CPE ให้ยืม

ระบบยืม-คืนอุปกรณ์

Version: 1.0

Date: 2/10/2563

By

กลุ่มที่ 5

นายธนาธร ทรงพินิจ 6020500365

นายธนานพ ครองชีพ 602051311

นายปริญญา พ่วงโสม 602051370

นายธนะเมศฐ์ ภัคโชตน์ธนิน 602052082

**สารบัญ**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | จัดทำโดย | Revision Date | การเปลี่ยนแปลง |
|  |  |  |  |
|  |  |  |  |

1. **บทนำ**

<บอกที่มา ความสำคัญ เหตุผล และวัตถุประสงค์ของการพัฒนาระบบ บรรยายบริบทและขอบเขตของผลิตภัณฑ์ที่จะพัฒนาโดยย่อ โดยจะต้องระบุชื่อระบบ อธิบายว่าระบบควรทำอะไร หรือไม่ทำอะไร>

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

2.1.1 การ login เพื่อเข้าสู้ระบบด้วยการใช้ระบบนนทรี

2.1.2 การใช้ฐานข้อมูลในการเก็บข้อมูลต่าง ๆ

2.1.3 การส่ง E-mail แจ้งเตือนอัตโนมัติ

* 1. ส่วนต่อประสานกับผู้ใช้

<บรรยายคุณสมบัติของส่วนต่อประสานระหว่างระบบและผู้ใช้ที่**ผู้ใช้กำหนดเป็นพิเศษ**ในเชิงตรรกะ จะเป็นการบรรยายว่าระบบจะมีปฏิสัมพันธ์กับผู้ใช้อย่างไร เช่น รูปแบบหน้าจอที่ต้องการ การวางตำแหน่งของปุ่มหรือองค์ประกอบต่างๆ ทางหน้าจอ ปุ่ม shortcuts ที่จะใช้ วิธีการแสดงข้อความผิดพลาด เป็นต้น>

* 1. คุณลักษณะของผู้ใช้

<บอกว่าใครเป็นผู้ใช้ระบบนี้ โดยควรจัดกลุ่มเป็นประเภทของผู้ใช้ ผู้ใช้แต่ละประเภทมีคุณสมบัติอย่างไร (เช่น ความรู้พื้นฐานทางคอมพิวเตอร์ ระดับการศึกษา สิทธิ์ในการเข้าถึงระบบ ความบ่อยในการใช้งาน เหตุผลที่จะใช้ระบบ เป็นต้น)>

* 1. คุณสมบัติของระบบ

<ระบุความต้องการของผู้ใช้ (user requirements) ซึ่งบอกฟังก์ชั่นหลักของระบบที่ระบบต้องทำหรือระบบให้บริการใดกับผู้ใช้บ้าง ควรเขียนเป็นภาษาที่ผู้อ่านทั่วไปก็สามารถอ่านเข้าใจได้ อาจเขียนเป็นรายการ อาจใช้แผนภาพอย่างง่ายประกอบเพื่อแสดงความสัมพันธ์ของส่วนต่างๆ แต่ไม่บอกถึงการออกแบบ>.

* 1. สมมติฐานและข้อจำกัดในการพัฒนา

<ลิสต์ปัจจัยต่างๆ ที่อาจส่งผลกระทบกับความต้องการในเอกสารนี้ ซึ่งหากมีการเปลี่ยนแปลงในปัจจัยดังกล่าว มันสามารถส่งผลกับความต้องการ รวมถึงการออกแบบหรือการพัฒนาต่อไป เช่น หากผลิตภัณฑ์ที่พัฒนาจะต้องรันบน OS หรือฮาร์ดแวร์ใด แล้วหากไม่มี OS หรือฮาร์ดแวร์นั้นจะทำให้ความต้องการเปลี่ยน

หัวข้อนี้จะเก็บทุกประเด็นที่ส่งผลกับการออกแบบระบบ แต่ไม่สามารถใส่ได้ในหัวข้ออื่น>

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

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> |

\*/

* 1. ความต้องการแบบ Non-functional

/\* 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 นาทีต่อวันในช่วงเวลาดังกล่าว

เป็นต้น

\*/

1. **แผนภาพการวิเคราะห์ระบบ**

<อาจใส่แผนภาพการวิเคราะห์ระบบ เช่น DFD, Use case diagrams, flow chart, activity diagram เป็นต้น >

1. **ภาคผนวก**

<เป็นการให้ข้อมูลเพิ่มเติมที่จะเป็นประโยชน์ในการพัฒนา เช่น เอกสารรายละเอียดของโครงการพัฒนาระบบ เอกสารทางการตลาด บันทึกการประชุมกับลูกค้า เอกสารที่เกี่ยวข้องกับระบบเดิม (หากเป็นการปรับปรุงระบบใหม่) รายการของผลิตภัณฑ์ที่เกี่ยวข้องกับระบบนี้ (ที่มีการวิเคราะห์ความแตกต่างกับระบบที่จะพัฒนา) เป็นต้น>