

M2: Architectural Design, UML diagrams & Implementation Plan

โปรเจค: ระบบการจัดการรถไฟ(Train-Management-System)

กลุ่ม: The-Fox

ผู้จัดทำ

นายธนวัฒน์ โกวิทรัตนกรกุล รหัส 55010511

นายบุญฤทธิ์ งามวิรรจน์เจริญ รหัส 55010683

Problem Analysis

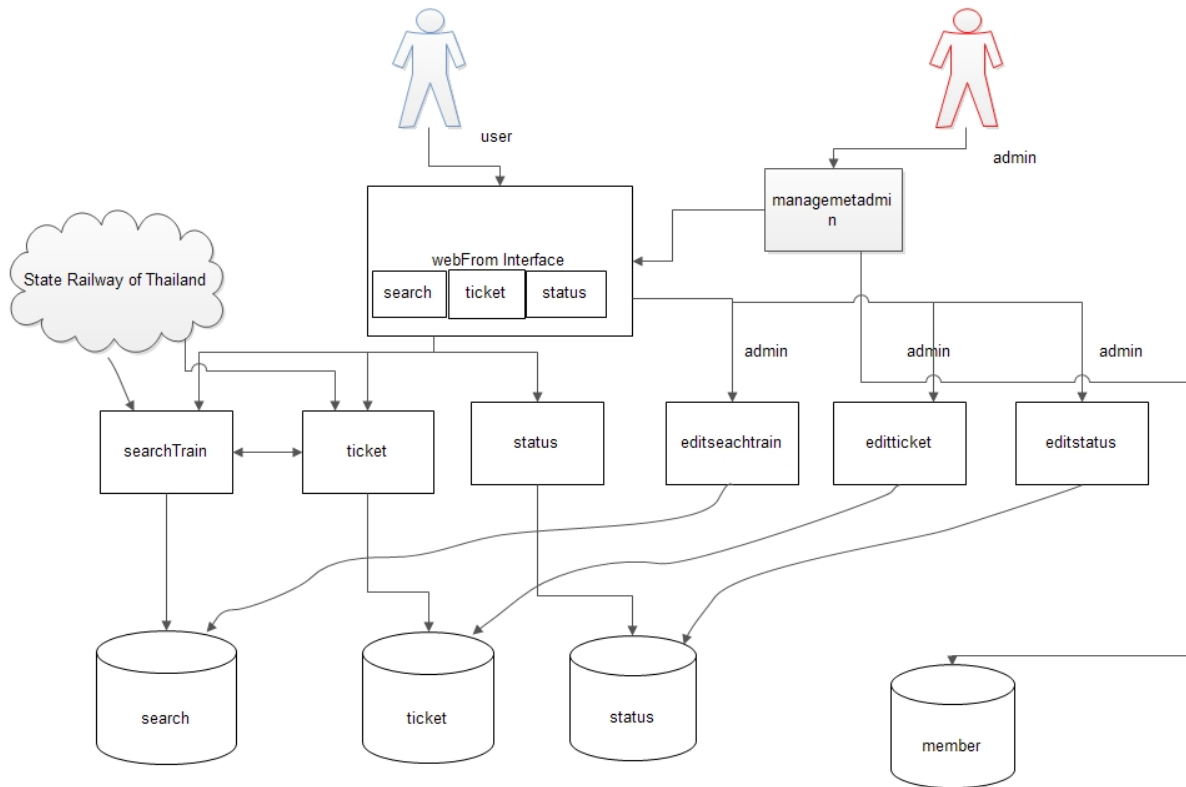
Abstraction

- 1.แผนที่เส้นทางรถไฟ เพื่อใช้ดูเส้นทางที่รถไฟวิ่งผ่าน
- 2.ค้นหาเส้นทางรถไฟ ใช้เพื่อดู สาย,เวลา,วันที่,ชนิดของรถไฟและเที่ยวไป เที่ยวกลับของรถไฟ
- 3.ราคาตั๋ว จะใช้เพื่อแสดงราคาตั๋วรถไฟและส่วนที่เกี่ยวข้องกับราคาตั๋ว
- 4.การแจ้งสถานะ ใช้เพื่อแจ้งสถานะของรถไฟได้
- 5.การเพิ่ม,ลบ,แก้ไขระบบค้นหาเส้นทาง จะสามารถ เพิ่ม ลบ แก้ไข ข้อมูลในระบบค้นหาเส้นทางรถไฟได้
- 6.การแก้ไขราคาตั๋ว จะสามารถ แก้ไข ราคาตั๋วและสิ่งเกี่ยวข้องกับตั๋วได้

Component

- 1.การแจ้งสถานะ
- 2.ราคาตั๋วและส่วนที่เกี่ยวข้องกับราคา
- 3.การค้นหาเส้นทาง
- 4.แก้ไขราคาตั๋ว
- 5.การแก้ไข การค้นหาเส้นทาง
- 6.แก้ไขสถานะรถไฟ

Application Architecture



รายละเอียดของ Application Architecture

อินเทอร์เฟซ(interface)

-WebFrom ทำหน้าที่ เป็น user interface ของระบบ ใช้สำหรับรับข้อมูลจาก ผู้ใช้งานและแสดงข้อมูลข้อมูลที่ ผู้ใช้งาน ต้องการต้องการ

Application

ผู้ใช้งาน(User)

-Ticket ทำหน้าที่ ดูราคาของตั๋วรถไฟ ซึ่งประกอบด้วยส่วนที่เกี่ยวข้องกับตัว คือสายรถไฟ เวลา วันที่ และชนิดของรถไฟ และเที่ยวไป เทียบกลับ

-Status จะส่วนที่ติดต่อกับผู้ใช้งานไว้สำหรับใช้แจ้งสถานะของรถไฟ

-Search จะใช้ในการค้นหาเส้นทางของรถไฟ ประกอบไปด้วย สาย,เวลา,วันที่,ชนิดของรถไฟและเที่ยวไป เทียบกลับของรถไฟ

ผู้ดูแลระบบ(Admin)

-EditSearch ใช้ในการเพิ่มหรือลบข้อมูลในฐานข้อมูลของ Search ซึ่งประกอบไปด้วย สาย,เวลา,วันที่,ชนิดของรถไฟและ
เที่ยวไป เทียบกลับของรถไฟ

-EditTicket ใช้ในการเพิ่มหรือลบข้อมูลในฐานข้อมูลของ Ticket ซึ่งประกอบไปด้วย สาย,เวลา,วันที่,ชนิดของรถไฟ,ราคาและ
เที่ยวไป เทียบกลับของรถไฟ

-Editstatusใช้ในการลบข้อมูลในฐานข้อมูลของ Status ซึ่งประกอบไป ข้อมูลที่ผู้ใช้งานแจ้งสถานะเข้ามา

Subsystem/Component

ส่วนของผู้ใช้งาน

-ส่วนที่ใช้ในการค้นหาเส้นทางของรถไฟซึ่งประกอบด้วย สาย ,วันเวลา,สถานี,เที่ยวไปเทียบกลับของรถไฟ

-ส่วนที่ใช้ในการดูราคาตั๋วของรถไฟและส่วนที่เกี่ยวข้องกับราคาตั๋ว

-ส่วนที่ใช้ในการติดต่อกับผู้ใช้งานซึ่งประกอบ ส่วนที่ใช้ในการรับข้อมูลและแสดงผลของข้อมูล

-ส่วนของผู้ดูแลระบบ

-ส่วนของการสมัครสมาชิกคือส่วนที่ใช้ในการเก็บข้อมูลต่างๆของสมาชิกลงในฐานข้อมูล

-ส่วนของการล็อกอิน(login) คือส่วนที่ใช้ในการเข้าสู่ระบบ

-ส่วนของการเพิ่มลบแก้ไขระบบค้นหาเส้นทางรถไฟจะใช้ในการเพิ่มหรือลบข้อมูลในฐานข้อมูล

-ส่วนของการเพิ่มลบแก้ไขราคาตั๋วจะใช้ในการแก้ไขฐานข้อมูลของราคาตั๋วและส่วนที่เกี่ยวข้องในตัว

ส่วนฐานข้อมูล

-ฐานข้อมูลของผู้ดูแลระบบใช้ในการเก็บข้อมูลของผู้ดูแลระบบ

-ฐานข้อมูลค้นหาเส้นทางใช้ในการเก็บข้อมูลของสาย,วันเวลา เป็นต้น

-ฐานข้อมูลของราคาตั๋วและส่วนที่เกี่ยวข้องใช้ในการเก็บข้อมูลราคาตั๋ว ,สาย,วันเวลา เป็นต้น

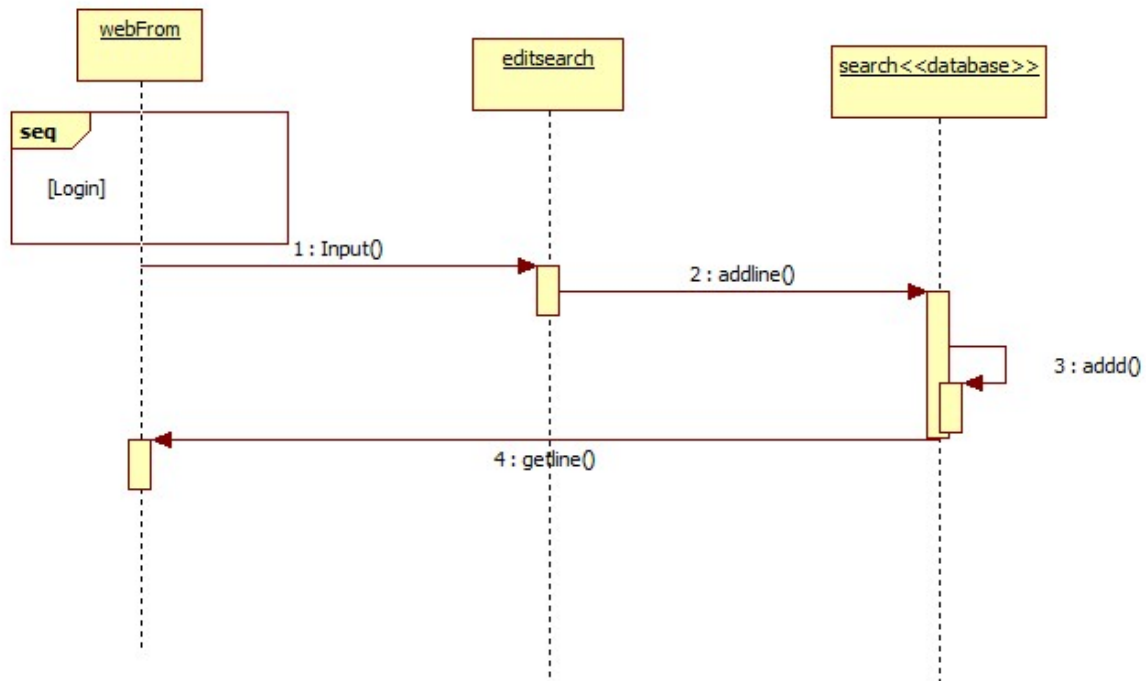
-ฐานข้อมูลเก็บสถานะใช้ในการเก็บสถานะของรถไฟ

ฐานข้อมูลจากภายนอก

-ฐานข้อมูลของการรถไฟแห่งประเทศไทย

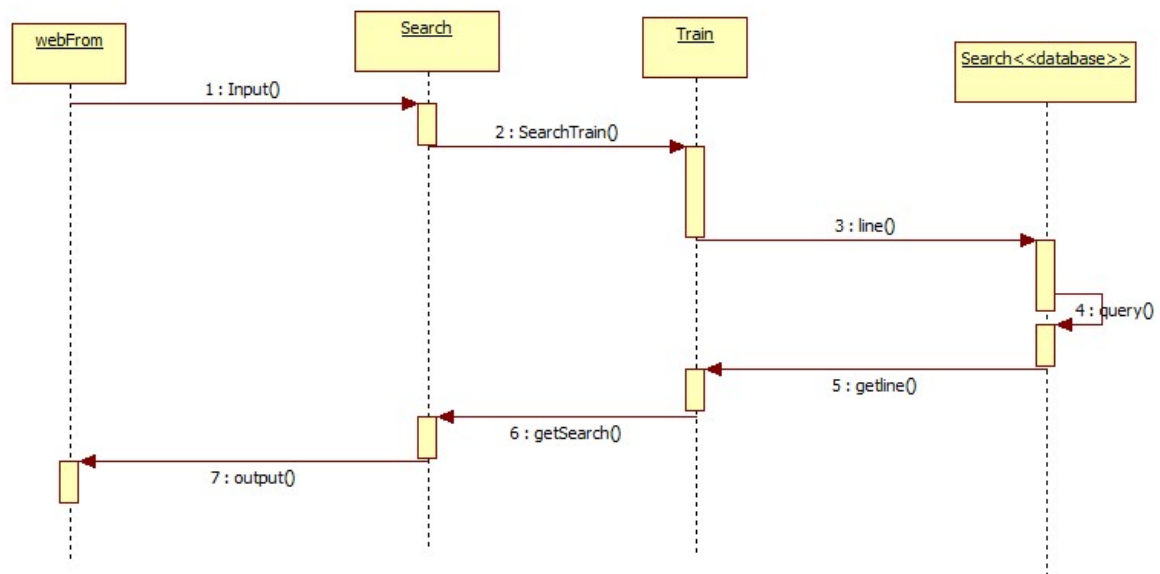
SequenceDiagram

Editsearch



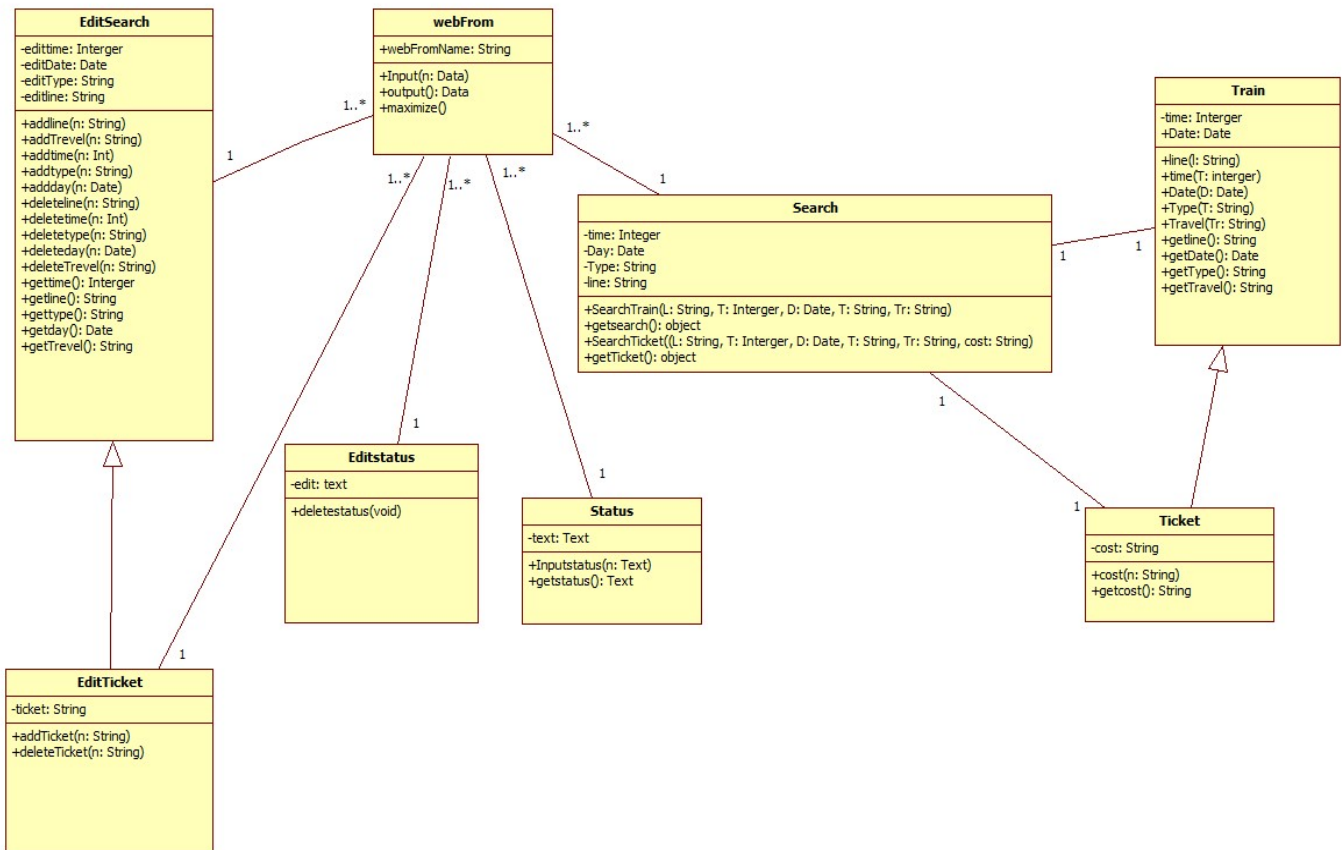
- 1.webFrom จะทำการ ล็อกอิน(login) เข้ามาในระบบและรับข้อมูลจาก ผู้ใช้งาน ผ่านทางฟังก์ชัน Input ()
- 2.เมื่อรับข้อมูลมาเข้าฟังก์ชัน addline() เพื่อ เพิ่มข้อมูลของสายรถไฟ
- 3.เพิ่มข้อมูลลงในฐานข้อมูลของsearch
- 4.จะใช้ฟังก์ชัน getline () เพื่อบอกว่าเพิ่มข้อมูลสายรถไฟนั้นแล้ว

Search



- 1.webFrom จะทำการรับข้อมูลจาก ผู้ใช้งานผ่านทางฟังก์ชัน input ()
- 2.เมื่อรับข้อมูลมาเข้าฟังก์ชัน SearchTrain() เพื่อหาข้อมูลของสายรถไฟ
- 3.โดยใช้ฟังก์ชัน line() จะทำการตรวจสอบข้อมูลในฐานข้อมูล
- 4.ฐานข้อมูลจะทำการหาข้อมูลที่ตรงกับที่ ผู้ใช้(user) ต้องการ
- 5.จะใช้ฟังก์ชัน getline () เพื่อนำข้อมูลไปให้object Train
6. object Train ก็จะนำข้อมูลไปแสดงผลผ่านฟังก์ชัน getSearch()

Domain class



Class ในระบบการจัดการรถไฟ ประกอบด้วยดังนี้

Class WebFrom

-ทำหน้าที่ เป็น user interface โดยจะเป็นส่วนที่ติดต่อกับผู้ใช้งานโดยตรง โดยจะทำการรับข้อมูลและแสดงผลข้อมูลและมีความสัมพันธ์กับคลาสอื่นดังนี้

-คลาส Editsearch โดยมีความสัมพันธ์แบบหลายต่อหนึ่งคือ รับข้อมูลจากผู้ใช้งานหลายคน สามารถเลือกได้ว่าจะเพิ่มข้อมูลหรือลบข้อมูล

-คลาส EditTicket โดยมีความสัมพันธ์แบบหลายต่อหนึ่งคือรับข้อมูลจากผู้ใช้งานหลายคน สามารถเลือกได้ว่าจะเพิ่มข้อมูลหรือลบข้อมูล

-คลาส Editstatus โดยมีความสัมพันธ์แบบหลายต่อหนึ่งคือรับข้อมูลจากผู้ใช้งานหลายคน จะทำได้แค่ลบข้อมูล

-คลาส Search โดยมีความสัมพันธ์แบบหลายต่อหนึ่งคือรับข้อมูลจากผู้ใช้งานหลายคน จะทำได้ว่าจะค้นหาเส้นทางรถไฟหรือค้นหาราคาตั๋ว

-คลาส Status โดยมีความสัมพันธ์แบบหลายต่อหนึ่งรับข้อมูลจากผู้ใช้งานหลายคนจะทำได้เพียงการเพิ่มสถานะของรถไฟ

Method ภายใน Class WebFrom

Input(n:Data) ใช้ในการรับข้อมูลจากผู้ใช้งาน

Output(N:Data) ใช้ในการแสดงผลให้กับผู้ใช้งาน

Maximize() ใช้ในตรวจสอบผู้ใช้งานสูงสุดที่ระบบจะรับได้

Class Search

จะทำหน้าที่รับข้อมูลจากผู้ใช้งานว่าจะทำการค้นหาเส้นทางรถไฟหรือค้นหาราคาตั๋ว

จะมีความสัมพันธ์กับคลาสอื่นดังนี้

คลาส webFrom โดยมีความสัมพันธ์แบบหนึ่งต่อหลายคือจะเลือกที่จะแสดงข้อมูลการค้นหาเส้นทางรถไฟหรือข้อมูลการค้นหาราคาตั๋ว

คลาส Train โดยมีความสัมพันธ์แบบหนึ่งต่อหนึ่งคือถ้าเลือกการค้นหาเส้นทางรถไฟคลาส Train ก็รับข้อมูลไปค้นหาในฐานข้อมูลและแสดงข้อมูลเส้นทางรถไฟนั้นกลับมา

คลาส Ticket โดยมีความสัมพันธ์แบบหนึ่งต่อหนึ่งคือถ้าเลือกการค้นหาราคาตั๋วรถไฟคลาส Ticket ก็รับข้อมูลไปค้นหาในฐานข้อมูลและแสดงข้อมูลราคาตั๋วรถไฟนั้นกลับมา

Method ที่ใช้ในคลาส Search

SearchTrain(L: String, T: Integer, D: Date, Tr: String) ใช้ในการค้นหาเส้นทางรถไฟโดยรับข้อมูลจากผู้ใช้งาน

getsearch(): object ใช้ในการแสดงผลของการค้นหาเส้นทาง

SearchTicket((L: String, T: Integer, D: Date, Tr: String, cost: String) ใช้ในการค้นหาราคาตั๋วโดยรับข้อมูลจากผู้ใช้งาน

getTicket(): object ใช้ในการแสดงผลของราคาตั๋ว

Class Editsearch

ทำหน้าที่เพิ่มหรือลบข้อมูลที่เกี่ยวข้องกับการค้นหาเส้นทางรถไฟ

จะมีความสัมพันธ์กับคลาสอื่นดังนี้

คลาส webFrom โดยมีความสัมพันธ์แบบหนึ่งต่อหลายคือจะเลือกได้ว่าจะเพิ่มหรือลบข้อมูลในการค้นหาเส้นทางรถไฟ

Method ใน Class EditSearch

โดยแบ่งเป็นส่วนของการทำงานดังนี้ เช่น

addline(n: String) ใช้ในการเพิ่มสายรถไฟ

addTrevel(n: String) ใช้ในการเพิ่มการเที่ยวไปหรือเที่ยวกลับ

addtype(n: String) ใช้ในการเพิ่มชนิดของรถไฟ

deleteline(n: String) ใช้ในการลบสายรถไฟ

getline(): String ใช้ในการแสดงสายรถไฟ

Class Ticket

ทำหน้าที่ในการรับข้อมูลจาก คลาส Search มาค้นหาราคาตั๋วของรถไฟและส่วนที่เกี่ยวข้องกับราคาตั๋วรถไฟ

จะมีความสัมพันธ์กับคลาสอื่นดังนี้

คลาส Search โดยมีความสัมพันธ์แบบหนึ่งต่อหนึ่งคือจะเลือกกว่าค้นหาราคาตั๋วรถไฟ

คลาส Train แบบสืบทอดคุณสมบัติกันคือจะต้องมีส่วนที่เกี่ยวข้องกับราคาตั๋วของรถไฟ

Method ใน คลาส Ticket เช่น

cost(n: String) ใช้ในการค้นหาราคาตั๋ว

getcost(): String ใช้ในการแสดงราคาตั๋ว

ine(l: String) ใช้ในการค้นหาสายรถไฟที่เกี่ยวข้องกับราคาตั๋ว

Class Editticket

ทำหน้าที่ การเพิ่มและลบ ราคาตั๋วรถไฟและส่วนที่เกี่ยวข้องกับราคาตั๋วของรถไฟ

จะมีความสัมพันธ์กับคลาสอื่นดังนี้

คลาส Editsearch เป็นความสัมพันธ์แบบสืบทอดคุณสมบัติกันโดย คลาส EditTicket จะต้องมีส่วนที่เกี่ยวข้องคือ การเพิ่มสาย การเพิ่มวันเวลา ซึ่งเกี่ยวข้องกับตัวเป็นต้น

คลาส webFrom โดยมีความสัมพันธ์แบบหนึ่งต่อหลายคือจะเลือกได้ว่าจะเพิ่มหรือลบข้อมูลในตัวขบวนรถไฟ

Method ใน คลาส EditTicket เช่น

addTicket(n: String) ใช้ในการเพิ่มราคาตั๋ว

deleteTicket(n: String) ใช้ในการลบราคาตั๋ว

addline(n: String) ใช้ในการเพิ่มสายรถไฟของตั๋วรถไฟ

Class Status

ทำหน้าที่การแจ้งสถานะของรถไฟ

จะมีความสัมพันธ์กับคลาสอื่นดังนี้

คลาส webFrom โดยมีความสัมพันธ์แบบหนึ่งต่อหลายคือจะนำข้อมูลแจ้งสถานะไปแสดงผล

Method ใน Status

Inputstatus(n: Text) ใช้การรับสถานะของรถไฟ

getstatus(): Text ใช้ในการแสดงผลของสถานะรถไฟ

Class Editstatus

ทำหน้าที่ลบสถานะของรถไฟ

จะมีความสัมพันธ์กับคลาสอื่นดังนี้

คลาส webFrom โดยมีความสัมพันธ์แบบหนึ่งต่อหลายคือลบข้อมูลแจ้งสถานะ

Method ใน Editstatus deletestatus(void)ใช้ในการลบสถานะ

Deployment

-ใช้ ภาษา Mysql ใช้จัดการฐานข้อมูล

Application Server

-ใช้ Microsoft Azure(VM เลข IP: 23.101.28.14)

-ใช้ Apache (Visual Server)

-ใช้ phpmyadmin เป็นตัวจัดการฐานข้อมูล

-ใช้ CodeIgniter เป็น frame work ที่ใช้ในการพัฒนา

-ใช้ HTML, CSS และ JavaScript

Implementation plan

ลำดับ	งานที่ได้รับผิดชอบ	วันเริ่ม	วันสำเร็จ	ระยะเวลา	ผู้รับผิดชอบ	19/9/2014	1/10/2014	1/11/2014
						12222222223 901234567890	1111111111222222222233 0123456789012345678901	111111 01234
1	ศึกษา CodeIgniter frame work	19/9/2014	24/9/2014	6 วัน	Thanawat Boonrit	<div></div>		
2	ออกแบบโครงสร้างหน้า เว็บ	25/9/2014	26/9/2014	2 วัน	Thanawat Boonrit	<div></div>		
3	ทำระบบเพิ่ม ลบ แก้ไข และออกแบบฐานข้อมูล	26/9/2014	7/10/2014	12 วัน	Thanawat	<div></div>		
4	ทำระบบค้นหาสินค้า ทางและการดูตัว	26/9/2014	7/10/2014	12 วัน	Boonrit	<div></div>		
5	ทำระบบแจ้งสถานะ	8/10/2014	19/10/2014	12 วัน	Thanawat Boonrit		<div></div>	
6	ทำระบบล็อกอิน	20/10/2014	30/10/2014	11วัน	Thanawat Boonrit		<div></div>	
7	ทดสอบระบบต่าง ๆ และ แก้ไขข้อผิดพลาด	30/10/2014	14/11/2014	16 วัน	Thanawat Boonrit			<div></div>

