#### 4. ตารางแผนงานปัจจุบันพิเศษ (Timetable)

ขั้นตอน	เดือนที่ 1	เดือนที่ 2	เดือนที่ 3	เดือนที่ 4
ศึกษาความต้องการ	↔			
วิเคราะห์ระบบ	↔			
ออกแบบระบบ		↔		
พัฒนาระบบ		↔	→	
ทดสอบและแก้ไข			↔	
สรุปผลและจัดทำเอกสาร				↔

#### สถานที่และระยะเวลา (Places and Duration)

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน

#### ระยะเวลา (Duration)

เริ่มตั้งแต่เดือนมิถุนายน ถึง เดือนตุลาคม 2568

#### ประโยชน์ที่คาดว่าจะได้รับ (Benefits)

ประโยชน์ที่คาดว่าจะได้รับจากการทำ เว็บแอประบบบริหารการจัดการในชุมชน มีดังนี้

- ลดความซ้ำซ้อนของเอกสาร และการจัดเก็บแบบกระดาษ
- ข้อมูลเป็นระบบ ค้นหา/อกรายงานได้ง่าย
- เพิ่มความโปร่งใสในการบริหารงบประมาณ
- เพิ่มการมีส่วนร่วมของสมาชิกในกิจกรรมชุมชน

แหล่งทุนสนับสนุน  
(Funding Source)

สาขาวิชาเทคโนโลยีสารสนเทศ คณะศิลปศาสตร์และวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์  
วิทยาเขตกำแพงแสน

เอกสารและสิ่งอ้างอิง  
(Literature Cited)

Mozilla Developer Network (MDN). HTML, CSS, and JavaScript Guides. [ออนไลน์].

แหล่งที่มา: <https://developer.mozilla.org>

Vue.js. The Progressive JavaScript Framework. [ออนไลน์].

แหล่งที่มา: <https://vuejs.org>

Node.js. Node.js Official Documentation. [ออนไลน์].

แหล่งที่มา: <https://nodejs.org>

Express.js. Fast, unopinionated, minimalist web framework. [ออนไลน์].

แหล่งที่มา: <https://expressjs.com>

MySQL. MySQL Reference Manual. [ออนไลน์].

แหล่งที่มา: <https://dev.mysql.com>

Firebase Authentication. Firebase Authentication Documentation. [ออนไลน์].

แหล่งที่มา: <https://firebase.google.com>

Git SCM. Git - Distributed Version Control System. [ออนไลน์].

แหล่งที่มา: <https://git-scm.com>

GitHub Docs. Introduction to GitHub. [ออนไลน์].

แหล่งที่มา: <https://docs.github.com>

Tailwind CSS. A utility-first CSS framework for rapidly building custom user interfaces.

[ออนไลน์].

แหล่งที่มา: <https://tailwindcss.com>

W3C. HTML5 Specification. [ออนไลน์]. แหล่งที่มา:

<https://www.w3.org/TR/html5/>

Tailwind CSS. A utility-first CSS framework. [ออนไลน์]. แหล่งที่มา:

<https://tailwindcss.com>