**แบบนําเสนอโครงงานและแผนดำเนินโครงงาน**

**รายวิชา Programming Fundamentals II รหัสวิชา 03603112**

**มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา**

**ภาคปลาย ปีการศึกษา 2567**

1. ชื่อโครงงาน

MeChat: Real-Time Chat Application in Java

1. สมาชิก
   1. นายชนกันต์ เขียวภักดี 6730300108
   2. นายทรงวุฒิ โคนัก 6730300175
2. รายละเอียดโดยย่อ

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

1. คุณลักษณะและขอบเขต
   1. คุณลักษณะขั้นต่ำ
      1. ผู้ใช้สามารถลงทะเบียนและเข้าสู่ระบบได้
      2. รองรับการแชทแบบ 1-ต่อ-1 (Private Chat)
      3. รองรับการแชทเป็นกลุ่ม (Group Chat)
      4. ใช้งานผ่านเครือข่ายอินเทอร์เน็ตได้
      5. ระบบแจ้งเตือนเมื่อมีข้อความใหม่
   2. คุณลักษณะเพิ่มเติม
      1. รองรับการส่งไฟล์และรูปภาพ
      2. การเข้ารหัสข้อความเพื่อเพิ่มความปลอดภัย
      3. ระบบแสดงสถานะออนไลน์/ออฟไลน์ของผู้ใช้
      4. ระบบจัดการเพื่อน
2. แผนดำเนินโครงงาน

|  |  |  |
| --- | --- | --- |
| **ระบบ** | **ช่วงเวลา** | **ผลลัพธ์ที่คาดหวัง** |
| เริ่มต้นโครงงาน | สัปดาห์ที่ 1 | - ส่งข้อเสนอโครงงาน |
| วิเคราะห์ความต้องการ | สัปดาห์ที่ 1 | - ศึกษาความต้องการของโครงงาน |
| 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 | - โปรแกรมมีเจอร์ครบถ้วนตามข้อกำหนด  - เอกสารประกอบสมบูรณ์ พร้อมสำหรับการนำเสนอ |

1. การแบ่งงาน
   1. นายชนกันต์ เขียวภักดี
      1. ออกแบบ REST API และ WebSocket
      2. ออกแบบ Database
      3. จัดการ GitHub repository
      4. ทำระบบส่วน Client
      5. ทำระบบส่วน Server
      6. ทำเอกสาร
   2. นายทรงวุฒิ โคนัก
      1. ออกแบบ UX/UI
      2. ทำระบบส่วน Client
      3. ทำระบบส่วน Server
      4. ทดสอบระบบ
      5. ทำเอกสาร
2. ความท้าทายและความเสี่ยง
   1. ความท้าทายด้านเทคนิค
      1. การพัฒนาระบบให้ทำงานได้แบบเรียลไทม์
      2. การจัดการฐานข้อมูลขนาดใหญ่และการเข้ารหัสข้อมูลเพื่อความปลอดภัย
      3. การออกแบบ UI/UX ให้รองรับการใช้งานที่สะดวกและรวดเร็ว
      4. การทดสอบและแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นจากการสื่อสารระหว่าง Client และ Server
      5. การพัฒนาและเชื่อมต่อ API และ WebSocket ที่อาจเกิดปัญหาด้านความเข้ากันได้
   2. ความเสี่ยงด้านเวลา
      1. การพัฒนาระบบที่ต้องใช้เวลาในการแก้ไขบั๊กและปรับปรุงฟังก์ชันการทำงาน
      2. การประสานงานระหว่างสมาชิกทีมที่อาจทำให้เกิดความล่าช้า
      3. การทดสอบและปรับปรุง UI/UX ที่อาจใช้เวลามากกว่าที่คาดการณ์
3. ต้นแบบและเอกสารอ้างอิง
   1. Spring Boot <https://docs.spring.io/spring-boot>
      1. Spring Boot Web
      2. Spring Boot WebFlux
      3. Spring Boot WebSocket
      4. Spring Boot Data JPA
   2. JavaFx <https://fxdocs.github.io/docs/html5>
   3. MySQL <https://w3schools.com/MySQL>
   4. MariaDB <https://mariadb.com/kb/en/about-mariadb-connector-j>
   5. Maven Repository <https://mvnrepository.com>
   6. Java 21 <https://docs.oracle.com/en/java/javase/21>