**All Learn**

ข้อกำหนดความต้องการของระบบ

Version: 0.01

Date: 10/2/2566

By

1.นายศุภณัฏฐ์ อภิพัฒน์จิรา 6320500662

2.นางสาวศุภลักษณ์ เหลือบุญชู 6320500671

3.นายอัษฎาวุธ คล้ายเมือง 6320500719

4.นายปวีณวิชญ์ ท่าดี 6320502461

5.นางสาวสรัลนุช ลือนภา 6320502517

6.นายปิยะ รัตนไชยดำรง 6320503068

**สารบัญ**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | จัดทำโดย | Revision Date | การเปลี่ยนแปลง |
| 0.01 Draft |  | 10/02/2023 |  |
|  |  |  |  |

1. **บทนำ**

ในปัจจุบันการเรียนรู้มีหลากหลายรูปแบบ รูปแบบที่เป็นที่นิยมมากที่สุดคือการเรียนรู้จากอินเทอร์เน็ตและผู้ที่มีความรู้และมีประสบการณ์ในหลายๆด้านที่อยากจะแบ่งปันความรู้ของตนเองให้กับคนทั่วไปและคนส่วนใหญ่ในปัจจุบันหาความรู้จากอินเทอร์เน็ตเพราะมีความสะดวกรวดเร็วและเข้าถึงง่าย แต่ความรู้ในหลายๆด้านยังกระจัดกระจายกันอยู่ในหลายๆแหล่ง ทางคณะผู้จัดทำจึงเล็งเห็นปัญหาในด้านนี้จึงได้จัดทำเว็บไซต์...นี้ขึ้นมาเพื่อแก้ปัญหาในด้านนี้ เว็บไซต์...สามารถให้ผู้ที่มีความรู้ในด้านต่างๆมาแชร์ความรู้ในเว็บและแบ่งหมวดหมู่เพื่อให้ผู้ใช้หาความรู้ได้ง่ายขึ้น และยังสามารถเปิดกิจกรรมให้ผู้ที่สนใจกดเข้าร่วมกิจกรรมได้

1. **ความต้องการของผู้ใช้ (User Requirements)**
   1. ส่วนต่อประสานของซอฟต์แวร์

* React.Js
* Java
* Node.js
* phpMyAdmin
  1. คุณลักษณะของผู้ใช้

- ผู้ใช้มีความคุ้นเคยกับเว็บเพจประเภท Community page อาทิ Facebook, Medium, Blockdit

* 1. คุณสมบัติของระบบ

-สามารถ Post ข้อความได้้

-สามารถแก้ไขหรือลบ Post ได้้

-สามารถสร้างกิจกรรมให้ผู้ใช้งานสามารถเข้าร่วมได้

-สามารถกด Like และ Comment ได้

-สามารถ Donate ให้กับผู้สร้างโพสต์ได้

-สามารถ Save Post ที่ถูกใจเก็บไว้ในคลังของตัวเองได้

-สามารถยื่นขอสิทธิ Verification เพื่อสร้างความน่าเชื่อถือให้กับผู้ใช้ได้

* 1. สมมติฐานและข้อจำกัดในการพัฒนา
* การทำกิจกรรมในระบบจะต้องอยู่ภายใต้การใช้งาน Internet
* จำนวนผู้ใช้ หากมีคนเข้าใช้ระบบมากเกินไปอาจจะเกิดการล่มของเว็บไซต์ได้

1. **ความต้องการของระบบ**

/\* หัวข้อนี้เป็นหัวข้อที่สำคัญและมีเนื้อหาเยอะที่สุด ซึ่งเป็นการเขียนความต้องการของระบบ (system requirements) ในระดับที่ละเอียดพอที่จะส่งต่อให้นักออกแบบและนักทดสอบทำงานต่อไปได้ เราจะต้องเขียนความต้องการอย่างชัดเจน และอาจระบุหมายเลขความต้องการเพื่อใช้อ้างอิงต่อไปได้\*/

* 1. ความต้องการแบบ functional

/\* ระบุความต้องการของระบบแบบ functional เป็นข้อๆ โดยควรมีการแบ่งหมวดหมู่หรือจัดประเภทตามความเหมาะสม เช่น จัดตาม mode การทำงาน ตามประเภทผู้ใช้ ตามคุณสมบัติของระบบ (system feature) ตามกรณีใช้งาน (use case) สถานการณ์การใช้งาน (scenario, user story) หรือตามวัตถุในโลกความจริงที่เกี่ยวข้องในระบบ (เช่น sensors, ยา, หมอ, พยาบาล, ผู้ป่วย, เครื่องวัดคลื่นหัวใจ เป็นต้น) เป็นต้น

อาจเขียนในรูปแบบการ์ดความต้องการ

1. ความต้องการแต่ละข้อควรเขียนอย่างถูกต้อง ไม่คลุมเครือ ไม่ขัดแย้งกัน ตรวจสอบได้ จัดลำดับความสำคัญ

ในการเขียนความต้องการแต่ละข้อควรระบุ ชื่อ หมายเลข ลำดับความสำคัญ อินพุต กระบวนการการทำงาน เอาท์พุต การจัดการกับความผิดพลาด

|  |  |
| --- | --- |
| ตัวอย่างการลำดับหัวข้อแยกตาม system feature | ตัวอย่างการลำดับหัวข้อแยกตามประเภทผู้ใช้ |
| 3.2.1 System Feature 1  3.2.1.1 คำบรรยายและความสำคัญ: อธิบายว่า feature นี้หมายถึงอะไร ใส่ความสำคัญเป็นระดับ เช่น Mandatory, desirable เป็นต้น  3.2.1.2 ลำดับการทำงาน: บอกลำดับการทำงานและการตอบสนองของระบบต่อผู้ใช้หรือโลกภายนอก  3.2.1.3 ความต้องการ  3.2.1.3.1 functional req #1.1  1) description:  2) inputs:  3) process:  4) outputs:  5) errors handling:  3.2.1.2 functional req #1.2  1) description:  2) inputs:  3) process:  4) outputs:  3.2.2 System Feature 2 | 3.2.1 User Class 1  3.2.1.1 functional req #1.1  1) description: อธิบายพร้อมบอกความสำคัญ  2) inputs: บอกข้อมูลที่ต้องใช้ในการทำฟังก์ชั่นโดยละเอียด เพื่อส่งต่อให้ฝ่ายออกแบบทำงานต่อได้  3) process:  4) outputs:  5) errors handling:  3.2.1.2 function req #1.2  …  3.2.2 User Class 2  … |
| การเขียนตาม use case ดู <http://tynerblain.com/blog/2007/04/09/sample-use-case-example/> | การเขียนตาม user story ดู   * <https://opkey.com/user-guide/pages.php?subid=151&accordid=10&val=TEAM> * <https://goo.gl/images/4YVG3S> |

เมื่อรวบรวม requirement ควรมีการจำลำดับความสำคัญของ requirement ด้วย\*/

Use Case 1 : ลงทะเบียนผู้ใช้

Primary Actor: ผู้ใช้

Pre Condition: การเชื่อมต่ออินเทอร์เน็ต

Main Scenario :

1.ผู้ใช้ต้องกรอกข้อมูล ชื่อ-นามสกุล ชื่อบัญชีผู้ใช้ e-mail รหัสผ่าน

2.สร้างโปรไฟล์ ใส่ข้อมูล

Use Case 2 : Login

Primary Actor: ผู้ใช้

Pre Condition: การเชื่อมต่ออินเทอร์เน็ตและการลงทะเบียนผู้ใช้

Main Scenario :

1.ผู้ใช้ต้องกรอก ชื่อบัญชีผู้ใช้ และ รหัสผ่าน เพื่อlogin

Use Case 3 : สร้าง ลบ แก้ไขโพสต์และกิจกรรม

Primary Actor: ผู้ใช้

Pre Condition: การเชื่อมต่ออินเทอร์เน็ตและการลงทะเบียนผู้ใช้

Main Scenario :

1.การสร้างโพสต์ ผู้ใช้กดปุ่มโพสต์ หน้าเว็บจะเด้งขึ้นมาให้ผู้ใช้พิมพ์ถ้าผู้ใช้ยังไม่ได้พิมพ์อะไรจะไม่สามารถกดโพสต์ลงในเว็บได้

2.การแก้ไขโพสต์ ผู้ใช้กดตรงโพสต์และจะรายการขึ้นมาให้เลือก ให้กดตรงแก้ไขโพสต์ หน้าเว็บจะเด้งให้แก้ไข ถ้าผู้ใช้ไม่แก้ไขอะไรจะไม่สามารถกดปุ่มแก้ไขได้

3.การลบโพสต์ ผู้ใช้กดตรงโพสต์และจะรายการขึ้นมาให้เลือก ให้กดตรงลบโพสต์หน้าเว็บจะเด้งการยืนยันการลบโพสต์ให้ผู้ใช้กดยืนยันลบเพื่อลบ กดยกเลิกเพื่อยกเลิกการลบ

Use Case 4 : การเข้าร่วมกิจกรรม

Primary Actor: ผู้ใช้

Pre Condition: การเชื่อมต่ออินเทอร์เน็ตและการลงทะเบียนผู้ใช้

Main Scenario :

1.โพสต์ที่เป็นกิจกรรมจะมีปุ่มการเข้าร่วมให้ผู้ใช้กด ผู้ใช้สามารถกดเข้าร่วมกิจกรรมที่สนใจ

Use Case 5 : ค้นหา

Primary Actor: ผู้ใช้ ผู้เยี่ยมชม

Pre Condition: การเชื่อมต่ออินเทอร์เน็ต

Pre Condition

1.ผู้ใช้และผู้เยี่ยมชมสามารถค้นหาโพสต์ด้วยคำที่พิมพ์ลงไปและหมวดหมู่ของโพสต์ได้

Use Case 6 : donate

Primary Actor: ผู้ใช้

Pre Condition: การเชื่อมต่ออินเทอร์เน็ตและการลงทะเบียนผู้ใช้

Main Scenario :

1.ผู้ใช้สามารถ donate เงิน ให้กับผู้สร้างโพสต์ได้ โดยกดปุ่ม donate ในโพสต์

Use Case 7 : Verification

Primary Actor: ผู้ใช้

Pre Condition: การเชื่อมต่ออินเทอร์เน็ตและการลงทะเบียนผู้ใช้

Main Scenario :

1.ผู้ใช้ที่มีผู้ติดตามมากกว่า10,000คน สามารถทำเรื่องยื่นขอ Verification จากแอดมินได้

Use Case 7 : แก้ไขบัญชีผู้ใช้

Primary Actor: ผู้ใช้

Pre Condition: การเชื่อมต่ออินเทอร์เน็ตและการลงทะเบียนผู้ใช้

Main Scenario :

1.ผู้ใช้กดไปยังหน้าบัญชีผู้ใช้ และ กดปุ่มแก้ไขบัญชีผู้ใช้ เมื่อแก้ไขข้อมูลเรียบร้อยแล้ว ต้องใส่รหัสผ่านเพื่อเป็นการยืนยันการแก้ไข

* 1. ความต้องการแบบ Non-functional
* เว็บไซต์ลองรับผู้ใช้งานพร้อมกันได้มากกว่า 10,000 คน
* เว็บไซต์ทำงานได้โดยไม่เกิดการล้มเหลวอย่างน้อย 1 ใน 1,000 ของชั่วโมงการทำงาน

/\* ระบุความต้องการแบบ Non-functional ที่วัดได้ โดยควรบอกคุณสมบัติที่ต้องการ (เช่น reliability) อธิบายว่ามันสำคัญกับระบบอย่างไร และวิธีการวัด อาจมีการอ้างอิง functional requirements ซึ่งอาจแบ่งเป็นหัวข้อย่อยตามคุณสมบัติที่ระบบควรมีหรือต้องคำนึงถึง เช่น

3.3.1 Performance: บอกประสิทธิภาพของการทำงานภายใต้สภาพแวดล้อมต่างๆ เช่น

ระบบรองรับคำขอการลงทะเบียนเรียนจากนิสิตพร้อมกันได้ โดยระบบสามารถรองรับคำขอได้ 250 คำขอในเวลา 1 วินาที

เหตุผล: นิสิตมักจะส่งคำร้องขอการลงทะเบียนในเวลาเดียวกัน ซึ่งมหาวิทยาลัยมีนิสิตประมาณ 30,000 คน ซึ่งประมาณการว่านิสิตจะใช้เวลาในการลงทะเบียนประมาณ 10 นาทีต่อคน ทำให้สามารถรองรับการลงทะเบียนทั้งหมดได้ภายใน 24 ชม.

วิธีการวัด: …

3.3.2 Reliability: บอกความน่าเชื่อถือของระบบ เช่น ระบบจะต้องมีความน่าเชื่อถือและไม่ล้มเหลวถึงแม้ว่าจะเป็นช่วงลงทะเบียนของนิสิต โดยหากระบบล้มเหลว จะต้องกลับมาใช้งานได้ภายใน 10 นาที

3.3.3 Availability: บอกการใช้งานได้ของระบบ เช่น ระบบสามารถใช้งานได้ตลอดเวลาในช่วง 8.00 – 17.00 น. โดยเวลาที่ระบบดาวน์รวมกันจะต้องไม่เกิน 5 นาทีต่อวันในช่วงเวลาดังกล่าวเป็นต้น

\*/

**4.** **แผนภาพการวิเคราะห์ระบบ**