

แบบนำเสนอโครงการและแผนดำเนินโครงการ  
รายวิชา Programming Fundamentals II รหัสวิชา 03603112  
มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา  
ภาคปลาย ปีการศึกษา 2567

1. ชื่อโครงการ

MeChat: Real-Time Chat Application in Java

2. สมาชิก

- 2.1. นายชนกันต์ เขียวภักดี 6730300108
- 2.2. นายทรงวุฒิ โคนัก 6730300175

3. รายละเอียดโดยย่อ

โครงการนี้มีวัตถุประสงค์เพื่อพัฒนาแอปพลิเคชันแชทที่สามารถใช้งานได้แบบเรียลไทม์โดยใช้ภาษา Java เป็นหลัก โดยจะมีฟังก์ชันพื้นฐานเช่น การส่งข้อความระหว่างผู้ใช้ การสร้างห้องแชท และการแจ้งเตือนข้อความใหม่ ระบบจะถูกพัฒนาให้รองรับการทำงานแบบ Client-Server และสามารถเชื่อมต่อผ่านเครือข่ายอินเทอร์เน็ตได้

4. คุณลักษณะและขอบเขต

4.1. คุณลักษณะขั้นต่ำ

- 4.1.1. ผู้ใช้สามารถลงทะเบียนและเข้าสู่ระบบได้
- 4.1.2. รองรับแชทแบบ 1-ต่อ-1 (Private Chat)
- 4.1.3. ใช้งานผ่านเครือข่ายอินเทอร์เน็ตได้
- 4.1.4. ระบบแจ้งเตือนเมื่อมีข้อความใหม่
- 4.1.5. แก้ไข/ลบข้อความ

4.2. คุณลักษณะเพิ่มเติม

- 4.2.1. รองรับการส่งไฟล์และรูปภาพ
- 4.2.2. รองรับแชทเป็นกลุ่ม (Group Chat)
- 4.2.3. การเข้ารหัสข้อความเพื่อเพิ่มความปลอดภัย
- 4.2.4. ระบบแสดงสถานะออนไลน์/ออฟไลน์ของผู้ใช้
- 4.2.5. ระบบจัดการเพื่อน

5. แผนดำเนินโครงการ

ระบบ	ช่วงเวลา	ผลลัพธ์ที่คาดหวัง
เริ่มต้นโครงการ	สัปดาห์ที่ 1	- ส่งข้อเสนอโครงการ
วิเคราะห์ความต้องการ	สัปดาห์ที่ 1	- ศึกษาความต้องการของโครงการ

ออกแบบระบบ	สัปดาห์ที่ 1-2	- สร้าง GitHub repository - สร้างโครงงานด้วย Maven และเขียนโค้ดตั้งต้น
พัฒนาระบบ	สัปดาห์ที่ 3-4	- โปรแกรมทำงานได้ในเบื้องต้น - มีฟีเจอร์หลักที่ใช้งานได้
ทดสอบ	สัปดาห์ที่ 5-6	- ฟีเจอร์ส่วนใหญ่พร้อมใช้งาน - โปรแกรมสามารถใช้งานได้ แต่อาจต้องปรับแก้หรือเพิ่มเติม
ส่งมอบงาน	สัปดาห์ที่ 7-8	- โปรแกรมมีเจอรืครบถ้วนตามข้อกำหนด - เอกสารประกอบสมบูรณ์ พร้อมสำหรับการนำเสนอ

## 6. การแบ่งงาน

### 6.1. นายชนกันต์ เขียวภักดี

6.1.1. ออกแบบ REST API และ WebSocket

6.1.2. ออกแบบ Database

6.1.3. จัดการ GitHub repository

6.1.4. ทำระบบส่วน Client

6.1.5. ทำระบบส่วน Server

6.1.6. ทำเอกสาร

### 6.2. นายทรงวุฒิ โคนัก

6.2.1. ออกแบบ UX/UI

6.2.2. ทำระบบส่วน Client

6.2.3. ทำระบบส่วน Server

6.2.4. ทดสอบระบบ

6.2.5. ทำเอกสาร

## 7. ความท้าทายและความเสี่ยง

### 7.1. ความท้าทายด้านเทคนิค

7.1.1. การพัฒนาระบบให้ทำงานได้แบบเรียลไทม์

7.1.2. การจัดการฐานข้อมูลขนาดใหญ่และการเข้ารหัสข้อมูลเพื่อความปลอดภัย

7.1.3. การออกแบบ UI/UX ให้รองรับการใช้งานที่สะดวกและรวดเร็ว

7.1.4. การทดสอบและแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นจากการสื่อสารระหว่าง Client และ Server

7.1.5. การพัฒนาและเชื่อมต่อ API และ WebSocket ที่อาจเกิดปัญหาด้านความเข้ากันได้

### 7.2. ความเสี่ยงด้านเวลา

7.2.1. การพัฒนาระบบที่ต้องใช้เวลาในการแก้ไขบั๊กและปรับปรุงฟังก์ชันการทำงาน

7.2.2. การประสานงานระหว่างสมาชิกทีมที่อาจทำให้เกิดความล่าช้า

7.2.3. การทดสอบและปรับปรุง UI/UX ที่อาจใช้เวลามากกว่าที่คาดการณ์

8.    ต้นแบบและเอกสารอ้างอิง

- 8.1.   Spring Boot <https://docs.spring.io/spring-boot>
  - 8.1.1. Spring Boot Web
  - 8.1.2. Spring Boot WebFlux
  - 8.1.3. Spring Boot WebSocket
  - 8.1.4. Spring Boot Data JPA
- 8.2.   JavaFx <https://fxdocs.github.io/docs/html5>
- 8.3.   MySQL <https://w3schools.com/MySQL>
- 8.4.   MariaDB <https://mariadb.com/kb/en/about-mariadb-connector-j>
- 8.5.   Maven Repository <https://mvnrepository.com>
- 8.6.   Java 21 <https://docs.oracle.com/en/java/javase/21>