



## Software Development Project Report

HeaUai

Date: October 12th, 2024

By: Group 4

1. 2213110451 นายณตฤณ ทองวิชิต (COORDINATOR)
2. 2213110022 นายภูติศ โขติช่วง
3. 2213110204 นายชนนัช เพ็ชรโยธิน
4. 2213110436 นายชิตวร ศิริเอกสถิต
5. 2213110535 นายนิรภัฏ คงห้วยรอบ
6. 2213110741 นายธฤต รวงผึ้งหลวง

Developed for Group 6

Thai-Nichi Institute of Technology

INT- 303 Software Engineering

## Table of Contents

Document Overview	1
Objective	1
Software Development Activities	2
Software Development Process	2
Overview of Process Phases	2
Deliverables	2
Software Development Tools	2
Workstation	2
Coding and Testing	2
Version Control Systems	3
Database Management Systems	3
Languages	3
Software Design (UX/UI)	3
Requirement Management Tools	3
Collaboration Tools	4
Points of Contact	4
Schedule	5
Risk Assessment & Management	6
Risk Log	6
Risk Event	6
Risk Control	7
Risk Management Preparation	8
Risk Response Control	8
Requirement Specification Document	9
Overall Description	9

Product Perspective	9
Product Functions	9
Specific Requirements	9
Functional Requirements	9
Non-Functional Requirements	10
User Interface Requirements	11
MBTI Quiz Interface	11
Book Recommendation	11
Book Detail Page	11
Favorites and History Management	11
Personality Type Information	11
Design Specification Document	12
Architecture Design	12
Major Architectural Components	12
Data Flow	13
Module Design	13
MBTI Quiz Module	13
Book Recommendation Module	13
Book Detail Page Module	14
Personality Information Module	14
Favorites and History Management Module	15
User Interface (UI/UX) Design	16
MBTI Quiz Interface	16
Book Recommendation	16
Book Detail Page	16
Personality Information Interface	16

Favorite and History Management Interface	16
Database Design	17
Implementation Note	19
Frontend Implementation	19
Backend Implementation	20
Testing and Debugging	20
Improvement and Correction	21
Test Plan & Test Cases	22
Test Plan	22
Introduction	22
Approach	22
Testing Methods	22
Item Pass/Fail Criteria	22
Responsibilities	22
Schedule	23
Risk and Contingencies	23
Approvals	23
Test Cases	24
Use Case Description	24
Test Case Summary	29
Test Cases for UC-0001: MBTI Quiz	29
Test Cases for UC-0002: Book Recommendation	30
Test Cases for UC-0003: Personality Information	31
Test Cases for UC-0004: Favorites and History Management	32
Test Cases for UC-0005: Book Detail Page	33
Summary	34

Appendices	35
References	35
Template	35
Further Readings	35
Additional Document	35

## Document Overview

เอกสารนี้เป็นเอกสารการวางแผนการออกแบบและพัฒนาซอฟต์แวร์ HeaUai ซึ่งจัดทำโดยกลุ่ม 4 โดยเนื้อหาของเอกสารครอบคลุมวิธีการพัฒนาเครื่องมือที่ใช้ และรายละเอียดในแต่ละขั้นตอน รวมถึงการแบ่งหน้าที่สมาชิกในทีมพัฒนา และ Timeline สำหรับตรวจสอบและดำเนินการตามแผนงาน

เอกสารนี้จัดทำขึ้นสำหรับกลุ่ม 6 เป็นหลัก ซึ่งเป็นผู้มีส่วนได้ส่วนเสีย รวมถึงอาจารย์ และทีมผู้พัฒนาระบบ เพื่อช่วยให้การดำเนินงานเป็นไปอย่างเข้าใจตรงกัน และเป็นระบบ โดยมีเป้าหมายให้แอปพลิเคชัน HeaUai สามารถแนะนำหนังสือที่ตรงกับบุคลิกภาพ MBTI ของผู้ใช้ได้อย่างมีประสิทธิภาพ โดยแนะนำหนังสือตามประเภทบุคลิกภาพ MBTI ผ่านการทำแบบทดสอบในแอปพลิเคชันมือถือ นอกจากนี้ยังมุ่งหวังให้ผู้ใช้ได้รับประสบการณ์การอ่านที่ดียิ่งขึ้นจากการค้นหาหนังสือที่เหมาะสมกับตนเอง

## Objective

- แอปพลิเคชันสำหรับแนะนำหนังสือตามบุคลิกภาพ (MBTI)
- สามารถเลือกชมหนังสือ และเพิ่มไปยังลิสต์หนังสือที่ชอบได้
- แอปพลิเคชันสามารถบันทึกประวัติการเข้าชมหนังสือได้
- ลิสต์หนังสือที่ชอบได้สามารถเพิ่มหรือลบหนังสือได้
- สามารถทำแบบทดสอบบุคลิกภาพ MBTI เพื่อระบุประเภทบุคลิกภาพของตนได้
- สามารถดูข้อมูลรายละเอียดเกี่ยวกับประเภทบุคลิกภาพต่าง ๆ

## Software Development Activities

### Software Development Process

ทีมพัฒนาเลือกใช้ Development approach แบบ Plan-driven เพราะโครงการพัฒนาซอฟต์แวร์ HeaUai มีข้อกำหนดและความต้องการที่ชัดเจนจากผู้ใช้งานตั้งแต่เริ่มต้น ทำให้การวางแผนสามารถทำได้อย่างละเอียดและเป็นระบบ แนวทางนี้ช่วยให้ทีมสามารถแบ่งงานออกเป็นขั้นตอนชัดเจนและดำเนินการตามแผนได้อย่างมีประสิทธิภาพ และเหมาะกับโครงการที่ข้อกำหนดไม่เปลี่ยนแปลงบ่อย ซึ่งทำให้การพัฒนามีเสถียรภาพและสามารถติดตามความคืบหน้าได้ง่าย

### Overview of Process Phases

ทีมพัฒนาได้มีการแบ่ง Phase เป็น 4 ขั้นหลัก ๆ ดังต่อไปนี้

- Software Development Project Plan
- Software Specification
- Software Design
- Software Implementation and Testing

### Deliverables

ในแต่ละขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์ (Software Development Process) เมื่อเสร็จสิ้นแล้วในแต่ละขั้นตอนจะมีการทำเอกสารที่เกี่ยวข้องเพื่อเป็นเอกสารประกอบความเข้าใจในการทำงานเพื่อเป็นการบันทึกข้อมูลและใช้อ้างอิงในส่วนต่าง ๆ

## Software Development Tools

### Workstation

- Discord
- ที่อยู่อาศัยของแต่ละสมาชิกทีมผู้พัฒนา
- สถาบันเทคโนโลยีไทย-ญี่ปุ่น

### Coding and Testing

- Visual Studio Code: ใช้สำหรับการพัฒนาแอปพลิเคชันและการดีบั๊ก
- Android Studio: ใช้สำหรับพัฒนาและจำลองแอปพลิเคชันบนอุปกรณ์ Android
- Postman: เครื่องมือสำหรับการทดสอบ API และการจัดการคำขอ HTTP

## Version Control Systems

- Git: ระบบควบคุมเวอร์ชันที่ติดตามการเปลี่ยนแปลงของโค้ดและช่วยให้สามารถย้อนกลับไปยัง states ก่อนหน้าได้
- GitHub: แพลตฟอร์มออนไลน์ที่โฮสต์รีโพสิทอรี Git และช่วยให้ทีมสามารถทำงานร่วมกันได้โดยการแชร์โค้ด การทำรีวิว และการจัดการปัญหา

## Database Management Systems

- MongoDB: ฐานข้อมูล NoSQL สำหรับการจัดการข้อมูลที่มีโครงสร้างที่ยืดหยุ่น

## Languages

- JavaScript: ภาษาโปรแกรมที่ใช้สำหรับการเขียนสคริปต์
- TypeScript: JavaScript ที่ถูกพัฒนาแล้วที่เพิ่มการตรวจสอบชนิดข้อมูลเพื่อช่วยในการตรวจหาข้อผิดพลาดและเพิ่มให้มีประสิทธิภาพมากขึ้น
- React Native: เฟรมเวิร์กที่ใช้สำหรับการพัฒนาแอปมือถือข้ามแพลตฟอร์ม โดยสามารถสร้างแอปสำหรับ iOS และ Android ด้วยโค้ดเดียว
- Node.js: รันไทม์ที่ใช้สำหรับการรัน JavaScript บนเซิร์ฟเวอร์ ซึ่งช่วยในการสร้างแอปพลิเคชันและ API
- Express.js: เฟรมเวิร์กสำหรับ Node.js ที่ช่วยในการสร้างแอปพลิเคชันและ API ฟังก์ชันเซิร์ฟเวอร์
- Expo Go: เครื่องมือสำหรับการพัฒนาและทดสอบแอป React Native บนอุปกรณ์มือถือ โดยสามารถดูตัวอย่างและทดสอบแอปพลิเคชันได้

## Software Design (UX/UI)

- Figma: ใช้สำหรับการออกแบบ User interfaces และสร้าง UX/UI prototypes

## Requirement Management Tools

- ClickUp: ใช้สำหรับการจัดการโครงการและติดตามความก้าวหน้าของงาน รวมถึงการจัดการงานที่ต้องทำ (tasks) การตั้งเป้าหมาย (goals) การจัดการเวลา (time management) และการติดตามความก้าวหน้า (progress tracking)
- Google Docs: ใช้สำหรับการสร้างและแก้ไขเอกสารร่วมกันแบบเรียลไทม์ มีฟีเจอร์สำหรับการทำงานร่วมกัน การตรวจสอบคำสะกด และการแชร์เอกสารผ่านลิงก์



### Collaboration Tools

- Discord: ใช้สำหรับการสื่อสารและการทำงานร่วมกันในทีม รวมถึงการประชุมออนไลน์ การสนทนาแบบเรียลไทม์ และการแลกเปลี่ยนความคิดเห็นเกี่ยวกับการพัฒนาและข้อกำหนดของงาน

### Points of Contact

Contact	Role	Responsibility
ณตฤณ ทองวิชิต	Project Manager & Developer	ดูแลโครงการในภาพรวม และรับผิดชอบการพัฒนาแอปพลิเคชัน
ภูติศ โขติช่วง	Developer & UX/UI Designer	ออกแบบส่วนต่อประสานผู้ใช้ (UI) และประสบการณ์ผู้ใช้ (UX) และพัฒนาแอปพลิเคชัน
ชนนัช เพ็ชรโยธิน	Developer & UX/UI Designer	ออกแบบส่วนต่อประสานผู้ใช้ (UI) และประสบการณ์ผู้ใช้ (UX) และพัฒนาแอปพลิเคชัน
ชิตวร ศิริเอกสถิต	Lead Developer & Document	ดูแลการออกแบบเชิงเทคนิค การนำระบบไปใช้ในเชิงเทคนิค และจัดทำเอกสาร
นิรภัฏ คงห้วยรอบ	Developer & Document	รับผิดชอบการพัฒนาแอปพลิเคชัน และจัดทำเอกสาร
ธฤต รวงผึ้งหลวง	Developer & QA Engineer	รับผิดชอบการพัฒนาแอปพลิเคชันและทดสอบซอฟต์แวร์

Schedule

PROJECT NAME	HeaUai			PROJECT MANAGER	นายณตฤณ ทองวีรจิต
START DATE	01/08/2024	END DATE	12/10/2024	OVERALL PROGRESS	100%

TASK NAME	ASSIGNED TO	START DATE	END DATE	% COMPLETE	STATUS	NOTES
Planning		01/08/2024	06/08/2024	100%		
Get a topic	จิตพร,นิรภัฏ	01/08/2024	02/08/2024	100%	Complete	
Gather stakeholders	จิตพร,นิรภัฏ	03/08/2024	06/08/2024	100%	Complete	
Requirements Analysis		08/08/2024	12/08/2024	100%		
Define business objectives and goals	จิตพร,นิรภัฏ	08/08/2024	10/08/2024	100%	Complete	
Formalize project requirements	จิตพร,นิรภัฏ	10/08/2024	12/08/2024	100%	Complete	
Define project	จิตพร,นิรภัฏ	10/08/2024	12/08/2024	100%	Complete	
Design		12/09/2024	19/09/2024	100%		
Research	ภูติศ,ชนนัช,ณตฤณ	12/09/2024	17/09/2024	100%	Complete	
Brainstorming	ภูติศ,ชนนัช,ณตฤณ	12/09/2024	17/09/2024	100%	Complete	
Prototyping	ภูติศ,ชนนัช,ณตฤณ	13/09/2024	19/09/2024	100%	Complete	
Handoff	ภูติศ,ชนนัช,ณตฤณ	13/09/2024	19/09/2024	100%	Complete	
Development		20/09/2024	10/10/2024	100%		
Execution	Member All	20/09/2024	10/10/2024	100%	Complete	
Creation	Member All	21/09/2024	10/10/2024	100%	Complete	
Testing	Member All	22/09/2024	10/10/2024	100%	Complete	
Demo	Member All	23/09/2024	10/10/2024	100%	Complete	
Retrospective	Member All	24/09/2024	10/10/2024	100%	Complete	
Implementation	Member All	25/09/2024	10/10/2024	100%	Complete	
Quality Assurance		25/09/2024	10/10/2024	100%		
Reviewing requirements	ชนนัช,ธฤต	25/09/2024	25/09/2024	100%	Complete	
Assuring quality	ชนนัช,ธฤต	25/09/2024	27/09/2024	100%	Complete	
Test case creation	ชนนัช,ธฤต	25/09/2024	29/09/2024	100%	Complete	
Testing	ชนนัช,ธฤต	08/10/2024	10/10/2024	100%	Complete	
Delivery		11/10/2024	12/10/2024	100%		
Showcase	Member All	11/10/2024	12/10/2024	100%	Complete	
Launch	Member All	11/10/2024	12/10/2024	100%	Complete	

## Risk Assessment & Management

การประเมินและจัดการความเสี่ยงในโครงการพัฒนาแอปพลิเคชัน HeaUai เป็นสิ่งสำคัญเพื่อให้สามารถรับมือกับปัญหาที่อาจเกิดขึ้นและจัดการให้โครงการดำเนินไปอย่างราบรื่น โดยการประเมินและควบคุมความเสี่ยง จะมีการจัดการดังนี้

### Risk Log

ID	Risk	Impact	Risk Response	Risk Level	Risk Owner
1	สมาชิกไม่เข้าใจในภาษาที่เขียนดีพอ	Whole app's functionality	ศึกษาหาข้อมูลเพิ่มเติม และให้คนที่ถนัดมาพัฒนาเป็นหลัก	High	All member
2	ความล่าช้าในการอัปเดตความคืบหน้าของโครงการ	Whole app's functionality	สอบถามและอัปเดตสถานะของงานเป็นระยะๆ เพื่อให้แน่ใจว่างานจะดำเนินไปตามแผนที่กำหนด	Medium	ณตฤณ
3	ความยากของบางระบบในโปรแกรม	Show recent book functionality	มุ่งเน้นการพัฒนาฟังก์ชันหลักและส่วนที่สำคัญที่สุดของแอปพลิเคชันให้เสร็จสมบูรณ์ก่อน	Medium	ณตฤณ นิรภัฏ

### Risk Event (เหตุการณ์ความเสี่ยง)

#### 1. ความเสี่ยงจากการขาดทักษะในการพัฒนาแอปพลิเคชันของบางคนในกลุ่ม

- บางคนในทีมไม่มีความถนัดในด้านการเขียนโค้ดหรือการพัฒนาแอปพลิเคชัน ทำให้การแบ่งงานและการดำเนินงานในส่วนต่าง ๆ เป็นเรื่องที่ยากลำบากและอาจเกิดความไม่สมดุลระหว่างการทำงานของสมาชิกในทีม
- ผลกระทบ: อาจทำให้โครงการมีความล่าช้าและไม่สามารถทำงานให้เสร็จตามเวลาที่กำหนด

## 2. ความล่าช้าในการอัปเดตความคืบหน้าของโครงการ

- การอัปเดตสถานะของโครงการหรือการสื่อสารระหว่างสมาชิกอาจล่าช้า ทำให้การตรวจสอบและการแก้ไขปัญหาไม่ทันเวลา
- ผลกระทบ: ทำให้โครงการไม่สามารถพัฒนาได้อย่างต่อเนื่อง และอาจทำให้ขั้นตอนการพัฒนาเกิดความล่าช้า

## 3. ไม่สามารถพัฒนาฟังก์ชันบางส่วนของแอปพลิเคชันให้สมบูรณ์ได้

- เนื่องจากความซับซ้อนของบางฟังก์ชัน หรือข้อจำกัดทางเทคนิค ทำให้บางส่วนของแอปพลิเคชันไม่สามารถทำงานได้เต็มประสิทธิภาพตามที่คาดหวัง
- ผลกระทบ: ผู้ใช้จะไม่ได้รับประสบการณ์การใช้งานที่ดี อาจทำให้เกิดข้อผิดพลาดในระบบหรือการใช้งานไม่สมบูรณ์

## Risk Control (การควบคุมความเสี่ยง)

### 1. การจัดการปัญหาด้านทักษะในการพัฒนา

- ทำการแบ่งงานโดยให้คนที่มีความถนัดในด้านการพัฒนาแอปพลิเคชันทำหน้าที่หลักในด้านนี้ ส่วนคนที่ไม่มีความถนัดให้ทำหน้าที่ในด้าน UX/UI Design หรือ อื่น ๆ ในการพัฒนาแอปพลิเคชัน วิธีนี้จะช่วยให้การทำงานมีประสิทธิภาพมากขึ้นและลดความเสี่ยงจากการพัฒนาที่ไม่สมบูรณ์

### 2. การติดตามความคืบหน้า

- มีการติดตามความคืบหน้าของแต่ละคนในทีมอย่างใกล้ชิด โดยการสอบถามและอัปเดตสถานะของงานเป็นระยะๆ เพื่อให้แน่ใจว่างานจะดำเนินไปตามแผนที่กำหนดไว้ วิธีนี้จะช่วยให้สามารถแก้ไขปัญหาที่เกิดขึ้นได้อย่างรวดเร็ว

### 3. การพัฒนาฟังก์ชันสำคัญก่อน

- มุ่งเน้นการพัฒนาฟังก์ชันหลักและส่วนที่สำคัญที่สุดของแอปพลิเคชันให้เสร็จสมบูรณ์ก่อน ส่วนฟังก์ชันที่ยากหรือมีความซับซ้อนสามารถพัฒนาเพิ่มเติมในภายหลัง วิธีนี้จะช่วยให้แอปพลิเคชันสามารถใช้งานได้จริงในระยะเวลาอันสั้น และสามารถปรับปรุงเพิ่มเติมภายหลังได้

## Risk Management Preparation (การเตรียมการจัดการความเสี่ยง)

### 1. การแจ้งเตือนและติดตามความเสี่ยง

- มีการแจ้งเตือนทุกคนในทีมให้รับทราบถึงความเสี่ยงที่อาจเกิดขึ้น และกระตุ้นให้ทุกคนติดตามสถานะและความเสี่ยงของงานอย่างสม่ำเสมอ
- ใช้ช่องทางการสื่อสารหลายช่องทาง เช่น Messenger LINE Discord เพื่อให้ทีมสามารถติดต่อและแจ้งเตือนกันได้อย่างรวดเร็ว

## Risk Response Control (การควบคุมการตอบสนองต่อความเสี่ยง)

### 1. การตอบสนองต่อความเสี่ยง

- ให้ทุกคนในทีมรับทราบและติดตามความเสี่ยงที่อาจเกิดขึ้นได้ตลอดระยะเวลาของการพัฒนาแอปพลิเคชัน เพื่อให้ทุกคนตระหนักถึงปัญหาและสามารถปรับตัวเพื่อรับมือกับความเสี่ยงได้อย่างเหมาะสม
- ควรมีการประชุมทีมเป็นระยะ ๆ เพื่อติดตามปัญหาและพัฒนาแนวทางในการจัดการกับความเสี่ยงใหม่ๆ ที่อาจเกิดขึ้น

## Requirement Specification Document

### 1. Overall Description

#### 1.1. Product Perspective

แอปพลิเคชันจะเป็นแอปพลิเคชันที่ใช้งานบนอุปกรณ์มือถือ โดยใช้ React Native สำหรับการพัฒนาและ MongoDB สำหรับการจัดเก็บข้อมูล

#### 1.2. Product Functions

1. MBTI Quiz: ผู้ใช้สามารถทำแบบทดสอบเพื่อระบุประเภทบุคลิกภาพของตัวเอง
2. Book Recommendations: ระบบแสดงหนังสือที่เหมาะสมกับบุคลิกภาพที่ได้จากการทดสอบ
3. Personality Information: ผู้ใช้สามารถดูข้อมูลบุคลิกภาพประเภทต่าง ๆ
4. Book Details:
  - บันทึกหนังสือเป็นรายการโปรด
  - คะแนนรีวิวจากระบบ
  - เก็บประวัติการเข้าชมหนังสือ
  - อ่านรายละเอียดของหนังสือ

### 2. Specific Requirements

#### 2.1. Functional Requirements

##### 2.1.1. MBTI Quiz

- ผู้ใช้ต้องสามารถทำแบบทดสอบบุคลิกภาพ MBTI ภายในแอปพลิเคชันได้
- แบบทดสอบจะประกอบด้วยชุดคำถามที่ออกแบบมาเพื่อระบุประเภทบุคลิกภาพของผู้ใช้
- ระบบต้องคำนวณและแสดงประเภทบุคลิกภาพ MBTI ของผู้ใช้ตามคำตอบที่ให้

##### 2.1.2. Recommendations Book

- ระบบต้องแสดงรายการหนังสือที่แนะนำตามประเภทบุคลิกภาพ MBTI ของผู้ใช้
- การแนะนำควรได้รับการอัปเดตเป็นระยะ ๆ หรืออิงจากความคิดเห็นของผู้ใช้

### 2.1.3. Personality Type Information

- ผู้ใช้ต้องสามารถดูรายละเอียดเกี่ยวกับประเภทบุคลิกภาพ MBTI ต่าง ๆ ได้

### 2.1.4. Book Details

- ผู้ใช้ต้องสามารถดูข้อมูลรายละเอียดเกี่ยวกับหนังสือแต่ละเล่มได้ รวมถึงชื่อเรื่อง ผู้เขียน ประเภท สรุปเนื้อหา และภาพปก
- หนังสือแต่ละเล่มต้องมีระบบการให้คะแนนสำหรับการรีวิวของผู้ใช้

### 2.1.5. Favorites and History

- ผู้ใช้ต้องสามารถบันทึกหนังสือลงในรายการโปรด เพื่อการเข้าถึงในภายหลังได้
- ระบบต้องเก็บประวัติหนังสือที่ผู้ใช้ได้เข้าชม
- ผู้ใช้ต้องสามารถจัดการรายการโปรดและประวัติการเข้าชม (เช่น ลบออกจากรายการโปรด ล้างประวัติ)

### 2.1.6. Ratings

- หนังสือแต่ละเล่มต้องมีระบบการให้คะแนนที่อนุญาตให้ผู้ใช้ให้คะแนนได้
- ผู้ใช้ต้องสามารถดูคะแนนเฉลี่ยแต่ละรายการได้

## 2.2. Non-Functional Requirements

### 2.2.1. Performance

- แอปพลิเคชันควรตอบสนองต่อการกระทำของผู้ใช้ภายใน 6 วินาที

### 2.2.2. Usability

- แอปพลิเคชันควรมีการออกแบบที่เป็นมิตรต่อผู้ใช้และอ่านง่าย
- การออกแบบต้องใช้โทนสีพาสเทลและฟอนต์ที่อ่านง่าย

### 2.2.3. Security

- ข้อมูลบุคลิกภาพและหนังสือจะต้องถูกจัดเก็บอย่างปลอดภัยในฐานข้อมูล

### 3. User Interface Requirements

#### 3.1. MBTI Quiz Interface

อินเทอร์เน็ตทำแบบทดสอบที่ชัดเจน ใช้ธีมสีพาสเทลโทนเย็น มีแถบความคืบหน้าและปุ่มนำทางที่ชัดเจน ช่วยให้ผู้ใช้ทำแบบทดสอบได้ง่ายและรู้สึกผ่อนคลาย

#### 3.2. Book Recommendation

แสดงหนังสือที่แนะนำ มีปุ่มดูรายละเอียด บันทึกเป็นรายการโปรด และเครื่องหมายว่าอ่านแล้ว มีตัวเลือกกรองและเรียงลำดับหนังสือ

#### 3.3. Book Detail Page

หน้ารายละเอียดหนังสือ แสดงข้อมูลของหนังสือและคะแนนรีวิว มีปุ่มเพิ่ม/ลบจากรายการโปรด พร้อมปุ่มนำทางกลับไปยังแดชบอร์ด

#### 3.4. Favorites and History Management

อินเทอร์เน็ตจัดการรายการโปรดและประวัติการเข้าชม แสดงรายการหนังสือที่บันทึกเป็นโปรดและประวัติการเข้าชม พร้อมตัวเลือกกรองและเรียงลำดับ

#### 3.5. Personality Type Information

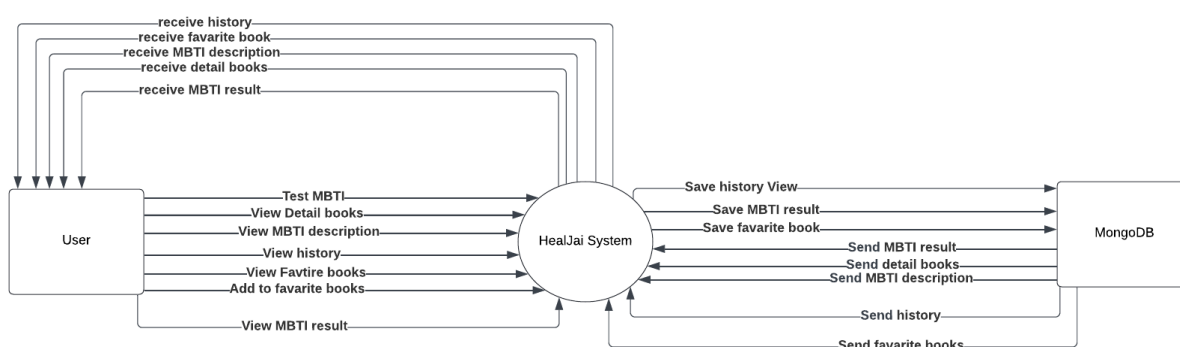
หน้าให้รายละเอียดประเภทบุคลิกภาพ MBTI ใช้สีสื่อความหมาย มีคำอธิบายลักษณะเด่นและคำแนะนำ พร้อมแถบเลื่อนเพื่อเลือกดูประเภทบุคลิกภาพต่าง ๆ



## Design Specification Document

### 1. Architectural Design

สถาปัตยกรรมของ HeaJai ใช้ Architectural Design แบบ client-server โดยแอปพลิเคชันมือถือ (Client) จะทำงานร่วมกับเซิร์ฟเวอร์ (Backend) เพื่อดึงข้อมูลและส่งคำขอ ผู้ใช้จะสามารถทำแบบทดสอบ MBTI และรับคำแนะนำหนังสือที่เหมาะสมผ่านแอปพลิเคชันมือถือ โดยเซิร์ฟเวอร์จะจัดการข้อมูลผ่าน MongoDB และจัดการ API ด้วย Express.js



#### 1.1. Major Architectural Components

- Mobile Application (Client):
  1. ใช้ React Native สำหรับการพัฒนาแอปพลิเคชัน
  2. จัดการแบบทดสอบ MBTI แสดงคำแนะนำหนังสือ และแสดงข้อมูลบุคลิกภาพ
- Backend Server (Server):
  1. ใช้ Node.js และ Express.js สำหรับการจัดการ API และการดำเนินการของเซิร์ฟเวอร์
  2. เชื่อมต่อกับฐานข้อมูล MongoDB เพื่อจัดเก็บข้อมูลผู้ใช้และหนังสือ
- Database: ฐานข้อมูลที่ใช้คือ MongoDB ซึ่งเป็นฐานข้อมูลแบบ NoSQL ใช้จัดเก็บข้อมูลที่เกี่ยวข้องกับผู้ใช้ บุคลิกภาพ MBTI และหนังสือแนะนำ

## 1.2. Data Flow

- ผู้ใช้ทำการเข้าสู่แอปพลิเคชัน
- ผู้ใช้สามารถทำแบบทดสอบ MBTI (ส่งคำตอบผ่าน API ไปยังเซิร์ฟเวอร์)
- เซิร์ฟเวอร์ประมวลผลผลลัพธ์ MBTI และจัดเก็บข้อมูลบุคลิกภาพของผู้ใช้ใน MongoDB
- เซิร์ฟเวอร์ส่งคำแนะนำหนังสือที่ตรงกับบุคลิกภาพไปยังแอปพลิเคชัน
- ผู้ใช้สามารถบันทึกหนังสือเป็นรายการโปรด และเรียกดูรายละเอียดของหนังสือได้

## 2. Module Design

### 2.1. MBTI Quiz Module

Primary Function: ให้ผู้ใช้ตอบแบบทดสอบ MBTI

Workflow:

1. แสดงคำถาม MBTI ทีละคำถามให้ผู้ใช้ตอบ
2. เมื่อผู้ใช้ตอบครบทุกคำถาม ระบบจะคำนวณผลลัพธ์และแสดงประเภทบุคลิกภาพ MBTI
3. ข้อมูลบุคลิกภาพจะถูกส่งไปยัง backend เพื่อจัดเก็บ

Managed Data:

1. คำถาม MBTI
2. ประเภทบุคลิกภาพของผู้ใช้

### 2.2. Book Recommendation Module

Primary Function: แสดงรายการหนังสือที่เหมาะสมกับประเภทบุคลิกภาพของผู้ใช้

Workflow:

1. ดึงข้อมูลประเภทบุคลิกภาพของผู้ใช้จากฐานข้อมูล
2. ใช้อัลกอริทึมการแนะนำเพื่อแสดงหนังสือที่ตรงกับบุคลิกภาพ MBTI ของผู้ใช้

3. แสดงรายการหนังสือพร้อมรายละเอียด เช่น ชื่อเรื่อง ผู้เขียน คะแนนรีวิว

Managed Data:

1. รายละเอียดหนังสือ (ชื่อ, ผู้เขียน, คำอธิบาย, คะแนนรีวิว)
2. รายการหนังสือที่แนะนำสำหรับประเภทบุคลิกภาพ MBTI แต่ละแบบ

### 2.3. Book Detail Page Module

Primary Function: แสดงรายละเอียดของหนังสือให้ผู้ใช้สามารถเข้าถึงข้อมูลที่สำคัญ เช่น ชื่อเรื่อง ผู้เขียน คะแนนรีวิว คำอธิบาย

Workflow:

1. แสดงข้อมูลหนังสือ เมื่อผู้ใช้เปิด Book Detail Page ระบบจะดึงข้อมูลหนังสือจากฐานข้อมูลและแสดงข้อมูลต่าง ๆ บนหน้าจอ เช่น ชื่อหนังสือ ผู้เขียน คะแนนรีวิว คำอธิบาย ประเภทหนังสือ
2. การโต้ตอบกับข้อมูลหนังสือ ผู้ใช้สามารถเลือกเพิ่มหนังสือใน รายการโปรด ได้โดยคลิกปุ่มที่จัดไว้

Managed Data:

1. ข้อมูลหนังสือ
  - ชื่อหนังสือ (Book Title)
  - ชื่อผู้เขียน (Author)
  - คะแนนรีวิว (Rating)
  - คำอธิบายของหนังสือ (Description)
  - ประเภทหนังสือ (Genre)
  - รูปภาพปกหนังสือ (Cover Image)

### 2.4. Personality Information Module

Primary Function: แสดงข้อมูลเกี่ยวกับบุคลิกภาพ MBTI แต่ละประเภท

Workflow:

1. แสดงรายละเอียดของประเภทบุคลิกภาพทั้ง 16 แบบ
2. ผู้ใช้สามารถเลือกดูข้อมูลเพิ่มเติมเกี่ยวกับบุคลิกภาพของตนเองหรือประเภทอื่น ๆ ได้

Managed Data:

1. ข้อมูลเชิงลึกเกี่ยวกับบุคลิกภาพแต่ละประเภท (เช่น จุดแข็ง จุดอ่อน ความชอบในการอ่าน)

## 2.5. Favorites and History Management Module

Primary Function: จัดการรายการโปรดและประวัติการอ่านของผู้ใช้

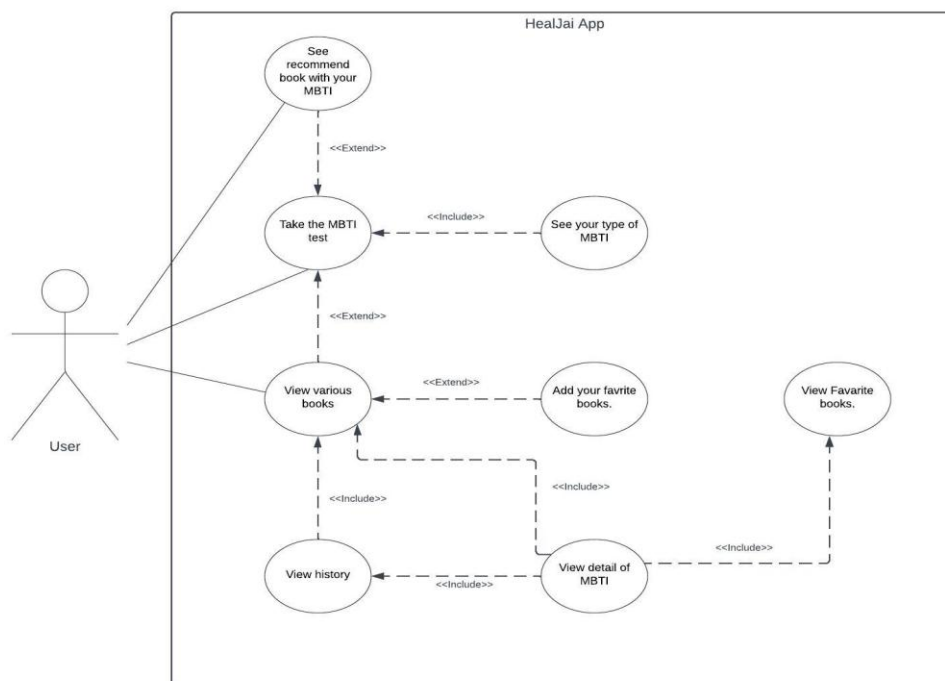
Workflow:

1. ผู้ใช้สามารถบันทึกหนังสือในรายการโปรดได้
2. สามารถดูและจัดการรายการโปรดและประวัติการเข้าชมหนังสือได้ (เช่น ลบหนังสือออกจากรายการโปรดหรือล้างประวัติการเข้าชม)

Managed Data:

1. ประวัติการเข้าชมหนังสือ
2. รายการโปรดของผู้ใช้

Use Cases



### 3. User Interface (UI/UX) Design

การออกแบบ UI จะเน้นการใช้งานง่าย ใช้โทนสีพาสเทลที่ช่วยให้ผู้ใช้รู้สึกผ่อนคลายและให้ประสบการณ์การใช้งานที่ราบรื่น

#### 3.1. MBTI Quiz Interface

Design: หน้าจอจะแสดงคำถามทีละข้อ พร้อมแถบแสดงความคืบหน้าของแบบทดสอบ มีปุ่มเลือกคำตอบขนาดใหญ่และชัดเจน

Color Scheme: ใช้โทนสีพาสเทลที่เป็นมิตรกับผู้ใช้ และมีปุ่มที่ออกแบบให้ใช้งานง่าย

#### 3.2. Book Recommendation

Design: แสดงรายการหนังสือที่แนะนำ มีปุ่มสำหรับบันทึกหนังสือเป็นรายการโปรดและเครื่องหมายที่บ่งบอกว่าอ่านแล้ว

Color Scheme: สีพื้นหลังโทนอ่อนและใช้ภาพปกหนังสือที่ชัดเจนเพื่อดึงดูดสายตา

#### 3.3. Book Detail Page

Design: แสดงข้อมูลหนังสือ เช่น ชื่อเรื่อง ผู้เขียน และคะแนนรีวิว

Color Scheme: ใช้สีที่เน้นการอ่านง่ายและสะอาดตา

#### 3.4. Personality Information Interface

Design: แสดงข้อมูลบุคลิกภาพ MBTI ทั้ง 16 แบบเป็นการ์ดแต่ละประเภท พร้อมข้อมูลสรุปจุดแข็ง จุดอ่อน และความชอบ มีให้เลือกอ่านเพิ่มเติม

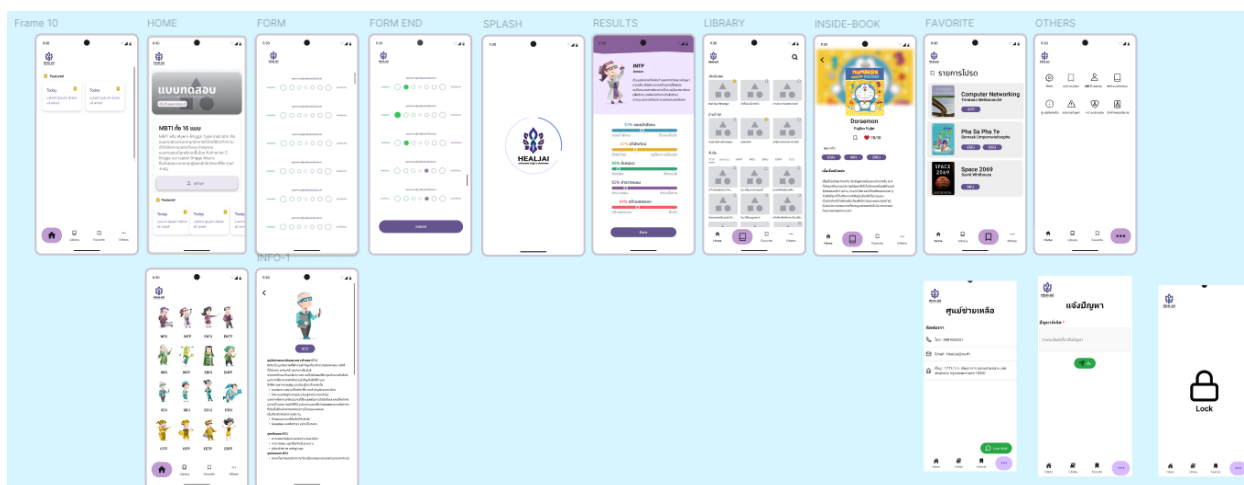
Color Scheme: ใช้โทนสีพาสเทลที่แตกต่างกันเล็กน้อยสำหรับแต่ละบุคลิกภาพ เพื่อความเป็นมิตรและอ่านง่าย

#### 3.5. Favorite and History Management Interface

Design: แสดงรายการหนังสือที่บันทึกเป็นโปรดและประวัติการเข้าชม พร้อมตัวเลือกในการจัดการ เช่น การลบหนังสือออกจากรายการโปรดหรือการล้างประวัติ

Color Scheme: ใช้สีที่แบ่งแยกชัดเจนระหว่างรายการโปรดและประวัติการเข้าชม

สามารถชมตัวอย่าง UI ได้ทางลิงก์นี้ [คลิก](#)



#### 4. Database Design

ฐานข้อมูลจะใช้ MongoDB สำหรับการจัดเก็บข้อมูลต่าง ๆ ดังนี้:

- ข้อมูลผู้ใช้ (รหัสผู้ใช้, ประเภทบุคลิกภาพ MBTI)
- ข้อมูลหนังสือ (ชื่อ, ผู้เขียน, ประเภท, คะแนนรีวิว)
- รายการโปรดและประวัติการเข้าชมหนังสือ

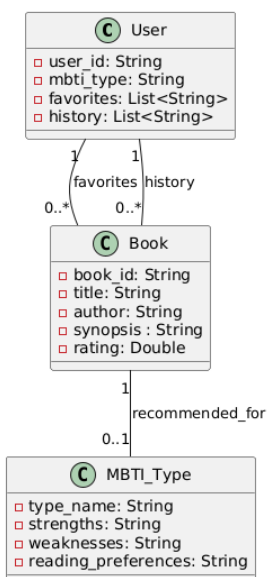
Database Structure:

##### 1. Users Collection:

- user\_id: รหัสผู้ใช้ (Unique)
- mbti\_type: ประเภทบุคลิกภาพ MBTI
- favorites: รายการโปรด (Array ของ book\_id)
- history: ประวัติการเข้าชม (Array ของ book\_id)

## 2. Books Collection:

- book\_id: รหัสหนังสือ (Unique)
- title: ชื่อหนังสือ
- author: ผู้เขียน
- synopsis: เนื้อเรื่องย่อหนังสือ
- rating: คะแนนรีวิวเฉลี่ย
- mbti\_type\_recommended: ประเภทบุคลิกภาพที่แนะนำ



## Implementation Note

ในส่วนของการพัฒนาแอปพลิเคชัน HealUai เราได้มีการวางแผนและดำเนินงานตามโครงสร้างที่กำหนดไว้อย่างชัดเจน โดยการพัฒนาครอบคลุมทั้ง Frontend และ Backend ซึ่งรายละเอียดของการ Implement แต่ละส่วนมีดังนี้

### 1. Frontend Implementation

เทคโนโลยีที่ใช้:

- Framework: React Native
- Language: TypeScript
- State Management: React State
- UI Components: ใช้การออกแบบ UI ตามดีไซน์สเปกที่ได้รับจากทีม UX/UI โดยใช้ Component-based architecture ในการแบ่งการพัฒนาฟีเจอร์แต่ละส่วน

ฟีเจอร์ที่ Implement:

1. การแสดงผลแบบทดสอบ MBTI:
  - พัฒนาหน้าจอแบบทดสอบที่ผู้ใช้สามารถเลือกคำตอบจากตัวเลือกที่กำหนดไว้ได้ โดยใช้ useState เพื่อจัดการ state ภายในแต่ละ component
2. การแสดงผลคำแนะนำหนังสือ:
 

ฟีเจอร์นี้เชื่อมต่อกับ Backend เพื่อดึงผลลัพธ์ MBTI และคำแนะนำหนังสือ โดยใช้ axios ในการเรียก API ข้อมูลที่ได้รับจะถูกเก็บใน React State เพื่อนำมาแสดงผลใน component ต่าง ๆ
3. การนำเสนอข้อมูลหนังสือ:
  - Implement การแสดงผลหนังสือที่แนะนำหลังจากทำแบบทดสอบ โดยข้อมูลจะถูกดึงมาจาก Backend และแสดงผลในรูปแบบของรายการหนังสือที่คลิกดูรายละเอียดเพิ่มเติมได้

Challenges:

- การจัดการกับข้อมูลที่แสดงผลในแต่ละขั้นตอน เช่น การแสดงผลแบบทดสอบที่มีหลายคำถาม และตัวเลือก ต้องการการออกแบบที่สามารถจัดการกับข้อมูลจำนวนมากได้อย่างมีประสิทธิภาพ



- การรักษาประสิทธิภาพของแอปพลิเคชันในการดึงข้อมูลและการจัดการการแสดงผล เมื่อมีการเชื่อมต่อกับ Backend

## 2. Backend Implementation

เทคโนโลยีที่ใช้:

- Framework: Node.js + Express.js
- Database: MongoDB
- API Development: RESTful API

ฟีเจอร์ที่ Implement:

1. การพัฒนา API สำหรับแบบทดสอบ MBTI:
  - Implement ฟังก์ชันที่ให้ผู้ใช้งานส่งข้อมูลคำตอบในแบบทดสอบผ่าน API และจัดการการประมวลผลเพื่อส่งผลลัพธ์กลับไปยัง Frontend
2. การพัฒนา API สำหรับการแสดงผลหนังสือ:
  - Implement API สำหรับการดึงข้อมูลหนังสือที่แนะนำตามผลลัพธ์ MBTI ข้อมูลหนังสือถูกจัดเก็บใน MongoDB โดยใช้ Mongoose schema สำหรับการสร้างโมเดลข้อมูล และ Express.js สำหรับการรับคำขอ API

Challenges:

- การจัดการกับความซับซ้อนของการเชื่อมต่อกับฐานข้อมูลที่มีหลายความสัมพันธ์ และการตอบสนองของ API ที่ต้องทำงานได้รวดเร็วและมีประสิทธิภาพ
- การรักษาความปลอดภัยของข้อมูลส่วนบุคคลของผู้ใช้และการตรวจสอบสิทธิ์ผ่านระบบ Authentication

## 3. Testing and Debugging

- การทดสอบในส่วน Frontend มีการใช้ React Developer Tools สำหรับการตรวจสอบ state และ Postman สำหรับทดสอบการเชื่อมต่อ API ของ Backend

- มีการทดสอบแบบ Manual Testing ในหลายรอบเพื่อให้แน่ใจว่าแต่ละฟีเจอร์ทำงานได้อย่างถูกต้องและตรงตามข้อกำหนด

เครื่องมือที่ใช้ในการทดสอบ:

- Postman: สำหรับทดสอบ API
- React Native Debugger: สำหรับตรวจสอบการทำงานของแอป

#### 4. Improvement and Correction

- ในระหว่างการพัฒนา มีการปรับแก้ไขข้อผิดพลาดที่พบในระหว่างการทดสอบ เช่น การเชื่อมต่อข้อมูลระหว่าง Frontend กับ Backend ที่ยังมีปัญหาบ้างในบางกรณี หรือการแสดงผลที่ไม่สมบูรณ์ตามดีไซน์
- มีการจัดการ Defects และแก้ไขโดยการปรับปรุงโค้ด และทำการทดสอบซ้ำหลายครั้งเพื่อให้มั่นใจว่าฟังก์ชันทุกส่วนทำงานได้อย่างสมบูรณ์

## Test Plan & Test Cases

### Test Plan

#### Introduction

Test Plan นี้จัดทำขึ้นเพื่ออธิบายแผนการทดสอบระบบของแอปพลิเคชัน HeaUai ซึ่งมีวัตถุประสงค์ในการตรวจสอบว่าระบบที่พัฒนาตรงตามข้อกำหนดของลูกค้าหรือไม่ รวมทั้งบันทึกข้อผิดพลาดที่เกิดขึ้นระหว่างการทดสอบเพื่อนำไปปรับปรุงแก้ไขตัวระบบในขั้นตอนต่อไป การทดสอบประกอบด้วยรายละเอียดของฟังก์ชันที่ต้องทำการทดสอบ เกณฑ์การประเมินผล ตลอดจนตารางเวลาของผู้ที่เกี่ยวข้องในกระบวนการทดสอบ

#### Approach

การทดสอบของ HeaUai จะแบ่งออกเป็นสองส่วนหลัก ได้แก่ การทดสอบ Frontend และการทดสอบ Backend โดยการทดสอบทั้งหมดจะใช้ข้อมูลจำลองและดำเนินการทดสอบผ่านการ Share Screen ในการประชุมบน Discord

- Frontend Testing: ใช้การทดสอบแบบ Manual Unit Testing โดยมุ่งเน้นที่การแสดงผลและการโต้ตอบของ UI ในแต่ละส่วน
- Backend Testing: ใช้การทดสอบแบบ Manual เพื่อยืนยันความถูกต้องของ API และการเชื่อมต่อข้อมูล โดยตรวจสอบฟังก์ชันการทำงานของระบบและฐานข้อมูล

#### Testing Methods

- การทดสอบ Unit ในแต่ละส่วนจะถือว่าผ่านเมื่อผลลัพธ์ของ Test Case ตรงตามที่คาดหวังไว้ และพบ Defects ไม่เกิน 2 จุด ในกรณีที่พบ Defects เกินเกณฑ์ที่กำหนด จะมีการทำรายงานและปรับปรุงฟังก์ชันก่อนที่จะดำเนินการทดสอบอีกครั้ง

#### Item Pass/Fail Criteria

การประเมินว่า Unit ใดจะผ่านหรือไม่ จะพิจารณาจากผลลัพธ์ที่ได้จากแต่ละ Test Case ถ้าผลลัพธ์ตรงตามที่คาดหวัง และพบข้อผิดพลาดไม่เกิน 2 จุดจะถือว่าผ่าน สำหรับข้อผิดพลาดใดที่เกิน 2 จุดขึ้นไป จะถือว่าฟังก์ชันนั้นไม่ผ่านเกณฑ์และต้องแก้ไขก่อนทำการทดสอบซ้ำ

#### Responsibilities

ผู้ที่มีสิทธิ์ตัดสินใจว่าฟังก์ชันใดจะทำการทดสอบหรือไม่

- Project Manager: นายณตฤณ ทองวิชิต

### ผู้ทดสอบ

- Tester: นายธฤต รวงผึ้งหลวง

### Schedule

- Unit Test (Frontend): จะทำการทดสอบทันทีหลังจากที่การพัฒนาแต่ละหน้าเสร็จสมบูรณ์
- Integration Testing (Backend): หลังจากเชื่อมต่อกับ Backend แล้ว การทดสอบจะต้องเสร็จสิ้นภายในวันที่ 7 ตุลาคม โดยมีการเผื่อเวลาล่าช้าถึงวันที่ 10 ตุลาคม

### Risk and Contingencies

- การเปลี่ยนแปลงของ Requirements: หากข้อกำหนดของลูกค้ามีการเปลี่ยนแปลง อาจทำให้การทดสอบและการพัฒนาเกิดความล่าช้า
- ความแตกต่างของ Environment: ความแตกต่างระหว่างสภาพแวดล้อมการพัฒนากับการทดสอบอาจส่งผลให้เกิดข้อผิดพลาดที่ไม่สามารถคาดการณ์ได้
- การขาดทรัพยากร: ทรัพยากรที่ไม่เพียงพอในส่วนของทีมพัฒนา Frontend อาจทำให้เกิดการล่าช้า
- การส่งมอบ Backend ล่าช้า: การส่งมอบจากทีม Backend ล่าช้าอาจทำให้กระบวนการทดสอบไม่เป็นไปตามกำหนด
- การทดสอบที่ล่าช้า: หากการทดสอบเกิดความล่าช้า อาจทำให้กระบวนการแก้ไขและพัฒนาแอปพลิเคชันล่าช้าตามไปด้วย Approvals

### Approvals

Project Manager: นายณตฤณ ทองวิชิต จะทำการตรวจสอบและอนุมัติ Test Plan หลังจากนั้นจะดำเนินงานในขั้นต่อไป

## Test Case

## Use Case Description

Use Case ID	Use Case Name	Description
UC-0001	MBTI Quiz	ผู้ใช้ทำแบบทดสอบเพื่อระบุประเภทบุคลิกภาพตามแบบ MBTI
UC-0002	Book Recommendation	ระบบแนะนำหนังสือตามประเภทบุคลิกภาพ
UC-0003	Personality Information	ระบบแสดงข้อมูลและลักษณะต่าง ๆ ของแต่ละประเภทบุคลิกภาพ MBTI
UC-0004	Favorites and History Management	ผู้ใช้สามารถจัดการหนังสือที่บันทึกเป็นรายการโปรดและดูประวัติการอ่านหนังสือ
UC-0005	Book Detail Page	แสดงข้อมูลรายละเอียดเกี่ยวกับหนังสือที่เลือก เช่น ชื่อเรื่อง ผู้เขียน คะแนนรีวิว และคำอธิบาย

Use Case ID	UC-0001
Use Case Name	MBTI Quiz
Actor	User
Description	ผู้ใช้ทำแบบทดสอบเพื่อระบุประเภทบุคลิกภาพตามแบบ MBTI
Normal course	<ol style="list-style-type: none"> <li>1. เปิดเข้าแบบทดสอบ MBTI จากหน้าหลัก</li> <li>2. ผู้ใช้ตอบคำถามที่มีอยู่ในแบบทดสอบทั้งหมด</li> <li>3. ผู้ใช้กดปุ่ม "Submit"</li> <li>4. ระบบประมวลผลคำตอบและแสดงผลประเภทบุคลิกภาพของผู้ใช้</li> </ol>
Alternate course	AC-1: ผู้ใช้กรอกข้อมูลไม่ครบถ้วน <ul style="list-style-type: none"> <li>- ระบบแสดงข้อความแจ้งเตือนให้กรอกข้อมูลให้ครบ</li> <li>- ผู้ใช้กลับไปแก้ไขข้อมูลและกดปุ่ม "Submit" อีกครั้ง</li> </ul>

Use Case ID	UC-0002
Use Case Name	Book Recommendation
Actor	User
Description	ระบบแนะนำหนังสือตามประเภทบุคลิกภาพ
Normal course	<ol style="list-style-type: none"> <li>1. ผู้ใช้เข้าสู่หน้าแนะนำหนังสือ</li> <li>2. ระบบตรวจสอบประเภทบุคลิกภาพของผู้ใช้</li> <li>3. ระบบแสดงรายการหนังสือที่แนะนำตามบุคลิกภาพ</li> <li>4. ผู้ใช้สามารถเลือกดูรายละเอียดหนังสือแต่ละเล่มหรือบันทึกลงในรายการโปรด</li> </ol>
Alternate course	<p>AC-1: ผู้ใช้ยังไม่ได้ทำแบบทดสอบ MBTI</p> <ul style="list-style-type: none"> <li>- ระบบจะแสดงรายการหนังสือตามความนิยม</li> </ul> <p>AC-2: ผู้ใช้ทำแบบทดสอบ MBTI แล้ว</p> <ul style="list-style-type: none"> <li>- ระบบจะแสดงรายการหนังสือตาม MBTI ของผู้ใช้เป็นหลัก</li> </ul>

Use Case ID	UC-0003
Use Case Name	Personality Information
Actor	User
Description	ระบบแสดงข้อมูลและลักษณะต่าง ๆ ของแต่ละประเภทบุคลิกภาพ MBTI
Normal course	<ol style="list-style-type: none"> <li>1. ผู้ใช้เลือกดูข้อมูลประเภทบุคลิกภาพ (เช่น INTJ, ENFP)</li> <li>2. ระบบแสดงข้อมูลเชิงลึกเกี่ยวกับบุคลิกภาพที่เลือก เช่น จุดแข็ง จุดอ่อน อาชีพที่เหมาะสม ฯลฯ</li> </ol>
Alternate course	AC-1: ไม่สามารถดึงข้อมูลบุคลิกภาพได้ <ul style="list-style-type: none"> <li>- หากระบบไม่สามารถดึงข้อมูลบุคลิกภาพได้ระบบจะแจ้งผู้ใช้งานว่าไม่สามารถแสดงข้อมูลในขณะนี้</li> </ul>

Use Case ID	UC-0004
Use Case Name	Favorites and History Management
Actor	User
Description	ผู้ใช้สามารถจัดการหนังสือที่บันทึกเป็นรายการโปรดและดูประวัติการอ่านหนังสือ
Normal course	<ol style="list-style-type: none"> <li>1. ผู้ใช้เปิดหน้า "รายการโปรด" หรือ "ประวัติการอ่าน"</li> <li>2. ระบบแสดงหนังสือที่ผู้ใช้บันทึกไว้หรือดูย้อนหลัง</li> <li>3. ผู้ใช้สามารถลบหนังสือจากรายการโปรดหรือประวัติการอ่านได้</li> <li>4. ระบบอัปเดตข้อมูลตามการเปลี่ยนแปลงของผู้ใช้</li> </ol>
Alternate course	AC-1: ไม่สามารถดึงข้อมูลได้ <ul style="list-style-type: none"> <li>- หากไม่มีหนังสือในรายการโปรดหรือประวัติการอ่านระบบจะแจ้งว่าไม่มีข้อมูล</li> <li>- หากระบบไม่สามารถลบหรือแก้ไขรายการได้ ระบบแจ้งข้อผิดพลาดให้ผู้ใช้งานทราบ</li> </ul>



Use Case ID	UC-0005
Use Case Name	Book Detail Page
Actor	User
Description	แสดงข้อมูลรายละเอียดเกี่ยวกับหนังสือที่เลือก เช่น ชื่อเรื่อง ผู้เขียน คะแนนรีวิว และคำอธิบาย
Normal course	<ol style="list-style-type: none"> <li>1. ผู้ใช้เลือกหนังสือ</li> <li>2. ระบบดึงข้อมูลรายละเอียดหนังสือจากฐานข้อมูล</li> <li>3. ระบบแสดงรายละเอียดของหนังสือ ได้แก่: <ul style="list-style-type: none"> <li>• ชื่อเรื่อง</li> <li>• ผู้เขียน</li> <li>• คะแนนรีวิว</li> <li>• คำอธิบาย</li> <li>• รูปปกหนังสือ</li> </ul> </li> <li>4. ผู้ใช้สามารถเพิ่มหนังสือลงในรายการโปรด</li> </ol>
Alternate course	<p>AC-1: ไม่สามารถดึงข้อมูลหนังสือได้</p> <ul style="list-style-type: none"> <li>- หากระบบไม่สามารถดึงข้อมูลหนังสือได้ ระบบจะแจ้งผู้ใช้งานว่าไม่สามารถแสดงข้อมูลในขณะนี้</li> </ul>

Test Case Summary: รายงานการทดสอบนี้มี Test Case ทั้งหมด 16 รายการ แบ่งออกเป็น 5 หมวดหมู่ตาม Use Case ที่กำหนดไว้

Test Cases for UC-0001: MBTI Quiz

Test Case ID: UC-0001

Use Case Name: MBTI Quiz

Pre-Condition: เปิดแอปพลิเคชัน

Test Case ID	Test Case	Test Step	Test Data	Expected Result
TC-001	กรอกข้อมูลครบถ้วน	1. User เปิดหน้าทดสอบ 2. User ตอบคำถามทั้งหมด	คำถามทั้งหมด	แสดงประเภทบุคลิกภาพ MBTI
TC-002	กรอกข้อมูลไม่ครบถ้วน	1. User เปิดหน้าทดสอบ 2. ไม่ตอบคำถาม	ข้ามคำถาม	ระบบแจ้งเตือนให้ตอบคำถามให้ครบ
TC-003	ทดสอบการส่งข้อมูลที่รวดเร็ว	1. User เปิดหน้าทดสอบ 2. User ตอบคำถามอย่างรวดเร็ว	คำถามทั้งหมด	แสดงประเภทบุคลิกภาพ MBTI
TC-004	ทดสอบการทำแบบทดสอบซ้ำ	1. User ทำแบบทดสอบเสร็จแล้ว 2. User เริ่มทำแบบทดสอบใหม่	คำถามทั้งหมด	ระบบแสดงคำถามใหม่
TC-005	ทดสอบการออกจากหน้าทดสอบ	1. User เปิดหน้าทดสอบ 2. User ออกจากหน้าทดสอบ	-	ระบบกลับไปยังหน้าหลัก

## Test Cases for UC-0002: Book Recommendation

Test Case ID: UC-0002

Use Case Name: Book Recommendation

Pre-Condition: เปิดแอปพลิเคชัน

Test Case ID	Test Case	Test Step	Test Data	Expected Result
TC-006	ไม่ได้ทำแบบทดสอบ MBTI	1. ระบบแนะนำหนังสือ	-	แสดงรายการหนังสือตามความนิยม
TC-007	ทำแบบทดสอบ MBTI แล้ว	1. User ทำแบบทดสอบ MBTI 2. ระบบแนะนำหนังสือ	ประเภทบุคลิกภาพ	แสดงหนังสือที่ตรงกับประเภทบุคลิกภาพ
TC-008	ไม่เคยเข้าชมหนังสือใดเลย	1. User เข้าหน้าแนะนำหนังสือ	-	ไม่แสดงหนังสือที่เคยอ่าน
TC-009	ตรวจสอบการแสดงผลหนังสือตามหมวดหมู่	1. User ดูหนังสือตามหมวดหมู่	หมวดหมู่ที่เลือก	ระบบแสดงรายการหนังสือตามหมวดหมู่
TC-010	ตรวจสอบการไปยังหน้ารายละเอียดหนังสือ	1. User คลิกที่หนังสือ	ชื่อหนังสือ	แสดงรายละเอียดหนังสือ

Test Cases for UC-0003: Personality Information

Test Case ID: UC-0003

Use Case Name: Personality Information

Pre-Condition: เปิดแอปพลิเคชัน

Test Case ID	Test Case	Test Step	Test Data	Expected Result
TC-011	แสดงข้อมูลบุคลิกภาพ	1. User ดูข้อมูลบุคลิกภาพ	-	แสดงข้อมูลบุคลิกภาพที่ถูกต้อง
TC-012	ทดสอบการแสดงผลข้อมูลรายละเอียด	1. User คลิกที่ข้อมูลบุคลิกภาพ	ประเภทบุคลิกภาพ	ระบบแสดงรายละเอียดบุคลิกภาพ

## Test Cases for UC-0004: Favorites and History Management

Test Case ID: UC-0004

Use Case Name: Favorites and History Management

Pre-Condition: User มีหนังสือในรายการโปรดและมีประวัติการอ่าน

Test Case ID	Test Case	Test Step	Test Data	Expected Result
TC-013	เพิ่มหนังสือในรายการโปรด	1. User ดูหนังสือ	ชื่อหนังสือ	ระบบเพิ่มหนังสือในรายการโปรด
TC-014	ลบหนังสือจากรายการโปรด	1. User ดูหนังสือในรายการโปรด 2. User ลบหนังสือ	ชื่อหนังสือ	ระบบลบหนังสือออกจากรายการโปรด
TC-015	ดูประวัติการอ่าน	1. User เปิดหน้าประวัติ	-	แสดงประวัติการอ่านที่ถูกต้อง
TC-016	ลบประวัติการอ่าน	1. User เปิดหน้าประวัติการอ่าน 2. User ลบประวัติ	ชื่อหนังสือ	ระบบลบประวัติการอ่าน

## Test Cases for UC-0005: Book Detail Page

Test Case ID: UC-0005

Use Case Name: Book Detail Page

Pre-Condition: User กำลังดูรายการหนังสือ

Test Case ID	Test Case	Test Step	Test Data	Expected Result
TC-017	ดูรายละเอียดหนังสือ	1. User คลิกที่หนังสือจากรายการ	ชื่อของหนังสือที่เลือก	ระบบแสดงรายละเอียดหนังสืออย่างถูกต้อง
TC-018	เพิ่มหนังสือลงในรายการโปรด	1. User คลิกที่ปุ่มเพิ่มในรายการโปรด	ชื่อหนังสือ	ระบบเพิ่มหนังสือในรายการโปรด
TC-019	ทดสอบการไม่สามารถดึงข้อมูล	1. ระบบไม่สามารถดึงข้อมูลหนังสือ	-	ระบบแสดงข้อความว่าไม่สามารถแสดงข้อมูลได้ในขณะนี้

## Summary

### ประเด็นที่พบ

ในการทดสอบระบบ HeaUai พบว่ามีความท้าทายในการเชื่อมต่อ Backend บางครั้งข้อมูลที่ดึงมาแสดงไม่สมบูรณ์ นอกจากนี้ยังพบปัญหาเล็กน้อยเกี่ยวกับการจัดเรียงข้อมูลใน UI ที่ไม่ตรงกับที่ออกแบบไว้

### วิธีการจัดการ

- มีการปรับปรุงการเชื่อมต่อ Backend เพื่อให้การดึงข้อมูลมีความเสถียรมากขึ้น
- ปรับแต่ง UI ให้ตรงกับดีไซน์ที่ได้ออกแบบไว้
- เพิ่มการทดสอบซ้ำในส่วนของการเชื่อมต่อ Backend และการแสดงผลของ Frontend

### บทเรียนที่ได้เรียนรู้

- การทดสอบและการปรับปรุงอย่างต่อเนื่องช่วยลด Defects และปัญหาที่เกิดขึ้นในกระบวนการพัฒนา
- การสื่อสารและทำงานร่วมกันระหว่างทีม Backend และ Frontend เป็นปัจจัยสำคัญที่ช่วยให้การพัฒนาเป็นไปอย่างราบรื่น
- การเตรียมข้อมูลและแผนการทดสอบล่วงหน้าช่วยให้การทำงานมีประสิทธิภาพและลดความเสี่ยงของการล่าช้าในกระบวนการทดสอบ

## Appendices

### References

### Templates

- Free Software Project Plan Templates: [Smartsheet](#)

### Further Readings

- React Native: [React Native Documentation](#)
- Stack Overflow: [Stack Overflow - Where Developers Learn, Share, & Build Careers](#)

### Additional Documents

### Documentation

- Software Requirements Specification (SRS): [Google Drive Link](#)
- Software Design Document (SDD): [Google Drive Link](#)
- Figma's User Interface Design Link: [Figma Project](#)
- Software Implementation & Testing Documentation: [Google Drive Link](#)