Java Project

แบบข้อแสนอโครงงาน

ชื่อโครงงาน Personal Expense Tracker

สมาชิก

1 นางสาว ปิยดารัตน์ สุพินิจ 6730301031

2 นางสาว ภูริชชยาพัชร์ สุพรรณ์ 6730301066

รายละเอียดโดยย่อ

โครงงานนี้เป็นโปรแกรม ติดตามรายรับรายจ่ายส่วนบุคคล ที่ช่วยให้ผู้ใช้สามารถบันทึก ติดตาม และวิเคราะห์พฤติกรรมการใช้จ่ายของตนเองได้อย่างมีประสิทธิภาพ ผู้ใช้สามารถเพิ่ม แก้ไข และลบรายการรายรับรายจ่าย ดูรายงานสรุป และกรองข้อมูลตามช่วงเวลาได้ โดยข้อมูลสามารถบันทึกลงในไฟล์ CSV หรือ JSON เพื่อให้สามารถดึงกลับมาใช้งานได้ในอนาคต

คุณลักษณะและขอบเขต

คุณลักษณะขั้นต่ำ

1. ระบบบันทึกรายรับ-รายจ่าย

- ผู้ใช้สามารถเพิ่ม แก้ไข และลบข้อมูลรายรับ-รายจ่ายได้
- มีหมวดหมู่ค่าใช้จ่าย เช่น อาหาร ค่าเดินทาง ค่าเช่าบ้าน ฯลฯ

2. แสดงผลข้อมูลการใช้จ่าย

- แสดงรายการรายรับ-รายจ่ายย้อนหลัง
- แสดงยอดรวมรายรับ รายจ่าย และคงเหลือ

3. การตั้งงบประมาณ

- ผู้ใช้สามารถกำหนดงบประมาณรายเดือน
- ระบบแจ้งเตือนเมื่อใช้จ่ายเกินงบประมาณ

4. ส่วนติดต่อผู้ใช้ (UI/UX) ที่ใช้งานง่าย

- รองรับการใช้งานบนมือถือหรือเว็บเบราว์เซอร์
- มีอินเทอร์เฟซที่เข้าใจง่าย ไม่ซับซ้อน

5. ฐานข้อมูลสำหรับจัดเก็บข้อมูล

- ใช้ฐานข้อมูลเช่น SQLite, Firebase, หรือ MySQL
- จัดเก็บข้อมูลรายการรายรับ-รายจ่ายของผู้ใช้

6. ระบบความปลอดภัยพื้นฐาน

- ผู้ใช้ต้องสามารถเข้าสู่ระบบ (Login/Signup) ได้
- มีการป้องกันข้อมูลส่วนตัวของผู้ใช้

คุณลักษณะเพิ่มเติม (หากมีเวลา)

1. ฟีเจอร์วิเคราะห์และรายงานการใช้จ่าย

- แสดงกราฟและสถิติการใช้จ่ายรายวัน รายเดือน รายปี
- วิเคราะห์แนวโน้มการใช้จ่ายของผู้ใช้
- แสดงรายงานการใช้จ่ายแยกตามหมวดหมู่

2. ระบบแจ้งเตือนอัจฉริยะ

- แจ้งเตือนเมื่อใช้จ่ายเกินงบประมาณที่ตั้งไว้
- ระบบแนะนำการปรับปรุงพฤติกรรมการใช้จ่าย

3. การซิงค์ข้อมูลและสำรองข้อมูล

- รองรับการซิงค์ข้อมูลระหว่างอุปกรณ์ (เช่น มือถือและเว็บ)
- มีระบบสำรองข้อมูลอัตโนมัติบน Cloud เช่น Google Drive หรือ Firebase

4. รองรับหลายสกุลเงิน

- สามารถเลือกสกุลเงินที่ต้องการใช้ได้
- มีระบบแปลงค่าเงินอัตโนมัติ

5. รองรับการจดจำใบเสร็จอัตโนมัติ (OCR)

- ผู้ใช้สามารถถ่ายรูปใบเสร็จและให้ระบบดึงข้อมูลค่าใช้จ่ายอัตโนมัติ
- ลดภาระการกรอกข้อมูลด้วยตนเอง

6. ระบบหมวดหมู่ที่ยืดหยุ่น

- ผู้ใช้สามารถสร้างและแก้ไขหมวดหมู่รายรับ-รายจ่ายเองได้
- รองรับการติดแท็กให้แต่ละรายการเพื่อการจัดหมวดหมู่ที่ดีขึ้น

7. ระบบแชร์งบประมาณร่วมกัน

- สามารถแชร์บัญชีรายรับ-รายจ่ายกับครอบครัวหรือเพื่อน
- มีฟีเจอร์แบ่งค่าใช้จ่าย (Split Bill) สำหรับการแชร์ค่าใช้จ่ายในกลุ่ม

8. ระบบเป้าหมายการออมเงิน

- ผู้ใช้สามารถตั้งเป้าหมายการออมเงิน เช่น เก็บเงินซื้อบ้าน รถยนต์ หรือท่องเที่ยว
- ระบบช่วยคำนวณและติดตามความคืบหน้าของเป้าหมายการออม

9. การเชื่อมต่อกับบัญชีธนาคารและ e-Wallets

- รองรับการเชื่อมต่อกับแอปพลิเคชันธนาคารหรือ e-Wallet เช่น TrueMoney, PayPal
- อัปเดตรายการใช้จ่ายอัตโนมัติจากบัญชีที่เชื่อมต่อ

แผนการทำเนินโครงงาน

ระยะ	ช่วงเวลา	ผลลัพธ์ที่คลาดหวัง
เริ่มต้นโครงงาน	สัปดาห์ที่ 1-2	-ส่งข้อเสนอโครงงาน -สร้าง GitHub repository -สร้างโครงงานด้วย Maven และเขียนโค้ดตั้งต้น
สร้างรูปแบบ	สัปดาห์ที่ 3-4	- สร้าง UI เบื้องต้นด้วย JavaFX - พัฒนา Core Feature (บันทึก, แสดง, แก้ไข, ลบรายการ)
ทดสอบการทำงาน	สัปดาห์ที่ 5-6	- เพิ่มระบบบันทึกข้อมูล CSV/JSON - ปรับปรุง UI และทดสอบการทำงาน
ส่งมอบงาน	สัปดาห์ที่ 7-8	- แก้ไขข้อผิดพลาดและเพิ่มฟีเจอร์เพิ่มเติม - จัดทำเอกสารประกอบและเตรียมนำเสนอ

การแบ่งงาน

นางสาว ปิยดารัตน์ สุพินิจ 6730301031

- จัดการ GitHub repository
- หาข้อมูลในการทำโครงงาน
- สร้าง Maven และเขียนโค้ดตั้งต้น

นางสาว ภูริชชยาพัชร์ สุพรรณ์ 6730301066

- ทำเอกสารใน Word
- หาข้อมูลในการทำโครงงาน
- ช่วยสร้าง Maven และเขียนโค้ดตั้งต้น

ความท้าทายและความเสี่ยงของโครงงาน “Personal Expense Tracker”

1. **ความท้าทายด้านเทคนิค**
   - ยังไม่คุ้นชินกับการใช้โปรแกรม Java
   - **แนวทางการแก้ไข :** ศึกษาเรียนรู้เพิ่มเติมและฝึกเขียนโค้ดมากขึ้น
2. **ความท้าทายด้านการออกแบบ**
   - ผู้จัดทำยังไม่มีประสบการณ์ในการทำโครงงานลักษณะนี้
   - **แนวทางการแก้ไข :** ค้นคว้าข้อมูลเพิ่มเติมและทำความเข้าใจเพื่อมาประยุกต์ใช้ในการทำโครงงาน
3. **ความเสี่ยงด้านเวลา**
   - ใช้เวลาในการศึกษาค้นความข้อมูลมากเกินไปอาจจะทำให้ไม่สามารถทำในส่วนอื่นได้ทันกำหนด
   - **แนวทางการแก้ไข :** วางแผนการทำงานให้รอบคอบแบ่งเวลาในการทำแต่ละพาร์ทให้ชัดเจน

เอกสารอ้างอิง

- **ศึกษาเรียนรู้วิธการใช้โปรแกรมเพิ่มเติม**

  (https://www.youtube.com/@borntodev **)**

- **ศึกษาการสร้าง GitHub repository**

  (ปฏิบัติการการใช้งาน-Git-และ-GitHub-เพื่อทำโครงงาน.pdf **)**

แนวทางการแบ่งงาน

1. **การแบ่งงานตามฟีเจอร์ (Feature-Based Division)**

   แนวทาง : แต่ละคนรับผิดชอบฟีเจอร์ของแอปพลิเคชันแยกจากกัน

   ข้อดี ง่ายต่อการบริหารงาน เพราะแต่ละคนรับผิดชอบเฉพาะส่วนของตัวเอง , สามารถทำงานพร้อมกันได้ ทำให้พัฒนาเสร็จเร็วขึ้น

2. **การแบ่งงานตามความเชี่ยวชาญของสมาชิก (Skill-Based Division)**

แนวทาง : ให้แต่ละคนทำงานตามความถนัดของตนเอง เช่น คนที่ถนัด UI จะพัฒนา Front-end ส่วนคนที่เก่งเรื่อง Database จะทำ Back-end

ตัวอย่างการแบ่งงาน :

ปิยดารัตน์ : พัฒนา Front-end

ภูริชชยาพัชร์ : ออกแบบ UI/UX

ข้อดี : ทำงานได้เร็ว เพราะแต่ละคนทำสิ่งที่ตนเองถนัด , คุณภาพของแต่ละส่วนมีโอกาสดีขึ้น เพราะทำโดยผู้เชี่ยวชาญ

คำแนะนำ

1.กำหนดขอบเขตให้เหมาะสม (Define Scope Clearly)

- โฟกัสที่ MVP (Minimal Viable Product) ก่อน ⟶ ทำ Core Features ให้เสร็จก่อน แล้วค่อยเพิ่มฟีเจอร์เพิ่มเติม
- แบ่งงานออกเป็นส่วนย่อย ๆ

2. ค่อย ๆ สร้างและเพิ่มทีละส่วน (Incremental Development)

- เริ่มจากฟีเจอร์ที่สำคัญและทำได้ง่ายก่อน
- พัฒนาให้ทำงานได้ในระดับพื้นฐานก่อน แล้วค่อยขยาย
- ทดลองใช้งานจริง เพื่อให้เห็นว่ามีจุดไหนต้องปรับปรุง

3. อย่ารอที่จะขอคำแนะนำ (Seek Help Early, Not at the Last Minute)

- หากพบปัญหา ให้รีบสอบถามเพื่อนร่วมทีม
- สื่อสารผ่านช่องทางที่ชัดเจน
- ใช้ Code Review ช่วยกันตรวจโค้ด

ประมาณการภาระงาน (ระยะเวลา 2 เดือน)

สัปดาห์ที่ 1-2: ตั้งต้นโครงการและทำความคุ้นเคย

กิจกรรม

- เรียนรู้พื้นฐาน Git/GitHub และตั้งค่า GitHub Repository
- ศึกษา JavaFX หรือ Spring Boot ผ่านแบบฝึกหัดง่าย ๆ
- ร่างและสรุป ข้อเสนอโครงงาน (Scope + MVP Features)
- ออกแบบ โครงสร้างโปรเจกต์ และ ER Diagram (ถ้ามี Database)

ผลลัพธ์ที่คาดหวัง

- มี โครงสร้างโปรเจกต์เริ่มต้น ใน GitHub
- มีความเข้าใจพื้นฐานเกี่ยวกับเครื่องมือและ Framework ที่ใช้
- มีเอกสารอธิบาย MVP และแผนการพัฒนา

สัปดาห์ที่ 3-4: เริ่มต้นการพัฒนา (ต้นแบบ)

กิจกรรม

พัฒนา ฟีเจอร์หลักของ MVP เช่น

- การบันทึกรายรับ-รายจ่าย
- ระบบหมวดหมู่ของค่าใช้จ่าย

จัดการปัญหาทางเทคนิค เช่น

- การเชื่อมต่อ Database (MySQL / SQLite / Firebase)
- การแสดงผลข้อมูลใน UI (ถ้ามี)

ทำ Commit โค้ดเป็นประจำ เพื่อแสดงความคืบหน้า

ผลลัพธ์ที่คาดหวัง

- แอปพลิเคชันสามารถทำงานได้บางส่วน (แต่ยังไม่สมบูรณ์)
- มีโครงสร้างพื้นฐานของโปรแกรมพร้อมใช้งาน

สัปดาห์ที่ 5-6: ปรับปรุงและทดสอบ

กิจกรรม

เพิ่ม ฟีเจอร์ที่ยังขาดไป เช่น

- ระบบสรุปรายจ่าย
- การค้นหาและกรองข้อมูล

ผลลัพธ์ที่คาดหวัง

- โปรแกรมสามารถใช้งานได้ และมีฟีเจอร์ครบถ้วนตาม MVP
- ปรับปรุง UX/UI ให้ใช้งานง่ายขึ้น

สัปดาห์ที่ 7-8: ปรับแต่งขั้นสุดท้ายและส่งโครงงาน

กิจกรรม

- ปรับแต่ง UI ให้สมบูรณ์ขึ้น (เพิ่มธีม, ปรับดีไซน์)
- เขียนเอกสารประกอบ

ผลลัพธ์ที่คาดหวัง

- โปรแกรมทำงานได้สมบูรณ์ (พร้อมส่ง)
- มีเอกสารครบถ้วนสำหรับผู้ใช้และการนำเสนอ