

SRS Checklist

แบบตรวจสอบเอกสาร Software Requirements Specification

สำหรับนักศึกษา — ใช้ตรวจสอบก่อนส่งงาน

ชื่อกลุ่ม:		วันที่ตรวจ:	___/___/___
ชื่อโปรเจกต์:		เวอร์ชัน:	___
ผู้ตรวจ:		บทบาท:	___

วิธีใช้: ทำเครื่องหมาย ✓ ในช่องที่ผ่านเกณฑ์ แล้วให้คะแนนรวมท้ายแต่ละหมวด

1

โครงสร้างเอกสาร (Document Structure)

อ้างอิง: IEEE 29148:2018

- ☐ มี Version History / Change Log อยู่ในส่วนต้นของเอกสาร
- ☐ มี Table of Contents ที่สอดคล้องกับเนื้อหาจริง
- ☐ มีส่วน Introduction ที่ระบุ Purpose และ Scope ชัดเจน
- ☐ มีส่วน Definitions / Glossary อธิบายคำศัพท์เฉพาะทาง
เช่น API, Supabase, RLS, JWT ฯลฯ ให้ผู้อ่านที่ไม่ใช่สาย IT เข้าใจ
- ☐ มีส่วน Overall Description อธิบายภาพรวมระบบ
- ☐ มี User Classes ระบุประเภทผู้ใช้งานและคุณลักษณะของแต่ละกลุ่ม
- ☐ มีส่วน Constraints, Assumptions, Dependencies
- ☐ รูปแบบ Markdown (.md) ถูกต้อง ไม่มี syntax error
Heading, table, list, code block แสดงผลได้ปกติ

คะแนน: ___ / 8

หมายเหตุ: _____

2

Functional Requirements (FR)

อ้างอิง: IEEE 29148 + Wiegiers' 3-Level Model

- ☐ แยกระดับ Requirements เป็น Business (BR), User (UR), Functional (FR) ชัดเจน
ตาม Wiegiers' Model: BR = ทำไม, UR = ผู้ใช้ทำอะไร, FR = ระบบทำอะไร
- ☐ ทุก Requirement มี ID ที่ไม่ซ้ำกัน (เช่น FR-001, FR-002)
- ☐ ทุก FR ระบุ Input, Processing, Output ครบถ้วน
- ☐ ทุก FR มี Acceptance Criteria แบบ Given-When-Then
Given [เงื่อนไขเริ่มต้น] When [การกระทำ] Then [ผลลัพธ์ที่คาดหวัง]
- ☐ ทุก FR ผ่านเกณฑ์ INVEST: Independent, Negotiable, Valuable, Estimable, Small, Testable
- ☐ ครอบคลุม Use Case / Flow หลักของระบบครบถ้วน

- ☐ ระบุ Error Cases และ Edge Cases สำหรับแต่ละ FR

เช่น กรอกข้อมูลผิด, session หมดอายุ, ข้อมูลซ้ำ

- ☐ มี Traceability ระหว่าง BR -> UR -> FR

สามารถอธิบายได้ว่าแต่ละ FR มาจาก UR ใด และ UR มาจาก BR ใด

คะแนน: ____ / 8

หมายเหตุ: _____

3

Non-Functional Requirements (NFR)

อ้างอิง: IEEE 830 Quality Attributes

- ☐ มี Performance requirement พร้อมตัวเลขที่วัดได้

เช่น Response time < 2s, Throughput > 100 req/s

- ☐ มี Availability requirement ระบุ uptime target

เช่น 99.9% uptime = downtime ไม่เกิน 8.76 ชม./ปี

- ☐ มี Security requirement ครอบคลุม Authentication & Authorization

- ☐ มี Scalability requirement ระบุจำนวนผู้ใช้งานพร้อมกัน

- ☐ มี Usability requirement

เช่น ผู้ใช้ใหม่ต้องใช้งานฟีเจอร์หลักได้ภายใน 5 นาทีโดยไม่ต้องอบรม

- ☐ มี Backup / Recovery policy

- ☐ ไม่ใช้คำคลุมเครือ เช่น 'เร็ว' 'ดี' 'เยอะ' — ทุกข้อวัดได้ด้วยตัวเลข

คะแนน: ____ / 7

หมายเหตุ: _____

4

Data Model / Database Schema

อ้างอิง: IEEE 29148 Data Requirements

- ☐ มี ER Diagram หรือ Entity-Relationship อธิบายความสัมพันธ์ของข้อมูล
- ☐ ระบุ Entity ทั้งหมดพร้อม Attributes และ Data Types
- ☐ ระบุ Primary Key, Foreign Key, Unique Constraints
- ☐ ระบุ CHECK Constraints สำหรับค่าที่เป็นไปได้ (เช่น status IN ('active','inactive'))
- ☐ มี SQL CREATE TABLE statements ที่ execute ได้จริง (สำหรับ AI-Ready SRS)
AI Agent ต้องการ schema ที่ชัดเจน ไม่ใช่แค่ ERD
- ☐ ระบุ Data Validation rules
เช่น email format, ความยาวสูงสุดของ field, ค่า default

คะแนน: ____ / 6

หมายเหตุ: _____

5

API & External Interface

อ้างอิง: OpenAPI Specification

- ☐ ทุก API Endpoint ระบุ HTTP Method, URL Path
เช่น GET /api/books, POST /api/loans
- ☐ ทุก Endpoint มีตัวอย่าง Request Body (JSON)
- ☐ ทุก Endpoint มีตัวอย่าง Response Body ทั้ง Success และ Error
- ☐ ระบุ Authentication method (เช่น Bearer Token, API Key)
- ☐ ระบุ Error Codes และความหมาย (400, 401, 403, 404, 409, 500)
- ☐ ระบุ Rate Limiting (ถ้ามี)

คะแนน: ____ / 6

หมายเหตุ: _____

6

UI/UX Requirements

อ้างอิง: Usability Standards

- ☐ มี Wireframe หรือ Mockup อย่างน้อยสำหรับหน้าหลัก
- ☐ มี Navigation Flow / Site Map
- ☐ ระบุ Component Hierarchy ของแต่ละหน้า (สำหรับ AI-Ready SRS)
เช่น LoginPage -> LoginForm -> EmailInput, PasswordInput, SubmitButton
- ☐ ระบุ Responsive Design breakpoints
เช่น Mobile 375px, Tablet 768px, Desktop 1280px
- ☐ ระบุ Accessibility requirements (WCAG level)

คะแนน: ____ / 5

หมายเหตุ: _____

7

Tech Stack & Architecture

สำคัญมากสำหรับ AI Agent

☐ ระบุ Programming Language / Framework พร้อมเวอร์ชัน

เช่น *Next.js 15.1 (App Router)*, *NOT* แต่ *'Next.js'*

☐ ระบุ Database system พร้อมเวอร์ชัน

☐ ระบุ UI Library / CSS Framework

เช่น *shadcn/ui* + *Tailwind CSS 4.0*

☐ ระบุ Authentication method

☐ ระบุ ORM / Database Client

☐ ระบุ Package Manager

☐ มี Constraints / DO NOT rules

เช่น *'DO NOT use Pages Router'*, *'DO NOT use Drizzle'*

☐ มี Project File Structure (folder tree)

ระบุว่าไฟล์แต่ละประเภทอยู่ folder ไหน

☐ มี Environment Variables list พร้อมคำอธิบาย

คะแนน: ____ / 9

หมายเหตุ: _____

8

AI-Ready SRS (สำหรับส่ง AI Agent พัฒนา)

Good Practice: SRS-to-AI Workflow

- ☐ มี Role-Permission Matrix (ตารางสิทธิ์การเข้าถึงของแต่ละ Role)
ครอบคลุมทุก Action x ทุก Role อย่างชัดเจน
- ☐ มี Implementation Order / Build Sequence เรียงลำดับการพัฒนา
เช่น Phase 1: Schema -> Phase 2: Auth -> Phase 3: CRUD -> Phase 4: Business Logic
- ☐ ทุก API มีตัวอย่าง JSON จริง (ไม่ใช่แค่คำอธิบาย)
AI เรียนรู้จาก Example ได้ดีกว่าจาก Description
- ☐ Database Schema เป็น SQL ที่ execute ได้ทันที
ไม่ใช่แค่ ERD หรือตารางอธิบาย field
- ☐ ทุก Feature เขียนแบบ Self-contained (อ่านจบแล้วพัฒนาได้เลย)
ไม่ต้องอ้างอิงข้อมูลจากส่วนอื่นของเอกสาร
- ☐ มี Constraints Section ที่ระบุสิ่งที่ AI ห้ามทำ
เช่น 'DO NOT add social login', 'ALWAYS use server-side validation'
- ☐ เอกสารแบ่งเป็น Chunk ได้ (ส่งทีละส่วนให้ AI ได้ตามลำดับ)
- ☐ ไม่ใช่คำคลุมเครือ — 'ควรจะ' 'น่าจะ' 'อาจจะ' เปลี่ยนเป็น 'ต้อง' ทั้งหมด

คะแนน: ____ / 8

หมายเหตุ: _____

9

คุณภาพการเขียน (Writing Quality)

อ้างอิง: IEEE 830 Quality Attributes

- ☐ ทุก Requirement ไม่คลุมเครือ (Unambiguous) — อ่านแล้วตีความได้ทางเดียว
- ☐ ทุก Requirement ทดสอบได้ (Verifiable) — เขียน Test Case ได้ทันที
- ☐ ไม่มี Requirement ที่ขัดแย้งกัน (Consistent)
- ☐ ไม่มีการ Over-specification — เขียน 'อะไร' ไม่ใช่ 'อย่างไร'
เช่น ไม่ระบุ 'ใช้ for loop' แต่ระบุ 'คำนวณผลรวมทั้งหมด'
- ☐ ใช้ภาษาที่สม่ำเสมอตลอดทั้งเอกสาร
เช่น ใช้คำว่า 'ผู้ใช้' หรือ 'user' อย่างใดอย่างหนึ่ง ไม่สลับไปมา
- ☐ สะกดถูกต้อง ไม่มี Typo ในชื่อตาราง, field, endpoint
สำคัญมากสำหรับ AI — typo ทำให้ generate code ผิด
- ☐ มีการจัดลำดับความสำคัญของ Requirements (Priority: Must/Should/Could)

คะแนน: ____ / 7

หมายเหตุ: _____

สรุปผลการตรวจสอบ

หมวด	คะแนนเต็ม	คะแนนที่ได้	%	หมายเหตุ
1. โครงสร้างเอกสาร	8			
2. Functional Requirements	8			
3. Non-Functional Requirements	7			
4. Data Model / Schema	6			
5. API & Interface	6			
6. UI/UX Requirements	5			
7. Tech Stack & Architecture	9			
8. AI-Ready SRS	8			
9. คุณภาพการเขียน	7			
รวมทั้งหมด	64			

เกณฑ์การประเมิน

ระดับ	คะแนน	ความหมาย	การดำเนินการ
ดีเยี่ยม	90-100%	SRS พร้อมส่ง AI Agent พัฒนาได้ทันที	ส่งงานได้เลย
ดี	70-89%	SRS ครบถ้วนเป็นส่วนใหญ่ ขาดรายละเอียดบางส่วน	แก้ไขเล็กน้อยแล้วส่ง
พอใช้	50-69%	มีโครงสร้างพื้นฐาน แต่ขาดรายละเอียดสำคัญหลายจุด	ต้องปรับปรุงก่อนส่ง
ต้องปรับปรุง	< 50%	SRS ยังไม่สมบูรณ์ ส่ง AI Agent แล้วจะได้ code ที่ใช้งานไม่ได้	เขียนใหม่ / ปรับปรุงมาก

ความคิดเห็นเพิ่มเติม / จุดที่ต้องแก้ไขเร่งด่วน

ลงชื่อผู้ตรวจ: _____

ลงชื่อหัวหน้ากลุ่ม: _____

วันที่: ____/____/____

วันที่: ____/____/____

Checklist อ้างอิงจาก IEEE 29148:2018, Wiegers' Model, INVEST Criteria, และ AI-Ready SRS Best Practices