

ข้อเสนอโครงการ

1. ชื่อโครงการ (Project Name) Chula Course Table

2. บทนำ (Introduction)

สำหรับการศึกษาระดับมหาวิทยาลัยแล้ว มีหลายสิ่งที่ย้ายไปจากการศึกษาที่ผ่านมา ทั้งการเรียนการสอนที่ยกระดับขึ้นจากเดิม การที่ผู้เรียนต้องเดินไปเรียนตามห้องต่าง ๆ หรือแม้แต่การเก็บคะแนนที่เปลี่ยนแปลงไปจากเดิม และสิ่งที่ขาดไม่ได้เลยสำหรับการศึกษาระดับมหาวิทยาลัยที่ทุก ๆ คนต้องจัดแจงด้วยตนเองก็คือ การลงทะเบียนเรียน ซึ่งการลงทะเบียนเรียนให้ได้วันและเวลาที่ต้องการนั้นเป็นสิ่งที่ไม่แน่นอน ทำให้เกิดปัญหาในการจัดแจงตารางเรียน

3. ที่มาและความสำคัญของปัญหา (Organization Background)

การจัดแจงตารางเรียนให้ลงตัวแต่ละเทอมต่างเป็นปัญหาของนิสิตจำนวนมากไม่ว่าจะเป็นเลือก section ที่ต้องการจะเรียนในบางวิชาที่แต่ละ section มีเวลาเรียนต่างกัน การเลือก Approve และ/หรือ Gened โดยไม่ให้เวลาเรียนและวันสอบตรงกัน และสิ่งที่ยากกว่าที่กล่าวคือกรณีที่ลงทะเบียนไม่ติดในบางวิชา ทำให้เวลาที่วางไว้ใช้ไม่ได้ แล้วต้องมานั่งแก้ตารางเรียนใหม่ทั้งหมด โดยแต่ละคนก็วางแผนตารางเรียนต่างวิธีออกไป มีการใช้กระดานจดวิชาที่จะเรียน ลองหาวิชา Gened ที่เปิดเรียน หรือบางคนนำ program เช่น excel มาใช้ในการจัดการตารางเรียน แต่สิ่งเหล่านั้นต่างใช้เวลาในการทำมากเพราะไม่ใช่เครื่องมือที่สร้างมา เพื่อจัดการแก้ไขปัญหานี้โดยตรง

เนื่องจากปัญหาข้างต้น คณะผู้จัดทำจึงต้องการเสนอระบบที่จะช่วยแก้ไขปัญหานี้เพื่อทำให้นิสิตภายในจุฬาลงกรณ์มหาวิทยาลัยมีความสะดวกสบายในการจัดแจงตารางเรียนให้ตรงความต้องการ และประหยัดเวลาในการจัดแจงตารางเรียน

4. ลักษณะของปัญหาและแนวทางแก้ไข (Problem Statement)

4.1. นิยามศัพท์

นิสิต คือ นิสิตของจุฬาลงกรณ์มหาวิทยาลัย

ลงทะเบียนเรียน คือ การลงทะเบียนในรายวิชาที่ศึกษาในปีการศึกษานั้นๆ

ระบบลงทะเบียนเรียน คือ ระบบที่เปิดให้นิสิตเข้ามาลงทะเบียนเรียนได้

วิชาบังคับ คือ วิชาที่นิสิตจะต้องลงทุกตัวเพื่อผ่านหลักสูตรการศึกษา

Gened คือ วิชาที่ไม่ใช่วิชาบังคับ แต่กำหนดให้นิสิตต้องลงตามหน่วยกิตที่กำหนดไว้ จึงจะผ่านหลักสูตรการศึกษา

Approve คือ วิชาประจำภาควิชาที่ไม่ใช่วิชาบังคับ แต่กำหนดให้นิสิตต้องลงตามหน่วยกิต ที่กำหนดไว้ จึงจะผ่านหลักสูตรการศึกษา

Section คือ ตอนเรียนของวิชาบังคับ Gened และ Approve ซึ่งนิสิตสามารถเลือกลงได้ในระบบลงทะเบียนเรียน

4.2. ปัญหา

1. ตารางเรียนจะขึ้นหลังจากที่ลงทะเบียนสำเร็จแล้ว
2. นิสิตไม่สามารถค้นหาวัน และ เวลาเรียนของแต่ละรายวิชาได้โดยง่าย
3. ต้องใช้ Application อื่นช่วยในการวางแผนตารางเรียน
4. ตารางแบบเก่าไม่สามารถปรับแต่ง หรือ นำไปใช้ประโยชน์อื่นได้

4.3. แนวทางการแก้ไขปัญหาในข้อ

1. จัดทำระบบให้ผู้ใช้สามารถใช้งาน Application ได้ทั้งก่อนและหลังการลงทะเบียน
2. จัดทำส่วนติดต่อผู้ใช้ให้ใช้งานได้ง่าย
3. ระบบมีฟีเจอร์ในการ generate ตารางรวมถึงการให้คำแนะนำที่เหมาะสมในการลงทะเบียนเรียน

4. ระบบมีฟีเจอร์ในการส่งออก ตารางหลังจาก generate แล้ว รวมถึงความสามารถในการปรับแต่งตารางเรียน

5. ระบบในปัจจุบัน (Existing system)

การจัดตารางเรียนแบบเดิมนั้นมีขั้นตอนคือ ต้องแจกแจงวิชาบังคับของภาค การศึกษานั้นๆ ว่ามีเวลาเรียนใดบาง section ใดบ้าง จากนั้นจึงนำไปจัดตารางเรียน คร่าวๆ เลือก section และวิชาให้ไม่ตรงกัน แล้วจึงหาวิชาเลือก ที่ตนอยากเรียนและ เวลาเรียนไม่เหลื่อมกับวิชาบังคับ หรือบางที่อาจจะลองเปลี่ยน section ของวิชาบังคับ เป็นเวลาอื่นให้ตรงกับ วิชาเลือกที่ตนจะลง นอกจากนี้บางวิชาต้องการลง section หรือ ลงวิชาเลือกเดียวกับเพื่อน ก็จะต้องมีการปรับตารางเรียนกันใหม่อีกที จากความยุ่งยากในการจัดตารางเรียนในไม่มีเวลาเรียนเหลื่อมกัน ยังไม่รวมไปถึงเวลาที่ใช้ในการหา วิชาเลือก (Gened, Approve) ที่ตนพอใจ และเวลาไม่ชนกับที่ต้องเรียน

6. ความคาดหวังของระบบหลังพัฒนา (To-be system)

ผู้ใช้อกรอกข้อมูลคณะและสาขาเพื่อแสดงรายชื่อวิชาที่ตนต้องเรียนในเทอมนี้ และ เลือก section ที่ต้องการ จากนั้น application จะแสดงข้อมูลตารางเรียนจาก วิชาและ section ที่เลือกนั้น และผู้ใช้สามารถจัดการวิชาเรียนของตนได้ ไม่ว่าจะเป็นเพิ่ม ลด ย้าย section รวมไปถึงสามารถค้นหาวิชา และเพิ่มรายวิชานั้นเข้าไปในตารางสอนได้ โดย เลือกจาก เวลาเรียนที่ต้องการ ประเภทและหมวดหมู่ของวิชา หรือจะเจาะจงรายวิชา นั้นๆก็สามารถทำได้ โดยถ้าหากมีการเลือกวิชาที่มีเวลาเรียนเหลื่อมกันจะต้องมีการ แสดงให้เห็นได้อย่างชัดเจนว่า มีเวลาเหลื่อมกัน

โดยการเข้าใช้ application ได้นั้น ผู้ใช้ต้องลงทะเบียนเข้าสู่ระบบผ่าน facebook, google หรือ email password โดยผู้ใช้สามารถเข้าสู่ระบบเพื่อมาดูตาราง เรียนของตนจากที่ไหน เมื่อไรก็ได้ นอกจากนี้ยังสามารถส่งออกข้อมูลเข้าสู่ application จัดตารางเวลาอื่นๆได้ เช่น google calendar, apple calendar ได้

โดยผู้ใช้งานจะสามารถใช้ระบบนี้ได้ เพื่อทำการแสดง และจัดการตารางเรียนได้ตลอดเวลา รวมไปถึงช่วงก่อนจะทำการลงทะเบียนในระบบของจุฬาลงกรณ์โดยตรง

7. วัตถุประสงค์ (Objective)

เพื่อออกแบบและสร้างเครื่องมือในการจัดการตารางเรียน (Chula Course Table) ที่มีประสิทธิภาพให้แก่ นิสิตทุกคนภายในจุฬาลงกรณ์มหาวิทยาลัย เพิ่มความสะดวกสบายและความแม่นยำของข้อมูล รวมไปถึงลดความ ซ้ำซ้อน ทำให้ลดระยะเวลาที่ต้องเสียไปกับการจัดการตารางเรียนของนิสิต ทั้งก่อนลงทะเบียนและหลังลงทะเบียนเรียน อีกทั้งยังช่วยลดนิติตดการเข้าใจผิดในข้อมูลจากสำนักทะเบียนจุฬาลงกรณ์มหาวิทยาลัย อีกด้วย

8. ข้อจำกัดของระบบ (System development constraint)

8.1. ผู้ใช้ต้องสามารถเชื่อมต่อกับ Internet ได้ในขณะที่ใช้งานระบบ เนื่องจากตัวระบบอยู่ในรูปแบบของ Web Application

8.2. ผู้ใช้ต้องใช้เบราว์เซอร์ที่ทันสมัย เพื่อให้สามารถรองรับการทำงานของ script บน Web Application ของระบบ

8.3. ระบบจำเป็นต้องได้รับข้อมูลบางส่วนจากผู้ใช้งาน เพื่อใช้ในการประมวลผลข้อมูลเพื่อให้การแสดงผลเป็นไปได้อย่างถูกต้อง และแม่นยำ

8.4. ระบบอาจไม่สามารถรองรับการเข้าใช้งานจากผู้ใช้งานจำนวนมากภายในช่วงระยะเวลาเดียวกัน ส่งผลให้ผู้ใช้ไม่สามารถเข้าถึงบริการที่สมบูรณ์ของระบบได้

9. ความต้องการของระบบ (Requirement)

9.1. ความต้องการเชิงฟังก์ชันงาน (Functional Requirement)

9.1.1. การแสดงข้อมูลตารางเรียนของแต่ละบุคคล

- 9.1.2. การเพิ่ม ลด และย้าย section วิชาในตารางเรียน
- 9.1.3. การดึงข้อมูลวิชาบังคับจากระบบตาม ปีการศึกษา, ภาคเรียน, คณะ และสาขาวิชา
- 9.1.4. การค้นหาวิชาเลือก gened ตามเวลา และประเภท
- 9.1.5. การส่งข้อมูลออก เพื่อเชื่อมต่อกับ application อื่น

9.2. ความต้องการเชิงคุณภาพ (Non-functional Requirement)

9.2.1. Implementation requirements

- 9.2.1.1. ระบบจะสามารถทำงานได้ในทุกแพลตฟอร์มที่มี Web Browser ดังนี้: Google Chrome, Mozilla Firefox, Apple Safari และ Microsoft Edge
- 9.2.1.2. ระบบจะสามารถเข้าได้จาก Smartphone ที่เป็น iPhone และ iPad ที่มีระบบปฏิบัติการ iOS เวอร์ชัน 9 ขึ้นไป และ android ที่มีระบบปฏิบัติการ 5.0 ขึ้นไป
- 9.2.1.3. ระบบจะใช้ฐานข้อมูล MySQL ในการพัฒนาฐานข้อมูลของระบบ

10. ขอบเขตของระบบ (Scope of work)

Chula course table จัดทำขึ้นเพื่อรองรับการจัดตารางเรียนของนิสิตจุฬาลงกรณ์มหาวิทยาลัย

10.1. ระบบแสดงข้อมูลวิชาเรียนของแต่ละบุคคล

- 10.1.1. ผู้ใช้สามารถ เข้าสู่ระบบ เพื่อเข้าถึงข้อมูลวิชาของตนได้ โดยผ่านช่องทาง facebook, google และ e-mail password
- 10.1.2. ผู้ใช้สามารถเลือกวิชาบังคับที่ตนจะต้องเรียนในเทอมนี้ได้ โดยสามารถเลือกแบบรายวิชา หรือแบบกลุ่มตามข้อมูลภาค และสาขาของตนได้

10.1.3. ผู้ใช้สามารถมองเห็นรายการของวิชาที่ตนเรียนได้ โดยแต่ละวิชา
จะแสดงข้อมูลดังนี้

10.1.3.1. รหัสวิชา

10.1.3.2. ชื่อวิชา

10.1.3.3. ช่วงเวลาเริ่มถึงจบ

10.1.3.4. Section

10.1.4. ผู้ใช้สามารถเห็นวิชาที่เหลื่อมซ้อนกันของวิชาที่ตนจัดไว้ได้

10.1.5. ผู้ใช้สามารถเพิ่ม, ลบวิชาเรียนที่จัดไว้ได้

10.1.6. ผู้ใช้สามารถค้นหาวิชาตาม รหัสวิชา, ชื่อวิชา, เวลาเรียน, หรือ
ประเภทวิชาได้ โดยประเภทวิชาได้แก่

10.1.6.1. วิชาบังคับ

10.1.6.2. วิชา Gened

10.1.6.2.1. หมวดวิทยาศาสตร์ และ คณิตศาสตร์

10.1.6.2.2. หมวดสหศาสตร์

10.1.6.2.3. หมวดมนุษยศาสตร์

10.1.6.2.4. หมวดสังคมศาสตร์

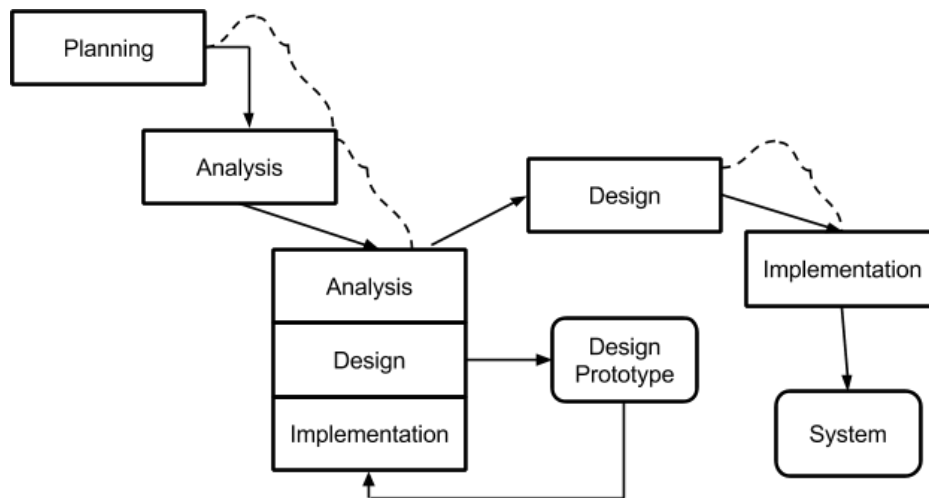
10.1.6.3. วิชา Approve ตามแต่ละคณะและสาขาได้

10.1.7. ผู้ใช้สามารถลองเปลี่ยน section ของวิชา ของตนเองได้

11. Software Development Life Cycle

Throw-Away Prototyping, เลือกใช้ model นี้เพราะ เป็นแนวทางการพัฒนา
ที่ตอบโจทย์กับ Business requirements มากที่สุด โดยทำการ Analyse และ Design
เพื่อสร้าง designed prototype แล้วจึงนำมาสอบถามความพึงพอใจ กับผู้ใช้งาน แล้ว
วนไปทำการ Design และ Implement ใหม่ไปเรื่อยๆ จนเมื่อผู้ใจพึงพอใจก็คือนำ
design นั้นมา implement จริง

โดย model จะช่วยในการพัฒนาระบบไปพร้อมกับรับ feedback จากผู้ใช้ได้ แม้ว่าจะยังไม่เริ่ม implement จริง ทำให้ได้ prototype ที่พึงพอใจกับผู้ใช้มากที่สุด



12. Gantt Chart

12.1. วางแผนโครงการ

- วิเคราะห์ปัญหาที่ส่งผลกระทบต่อผู้ใช้ในระบบเดิม
- วางแผนการทำงาน
- พิจารณาความต้องการของระบบ
- นำเสนอความต้องการของระบบ

12.2. วิเคราะห์ความต้องการระบบ

- รวบรวมความต้องการของผู้ใช้
- สืบหาข้อมูลในระบบการค้นหาวินิจฉัยของจุฬาลงกรณ์
- วิเคราะห์ความเป็นไปได้ และความเสี่ยงของ Project
- มอบหมายหน้าที่ และงานแก่คนในทีม
- จัดทำข้อเสนอโครงการ
- นำเสนอโครงการ

12.3. วิเคราะห์โครงการระยะที่หนึ่ง

- ศึกษาและวิเคราะห์ Database ของวิชาในระบบทะเบียนจุฬา
- ศึกษาและวิเคราะห์การดึงข้อมูล และวิธีการดึงข้อมูล

- ศึกษาและวิเคราะห์ Website สำหรับแสดงผลในระยที่ 1

12.4. ออกแบบระยที่หนึ่ง

- ออกแบบ Prototype ตัวแรก
- ออกแบบ Logical Layer
- ออกแบบ Data Layer
- ออกแบบ Interface Layer (UI & UX)
- จัดทำและนำเสนอ Prototype ตัวแรก

12.5. วิเคราะห์โครงการระยที่สอง

- ศึกษาและวิเคราะห์ Database ของวิชาในระบบทะเบียนจุฬา
- ศึกษาและวิเคราะห์การดึงข้อมูล และวิธีการดึงข้อมูล
- ศึกษาและวิเคราะห์ Website สำหรับแสดงผลในระยที่ 2

12.6. ออกแบบระยที่สอง

- ออกแบบ Prototype ตัวที่ 2
- ออกแบบ Logical Layer
- ออกแบบ Data Layer
- ออกแบบ Interface Layer (UI & UX)
- จัดทำและนำเสนอ Prototype ตัวที่ 2

12.7. วิเคราะห์โครงการระยที่สาม

- ศึกษาและวิเคราะห์ Database ของวิชาในระบบทะเบียนจุฬา
- ศึกษาและวิเคราะห์การดึงข้อมูล และวิธีการดึงข้อมูล
- ศึกษาและวิเคราะห์ Website สำหรับแสดงผลในระยที่ 3

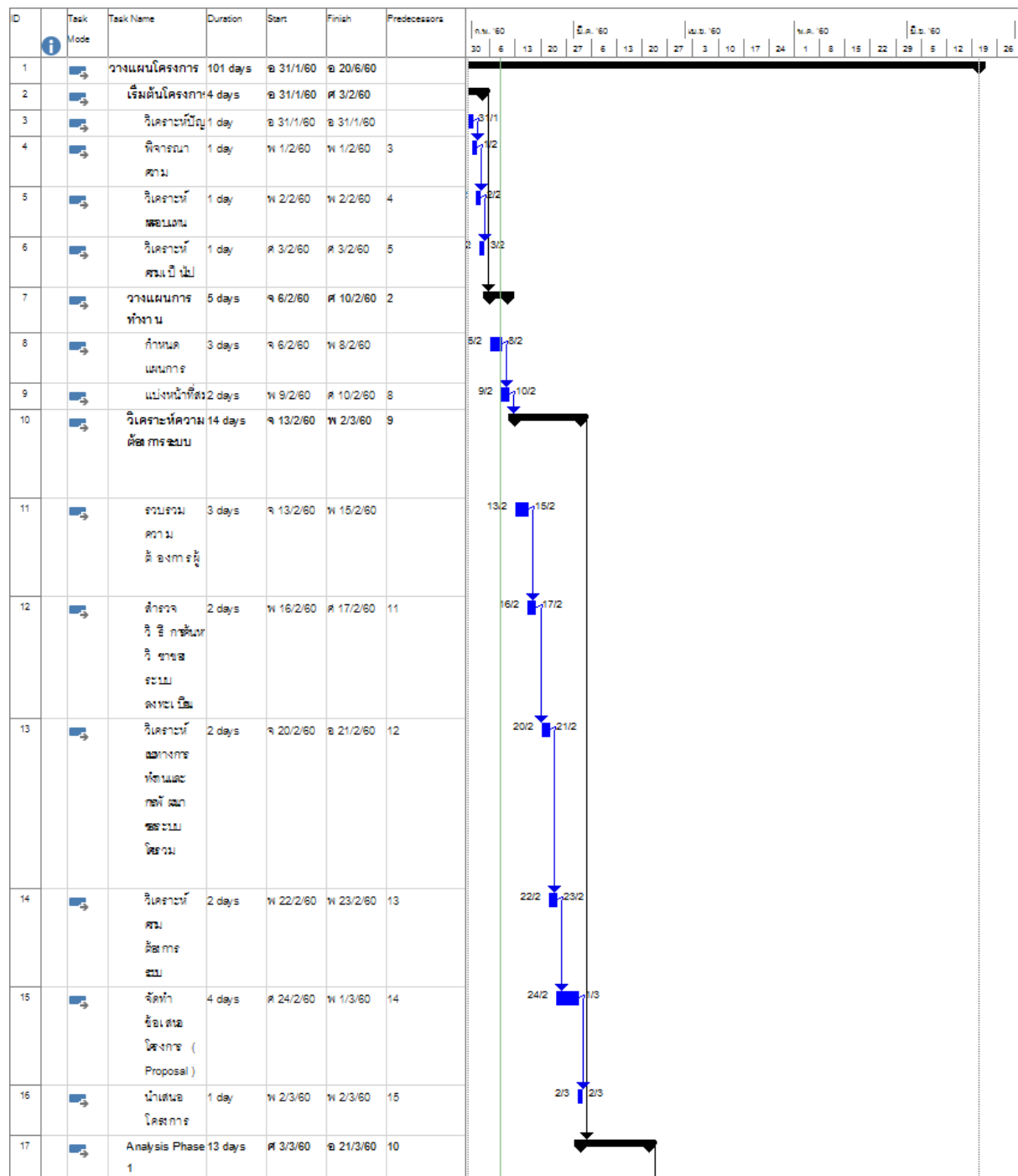
12.8. ออกแบบระยที่สาม


- ออกแบบ Prototype ตัวที่ 3
- ออกแบบ Logical Layer
- ออกแบบ Data Layer
- ออกแบบ Interface Layer (UI & UX)

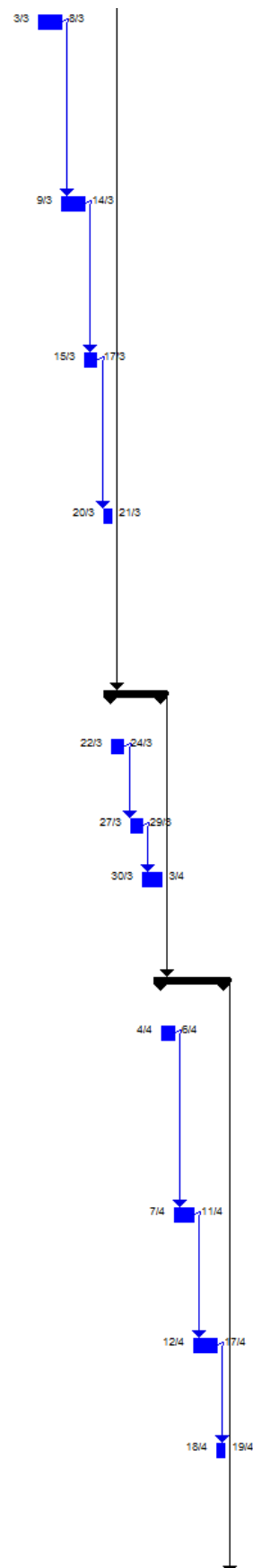
- จัดทำและนำเสนอ Prototype ตัวที่ 3

12.9. Implementation

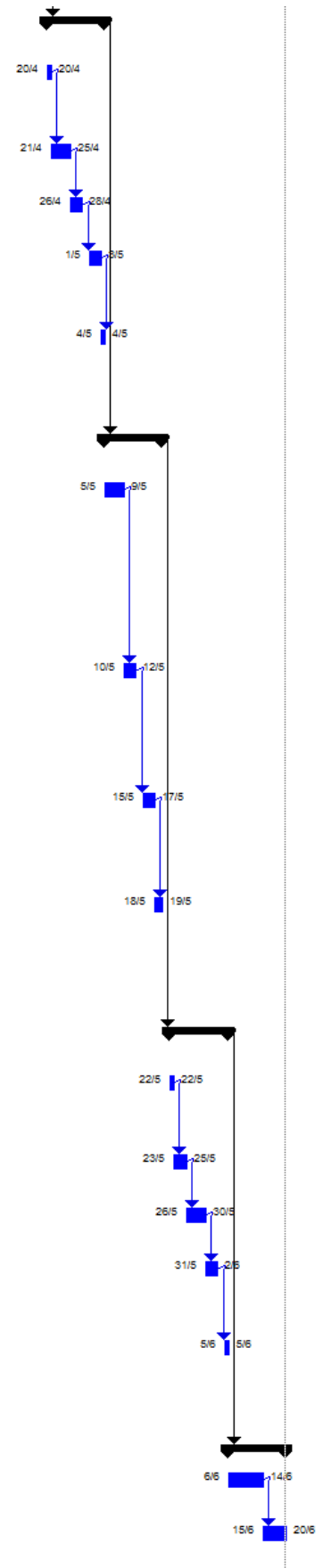
- สร้าง application
- final report



18		วิเคราะห์ เนื้อหา Database ของวิชาใน ระบบ ทะเบียน จุฬาลงกรณ์	4 days	ศ 3/3/60	พ 8/3/60	
19		ศึกษาและ วิเคราะห์ กรณี ข้อมูล และ วิธี การตั้ง ข้อมูล	4 days	พ 9/3/60	อ 14/3/60	18
20		วิเคราะห์ เนื้อหา เว็บไซต์ สำหรับ สอน ออนไลน์ ตอนที่ 1	3 days	พ 15/3/60	ศ 17/3/60	19
21		วิเคราะห์ เนื้อหา กรณี ข้อมูล และ วิธี การ ตอนที่ 1	2 days	จ 20/3/60	อ 21/3/60	20
22		ออกแบบระบบ ตอนที่ 1	9 days	พ 22/3/60	จ 3/4/60	17
23		ออกแบบ Logical Layer	3 days	พ 22/3/60	ศ 24/3/60	
24		ออกแบบ Data Layer	3 days	จ 27/3/60	พ 29/3/60	23
25		ออกแบบ Interface Layer (UI & UX)	3 days	พ 30/3/60	จ 3/4/60	24
26		Analysis Phase 2	12 days	อ 4/4/60	พ 19/4/60	22
27		วิเคราะห์ เนื้อหา Database ของวิชาใน ระบบ ทะเบียน จุฬาลงกรณ์	3 days	อ 4/4/60	พ 6/4/60	
28		ศึกษาและ วิเคราะห์ กรณี ข้อมูล และ วิธี การตั้ง	3 days	ศ 7/4/60	อ 11/4/60	27
29		วิเคราะห์ เนื้อหา เว็บไซต์ สำหรับ	4 days	พ 12/4/60	จ 17/4/60	28
30		วิเคราะห์ เนื้อหา กรณี ข้อมูล และ วิธี การ	2 days	อ 18/4/60	พ 19/4/60	29



31		ออกแบบระยะที่ 2	11 days	พ 20/4/60	พ 4/5/60	26
32		ออกแบบ Prototype ตัวที่ 2	1 day	พ 20/4/60	พ 20/4/60	
33		ออกแบบ Logical	3 days	ศ 21/4/60	อ 25/4/60	32
34		ออกแบบ Data Layer	3 days	พ 26/4/60	ศ 28/4/60	33
35		ออกแบบ Interface Layer (UI	3 days	จ 1/5/60	พ 3/5/60	34
36		จัดทำและนำเสนอ Prototype ตัวที่ 2	1 day	พ 4/5/60	พ 4/5/60	35
37		Analysis Phase 3	11 days	ศ 5/5/60	ศ 19/5/60	31
38		วิเคราะห์และออกแบบ Database ของวิชาใน ระบบทะเบียน จุฬาลงกรณ์	3 days	ศ 5/5/60	อ 9/5/60	
39		ศึกษาและวิเคราะห์ การดึง ข้อมูล และ การดึง	3 days	พ 10/5/60	ศ 12/5/60	38
40		วิเคราะห์และออกแบบ เว็บไซต์	3 days	จ 15/5/60	พ 17/5/60	39
41		วิเคราะห์และออกแบบ การดึงข้อมูล และ การดึง	2 days	พ 18/5/60	ศ 19/5/60	40
42		ออกแบบระยะที่ 3	11 days	จ 22/5/60	จ 5/6/60	37
43		ออกแบบ Prototype ตัวที่ 3	1 day	จ 22/5/60	จ 22/5/60	
44		ออกแบบ Logical	3 days	อ 23/5/60	พ 25/5/60	43
45		ออกแบบ Data Layer	3 days	ศ 26/5/60	อ 30/5/60	44
46		ออกแบบ Interface Layer (UI	3 days	พ 31/5/60	ศ 2/6/60	45
47		จัดทำและนำเสนอ Prototype ตัวที่ 3	1 day	จ 5/6/60	จ 5/6/60	46
48		Implementation	11 days	อ 6/6/60	อ 20/6/60	42
49		สร้าง Application	7 days	อ 6/6/60	พ 14/6/60	
50		จัดทำ Final report	4 days	พ 15/6/60	อ 20/6/60	49



13. ผู้รับผิดชอบโครงการ (Team organization and responsibility)

13.1. Project Manager	ณัฐภัทร กิระวิทยา
13.2. System Analysis	บรรณ อุทัยธีรตัน
13.3. Business Analysis	ชนะ กิจกุลอนันต์เอก
13.4. Technical Lead	นัทธ์ ศรีเรือนทอง
13.5. Programmer	ดนุภัทร คำนวนสินธุ์
13.6. External Coordinator	ณัฐสิทธิ์ มหากุลศิริกุล

14. ประโยชน์ที่คาดว่าจะได้รับ (Benefit)

- 14.1. เพิ่มประสิทธิภาพในการจัดตารางการเรียนการสอน
- 14.2. ลดระยะเวลาที่ใช้ในการวางแผนตารางการเรียน
- 14.3. ลดโอกาสความผิดพลาดที่จัดตารางการสอนออกมาแล้วมีเวลาเรียนทับกัน
- 14.4. เพิ่มประสิทธิภาพในการหาวิชาเรียนมาเพิ่มในช่วงเวลาว่างได้อย่าง

เหมาะสม

15. ดรรชนีชี้วัดความสำเร็จของโครงการ

15.1. เชิงปริมาณ

15.1.1. มีผู้ใช้บริการระบบใช้บริการเพื่อการลงทะเบียนอย่างน้อย 60%
ภายในระยะเวลา 1 ปี

15.1.2. ผู้ใช้บริการ สามารถใช้เวลาในการจัดสรรตารางเรียนเพื่อเรียนได้
เร็วขึ้น 70%

15.2 เชิงคุณภาพ

15.2.1. ผู้ใช้บริการสามารถค้นหารายวิชาที่ตนต้องการหาพบ และถูกต้อง
ตรงกับระบบลงทะเบียนเรียน

15.2.2. ผู้ใช้บริการสามารถใช้ตารางเพื่อช่วยในการจัดเวลาเรียนได้อย่างคล่องแคล่วและไม่ติดขัด

15.2.3. ผู้ใช้บริการสามารถส่งคำร้องเพื่อ เพิ่มเติม แก้ไข หรือลบ รายวิชาที่ไม่ถูกต้องในระบบได้

15.2.4. ผู้ใช้บริการมีความพึงพอใจในการใช้บริการระบบ

16. ความเสี่ยงและแนวทางการจัดการความเสี่ยง

16.1 ผู้ใช้บริการการลงทะเบียน ไม่ทราบถึงการมีอยู่ของระบบ

- โอกาสที่จะเกิดความเสี่ยง : สูง
- ผลกระทบที่อาจเกิดขึ้นได้ : ผู้ใช้บริการผ่านระบบลงทะเบียนปกติ ไม่ได้รับประโยชน์จากการจัดสรรตารางเรียนที่มีประสิทธิภาพมากขึ้น
- วิธีจัดการความเสี่ยง : ประชาสัมพันธ์ผ่านระบบสังคมออนไลน์ และการติดป้ายประชาสัมพันธ์

16.2 ระบบไม่มีความเสถียรหรือไม่สามารถตอบสนองต่อความต้องการของผู้ใช้บริการระบบเมื่อมีการ

เข้าใช้งานพร้อมกันปริมาณมาก

- โอกาสที่จะเกิดความเสี่ยง : ปานกลาง
- ผลกระทบที่อาจเกิดขึ้นได้ : ผู้ใช้บริการเสียความมั่นใจในบริการ เกิดความไม่พึงพอใจ
- วิธีจัดการความเสี่ยง : มีการทดสอบระบบอย่างต่อเนื่องเพื่อการแก้ปัญหาได้อย่างทันการ มีการเลือกใช้เวป Hosting ที่มีประสิทธิภาพ

17. Appendix (result of feasibility study)

เอกสารอ้างอิง

[1] Alan Dennis, Barbara Harley Wixom และ David Tegarden. System Analysis and Design An Object-Oriented Approach with UML. 5th Edition. John Wiley and Sons, inc, 2015

[2] จุฬาลงกรณ์มหาวิทยาลัย สำนักงานทะเบียน สนท. [ออนไลน์] แหล่งที่มา: <https://www2.reg.chula.ac.th/>

[3] มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา KU-PreRegis [ออนไลน์] แหล่งที่มา: <https://zugarzeeker.github.io/ku-courses-frontend/>

ภาคผนวก ก : การศึกษาความเป็นไปได้ของโครงการ

ก.1. Technical Feasibility

ก.1.1. Functional Area

ด้านผู้ใช้งาน

นิสิตที่ใช้งานมีความคุ้นเคยกับ web application ทำให้สามารถใช้งานได้ อย่างครบถ้วนสมบูรณ์ตามจุดประสงค์ของการสร้างระบบ

ด้านผู้พัฒนาระบบ

ผู้พัฒนาระบบมีความเข้าใจถึงฟังก์ชันงานของระบบตามความต้องการของ ผู้ใช้ทำให้ออกแบบ web application ให้มีความสอดคล้องกับความต้องการ และ มีการใช้งานที่ง่าย และ สะดวกรวดเร็ว

ก.1.2. Technology

ระบบพัฒนาในรูปแบบของ web application ที่ทำงานร่วมกับระบบฐานข้อมูล จากสำนักทะเบียนจุฬาลงกรณ์มหาวิทยาลัยซึ่งผู้พัฒนาระบบยังไม่เคยพัฒนาระบบโดย ใช้เทคโนโลยีการเชื่อมต่อ ดังกล่าว

ก.1.3. Project size

เนื่องจากมีผู้พัฒนาระบบจำนวนทั้งสิ้น 6 คน และ ใช้เวลาในการพัฒนา ระบบ 4 เดือน จึงถือว่าโครงการมีขนาดเล็ก

ก.1.4. Compatibility

เนื่องจากระบบพัฒนาในรูปแบบ web application ที่สามารถรองรับได้ในอุปกรณ์ที่เชื่อมต่ออินเทอร์เน็ตส่วนใหญ่ ทำให้ระบบมีความเข้ากันได้กับเทคโนโลยีก่อนหน้าสูง

ก.2. Organizational Feasibility

ก.2.1. นิสิตนักศึกษา มีความเป็นไปได้สูงที่นิสิตจะใช้งานระบบ เนื่องจากปัจจุบัน การจะจัดเวลาเรียนของตนต้องทำบนกระดาษหรือโปรแกรม เอกเซลบางตัวและต้องคอยจดหรือสลับหน้า เพื่อดูรหัสวิชาและเวลาเรียนแต่ตัวระบบทำให้ตัวนิสิตทำการค้นหาวิชาและเพิ่มลงบนตารางเรียนได้ทันทีถือเป็นการเพิ่มความสะดวกให้นิสิต

ก.2.2. สำนักทะเบียน มีความเป็นไปได้สูงที่จะใช้งานระบบ คือการให้ข้อมูลวิชาสู่ระบบ เพื่อให้ให้นิสิตได้เวลาเรียนที่พอใจ ลดภาระของระบบในเวลาที่นิสิตต้องแย่งกันเพื่อเอาวิชา รวมไปถึงเป็นการทำให้ระบบใช้งานได้ดีขึ้น ส่งผลให้นิสิตพอใจกับระบบลงทะเบียนมากขึ้น

ก.3. Economic Feasibility

ก.3.1. Benefits

ก.3.1.1. Tangible benefits

โครงการนี้ไม่ได้รับผลตอบแทนในรูปของรายได้ จึงไม่มี Tangible benefits

ก.3.1.2. Intangible benefits

ช่วยอำนวยความสะดวกและเพิ่มประสิทธิภาพในการจัดตารางสอนให้แก่ นิสิต

ก.3.2. Development Costs

ก.3.2.1. ค่าใช้จ่ายในการใช้งานซอฟต์แวร์ที่นำมาใช้ในการพัฒนาระบบ เป็นเงิน 7,500 บาท

ก.3.2.2. ค่าใช้จ่ายในการว่าจ้างผู้พัฒนาซอฟต์แวร์ เดือนละ 120,000 บาทเป็นระยะเวลา 3 เดือน เป็นเงิน 360,000 บาท

ก.3.3. Operational Costs

ก.3.3.1. ค่าใช้จ่ายในการว่าจ้างพนักงานดูแลระบบ เดือนละ 32,000 บาท เป็นเงิน 384,000 บาทต่อปี

ก.3.3.2. ค่าใช้จ่ายในการเช่าเซิร์ฟเวอร์จากผู้ให้บริการ เดือนละ 273.083 บาท หรือ ปีละ 3,277 บาท

ภาคผนวก ข: Project Effort Estimation: Use-case Point

ข.1. Unadjusted Use-Case Weighting Tables

Use-Case Classification	NO. of Transactions	Weight	Number	Result
Simple	1 to 3 transactions	5		
Average	4 to 7 transactions	10		
Complex	8 or more transactions	15		

Unadjusted Use-Case Weight Total (UUCW) = Sum of Result =

ข.2. Unadjusted Actor Weighting Tables

Actor Classification	Type of Actor	Weight	Number	Result
Simple	External system that must interact with the system using a well-defined API.	1		
Average	External system that must interact with the system using standard communication protocols. (e.g. TCP/IP, FTP, HTTP, database)	2		
Complex	Human actor using a GUI application interface.	3		

Unadjusted Actor Weight Total (UAW) = Sum of Result =

ข.3. Technical Complexity Factor (TCF)

Factor	Description	Weight	Assigned Value	Result
T1	Distributed system	2.0		

T2	Response time/performance objective	1.0		
T3	End-user efficiency	1.0		
T4	internal processing complexity	1.0		
T5	Code reusability	1.0		
T6	Easy to install	0.5		
T7	Easy to use	0.5		
T8	Portability to other platforms	2.0		
T9	System maintenance	1.0		
T10	Concurrent/parallel processing	1.0		
T11	Security features	1.0		
T12	Access for third parties	1.0		
T13	End user training	1.0		

Technical Factor Value (TFactor) =

Technical Complexity Factor (TCF) = Sum of Result =

ข.4. Environmental Complexity Factor (ECF)

Factor	Description	Weight	Assigned Value	Result
E1	Familiarity with system development process being use	1.5		
E2	Application experience	0.5		
E3	Object-oriented experience	1.0		
E4	Lead analyst capability	0.5		
E5	Motivation	1.0		
E6	Requirements stability	2.0		
E7	Part-time staff	-1.0		
E8	Difficult of programming language	-1.0		

Environmental Factor Value (EFactor) =

Environmental Complexity Factor (ECF) = Sum of Result =

Adjusted use case points (UCP) = UUCP * TCF * EF =

ภาคผนวก ค: มาตรฐานโครงการ

ค.1. มาตรฐานการเขียนเอกสาร

ค.1.1. การทำเอกสารทางทีมเลือกใช้ Google docs ซึ่งเป็นแอปพลิเคชันที่สามารถเขียนเอกสารได้แบบออนไลน์ สามารถทำงานพร้อมๆกันได้ และเมื่อคนในทีมแก้ไขอะไรก็สามารถเห็นได้ทันที

ค.1.2. จากนั้นจะตรวจสอบความถูกต้องเอกสาร และexportออกไปเพื่อทำงานในโปรแกรม Microsoft Word เพื่อทำการจัดหน้า และบันทึกเอกสารให้อยู่ในรูปแบบ .docx และ .pdf

ค.1.3. แบบอักษรที่ใช้คือ TH Sarabun New โดบหัวข้อเป็นตัวหนา ขนาด 22 หน่วย และเนื้อหาทั่วไปมีขนาด 20 หน่วย อักษรปกติ และมีระยะเว้นย่อหน้า 1.27 เซนติเมตร

ค.1.3.1. การจัดรูปแบบลำดับของหัวข้อมีรูปแบบดังนี้

1. **หัวข้อ**
 - 1.1. หัวข้อย่อยระดับที่ 1
 - 1.1.1. หัวข้อย่อยระดับที่ 2
 - 1.1.1.1. หัวข้อย่อยระดับที่ 3
 เนื้อหา

ค.1.3.2. หากมีข้อความที่เป็นรหัสคำสั่ง (code) อยู่ในเนื้อหาจะต้องถูกจัดวางในกล่องข้อความสี่เหลี่ยมที่มีขนาดเส้นขอบ 1 หน่วย และใช้อักษร Consolas ขนาด 10 หน่วย

ค.2. มาตรฐานการเขียนโปรแกรม

สำหรับแต่ละโมดูล (module) ในโปรแกรม จะต้องมีการเขียนหมายเหตุ (comment) กำกับโมดูลนั้น เพื่อให้ผู้ใช้งานได้สะดวก สามารถถามผู้ที่เป็นคนสร้างโมดูลนั้นได้ รวมทั้งแก้ปัญหาได้ง่ายขึ้นเวลามีปัญหา

ค.3. มาตรฐานขั้นตอนการทำงาน

ค.3.1. จัดการประชุมเพื่อติดตามความคืบหน้าของโครงการกับสมาชิกในกลุ่มทุก
วันศุกร์ เวลา 16.00 น.-19.00 น.

ค.3.2. เอกสาร และข้อมูลทั้งหมดถูกเก็บไว้ที่ Google Drive

ค.3.3. ใช้ระบบรุ่น (Version Control System) แบบกระจายโดยใช้บริการของ
Github

ค.3.4. การแจ้งข่าวสารข้อมูลให้แก่สมาชิกในกลุ่มจะดำเนินการผ่านระบบ
สนทนาแบบกลุ่มใน Facebook Messenger

ค.3.5. ผู้พัฒนาสามารถนัดหมายประชุมเพิ่มได้ตามโอกาสที่เหมาะสม