เอกสารข้อเสนอโครงการ
Project Proposal

ระบบบริหารจัดการ การยืม–คืนอุปกรณ์มหาวิทยาลัย

University equipment Lending Management system

(ULMs)

รายวิชาซอฟต์แวร์วิศวกรรม (Software Engineering)
รหัสวิชา 01204341 ภาคต้น ปีการศึกษา 2569

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์

ชื่อทีม: “miro ปั่น 14 บาท”

อาจารย์ที่ปรึกษาโครงการ
ปรีดา เลิศพงศ์วิภูษณะ (preeda.l@ku.th)
จิตร์ทัศน์ ฝักเจริญผล (jittat.f@ku.th)

รายชื่อสมาชิกในทีม

ลำดับ	รหัสนักศึกษา	ชื่อ – นามสกุล	บทบาทในทีม
1	6710503861	ซามีล ประยุรธเนศ	Project Manager
2	6710503739	กวินภพ จันเจือ	System Analyst
3	6710504077	ปัญญวัฒน์ เชื้อวัชรินทร์	Front-End – Head
4	6710503721	กฤษฎี ทองบ้านเกาะ	Front-End Developer
5	6710503798	จิรวัฒน์ สมรส	Front-End Developer
6	6610501971	กฤษกร สุทธิรักษ์	Front-End Developer
7	6710503780	จักรภัทร ชัยดิลกลาภ	UI/UX Designer
8	6710504310	รัชพล สนิทวงษ์	Back-End – Head
9	6710503887	ณภัทร โพธิบัวทอง	Back-End Developer
10	6710504069	ปวีณกฤษฏิ์ ท่าห้อง	Back-End Developer
11	6710504441	โสภณวิชญ์ นิลฟุ้งขจร	Back-End – API designer
12	6610502200	มูฮัมหมัดฮานีฟ เพ็งมูซอ	Tester – Head
13	6710504140	พฤฒิพงศ์ โขคธนอนันต์	Tester Security
14	6710504352	วีระศักดิ์ ทองปัสโน	Tester Quality

สารบัญ (Table of Contents)

1 ชื่อโครงการ (Project Name) 3

2 หลักการและเหตุผล (Principle & Rationale) 3
- 2.1 สภาพปัญหาในปัจจุบัน (Problem) 3
- 2.2 แนวทางการพัฒนา (Development Guidelines) 3
- 2.3 ประโยชน์ที่คาดว่าจะได้รับ (Expected Benefits) 3

3 วัตถุประสงค์ (Objectives) 4

4 เป้าหมาย (Goals) 4

5 ขอบเขตการดำเนินงาน (Scope: Features / Functions) 5
- 5.1 บทบาทผู้ใช้งาน (User Roles) 5
- 5.2 กระบวนการทำงานของระบบ (System Workflow) 5
- 5.3 ขอบเขตฟังก์ชันที่จัดทำ (Scope Functions) 5
- 5.4 สถาปัตยกรรมระบบ (System Architecture) 6
- 5.5 เทคโนโลยีที่ใช้ (Technology Stack) 7

6 แผนการดำเนินงาน (Project Plan) 7
- 6.1 แผนภูมิ Gantt แสดงระยะเวลา (Gantt Chart) 8

7 ทีมงานและหน้าที่ความรับผิดชอบ (Team & Responsibilities) 8

8 งบประมาณ (Project Budget) 9
- 8.1 ค่าใช้จ่าย (Cost) 9

1 ชื่อโครงการ (Project Name)

หัวข้อ	รายละเอียด
ชื่อโครงการ (ไทย)	ระบบบริหารจัดการการยืม–คืนอุปกรณ์มหาวิทยาลัย
ชื่อโครงการ (อังกฤษ)	University equipment Lending Management system
ชื่อย่อ	ULMs

2 หลักการและเหตุผล (Principle & Rationale)

รูปแบบการยืม-คืนของจากมหาวิทยาลัยในปัจจุบัน มักใช้รูปแบบการจัดเก็บข้อมูลด้วยการจดบันทึกเช่น การบันทึกลงกระดาษ หรือ spreadsheet ที่สะดวกสำหรับการยืมในปริมาณน้อย หรือใช้งานในขณะที่ยังมีอุปกรณ์ในการดูแลไม่มาก ซึ่งทำให้การดูแลและติดตามเป็นไปได้ยากขึ้นเมื่ออุปกรณ์หรือผู้ใช้งานมีจำนวนมากขึ้น

2.1 สภาพปัญหาในปัจจุบัน (Problem)

- บันทึกข้อมูลด้วยกระดาษ หรือ spreadsheet ทำให้ค้นหา ติดตาม และสรุปข้อมูลได้ยาก และมีโอกาสผิดพลาดสูง
- ไม่สามารถตรวจสอบสถานะอุปกรณ์แบบเรียลไทม์ได้ว่าอุปกรณ์ชิ้นใดว่าง ถูกยืม หรืออยู่ระหว่างซ่อม
- อุปกรณ์สูญหายหรือส่งคืนล่าช้า เนื่องจากไม่มีระบบแจ้งเตือนและติดตามผู้ยืมอย่างเป็นระบบ
- ฐานข้อมูลที่กระจัดกระจายทำให้การจัดซื้อ และจัดสรรอุปกรณ์ขาดประสิทธิภาพ

2.2 แนวทางการพัฒนา (Development Guidelines)

โครงการต้องการที่จะปรับปรุงรูปแบบการยืม-คืนอุปกรณ์แบบเดิมให้มีประสิทธิภาพมากขึ้น โดยการออกแบบกระบวนการยืม-คืนใหม่ให้ประสานเข้ากับระบบดิจิทอล และปรับปรุงประสิทธิภาพของการจัดการอุปกรณ์ของมหาวิทยาลัย

2.3 ประโยชน์ที่คาดว่าจะได้รับ (Expected Benefits)

- ลดภาระงานเอกสารและงานซ้ำซ้อนของเจ้าหน้าที่ที่ดูแลอุปกรณ์
- เพิ่มความสะดวกและรวดเร็วให้นักศึกษาที่ต้องการขอยืม หรือติดตามสถานะ
- เพิ่มความโปร่งใสและความสามารถในการตรวจสอบประวัติย้อนกลับ ของทุกรายการยืม–คืน
- ลดอัตราการสูญหายและคืนล่าช้าผ่านระบบแจ้งเตือนอัตโนมัติ
- ข้อมูลสถิติเชิงลึกเพื่อสนับสนุนการตัดสินใจในการจัดซื้อหรือดูแลอุปกรณ์

3 วัตถุประสงค์ (Objectives)

- เพื่อพัฒนาระบบที่ให้นักศึกษาสามารถค้นหาและขอยืมอุปกรณ์ผ่านช่องทางออนไลน์ได้
- เพื่อให้เจ้าหน้าที่สามารถบริหารจัดการคำขอยืม อนุมัติ/ปฏิเสธ และติดตามการคืนอุปกรณ์ได้อย่างเป็นระบบ
- เพื่อให้ผู้ใช้งานทราบสถานะและจำนวนคงเหลือของอุปกรณ์แต่ละชิ้นแบบเรียลไทม์
- เพื่อลดปัญหาการสูญหายและการคืนอุปกรณ์ล่าช้า ด้วยระบบแจ้งเตือนและการติดตาม
- เพื่อจัดเก็บประวัติและข้อมูลสถิติการใช้งานสำหรับสนับสนุนการตัดสินใจในการจัดการอุปกรณ์
- เพื่อออกแบบและพัฒนาระบบให้มีสถาปัตยกรรมครบทั้ง 4 ส่วน (Client, Application Server, API Server, Database Server) ตามข้อกำหนดของโครงการ

4 เป้าหมาย (Goals)

- ระบบรองรับผู้ใช้งานหลายบทบาท (นักศึกษา, เจ้าหน้าที่ผู้ดูแลอุปกรณ์, ผู้ดูแลระบบ) ให้สามารถทำงานร่วมกันได้
- ระบบจัดการบัญชีรายการอุปกรณ์ได้ครบถ้วน
- มีกระบวนการยืม–คืนครบวงจร ตั้งแต่ส่งคำขอ อนุมัติ รับของ คืนของ จนถึงตรวจสอบสภาพ
- มีระบบแจ้งเตือนและติดตามผ่าน Email รวมถึงการแจ้งเตือนภายในแอป
- มี Dashboard และรายงานสรุปสถิติการใช้งานอุปกรณ์
- ออกแบบสถาปัตยกรรม 4 ส่วน โดยมี API Server แยกการทำงานออกจาก Database Server อย่างชัดเจน

5 ขอบเขตการดำเนินงาน (Scope: Features / Functions)

5.1 บทบาทผู้ใช้งาน (User Roles)

บทบาท (Role)	คำอธิบายและสิทธิ์การใช้งานหลัก
นักศึกษา (Student)	ผู้ขอยืม: ค้นหา จองอุปกรณ์ ส่งคำขอยืม ดูสถานะคำขอ ดูประวัติ แจ้งคืน รายงานปัญหา ความเสียหาย
เจ้าหน้าที่ผู้ดูแล (Staff)	ผู้ดูแลกระบวนการ: จัดการบัญชีรายการอุปกรณ์ อนุมัติคำขอ บันทึกการรับ–คืน ตรวจสอบสภาพ จัดการรายการเกินกำหนด
ผู้ดูแลระบบ (Admin)	ผู้ดูแลระบบโดยรวม: จัดการบัญชีผู้ใช้และสิทธิ์ ตั้งค่าระบบ และตรวจสอบบันทึกการใช้งาน
เจ้าหน้าที่ระดับสูง (Supervisor)	ผู้อนุมัติระดับสูง: จัดการอนุมัติสำหรับอุปกรณ์มูลค่าสูงที่ต้องการการอนุมัติหลายขั้น

5.2 กระบวนการทำงานของระบบ (System Workflow)

1. นักศึกษาเข้าสู่ระบบ ค้นหาอุปกรณ์ และส่งคำขอยืม (ระบุรายการและช่วงเวลา)
2. ระบบแจ้งเตือนเจ้าหน้าที่ให้ตรวจสอบคำขอ
3. เจ้าหน้าที่ตรวจสอบและอนุมัติหรือปฏิเสธคำขอ (พร้อมระบุเหตุผลกรณีปฏิเสธ)
4. เมื่ออนุมัติ นักศึกษามารับอุปกรณ์ เจ้าหน้าที่กดยืนยัน เพื่อบันทึกการรับ
5. นักศึกษาใช้งานอุปกรณ์ตามปรกติ ระบบจะแจ้งเตือนเมื่อใกล้ถึงกำหนดคืน
6. นักศึกษาคืนอุปกรณ์ เจ้าหน้าที่ตรวจสอบสภาพและยืนยัน เพื่อบันทึกการคืน
7. หากพบความเสียหาย เจ้าหน้าที่บันทึกรายการความเสียหาย จากนั้นปิดรายการ

5.3 ขอบเขตฟังก์ชันที่จัดทำ (Scope Functions)

Release 1 (R01) — ฟังก์ชันหลัก

- ระบบยืนยันตัวตนและจัดการสิทธิ์ตามบทบาท (Authentication & Role-based Access)
- บัญชีรายการอุปกรณ์: ค้นหา กรอง ดูรายละเอียดและสถานะคงเหลือ
- การส่งคำขอยืมรายการเดี่ยวและหลายรายการ พร้อมระบุช่วงเวลา
- รอบกระบวนการการอนุมัติ/ปฏิเสธโดยเจ้าหน้าที่
- การบันทึกการรับ–คืน และรายงานความเสียหายโดยเจ้าหน้าที่ผู้ดูแลอุปกรณ์
- การจัดการคลังอุปกรณ์แบบหลายหน่วยต่อประเภท
- การติดตามสถานะคำขอ: รออนุมัติ อนุมัติ ปฏิเสธ กำลังยืม คืนแล้ว และเกินกำหนด

- การแจ้งเตือนภายในแอป และทาง Email เมื่อสถานะเปลี่ยนหรือใกล้ครบกำหนด
- ประวัติการยืมรายบุคคล และ Dashboard พื้นฐาน

Release 2 (R02) — ฟังก์ชันเสริม

- การจองล่วงหน้าแบบเลือกช่วงเวลา พร้อมปฏิทิน
- การจัดการรายการเกินกำหนด
- การติดตามการซ่อมบำรุง และทำให้สถานะอุปกรณ์ไม่พร้อมใช้งานชั่วคราว
- การออกรายงานและส่งออกข้อมูล

5.4 สถาปัตยกรรมระบบ (System Architecture)

ชั้น (Tier)	หน้าที่	เทคโนโลยี
Client (Presentation)	สื่อสารกับผู้ใช้งานผ่าน web application เพื่อรับ request หรือข้อมูลจากผู้ใช้งานและส่งต่อให้ Application Server และในทางกลับกัน	React + TypeScript
Application Server (Business Logic)	ควบคุมและประมวลผลตรรกะทางธุรกิจของระบบ เช่น การยืนยันตัวตน การจัดการสิทธิ์ผู้ใช้ การตรวจสอบเงื่อนไขการยืม–คืน การอนุมัติคำขอ การแจ้งเตือน และประสานงานกับฐานข้อมูล	NestJS
API Server (API Layer)	เป็นช่องทางการสื่อสารระหว่าง Client กับ Application Server จัดการการรับ–ส่งข้อมูล กำหนด API Endpoint/Procedure ตรวจสอบชนิดข้อมูล (Type-safe) และส่งต่อคำขอไปยัง Business Logic	tRPC
Database Server (Data Layer)	จัดเก็บและบริหารข้อมูลของระบบในแบบทั้ง SQL และ NoSQL เช่น ผู้ใช้ อุปกรณ์ คำขอยืม ประวัติการยืม การแจ้งเตือน และข้อมูลการซ่อมบำรุง พร้อมรองรับการค้นหาและการทำธุรกรรม	PostgreSQL, MongoDB

ลำดับการไหลของข้อมูล (Data Flow): Client → API Server → Application Server → Database Server

5.5 เทคโนโลยีที่ใช้ (Technology Stack)

ด้าน	เทคโนโลยี / เครื่องมือ
Front-End	**[Core]** Vite + React + TypeScript + React Router **[UI Layer]** Tailwind CSS + shadcn/ui **[Data/Form]** TanStack Query + React Hook Form + Zod **[Global]** React Context **[Scale]** Server-side Pagination/Search + Code-splitting **[Real-time]** TanStack Query Polling
Back-End / API	**[Core]** NestJS **[API]** tRPC
Database	PostgreSQL, MongoDB, Prisma
Version Control	Git + GitHub
Design	Figma
Collaboration / PM	Miro (Whiteboard)

6 แผนการดำเนินงาน (Project Plan)

ระยะ (Phase)	กิจกรรมหลัก	ผู้รับผิดชอบหลัก	ผลผลิต (Deliverable)
1. เริ่มต้น & วางแผน	แบ่งทีม ตั้งหัวข้อ จัดทำข้อเสนอโครงการ นำเสนอและขออนุมัติ	PM	Proposal, PPT นำเสนอ
2. วิเคราะห์ความต้องการ R01	เก็บและวิเคราะห์ความต้องการ กำหนด Use Case	System Analyst	เอกสารขอบเขตงาน R01
3. ออกแบบ & ต้นแบบ R01	ออกแบบ UI/UX, API และสถาปัตยกรรม ทำ Prototype	UI/UX, API Designer	Design Doc, Prototype
4. พัฒนา R01	พัฒนาฟังก์ชันหลัก ทดสอบ และ Demo	Front-End, Back-End, QA	ระบบ R01 (Demo)
5. ปรับขอบเขต & ออกแบบ R02	ปรับขอบเขตรอบ 3 วิเคราะห์/ออกแบบฟังก์ชันเสริม	SA, API Designer	ขอบเขต & Design R02
6. พัฒนา R02	พัฒนาฟังก์ชันเสริม ทดสอบ Integration และ Demo	ALL Dev, QA	ระบบ R02 (Demo)
7. สรุป & ส่งมอบ	ทดสอบรวม จัดทำเอกสารสรุป และนำเสนอ Final Release	ALL	Final Release

6.1 แผนภูมิ Gantt แสดงระยะเวลา (Gantt Chart)

แผนพัฒนา	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
เริ่มต้น & วางแผน																
วิเคราะห์ความต้องการ R01																
ออกแบบ & ต้นแบบ R01																
พัฒนา R01																
ปรับขอบเขต & ออกแบบ R02																
พัฒนา Second Release R02																
สรุป & ส่งมอบ																

7 ทีมงานและหน้าที่ความรับผิดชอบ (Team & Responsibilities)

กลุ่มงาน	**ตำแหน่ง**	**จำนวน**	**หน้าที่และผลผลิตหลัก**
1. Project Management & System Analyst	Project Manager (PM)	1	บริหารโครงการโดยรวม แบ่งงาน ติดตาม ควบคุมคุณภาพ
	System Analyst	1	ออกแบบระบบภาพรวม เชื่อมต่อเข้ากับการใช้งานจริง
2. Front-End Development	Front-End Leader	1	ดูแลและพัฒนาส่วนติดต่อผู้ใช้ ประสานกับ UI/UX
	UI/UX Designer	1	ออกแบบส่วนติดต่อผู้ใช้ให้ใช้งานง่าย
	Front-End Developer	3	พัฒนาหน้าจอและเชื่อมต่อ API ตามที่ออกแบบ
3. Back-End Development	Back-End Team Leader	1	ดูแลและพัฒนาส่วนหลังบ้านและ API Server
	API designer	1	ออกแบบโครงสร้าง API
	Back-End Developer	3	พัฒนา API จัดการข้อมูลและตรรกะ
4. Quality Assurance (3 คน)	QA Team Leader	1	วางแผนและออกแบบการทดสอบ กำหนดเกณฑ์ผ่าน
	Quality Control	1	ทดสอบระบบและตรวจสอบคุณภาพก่อนส่งมอบ
	Security Tester	1	ทดสอบความปลอดภัยระบบ

8 งบประมาณ (Project Budget)

8.1 ค่าใช้จ่าย (Cost)

รายการ	รายละเอียด	ประมาณการ (บาท)
Cloud / Database	PostgreSQL Database	0
Hosting	Docker	0
Design Tool	Figma (free tier)	0
Claude Pro Plan	5 ID × 3 เดือน	—
รวมโดยประมาณ		**—**