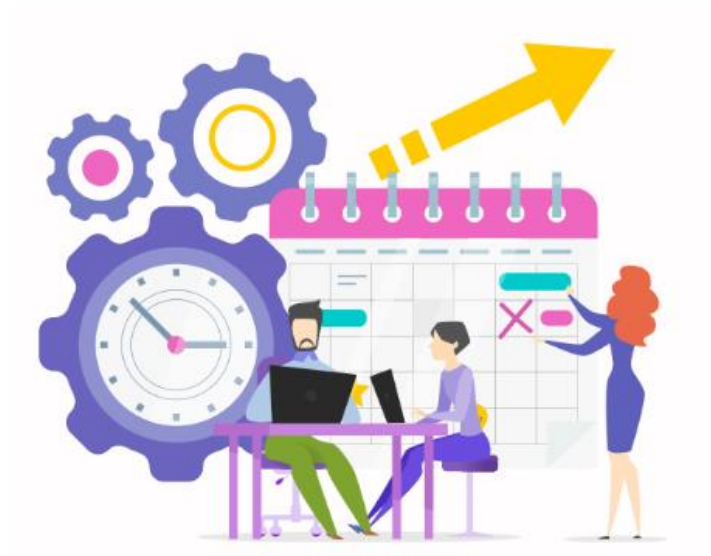




เว็บแอปบริหารเวลา จัดตารางเรียนตารางสอน
(Schedooo!)
CS 251 ระบบฐานข้อมูล 1



กลุ่ม 11

ยลวดี	แช่ต้ง	6509650070
พนพล	จุ่นเจิม	6509650146
กชพร	มีณรงค์	6509650195
กรกฤต	พงศ์ปัญญา	6509650203
จิรัฐญา	ทั้งจันทร์	6509650278
ปณณทัต	มณีนวษ์	6509650542
ปุณณภา	ตรีรัตน์ดิลกกุล	6509650559
ภัทรพล	ชาวฟ่อน	6509650625
ศิวพร	ลมสูงเนิน	6509650740

Section 650001
22/02/2567 Phase 3

สารบัญ

	หน้า
1. Problem Statement	3
2. Solution Requirement	4
3. Software Requirements Specification	4
4. System Limit	5
5. Conceptual Database Design	5
6. Functional Component Design	6
7. DBMS และ Developer Tools	8
8. ER-Diagram	9
9. Relational Schema	10
10. Data Dictionary	10
11. SQL Code	11
12. Use Case	17
13. Activity Diagram	19
14. User Interface	26

Problem Statement

★ ที่มาและความสำคัญ

ปัจจุบัน หลักสูตรการศึกษาในระดับมหาวิทยาลัยโดยเฉพาะ มหาวิทยาลัยธรรมศาสตร์ มีการเปลี่ยนแปลงและพัฒนาอยู่ตลอดเวลาเพื่อขัดเกลา นักศึกษาให้สามารถนำความรู้ไปต่อยอดทั้งในการประกอบอาชีพ และพัฒนา ประเทศชาติได้ (เกศินี วิฑูรชาติ, 2567) ซึ่งการเปลี่ยนแปลงหรือการพัฒนาหลักสูตรนั้น ส่งผลให้ทั้งนักศึกษาและคณาจารย์มีภาระหน้าที่เพิ่มมากขึ้น นอกจากนี้นักศึกษาหรือ อาจารย์บางท่านก็มียานอดิเรกหรือภาระหน้าที่อื่นๆ อีกมากมาย ความต้องการในการ จัดการเวลาและตารางการทำงานของแต่ละบุคคลในชีวิตประจำวันจึงมีเพิ่มมากขึ้น โดยเฉพาะอย่างยิ่งในช่วงเวลาที่มีกิจกรรมหลายอย่างต้องทำพร้อมกัน เช่น การทำงาน การศึกษา การฝึกอบรม หรือกิจกรรมส่วนตัวอื่นๆ การจัดตารางเรียนและ ตารางสอนจึงเป็นเรื่องที่สำคัญในการวางแผนชีวิตประจำวันของผู้คนที่ต้องการให้เวลา แต่ละช่วงถูกใช้ให้เกิดประโยชน์สูงสุดดังนั้นการมีเว็บแอปที่ช่วยในการจัดการและ วางแผนตารางเรียนหรือตารางสอนจึงมีความสำคัญอย่างมากในการช่วยให้ผู้ใช้สามารถ จัดการเวลาและกิจกรรมต่างๆ ได้อย่างมีประสิทธิภาพลดความเครียดในการ จัดการเวลาและเพิ่มประสิทธิภาพในการใช้ชีวิตอีกด้วย นอกจากนี้เว็บแอปที่ ทางคณะผู้จัดทำโครงการยังช่วยให้ผู้ใช้สามารถวางแผนเวลาได้เป็นระยะๆ เพื่อให้สามารถก้าวไปสู่ความสำเร็จในทุกด้านได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

★ วัตถุประสงค์

- เพื่อช่วยจัดการเวลาอย่างมีประสิทธิภาพ
- เพื่อเพิ่มประสิทธิภาพในการเรียนการสอน
- เพื่อเพิ่มความสะดวกสบายในการแสดงตารางเรียนตารางสอน
- เพื่อแก้ไขปัญหาการลืมนัด

★ ผลที่คาดว่าจะได้รับ

- ผู้ใช้สามารถจัดการเวลาได้อย่างมีประสิทธิภาพ
- ผู้ใช้จะมีเวลามากขึ้นสำหรับกิจกรรมที่ชอบ
- ตารางเรียนตารางสอนจะช่วยลดความสับสนในการจัดการกิจกรรมต่างๆ
- ช่วยลดความเครียดจากความไม่แน่ใจว่าควรทำอะไรก่อนหรือหลัง

Solution Requirement

1. การสร้างตารางเรียนและตารางสอน :
ผู้ใช้งานสามารถสร้างตารางเรียนหรือตารางสอนได้โดยมีการแสดงข้อมูลต่างๆ ซึ่งสามารถแก้ปัญหาการจัดการเวลาได้
2. การสร้าง task :
ผู้ใช้งานสามารถสร้าง task งานที่ต้องทำเมื่อสำเร็จก็สามารถ checklist ได้ โดยการสร้าง task จะป้องกันการสับสนในระหว่างดำเนินกิจกรรม ทำให้มีประสิทธิภาพในการใช้ชีวิตได้
3. การจัดการเวลา :
ผู้ใช้งานสามารถเพิ่ม ลบ แก้ไข หรือย้ายกิจกรรมในตารางเรียนหรือตารางสอน ทำให้เพิ่มความสะดวกในการจัดตารางเรียนหรือตารางสอน
4. การเตือน :
สามารถเตือนผู้ใช้เกี่ยวกับกิจกรรมที่กำลังจะเกิดขึ้นช่วยให้ผู้ใช้งานสามารถทำกิจกรรมหรือหน้าที่ต่างๆ ภายในระยะเวลาที่เหมาะสม
5. การตกแต่ง :
ผู้ใช้งานสามารถเปลี่ยนรูปแบบการแสดงผล การปรับแต่งสีรายวิชา เพื่อให้เหมาะสมกับความชอบของผู้ใช้ นอกจากนี้ยังลดความเครียด ในการจัดตารางเรียนหรือตารางสอนได้อีกด้วย
6. การใช้งานได้หลายแพลตฟอร์ม :
มีความสามารถในการใช้งานได้หลายแพลตฟอร์ม เพื่อให้ผู้ใช้งานเข้าถึง และใช้งานได้ทุกที่ทุกเวลา
7. ความปลอดภัยของข้อมูล :
สามารถรักษาความปลอดภัยของข้อมูลส่วนตัวของผู้ใช้ได้ ทั้งนี้เพื่อความสบายใจในการใช้เว็บแอปในการจัดตารางกิจกรรม และผู้ใช้งานสามารถมีอิสระในพื้นที่ส่วนตัวภายในเว็บแอปได้

Software Requirements Specification

★ Functional Requirement

- ผู้ใช้งานสามารถเรียกดู Task จากหน้าปฏิทินได้
- เมื่อผู้ใช้งานเข้าสู่ระบบจะสามารถแสดงแถบแจ้งเตือนของ Task ที่ต้องทำในวันนั้น ๆ ได้
- เมื่อผู้ใช้งานเพิ่ม แก้ไข หรือลบรายวิชา ข้อมูลนั้นจะถูกเพิ่ม แก้ไข หรือลบในฐานข้อมูลตามคำสั่งของผู้ใช้
- ผู้ใช้งานสามารถสร้าง หรือแก้ไขตารางเรียน/สอน โดยเลือกรายวิชา จากวิชาที่เพิ่มไว้ได้
- ผู้ใช้งานสามารถสร้าง แก้ไข หรือลบ Task ได้
- ผู้ใช้งานสามารถกดเสร็จสิ้น Task ที่ทำสำเร็จแล้วได้

★ Non-functional Requirement

- ระบบมีการทำงานได้ถูกต้อง ได้ผลลัพธ์ตามความต้องการของผู้ใช้
- ระบบมีความน่าเชื่อถือ สามารถทำงานได้อย่างถูกต้อง
- ระบบมีความปลอดภัย ข้อมูลของผู้ใช้จะไม่สูญหายหรือถูกแก้ไขเปลี่ยนแปลงโดยผู้อื่นได้
- เวลาตอบสนองของระบบมีความเหมาะสม ไม่ตอบสนองช้าจนเกินไป
- ระบบมี User Interface ที่ใช้งานได้ง่าย ผู้ใช้สามารถเข้าใจวิธีการใช้ได้ง่าย
- ระบบมี User Interface ที่มีความน่าใช้งาน ดึงดูดผู้ใช้

System Limit

- ผู้ใช้งานต้อง Register เข้าสู่ระบบสำเร็จก่อน จึงจะสามารถ Login ใช้งานได้
- ผู้ใช้ต้องเข้าถึงระบบผ่าน Internet

Conceptual Database Design

1. User Table (ตารางผู้ใช้) :

ตารางสำหรับจัดเก็บข้อมูลที่จำเป็นของผู้ใช้ประกอบด้วย

Username (Primary Key) : ชื่อผู้ใช้, Password : รหัสผ่าน

2. Course Table (ตารางรายวิชา) :

ตารางสำหรับจัดเก็บข้อมูลรายวิชาประกอบด้วย

CourseID (Primary Key): รหัสรายวิชา, CourseName : ชื่อรายวิชา, StartTime : เวลาเริ่มต้น, EndTime : เวลาสิ้นสุด, DayOfWeek: วันของสัปดาห์ (เช่น จันทร์, พุธ เป็นต้น), ProfessorName : ชื่ออาจารย์ผู้สอน, ColorCourse : สีรายวิชา

3. Task Table (ตารางกิจกรรม) :

ตารางสำหรับจัดเก็บข้อมูลกิจกรรมและงานต่าง ๆ ประกอบด้วย

TaskID(Primary Key) : รหัส Task ,NameTask : ชื่อ Task, DetailsTask : รายละเอียด Task, DeadlineTask: วันกำหนดการ, IsDone : สถานะความสำเร็จ(T/F)

Function Component Design

ส่วนยืนยันตัวตน

- Function : sign up (เป็นฟังก์ชันที่ใช้ในการสร้าง Account)
Input : ผู้ใช้ต้องสร้าง Username และ Password ของบัญชีขึ้นมา และเพิ่มข้อมูลส่วนตัว
Process : ระบบทำสร้างบัญชีขึ้นมาใหม่
Output : ระบบบันทึกข้อมูลว่ามีบัญชีนี้อยู่ในระบบแล้ว
- Function : login (เป็นฟังก์ชันที่ใช้เข้าสู่ระบบด้วย Account ที่มีอยู่แล้วในฐานข้อมูล)
Input : Username และ Password
Process : ตรวจสอบว่าชื่อผู้ใช้ และรหัสผ่านที่ป้อนถูกต้องหรือไม่
Output : ผลลัพธ์ว่าเข้าสู่ระบบสำเร็จหรือไม่
- Function : logout (เป็นฟังก์ชันที่ใช้ออกจากระบบ)
Input : ปุ่ม logout
Process : นำ User ออกจากระบบและกลับไปหน้า login
Output : ผลลัพธ์ว่าออกจากระบบสำเร็จหรือไม่

ส่วน Task

- Function : เพิ่ม task (เป็นฟังก์ชันที่ใช้เพิ่มกิจกรรม)
Input : ใส่ชื่อ task และใส่รายละเอียดของ task ที่ต้องการจะทำลงไป
Process : ระบบทำการสร้าง task ขึ้นมาพร้อมกับรายละเอียดของ task
Output : มี task ใหม่ถูกสร้างขึ้น และแสดงรายละเอียดของ task ด้วย
- Function : ลบ task (เป็นฟังก์ชันที่ใช้ลบกิจกรรม)
Input : ผู้ใช้เลือก task ที่ต้องการลบ
Process : ระบบทำการลบ task ที่ผู้ใช้เลือก
Output : task ที่ผู้ใช้เลือกจะถูกลบออกไปจากระบบ
- Function : แก้ไข task (เป็นฟังก์ชันที่ใช้แก้ไขกิจกรรม)
Input : ใส่ข้อความใหม่ เพื่อแก้ไขชื่อ task หรือข้อความรายละเอียดของ task อันเก่า
Process : ระบบทำบันทึกการแก้ไข task ที่ผู้ใช้เลือกตามข้อความ
Output : task ที่ผู้ใช้เลือกถูกแก้ไข

ส่วนรายวิชา

- Function : เพิ่มรายวิชา(เป็นฟังก์ชันที่ใช้เพิ่มรายวิชา)
Input : ใส่ชื่อ, รหัส และรายละเอียดของรายวิชาที่ต้องการจะทำการไป
Process : ระบบทำการสร้างรายวิชาขึ้นมาพร้อมกับรายละเอียดของรายวิชา
Output : รายวิชาใหม่ถูกสร้างขึ้น และแสดงรายละเอียดของรายวิชาด้วย
- Function : ลบรายวิชา(เป็นฟังก์ชันที่ใช้ลบรายวิชา)
Input : ผู้ใช้เลือกรายวิชาที่ต้องการลบ
Process : ระบบทำการลบรายวิชาที่ผู้ใช้เลือก
Output : รายวิชาที่ผู้ใช้เลือกจะถูกลบออกไปจากระบบ
- Function : แก้ไขรายวิชา(เป็นฟังก์ชันที่ใช้แก้ไขกิจกรรม)
Input : ใส่ข้อความใหม่เพื่อแก้ไขชื่อรายวิชา หรือข้อความรายละเอียดของรายวิชาอันเก่า
Process : ระบบทำบันทึกการแก้ไขรายวิชาที่ผู้ใช้เลือกตามข้อความ
Output : รายวิชาที่ผู้ใช้เลือกถูกแก้ไข

ส่วนปฏิทิน

- Function : ปฏิทิน(ขยาย)
Input : ผู้ใช้เลือกวันที่ต้องการ
Process : ระบบทำการดึงหน้าตาเป็นหน้าตา Task
Output : แสดง Task ในวันนั้นๆ

ส่วน Interface

- Function : หน้าหลัก(หน้าเริ่มต้น)
ทำหน้าที่เป็นหน้าเริ่มต้นและเป็นหน้าต่างเพื่อเข้าถึง function อื่น ๆ
- Function : ปฏิทิน
Input : ผู้ใช้คลิกปุ่มปฏิทิน
Process : ระบบทำการเปลี่ยนมุมมองเป็นตารางปฏิทิน
Output : ปฏิทินแสดงวันที่และกิจกรรม
- Function : ตารางเรียน/สอน
Input : ผู้ใช้คลิกปุ่มตารางเรียน/สอน
Process : ดึงข้อมูลรายวิชาที่มี นำมาประกอบเป็นตารางเรียน
Output : แสดงตารางเรียน/สอน

- Function : ตารางกิจกรรม
Input : ผู้ใช้คลิกปุ่มตารางกิจกรรม
Process : ดึงข้อมูลกิจกรรมที่มี นำมาประกอบเป็นตารางกิจกรรม
Output : แสดงตารางกิจกรรม
- Function : การแจ้งเตือน Task
Process : แจ้งเตือนในวันที่ต้องส่ง Task งานในวันนั้นๆ
Output : การแจ้งเตือน Task งานในวันนั้นๆ

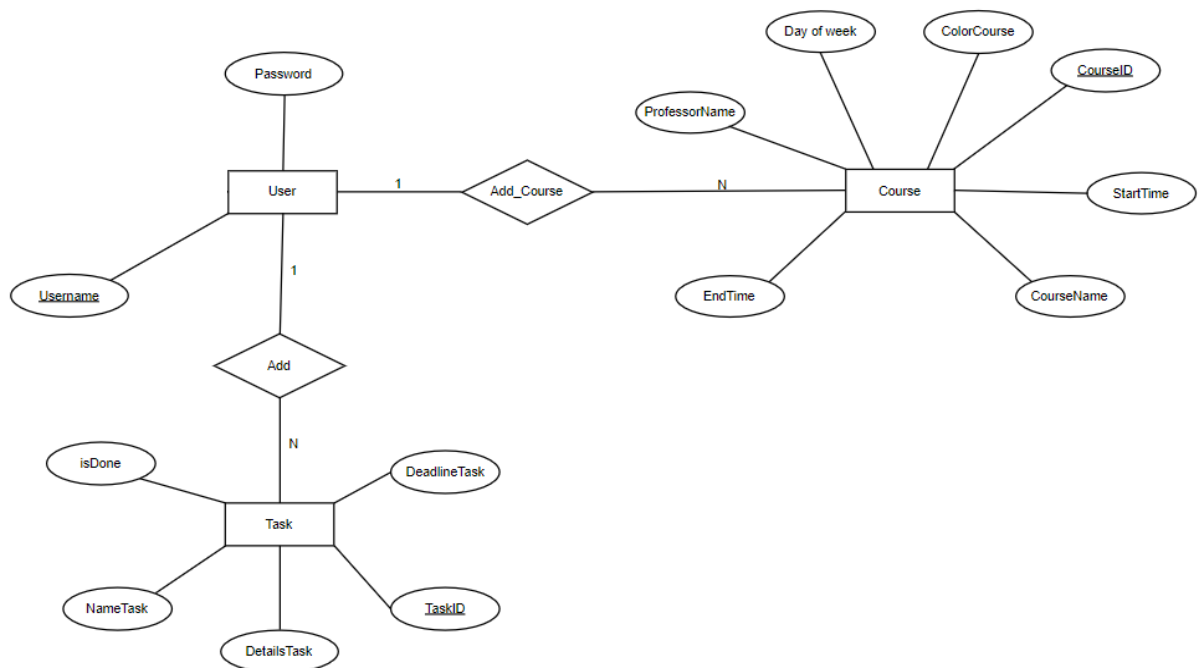
DBMS and Developer Tools

DBMS : SQL Server Management Studio (SSMS)

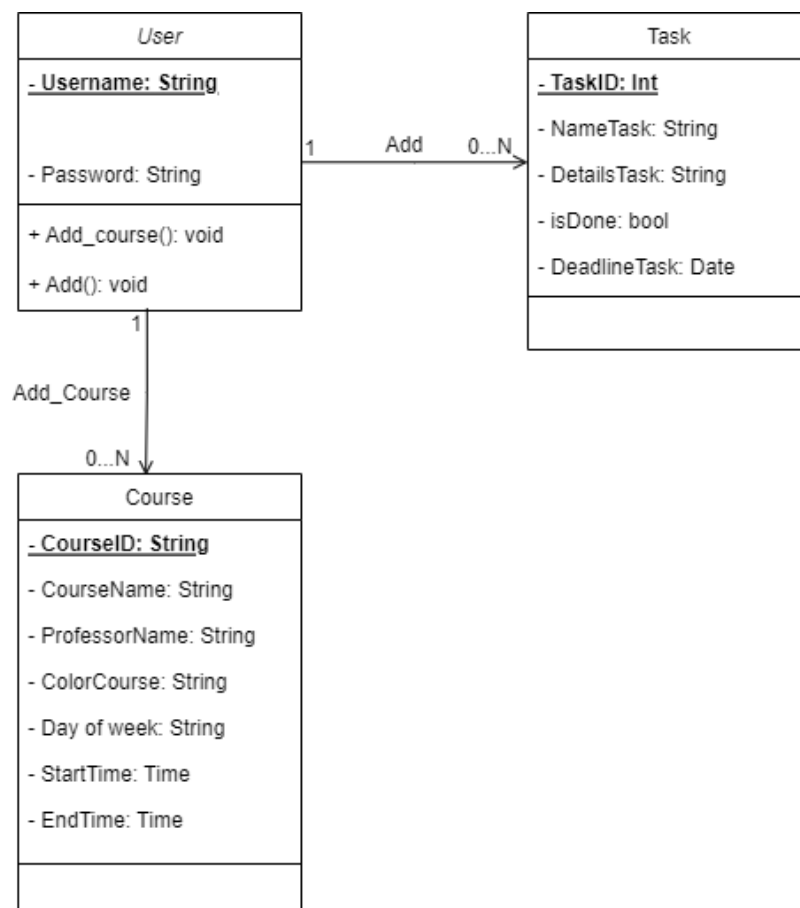
Developer Tools :

- Backend : IntelliJ IDEA Community Edition, Visual Studio Code
- Frontend : Sublime, Visual Studio Code
- Environment : Docker, Docker-compose
- Versioncontrol : GitHub
- Text Editor : Visual Studio Code
- Design : Canva
- Project Management : Trello

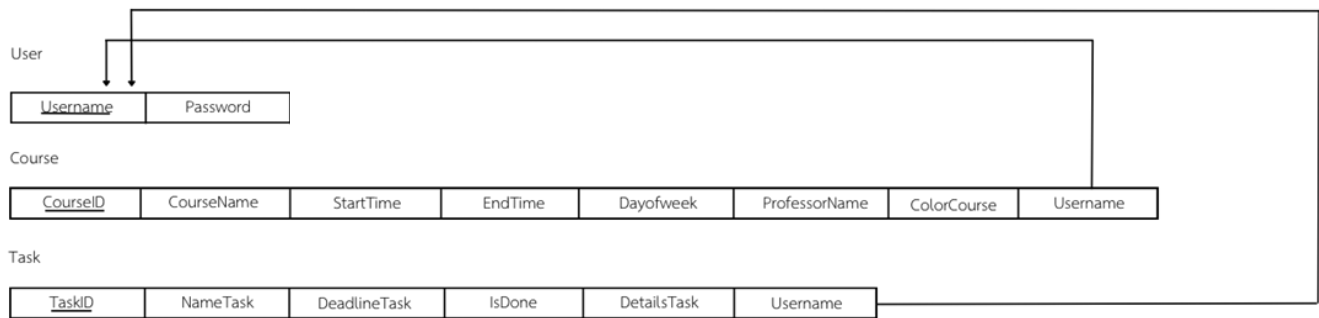
ER-Diagram



UML Diagram



Relational Schema



Data Dictionary

1. User

Attribute Name	Data Type	Data Format	Length	Description	Example
Username	NVARCHAR	-	20	ชื่อผู้ใช้	Panapol
Password	NVARCHAR	-	20	รหัสผ่าน	Panapol999

2. Course

Attribute Name	Data Type	Data Format	Length	Description	Example
CourseID	NVARCHAR	NNNNN	5	รหัสวิชา	CS251
CourseName	NVARCHAR	-	50	ชื่อวิชา	ระบบฐานข้อมูล 1
ProfessorName	NVARCHAR	-	50	ชื่อผู้สอน	อ. นุชจรินทร์
Dayofweek	CHAR	NNN	3	ตัวย่อวัน	ศ.
ColorCourse	NVARCHAR	-	10	สีวิชา	Blue
StartTime	TIME	NN:NN	4	เวลาเริ่มรายวิชา	09:30
EndTime	TIME	NN:NN	4	เวลาเลิกรายวิชา	12:30

3. Task

Attribute Name	Data Type	Data Format	Length	Description	Example
TaskID	INT	NNNN	4	รหัส Task	1
NameTask	NVARCHAR	-	20	ชื่อ Task	CS251 Phase 2
DetailsTask	NVARCHAR	-	100	รายละเอียด Task	ทำ Data Dictionary
isDone	BIT	X	1	งานเสร็จหรือยัง (เสร็จ = 1, ไม่เสร็จ = 0)	1
DeadlineTask	DATE	YYYY-MM-DD	8	วันกำหนดส่ง Task	2024-03-30

SQL Code

1. สร้าง Database สำหรับเก็บข้อมูล ชื่อ Schedooo!

```
CREATE DATABASE Schedooo!;
```

2. สร้างตาราง Task

2.1 สร้างตาราง ชื่อ User มีคอลัมน์ Username เป็น Key หลัก, Password และ Email

```
CREATE TABLE [User](  
    Username VARCHAR(20) PRIMARY KEY NOT NULL,  
    Password VARCHAR(20) NOT NULL,  
    Email VARCHAR(30)  
);
```

2.2 สร้างตาราง ชื่อ Task มีคอลัมน์ TaskID เป็น Key หลัก, Username เป็นทั้ง Key หลัก และ Key ที่อ้างอิงมาจากตาราง User, NameTask, DetailTask, isDone, และ DeadlineTask

```
CREATE TABLE Task(  
    TaskID INT NOT NULL,  
    Username VARCHAR(20) NOT NULL,  
    NameTask VARCHAR(50) NOT NULL,  
    DetailTask VARCHAR(200),  
    isDone BIT NOT NULL,  
    DeadlineTask DATE NOT NULL,  
    PRIMARY KEY (TaskID, Username),  
    FOREIGN KEY(Username)  
    REFERENCES [User](Username)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
);
```

และ เมื่อมีการอัปเดต และลบข้อมูลในตารางที่อ้างอิงมา (User) ข้อมูลที่เกี่ยวข้องในตาราง Task จะถูกเปลี่ยนแปลงด้วย

2.3 สร้างตาราง ชื่อ Course มีคอลัมน์ CourseID เป็น Key หลัก, Username เป็นทั้ง Key หลัก และ Key ที่อ้างอิงมาจากตาราง User, ColorCourse, CourseName, ProfessorName, DayOfWeek, StartTime, EndTime

```
CREATE TABLE Course(  
    CourseID VARCHAR(5) NOT NULL,  
    Username VARCHAR(20) NOT NULL,  
    ColorCourse VARCHAR(10) NOT NULL,  
    CourseName VARCHAR(50) NOT NULL,  
    ProfessorName VARCHAR(50) NOT NULL,  
    DayOfWeek CHAR(3) NOT NULL,  
    StartTime TIME(4) NOT NULL,  
    EndTime TIME(4) NOT NULL,  
    PRIMARY KEY (CourseID, Username),  
    FOREIGN KEY(Username)  
    REFERENCES [User](Username)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
);
```

และ เมื่อมีการอัปเดต และลบข้อมูลในตารางที่อ้างอิงมา (User) ข้อมูลที่เกี่ยวข้องในตาราง Task จะถูกเปลี่ยนแปลงด้วย

3. ฟังก์ชันของตาราง User

3.1 ลงทะเบียนใช้งาน

```
INSERT INTO User VALUES ('Username_enter_by_user', «- ชื่อผู้ใช้ของผู้ใช้  
'Password_enter_by_user'); «- รหัสผ่านของผู้ใช้
```

คำอธิบาย เป็นฟังก์ชัน เพิ่มชื่อผู้ใช้ และรหัสผ่านใหม่เข้าไปในตาราง User

3.2 Login ใช้งาน

```
SELECT Username
```

```
FROM [User]
```

```
WHERE Username = 'Username_enter_by_user' «- ชื่อผู้ใช้ของผู้ใช้
```

```
AND Password = 'Password_enter_by_user'; «- รหัสผ่านของผู้ใช้
```

คำอธิบาย เป็นฟังก์ชัน เรียกชื่อผู้ใช้ โดยเลือกชื่อผู้ใช้ที่มีอยู่ มาจากตาราง User โดยกำหนดให้ Username เท่ากับ ชื่อผู้ใช้ของผู้ใช้ที่กรอกมา และ Password เท่ากับ รหัสผ่านของผู้ใช้ที่กรอกมา

4. ฟังก์ชันของตาราง Task

4.1 เพิ่ม Task

```
INSERT INTO Task(TaskID, Username, NameTask, DetailTask,  
isDone, DeadlineTask)
```

```
VALUES ('TaskID', 'Username', 'NameTask', 'DetailTask', isDone, 'DeadlineTask')
```

คำอธิบาย เป็นฟังก์ชัน เพิ่มงานใหม่ ที่มีคอลัมน์ TaskID, Username, NameTask, DetailTask, isDone และ DeadlineTask ให้มีค่าตามที่ผู้ใช้เป็นคนกำหนด

4.2 หา Task ด้วย TaskID และ Username

```
SELECT TaskID, Username, NameTask, DetailTask, isDone,  
DeadlineTask
```

```
FROM Task
```

```
WHERE TaskID = 'TaskID' AND Username = 'Username';
```

คำอธิบาย เป็นฟังก์ชัน เรียกหางาน โดยเลือก TaskID, Username, NameTask, DetailTask, isDone และ DeadlineTask มาจากตาราง Task โดยกำหนดให้ TaskID เท่ากับ ไอดีงานที่ผู้ใช้เลือก และ Username เท่ากับ ชื่องานที่ผู้ใช้เลือก

4.3 หาทุก Task ด้วย Username

```
SELECT TaskID, Username, NameTask, DetailTask, isDone,  
DeadlineTask  
FROM Task  
WHERE Username = 'Username';
```

คำอธิบาย เป็นฟังก์ชัน เรียกหางาน โดยเลือก TaskID, Username, NameTask, DetailTask, isDone และ DeadlineTask มาจากตาราง Task โดยกำหนดให้ Username เท่ากับ ชื่องานที่ผู้ใช้เลือก

4.4 หา Task ทั้งหมดใน Database

```
SELECT TaskID, Username, NameTask, DetailTask, isDone,  
DeadlineTask  
FROM Task;
```

คำอธิบาย เป็นฟังก์ชัน เรียกหางานทั้งหมด โดยเลือก TaskID, Username, NameTask, DetailTask, isDone และ DeadlineTask มาจากตาราง Task

4.5 ลบ Task

```
DELETE FROM Task  
WHERE TaskID = 'TaskID' AND Username = 'Username';
```

คำอธิบาย เป็นฟังก์ชัน ลบงาน จากตาราง Task โดยกำหนดให้ TaskID เท่ากับ ไอดีงานที่ผู้ใช้เลือก และ Username เท่ากับ ชื่องานที่ผู้ใช้เลือก

4.6 ลบทุก Task ด้วย Username

```
DELETE FROM Task  
WHERE Username = 'Username';
```

คำอธิบาย เป็นฟังก์ชัน ลบงาน จากตาราง Task โดยกำหนดให้ Username เท่ากับ ชื่องานที่ผู้ใช้เลือก

4.7 ลบทุก Task ใน Database

```
DELETE FROM Task;
```

คำอธิบาย เป็นฟังก์ชัน ลบงานทั้งหมด จากตาราง Task

4.8 แก้ไข Task

```
UPDATE Task  
SET NameTask = 'NameTask_enter_by_user',  
DetailTask = 'Details_enter_by_user',  
DeadlineTask = 'Selected_Deadline_time_by_user',  
isDone = 'Selected_isDone_by_user'  
WHERE TaskID = 'Selected_TaskID_by_user' AND Username = 'Username';
```

คำอธิบาย เป็นฟังก์ชัน แก้งาน โดย SET ค่า NameTask ให้เท่ากับ ชื่องานที่ผู้ใช้กรอก ,DetailTask ให้เท่ากับ รายละเอียดเกี่ยวกับงานที่ผู้ใช้กรอก ,DeadlineTask ให้เท่ากับ วัน/เวลากำหนดส่งที่ผู้ใช้เลือก และ isDone ให้เท่ากับ งานเสร็จแล้วหรือไม่ ตามที่ผู้ใช้เลือก โดยกำหนดให้ TaskID เท่ากับ ID งานที่ผู้ใช้เลือก และ Username เท่ากับ ชื่องานที่ผู้ใช้เลือก

5. ฟังก์ชันของตาราง Course

5.1 เพิ่ม Course

```
INSERT INTO Course(CourseID, Username, ColorCourse, CourseName,
ProfessorName, DayOfWeek, StartTime, EndTime)
VALUES('CourseID', 'Username', 'ColorCourse', 'CourseName', 'ProfessorName',
'DayOfWeek', 'StartTime', 'EndTime')
```

คำอธิบาย เป็นฟังก์ชัน เพิ่มรายวิชา ที่มีคอลัมน์ CourseID, Username, ColorCourse, CourseName, ProfessorName, DayOfWeek, StartTime และ EndTime ให้มีค่าตามที่ผู้ใช้เป็นคนกำหนด

5.2 หา Course ด้วย CourseID และ Username

```
SELECT CourseID, Username, ColorCourse, CourseName, ProfessorName,
DayOfWeek, StartTime, EndTime
FROM Course
WHERE CourseID = 'CourseID' AND Username = 'Username'
```

คำอธิบาย เป็นฟังก์ชัน เรียกหารายวิชา โดยเลือก CourseID, Username, ColorCourse, CourseName, ProfessorName, DayOfWeek, StartTime และ EndTime มาจากตาราง Course โดยกำหนดให้ CourseID เท่ากับ รหัสรายวิชาที่ผู้ใช้เลือก และ Username เท่ากับ ชื่อรายวิชาที่ผู้ใช้เลือก

5.3 หา Course ทั้งหมดด้วย Username

```
SELECT CourseID, Username, ColorCourse, CourseName, ProfessorName,
DayOfWeek, StartTime, EndTime
FROM Course
WHERE Username = 'Username'
```

คำอธิบาย เป็นฟังก์ชัน เรียกหารายวิชา โดยเลือก CourseID, Username, ColorCourse, CourseName, ProfessorName, DayOfWeek, StartTime และ EndTime มาจากตาราง Course โดยกำหนดให้ Username เท่ากับ ชื่อรายวิชาที่ผู้ใช้เลือก

5.4 หา Course ทั้งหมดใน Database

```
SELECT CourseID, Username, ColorCourse, CourseName, ProfessorName,  
DayOfWeek, StartTime, EndTime  
FROM Course
```

คำอธิบาย เป็นฟังก์ชัน เรียกหารายวิชาทั้งหมด โดยเลือก CourseID, Username, ColorCourse, CourseName, ProfessorName, DayOfWeek, StartTime และ EndTime มาจากตาราง Course

5.5 ลบ Course ด้วย CourseID และ Username

```
DELETE FROM Course  
WHERE CourseID = 'CourseID' and  
Username = 'Username';
```

คำอธิบาย เป็นฟังก์ชัน ลบรายวิชา จากตาราง Course โดยกำหนดให้ CourseID เท่ากับ รหัสรายวิชาที่ผู้ใช้เลือก และ Username เท่ากับ ชื่อรายวิชาที่ผู้ใช้เลือก

5.6 ลบ Course ทั้งหมดด้วย Username

```
DELETE FROM Course WHERE Username = 'Username';
```

คำอธิบาย เป็นฟังก์ชัน ลบรายวิชา จากตาราง Course โดยกำหนดให้ Username เท่ากับ ชื่อรายวิชาที่ผู้ใช้เลือก

5.7 ลบ Course ทั้งหมดใน Database

```
DELETE FROM Course
```

คำอธิบาย เป็นฟังก์ชัน ลบรายวิชาทั้งหมด จากตาราง Course

5.8 แก้ไข Course

UPDATE Course

```
SET CourseID = 'NameTask_enter_by_user',  
    CourseName = 'Details_enter_by_user',  
    CourseColor = "Color",  
    ProfessorName = 'Selected_Deadline_time_by_user',  
    DayOfWeek = 'Selected_isDone_by_user',  
    StartTime = 'StartTime',  
    EndTime = 'EndTime'  
  
WHERE CourseID = 'Selected_CourseID_by_user' AND Username = 'Username';
```

คำอธิบาย เป็นฟังก์ชัน แก้ไขรายวิชา โดย SET ค่า

CourseID ให้เท่ากับ รหัสรายวิชาที่ผู้ใช้กรอก

,CourseName ให้เท่ากับ ชื่อรายวิชาที่ผู้ใช้กรอก

,CourseColor ให้เท่ากับ สีรายวิชาที่ผู้ใช้เลือก

,ProfessorName ให้เท่ากับ ชื่ออาจารย์ผู้สอนที่ผู้ใช้กรอก

,DayOfWeek ให้เท่ากับ วันที่มีเรียนรายวิชานี้ตามที่ใช้เลือก

,StartTime ให้เท่ากับ เวลาที่เริ่มเรียนรายวิชานี้ตามที่ใช้กรอก

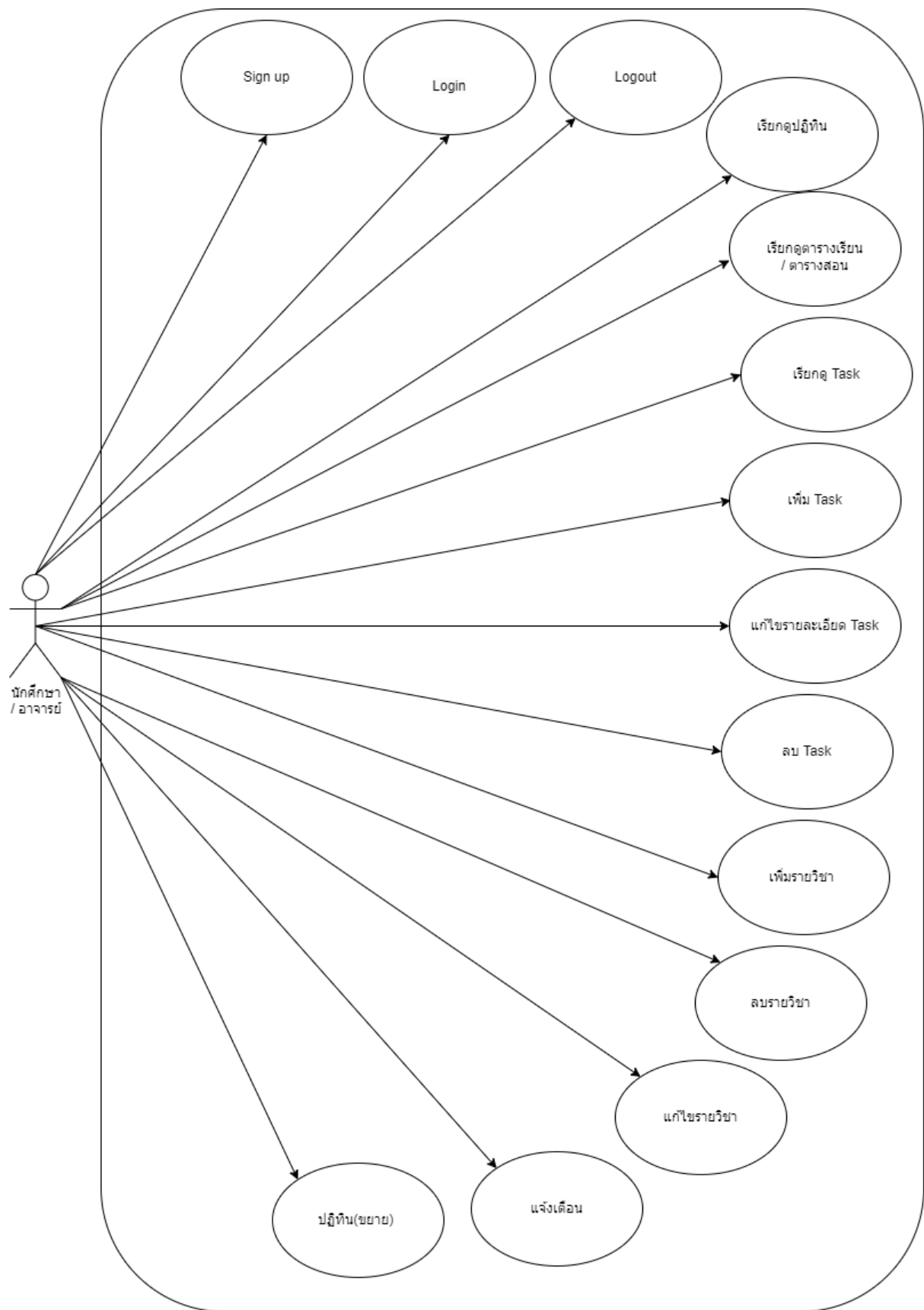
และ EndTime ให้เท่ากับ เวลาที่เลิกเรียนรายวิชานี้ตามที่ใช้กรอก

โดยกำหนดให้ CourseID เท่ากับ รหัสรายวิชาที่ผู้ใช้เลือก

และ Username เท่ากับ ชื่องานที่ผู้ใช้เลือก

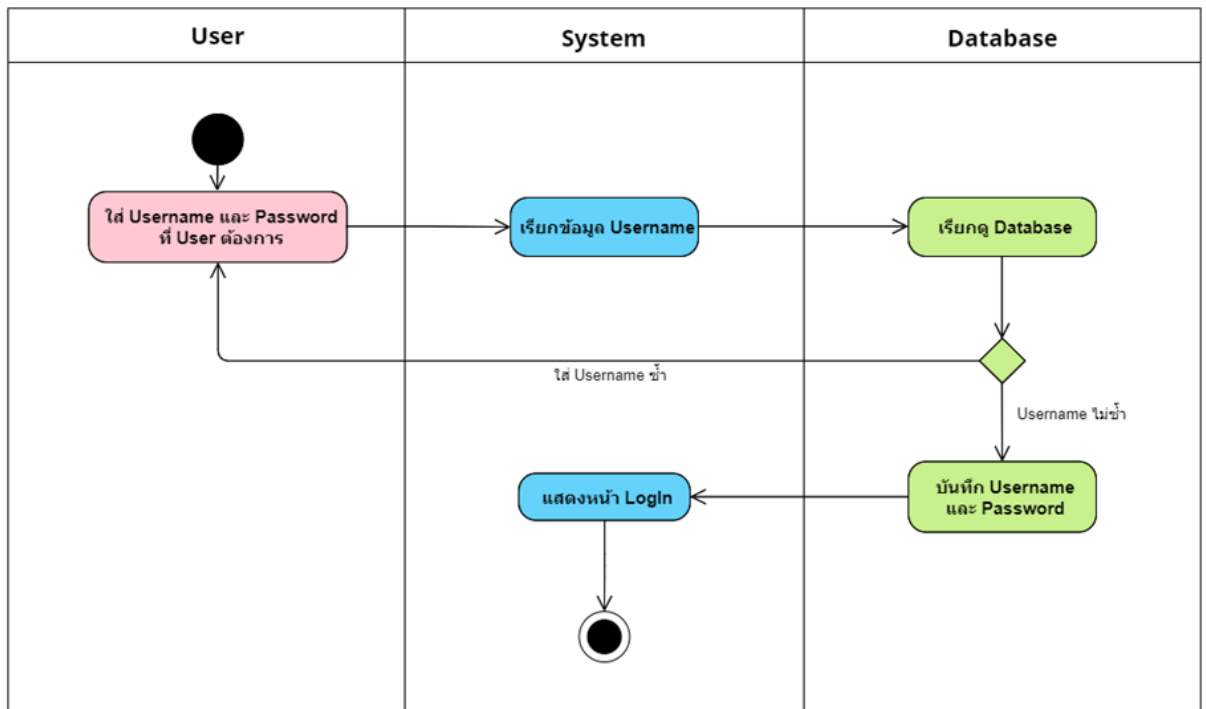
Use Case

Actor	Use case
★ User → นักศึกษา / อาจารย์	→ Sign up → Login → Logout → เรียกดูตารางเรียน/ตารางสอน → เพิ่มรายวิชา → แก้ไขรายวิชา → ลบรายวิชา → เรียกดู Task → เพิ่ม Task → แก้ไข Task → ลบ Task → เรียกดูปฏิทิน → เรียกดูปฏิทินขยาย (เพื่อดู Task) → แจ้งเตือน

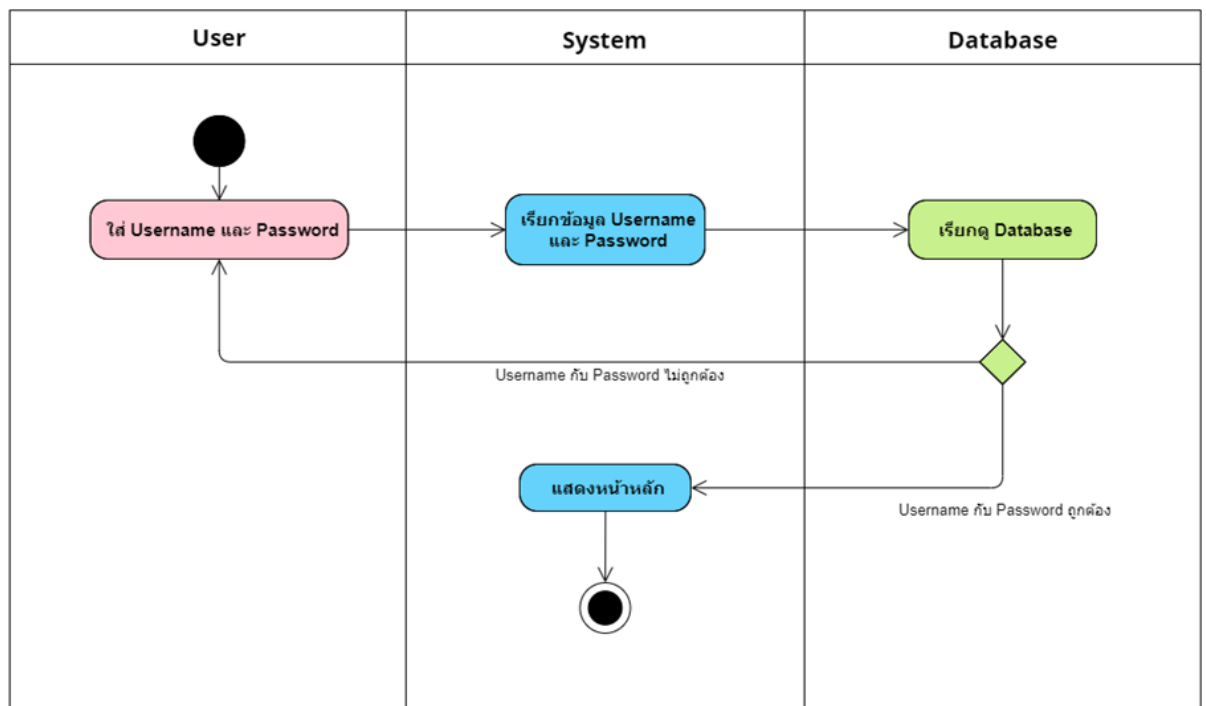


Activity Diagram

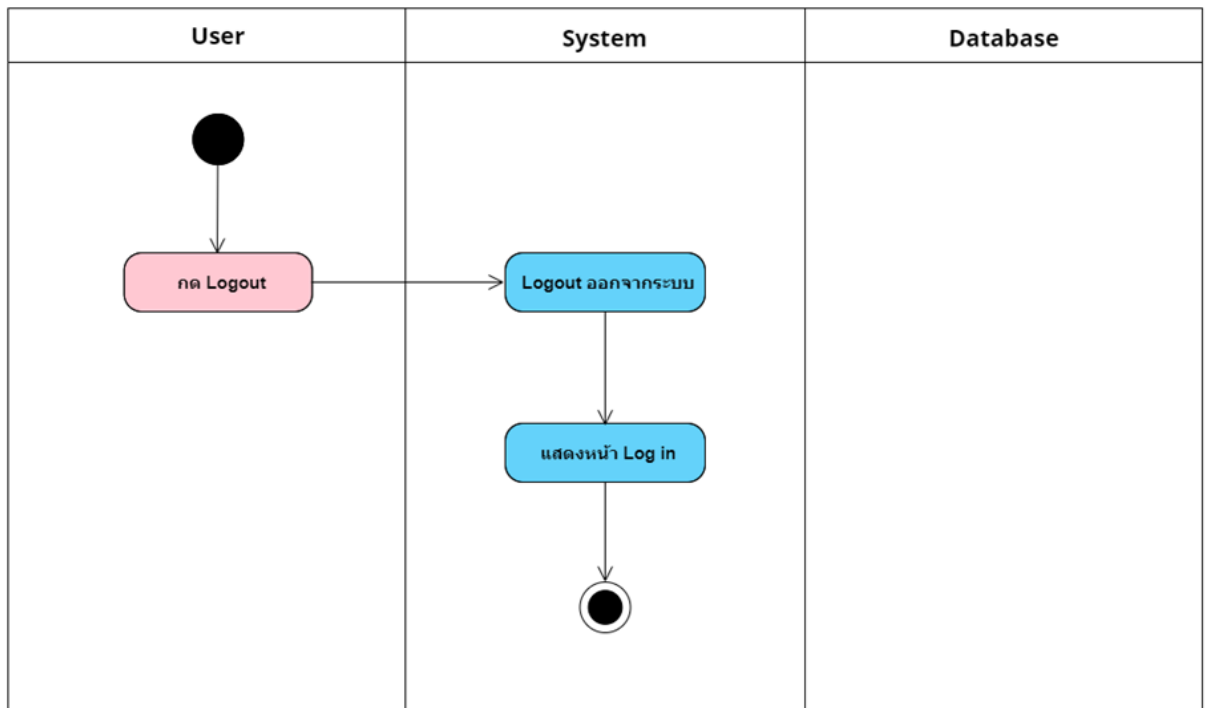
→ Sign up



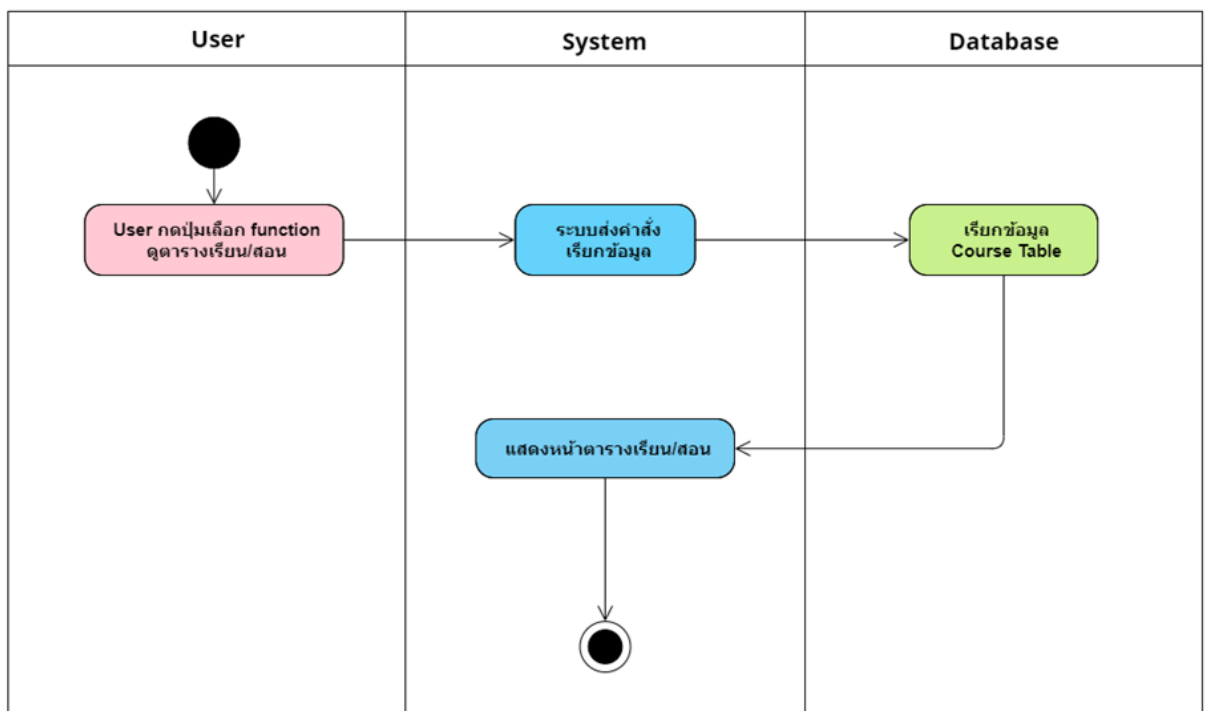
→ Login



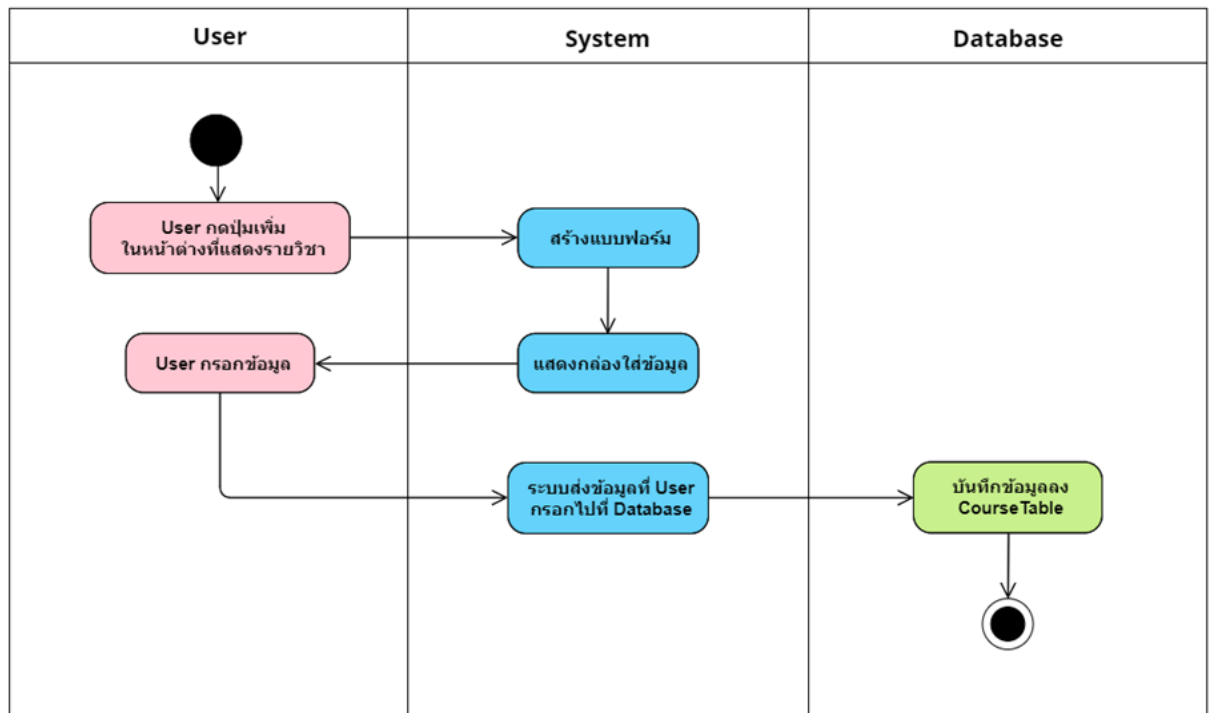
→ Logout



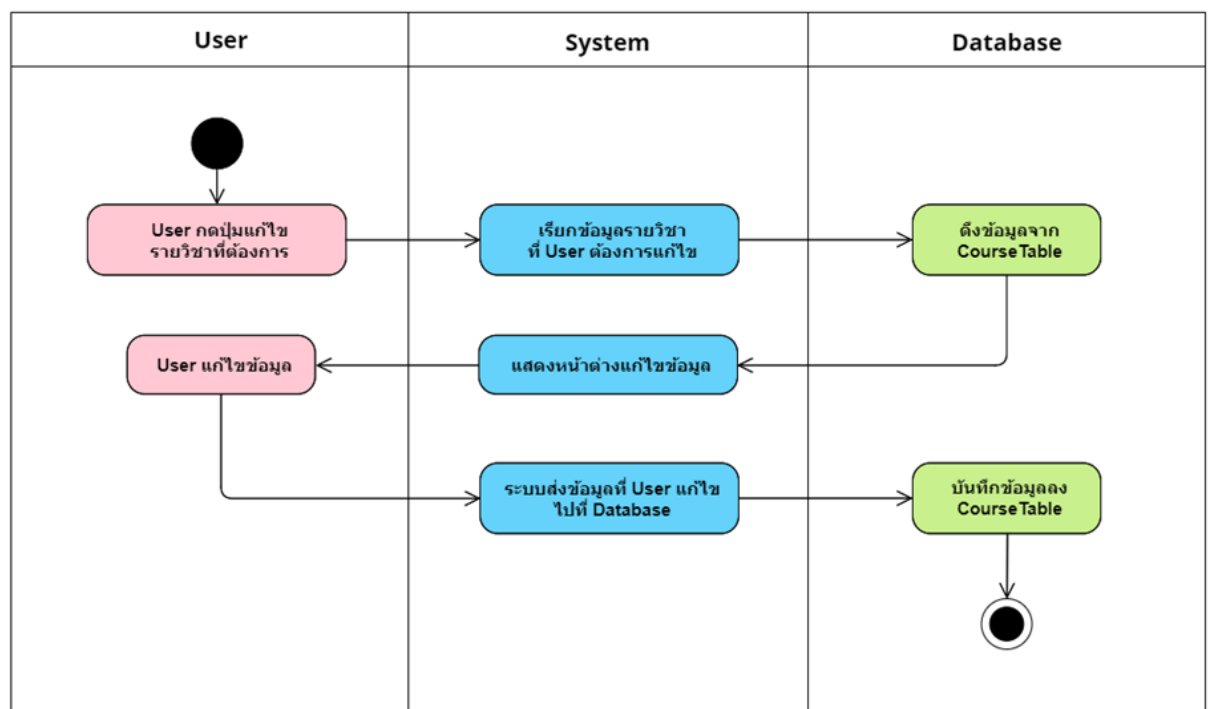
→ เรียกดูตารางเรียน/ตารางสอน



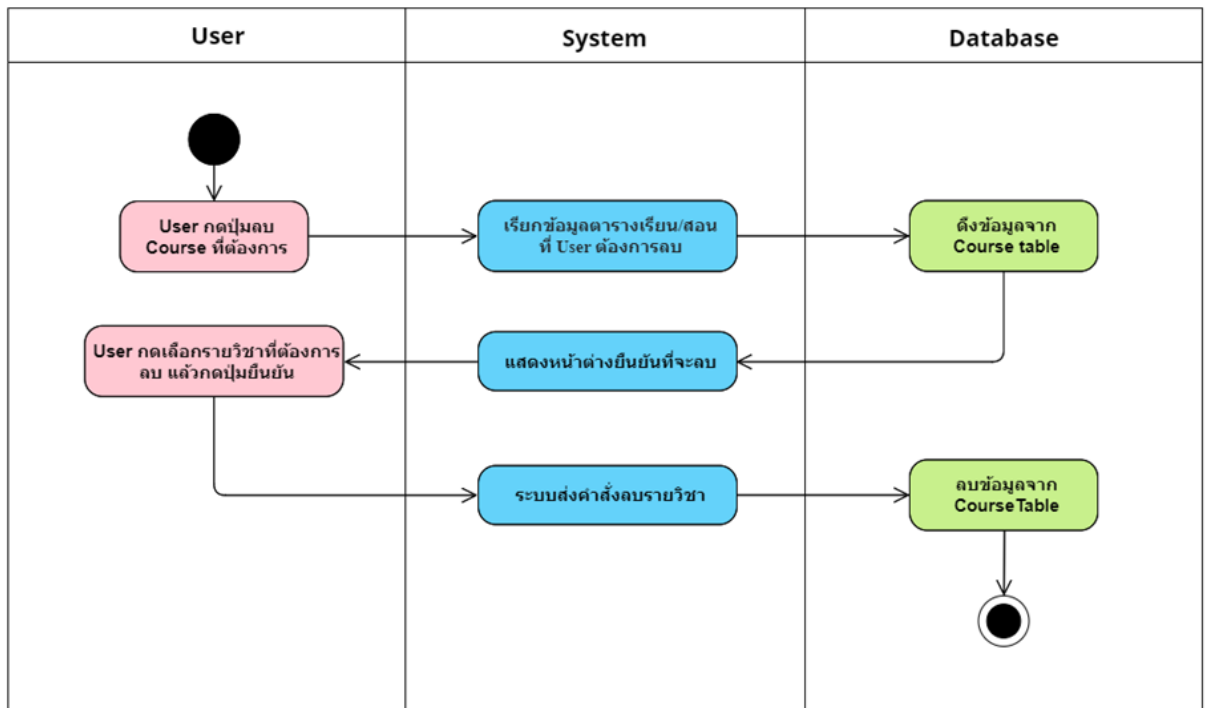
→ เพิ่มรายวิชา



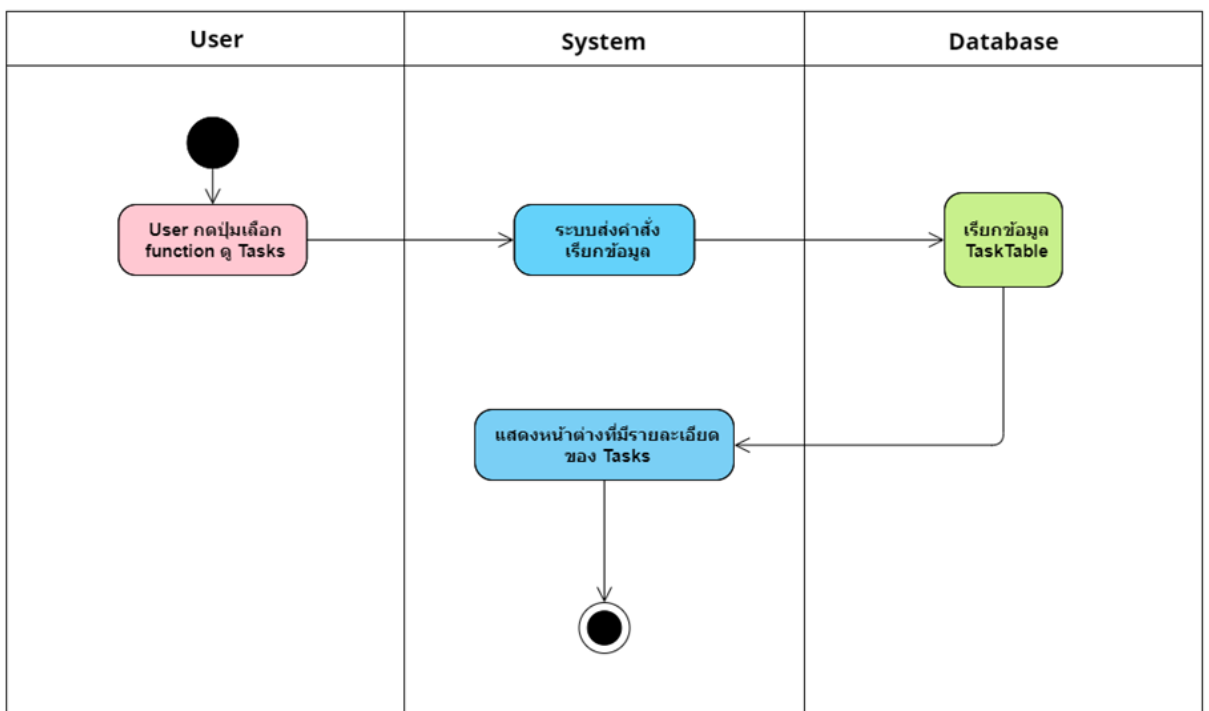
→ แก้ไขรายวิชา



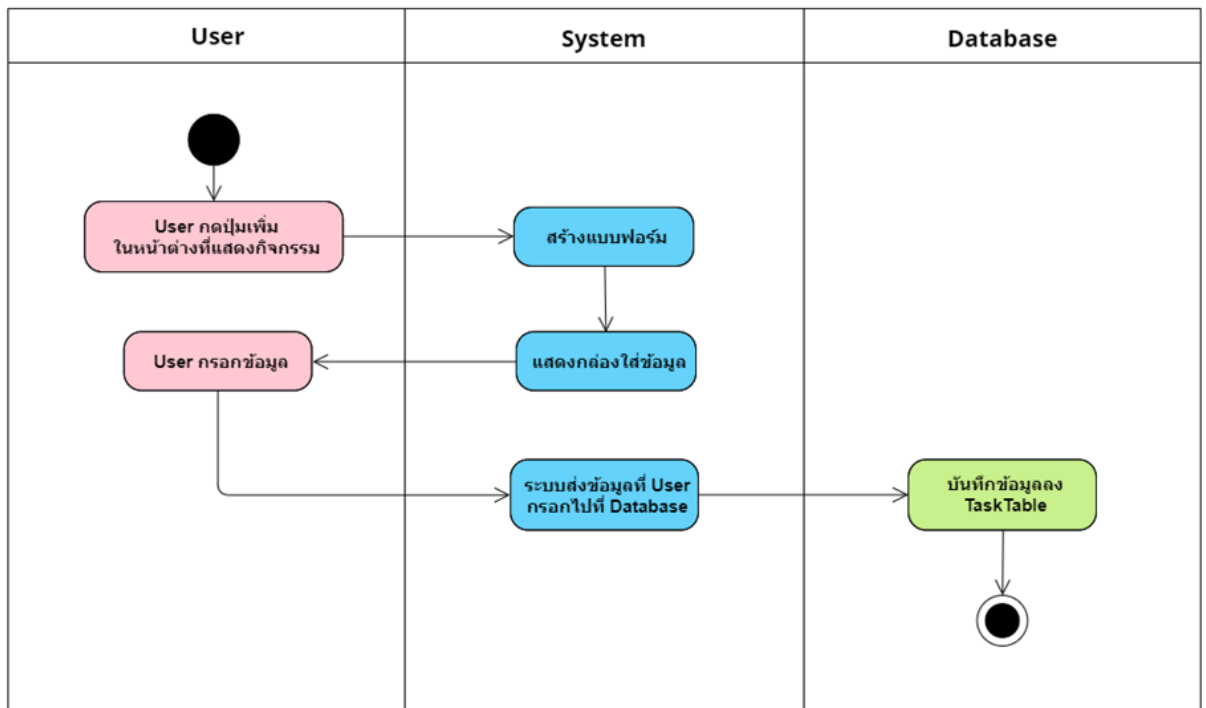
→ ลบรายวิชา



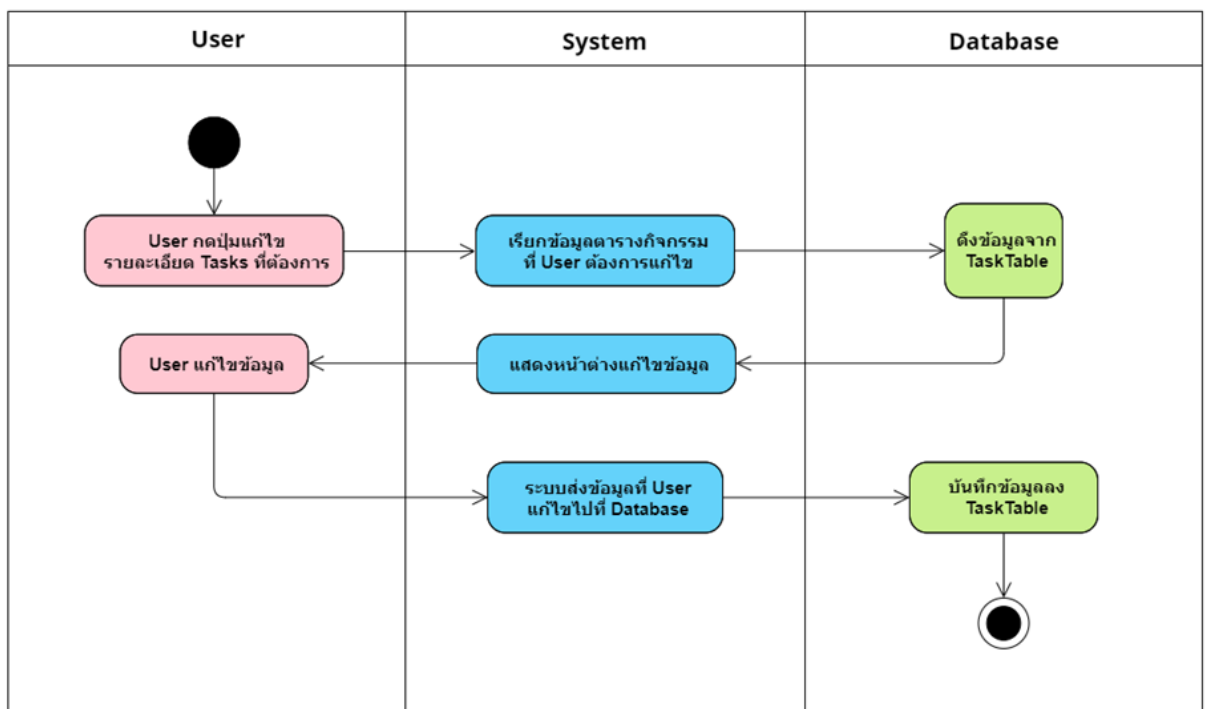
→ เรียกดู Task



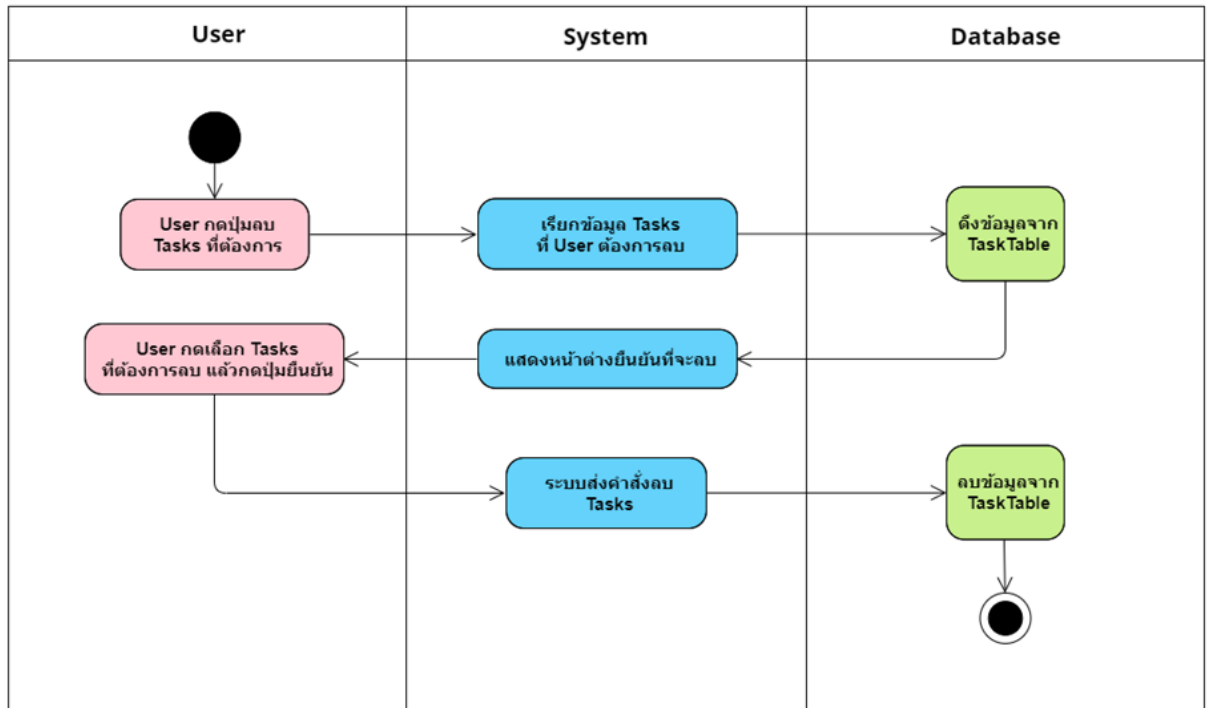
→ เพิ่ม Task



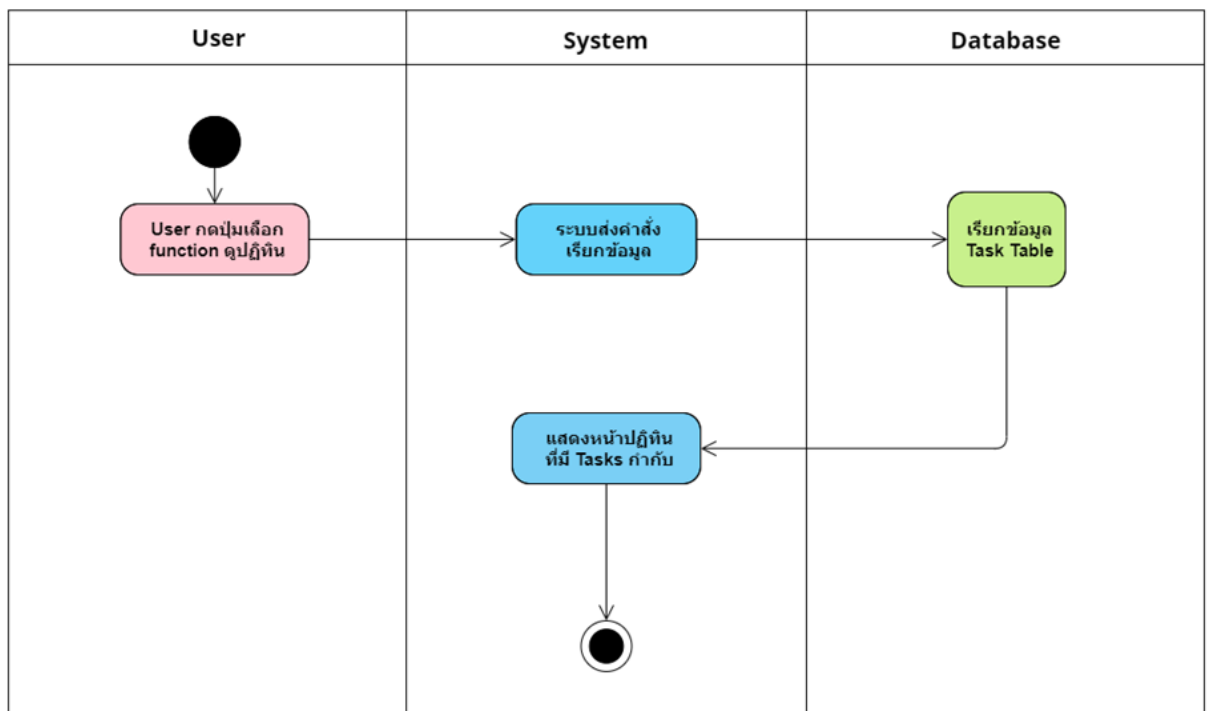
→ แก้ไข Task



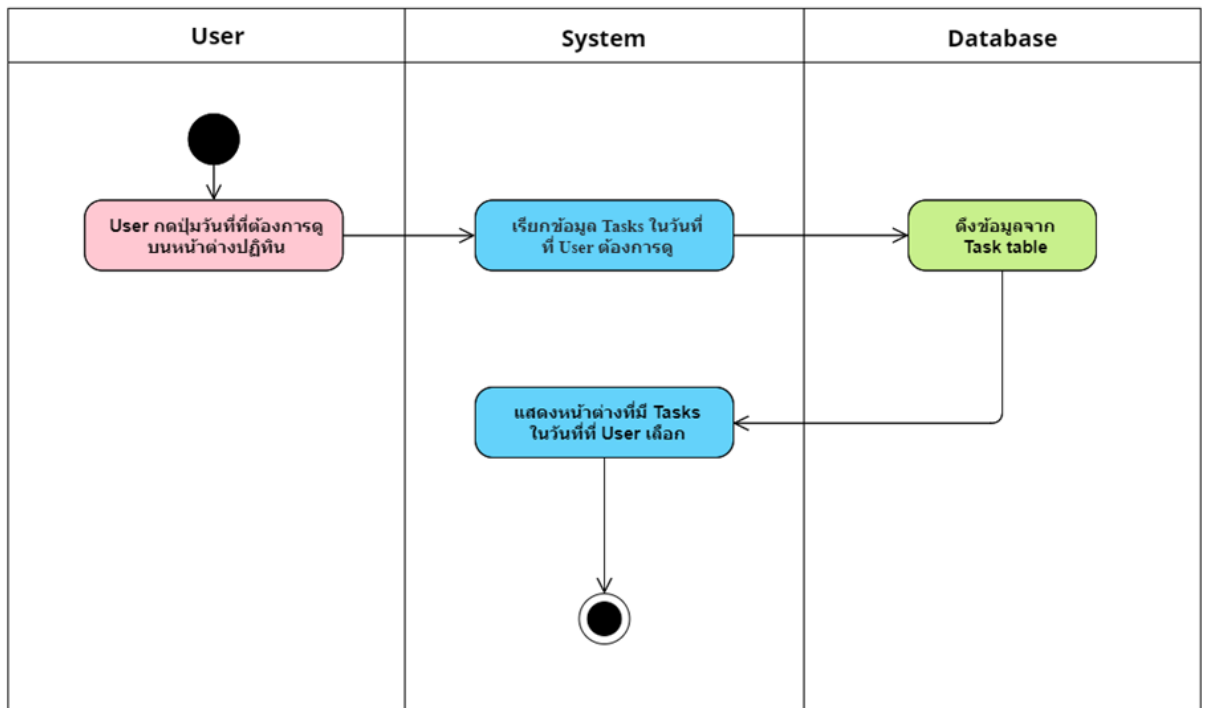
→ ลบ Task



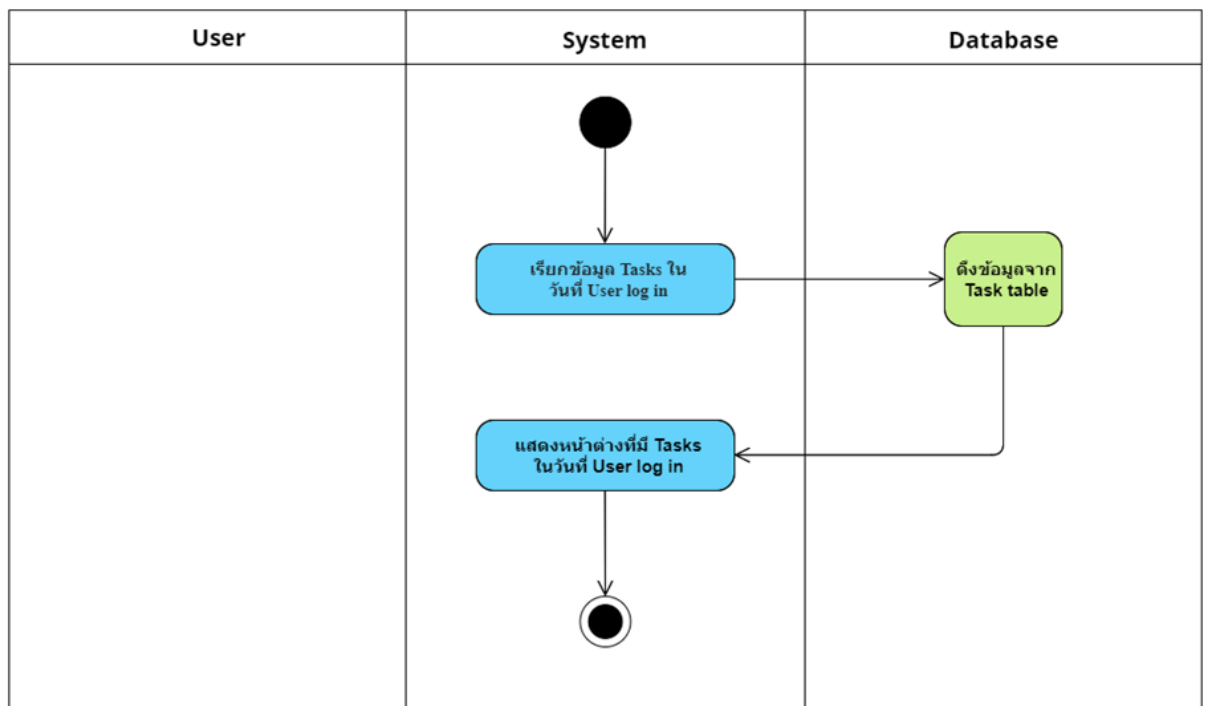
→ เรียกดูปฏิทิน



→ เรียกดูปฏิทินขยาย (เพื่อดู Task)



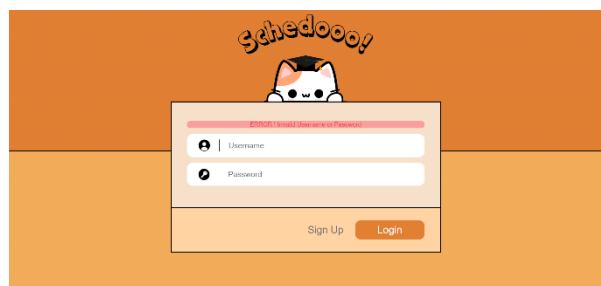
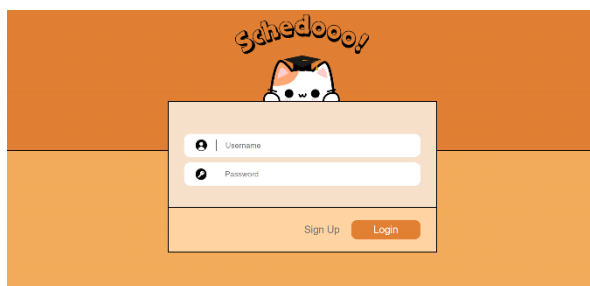
→ แจ้งเตือน



User Interface

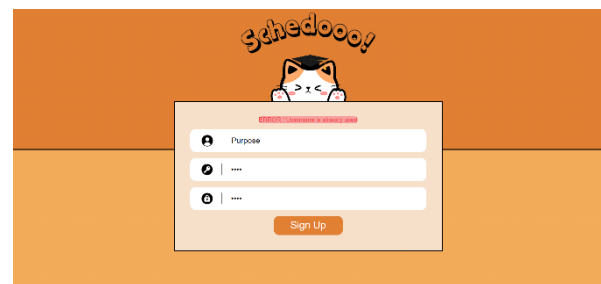
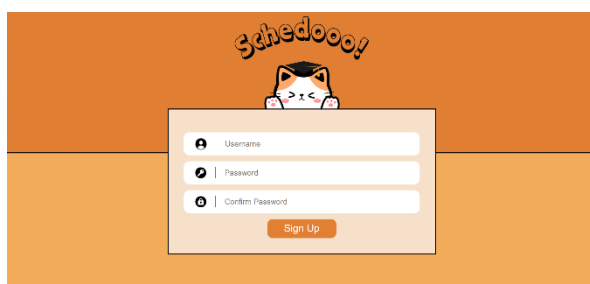
1. หน้า Login

ในหน้านี้ผู้ใช้ต้องใส่ Username และ Password ให้ถูกต้อง เมื่อผู้ใช้กดปุ่ม “Login” ที่ปุ่มขวาล่างจะสามารถเข้าใช้ web app นี้ได้ ถ้าใส่ Username หรือ Password ไม่ถูกต้องระบบจะแจ้งเตือนให้ใส่ใหม่ให้ถูกต้องดังภาพ แต่ถ้าหากผู้ใช้ยังไม่มีบัญชีผู้ใช้สามารถกดไปที่ Sign up เพื่อลงทะเบียนได้



2. หน้า Sign up

ในหน้านี้ผู้ใช้ที่ยังไม่มีบัญชีสามารถสร้างบัญชีได้โดยใส่ Username และ Password เมื่อผู้ใช้กดปุ่ม “Sign Up” ที่ปุ่มล่าง ก็จะลงทะเบียนเรียบร้อยแต่ Username ต้องไม่ซ้ำกับผู้ใช้คนอื่น หากผู้ใช้กรอก Username ซ้ำกับผู้อื่นระบบจะแจ้งเตือนให้ใส่ Username ใหม่ดังภาพ



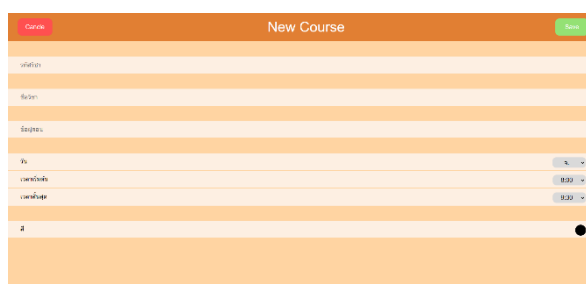
3. หน้าตารางเรียน/ตารางสอน

ในหน้านี้ผู้ใช้สามารถดู ตารางเรียน/ตารางสอน, เพิ่มรายวิชา, แก้ไขรายวิชา และลบรายวิชาได้ โดยสามารถคลิกที่ icon อันแรก (เมื่อกดแล้วจะเป็นสี่สีส้ม) เพื่อย้ายหน้านี้ได้เลย

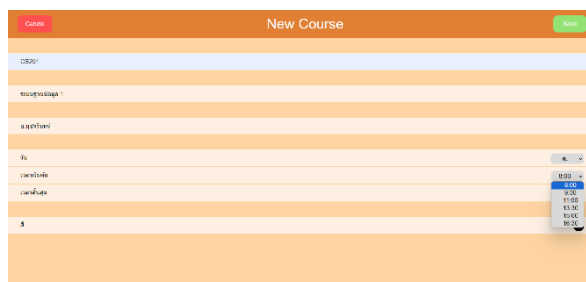
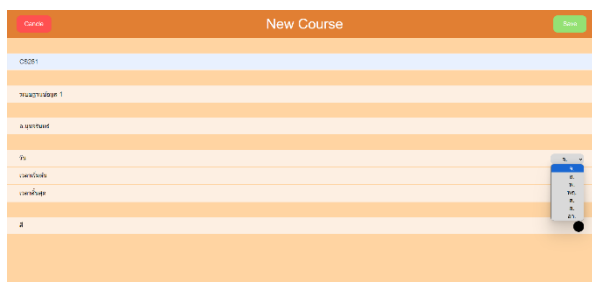


4. หน้าเพิ่มรายวิชา

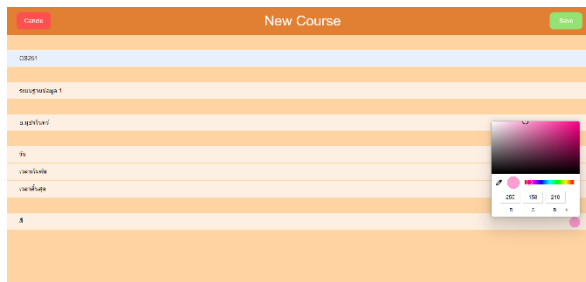
ในหน้านี้เมื่อผู้ใช้ต้องการเพิ่มรายวิชาลงตารางเรียน/ตารางสอน ผู้ใช้สามารถกดปุ่ม “+” ที่มุมขวาด้านบนได้เลย



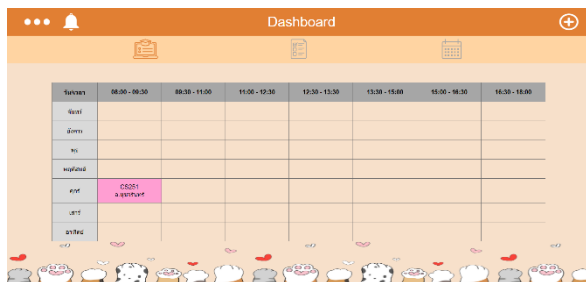
ต่อมาเมื่อผู้ใช้คลิกเข้ามาในหน้าเพิ่มรายวิชาแล้ว ก็จะแสดงหน้าให้กรอกรายละเอียดต่างๆ เริ่มจาก รหัสวิชา, ชื่อวิชา และชื่อผู้สอน ผู้ใช้สามารถพิมพ์ข้อมูลได้เลย



ในส่วนของวัน, เวลาเริ่มต้น และเวลาสิ้นสุด เมื่อผู้ใช้กดก็จะสามารถเลือกวัน และเวลา ได้ดังภาพ



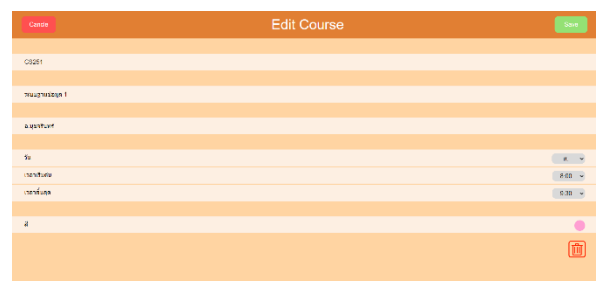
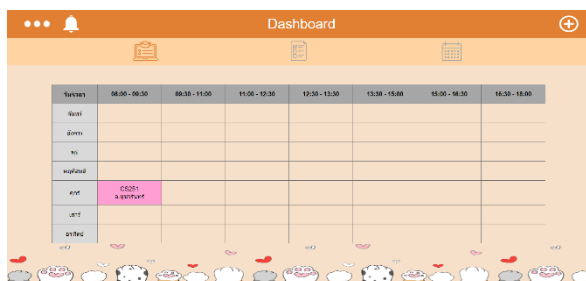
ในส่วนของสี เมื่อผู้ใช้งานก็สามารถลากเพื่อเลือกสีได้ตามต้องการ หรือหากมีสีที่ต้องการอยู่แล้ว ก็สามารถใส่รหัสสี RGB โดยการพิมพ์ได้ ดังภาพ



เมื่อเพิ่มรายวิชาเสร็จแล้วให้กดปุ่ม “Save” สีเขียวที่มุมขวาด้านบน รหัสวิชา และ ชื่ออาจารย์ผู้สอนจะมาขึ้นที่ตารางเรียน/ตารางสอน ดังภาพ

5. หน้าแก้ไขรายวิชา

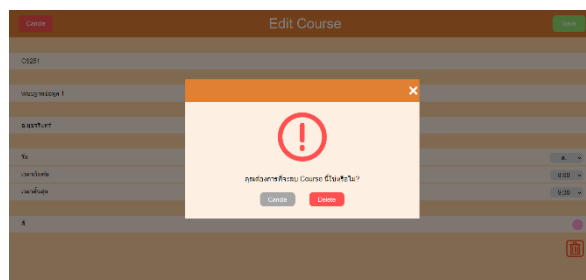
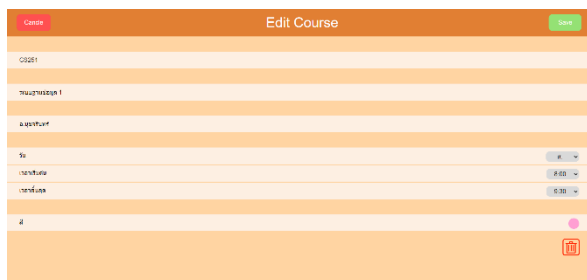
ในหน้านี้ผู้ใช้งานสามารถแก้ไข ตารางเรียน/ตารางสอนได้ โดยกดไปที่รายวิชานั้นๆ บนตารางเรียน/ตารางสอนได้เลย



เมื่อผู้ใช้งานแก้ไขรายวิชานั้นๆ เสร็จแล้วให้กดปุ่ม “Save” สีเขียวที่มุมขวาด้านบน ดังภาพ

6. หน้าลบบรายวิชา

ในหน้านี้ผู้ใช้สามารถลบ ตารางเรียน/ตารางสอนได้ โดยกดไปที่รายวิชานั้นๆ บนตารางเรียน/ตารางสอน



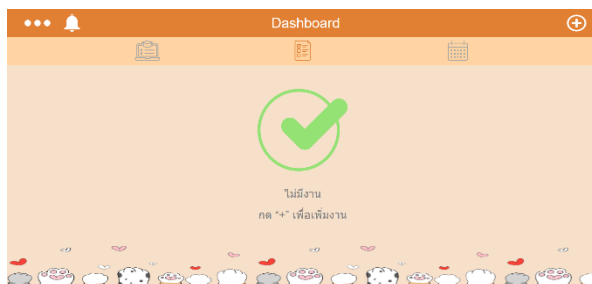
ต่อมาให้กดปุ่มรูป “ถังขยะ” สีแดงที่มุมขวาล่าง เมื่อกดแล้วระบบจะตั้งหน้าต่าง ยืนยันการลบบรายวิชานั้นๆ หากยืนยันให้กดปุ่ม “Delete” สีแดงที่มุมขวาล่าง ดังภาพ



เท่านี้รายวิชานั้นๆ ก็จะถูกลบเรียบร้อยแล้ว

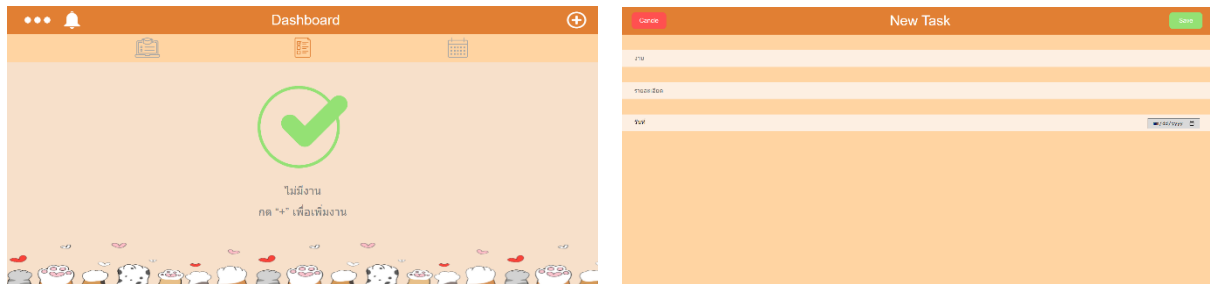
7. หน้าเรียกดู Task

ในหน้านี้ผู้ใช้สามารถดู Task, เพิ่ม Task, แก้ไข Task และลบ Task ได้ โดยสามารถกดที่ icon อันกลาง (เมื่อกดแล้วจะเป็นสีส้ม) เพื่อมายังหน้านี้ได้เลย

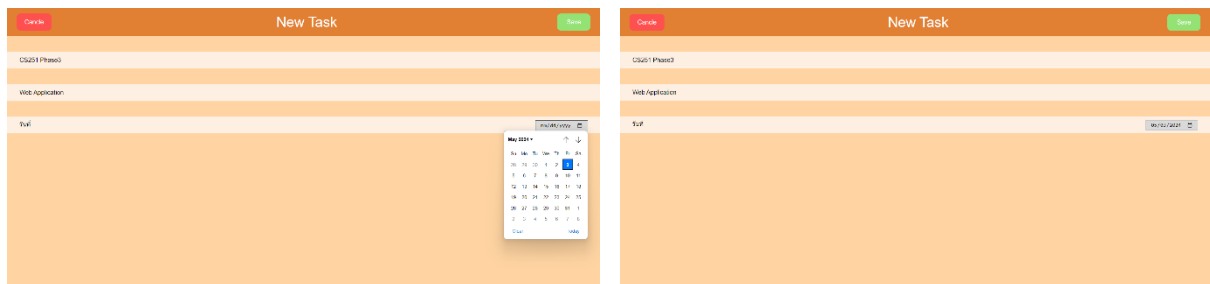


8. หน้าเพิ่ม Task

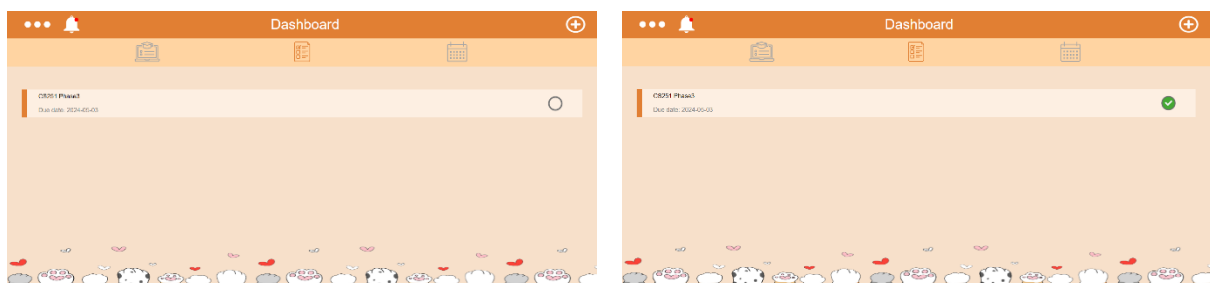
ในหน้านี้เมื่อผู้ใช้งานต้องการเพิ่ม Task ผู้ใช้งานสามารถกดปุ่ม “+” ที่มุมขวาบนได้เลย



ต่อมาเมื่อผู้ใช้งานคลิกเข้ามาในหน้าเพิ่ม Task ก็จะแสดงหน้าให้กรอกรายละเอียดต่างๆ เริ่มจาก งาน และรายละเอียดงาน ผู้ใช้งานสามารถพิมพ์ข้อมูลได้เลย



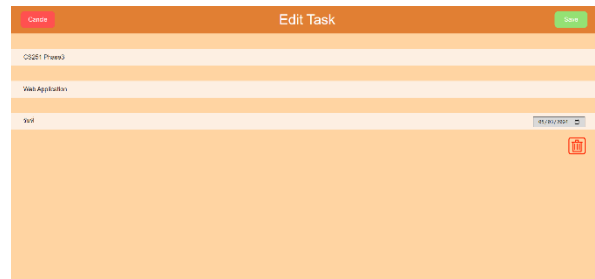
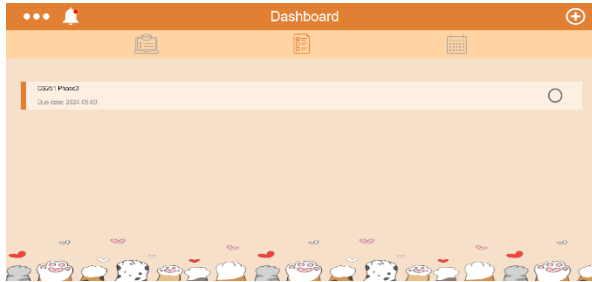
ในส่วนของวันที่ เมื่อผู้ใช้งานคลิกจะมีหน้าต่างวันที่ขึ้นมาให้ผู้ใช้งานเลือก เมื่อ Task เสร็จแล้วให้กดปุ่ม “Save” สีเขียวที่มุมขวาบน



ต่อมา งาน และวันกำหนดการส่งงานก็จะขึ้น ดังภาพ และเมื่อผู้ใช้งานทำงานนั้นๆ เสร็จแล้ว ก็จะมีช่องให้ผู้ใช้งานสามารถติ๊กข้างหลังได้ว่าทำงานนั้นเสร็จแล้ว

9. หน้าแก้ไข Task

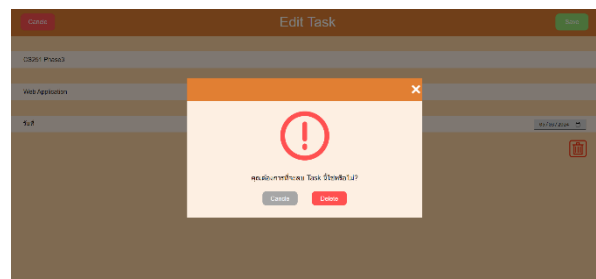
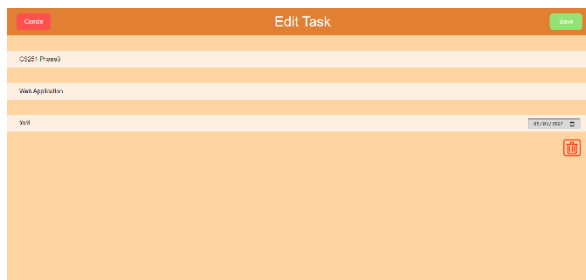
ในหน้านี้ผู้ใช้สามารถแก้ไข Task ได้ โดยกดไปที่ Task นั้นๆ



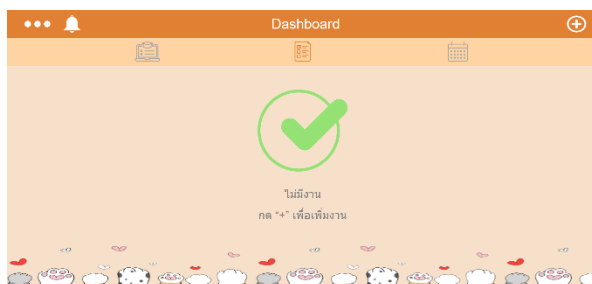
เมื่อผู้ใช้แก้ไข Task นั้นๆ เสร็จแล้วให้กดปุ่ม “Save” สีเขียวที่มุมขวามือ ดังภาพ

10. หน้าลบ Task

ในหน้านี้ผู้ใช้สามารถลบ Task ได้ โดยกดไปที่ Task นั้นๆ ได้เลย



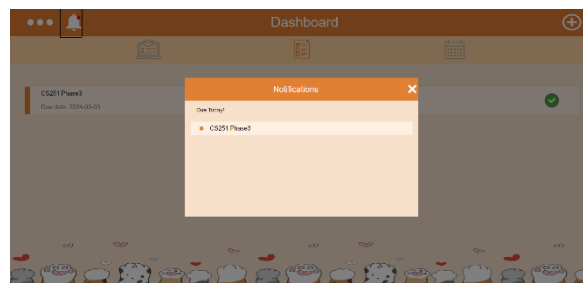
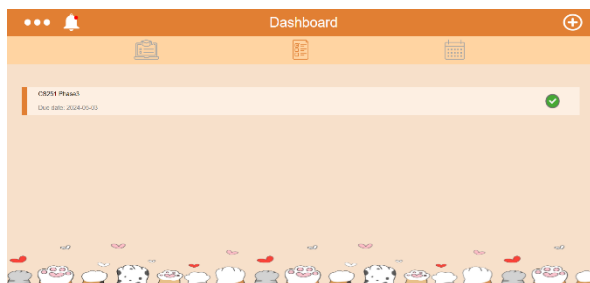
ต่อมาให้กดปุ่มรูป “ถังขยะ” สีแดงที่มุมขวาล่าง เมื่อกดแล้วระบบจะแจ้งเตือนหน้าต่างยืนยันการลบ Task หากยืนยันให้กดปุ่ม “Delete” สีแดงที่มุมขวาล่าง ดังภาพ



เท่านี้ Task นั้นๆ ก็จะถูกลบเรียบร้อยแล้ว

11. หน้าแจ้งเตือน

ที่แถบด้านบนจะมีสัญลักษณ์รูปกระดิ่ง หากถึงวันที่มีกำหนดส่งงานใดๆ ที่กระดิ่งจะมีจุดสีแดงดังภาพ



เมื่อกดเข้าไปที่ปุ่มกระดิ่งก็จะมีหน้าต่างแสดง Task งานที่ต้องส่งในวันนั้นๆ ดังภาพ

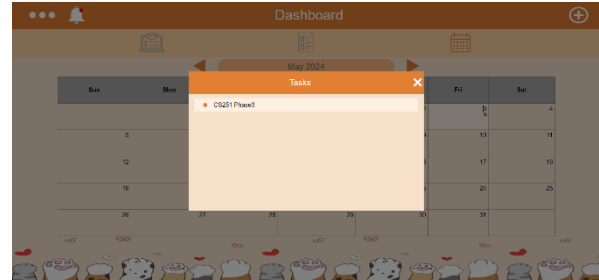
12. หน้าเรียกดูปฏิทิน

ในหน้านี้ผู้ใช้สามารถดูปฏิทิน และสามารถดูวันที่มี Task ที่ต้องส่งได้ โดยสามารถกดที่ icon อันที่สาม (เมื่อกดแล้วจะเป็นสีส้ม) เพื่อมายังหน้านี้ได้เลย



13. หน้าเรียกดูปฏิทินขยาย (เพื่อดู Task)

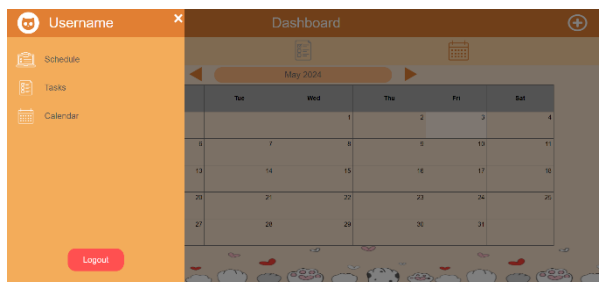
ในหน้านี้หากผู้ใช้มี Task งานที่ต้องส่งวันไหน ก็จะมีจุดสีส้มที่ไต้หวันั้นๆ



เมื่อกดเข้าไป ก็จะมีหน้าต่างเด้งขึ้นมาเพื่อแสดง Task งานที่ต้องส่งในวันนั้นๆ ดังภาพ

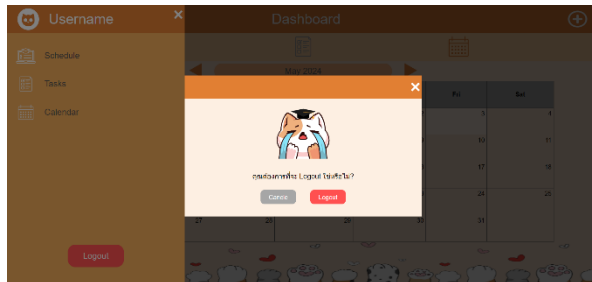
14. หน้าเมนู

ที่แถบด้านบนจะมีสัญลักษณ์เป็นปุ่มสามปุ่ม เมื่อกดเข้าไปก็จะสามารถไปที่หน้าตารางเรียน/ตารางสอน (Schedule), หน้า Task (Taks) และหน้าปฏิทิน (Calendar) ได้เหมือนกัน นอกจากนี้ ในหน้านี้ก็ยังมีปุ่มสำหรับ Logout ดังภาพ



15. หน้าLogout

ในหน้านี้หากผู้ใช้ต้องการออกจากระบบ ก็สามารถกดที่ปุ่ม “Logout” สีแดงที่ซ้ายล่าง



เท่านี้ก็จะออกจากระบบเรียบร้อยแล้ว