

Project No. 6

ระบบเชื่อมต่อข้อมูลการออกกำลังกายของ Strava<sup>(TM)</sup>  
สำหรับจักรยานอยู่กับที่และลู่วิ่ง  
(Strava<sup>(TM)</sup> Activity Data Gateway API for Stationary Cycling  
and Running Systems)

จัดทำโดย

นางสาว ธนัชพร ประกอบทรัพย์ 55070501065

ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร. มารอง ผดุงสิทธิ์

“ข้าพเจ้าขอรับว่าได้อ่านเนื้อหาภายในรายงานฉบับนี้แล้ว”

.....  
(.....)

อาจารย์ที่ปรึกษา



**ระบบเชื่อมต่อข้อมูลการออกกำลังกายของ Strava<sup>(TM)</sup>  
สำหรับจักรยานอยู่กับที่และลู่วิ่ง  
(Strava<sup>(TM)</sup> Activity Data Gateway API for Stationary Cycling  
and Running Systems)**

นางสาว ธนัชพร ประกอบทรัพย์

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
ปีการศึกษา 2558

ระบบเชื่อมต่อข้อมูลการออกกำลังกายของ Strava(TM) สำหรับจักรยานอยู่กับที่และลู่วิ่ง

นางสาว ธนัชพร ประกอบทรัพย์

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
ปีการศึกษา 2558  
ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

คณะกรรมการสอบโครงการ

..... ประธานกรรมการสอบวิทยานิพนธ์และอาจารย์ที่ปรึกษา  
(ผู้ช่วยศาสตราจารย์ ดร. มารอง ผดุงสิทธิ์)

..... กรรมการ  
(รองศาสตราจารย์ ดร. ชำรงรัตน์ อมรรักษ์)

..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ สนั่น สระแก้ว )

..... กรรมการ  
(อาจารย์ จุมพล พลวิชัย)

Strava<sup>(TM)</sup> Activity Data Gateway API for Stationary Cycling and Running Systems

Miss Thanatchaphorn Prakopsap

A Project Submitted in Partial Fulfillment of the Requirements  
for the Degree of Bachelor of Engineering  
Department of Computer Engineering, Faculty of Engineering  
King Mongkut's University of Technology Thonburi  
Academic Year 2015

Project Committee

..... Chairman and advisor  
(Asst. Prof Marong Phadoongsidhi, Ph.D.)

..... Committee  
(Assoc. Prof. Thumrongrat Amornraksa, Ph.D.)

..... Committee  
(Asst. Prof Sanan Srakaew )

..... Committee  
(Jumpol Polvichai, Ph.D.)

หัวข้อโครงการ	ระบบเชื่อมต่อข้อมูลการออกกำลังกายของ Strava(TM) สำหรับจักรยานอยู่กับที่และลู่วิ่ง
หน่วยกิตของโครงการ	7 หน่วยกิต
จัดทำโดย	นางสาว ธนัชพร ประกอบทรัพย์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. มารอง ผดุงสิทธิ์
ระดับการศึกษา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2558

### บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อแก้ไขปัญหาของผู้ที่ชื่นชอบในการออกกำลังกายและนักกีฬาที่ออกกำลังกายโดยใช้เครื่องออกกำลังกายในฟิตเนส เช่น ลู่วิ่ง จักรยานเสมือนและอื่น ๆ แต่ไม่สามารถที่จะนำข้อมูลที่ได้จากการออกกำลังกายในครั้งนั้นไปจัดเก็บรวบรวมข้อมูลเอาไว้เพื่อไปวิเคราะห์ผลในด้านต่างๆได้ โครงการนี้จึงจะทำหน้าที่เป็นตัวกลางในการเชื่อมต่อข้อมูลต่างๆที่ผู้ใช้ได้จากการออกกำลังกายไปเก็บไว้บนเว็บไซต์ที่มีไว้เพื่อการรวบรวมข้อมูลจากการออกกำลังกาย และเป็นตัวกลางในการนำข้อมูลของผู้ใช้ไปประยุกต์ใช้งานในด้านอื่นๆเช่น แอปพลิเคชันที่มีความเกี่ยวข้องกับสุขภาพหรือแอปพลิเคชันที่เกี่ยวกับการออกกำลังกาย โดยโครงการนี้จะจำกัดเครื่องออกกำลังกายที่จะรองรับและทำการเชื่อมต่อกับระบบเพียงแค่ ลู่วิ่งและจักรยานปั่นอยู่กับที่เท่านั้น

Project Title	Strava(TM) Activity Data Gateway API for Stationary Cycling and Running Systems
Project Credit	7 credits
Project Participant	Miss Thanatchaphorn Prakopsap
Advisor	Asst. Prof. Dr. Marong Phadoongsidhi
Degree of Study	Bachelor's Degree
Department	Computer Engineering
Academic Year	2015

### Abstract

This project is designed to solve the problems of those who like to exercise or athletes who exercise using exercise machines in the fitness such as a treadmill or virtual cycling and other. But the inability to use information gained from the exercise to collect data for analysis the results in various fields. This project will act as an intermediary in connection information received from the exercise of the user to keep it on the website is intended to collect information from exercise and as an intermediary in bringing user's information to apply on the other side such as application that associated with health or exercise. This project will limit the exercise machine to accommodate and connect to the system just treadmill and virtual cycling only.

## กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก ผู้ช่วยศาสตราจารย์ ดร. มารอง ผดุงสิทธิ์ ที่ปรึกษาโครงการ ที่กรุณาใช้เวลาให้คำปรึกษา ให้ข้อมูล แนวคิด แก้ไขปัญหา และข้อเสนอแนะต่างๆ ที่เป็นประโยชน์อย่างมาก ตลอดจนคอยติดตามดูแลเอาใจใส่ต่อการทำโครงการฉบับนี้จนสำเร็จลุล่วงได้ด้วยดี ผู้จัดทำโครงการจึงขอกราบขอบพระคุณเป็นอย่างสูง ณ ที่นี้ด้วย

ขอขอบพระคุณ รองศาสตราจารย์ ดร. ชำรงรัตน์ อมรรักษ์ ผู้ช่วยศาสตราจารย์ สนั่น สระแก้ว และอาจารย์ จุมพล พลวิชัย ที่ได้สละเวลาร่วมเป็นคณะกรรมการตรวจสอบโครงการในครั้งนี้

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีที่เอื้อเฟื้อสถานที่ห้อง Senior Project และอุปกรณ์ในการทำ โครงการในครั้งนี้

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และพี่ๆ ที่ให้กำลังใจ ให้ความช่วยเหลือ และให้คำปรึกษาตลอดมา ขอขอบคุณเพื่อนๆ พี่ๆ และน้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ทุกคนที่ให้กำลังใจ คอยดูแล และอยู่เคียงข้างกันเสมอมา

## สารบัญ

	หน้า
<b>บทที่ 1 บทนำ</b>	<b>1</b>
1.1 ความสำคัญและที่มาของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนการทำงาน	2
<b>บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง</b>	<b>4</b>
2.1 ระบบจักรยานเสมือน MixVR	4
2.2 การบริการเชื่อมต่อในการบันทึกข้อมูลและอุปกรณ์ Activity tracker	5
2.2.1 <a href="http://www.strava.com">http://www.strava.com</a>	5
2.2.2 <a href="http://www.endomondo.com">http://www.endomondo.com</a>	7
2.2.3 <a href="http://connect.garmin.com">http://connect.garmin.com</a>	7
2.2.4 Comparison features	8
2.3 เครื่องมือและเทคโนโลยีที่ใช้ในการพัฒนา	9
2.3.1 Node.js	9
2.3.2 RESTful Web Service	10
2.3.3 OAuth	10
2.3.4 ฐานข้อมูล MongoDB	11
2.3.5 Digitalocean	11
2.3.6 Postman	12
2.3.7 Bitbucket	12
<b>บทที่ 3 วิธีการดำเนินโครงการ</b>	<b>13</b>
3.1 อุปกรณ์ที่ใช้	13
3.2 ภาษาและชุดเครื่องมือที่ช่วยในการพัฒนา	13
3.2.1 ภาษาที่ใช้ในการพัฒนา	13
3.2.2 เครื่องมือที่ใช้ในการพัฒนา	13



## สารบัญ(ต่อ)

	หน้า
3.3 ขั้นตอนการดำเนินโครงการ	16
3.4 การออกแบบและพัฒนาโปรแกรมสำหรับเชื่อมต่อข้อมูล	16
3.4.1 Use Case diagram	17
3.4.2 Sequence diagram	19
3.4.3 การเชื่อมต่อในการอัปโหลดไฟล์ข้อมูล	19
3.5 การเก็บข้อมูล	20
3.5.1 userdetails	20
3.5.2 originalfile	20
<b>บทที่ 4 ผลการทดลองและการอภิปรายผลการทดลอง</b>	<b>21</b>
4.1 วิเคราะห์และออกแบบโครงสร้างในการทำงาน	21
4.2 ผลจากการให้ผู้ใช้ทำการยินยอมในการเข้าถึงข้อมูล	22
4.3 ผลจากการดึงข้อมูล การอัปโหลดไฟล์และการบันทึกข้อมูล	22
4.2.1 ผลการดึงข้อมูล Profile	23
4.2.2 ผลการดึงข้อมูล Activity	24
4.2.3 ผลการอัปโหลดข้อมูล	24
4.2.4 ผลการบันทึกข้อมูลลงฐานข้อมูล	26
<b>บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ</b>	<b>28</b>
5.1 สรุปผลการทดลอง	28
5.2 ปัญหาและอุปสรรคในการทดลอง	28
5.3 ข้อเสนอแนะและแนวทางในการพัฒนา	29
<b>บรรณานุกรม</b>	<b>30</b>

## รายการรูปภาพประกอบ

รูปภาพ	หน้า
รูปที่ 2.1 Virtual cycling MixVR	5
รูปที่ 2.2 รูปแบบหน้า <a href="http://www.strava.com">http://www.strava.com</a>	6
รูปที่ 2.3 รูปแบบหน้า <a href="http://www.endomondo.com">http://www.endomondo.com</a>	7
รูปที่ 2.4 รูปแบบหน้า <a href="http://connect.garmin.com">http://connect.garmin.com</a>	8
รูปที่ 2.5 การอัปโหลดข้อมูลที่ได้จากการออกกำลังกายผ่านทาง Device	10
รูปที่ 2.6 การอัปโหลดข้อมูลที่ได้จากการออกกำลังกายผ่านทางแอปพลิเคชัน	10
รูปที่ 2.7 การอัปโหลดข้อมูลที่ได้จากการออกกำลังกายแบบ Manual	11
รูปที่ 2.8 การอัปโหลดข้อมูลที่ได้จากการออกกำลังกายผ่านทางแอปพลิเคชัน	11
รูปที่ 2.9 ตัวอย่างของ Node.js ที่นำมาใช้งาน	12
รูปที่ 2.10 ตัวอย่างของ Digitalocean	13
รูปที่ 2.11 ตัวอย่างของ Postman	14
รูปที่ 3.1 การทำงานในภาพรวมของโครงการ	16
รูปที่ 3.2 ภาพรวมของระบบ API ในการทำงาน	17
รูปที่ 3.3 Use case diagram	17
รูปที่ 3.4 Sequence diagram	19
รูปที่ 4.1 ตัวอย่างหน้าที่ใช้ในการ Authentication	22
รูปที่ 4.2 ตัวอย่างผลการทำการ Authentication	22
รูปที่ 4.3 ตัวอย่างผลการบันทึกข้อมูลของผู้ใช้ลงในฐานข้อมูล	23
รูปที่ 4.4 ตัวอย่างผลการดึงข้อมูล Profile ของผู้ใช้จาก strava	23
รูปที่ 4.5 ตัวอย่างผลการดึงข้อมูล Activity ของผู้ใช้จาก strava	24
รูปที่ 4.6 การทดสอบการอัปโหลดไฟล์โดย Postman	25
รูปที่ 4.7 แสดงผลจากการอัปโหลดไฟล์ของผู้ใช้ไปยัง strava	25
รูปที่ 4.8 แสดงไฟล์ข้อมูลของผู้ใช้ก่อนการอัปโหลด	26
รูปที่ 4.9 แสดงไฟล์ข้อมูลของผู้ใช้หลังการอัปโหลด	26
รูปที่ 4.10 แสดงผลการบันทึกข้อมูลในส่วน of userdetails	27
รูปที่ 4.11 แสดงผลการบันทึกข้อมูลในส่วน of originalfile	27

## รายการตารางประกอบ

ตาราง	หน้า
ตารางที่ 2.1 ชนิดของการออกกำลังกายที่รองรับ	6
ตารางที่ 2.2 Comparison features	8
ตารางที่ 2.3 แสดงลักษณะการทำงานของ Method ใน RESTful Web service	12
ตารางที่ 2.4 เปรียบเทียบระยะเวลาที่ใช้ในการทำงานระหว่าง MySQL และ MongoDB	13
ตารางที่ 3.1 Parameters ที่ใช้ในการรับส่งข้อมูลในการอัปโหลดไฟล์	19
ตารางที่ 3.2 แสดงคุณลักษณะต่างๆของฐานข้อมูลที่เป็นประเภท userdetails	20
ตารางที่ 3.3 แสดงคุณลักษณะต่างๆของฐานข้อมูลที่เป็นประเภท originalfile	20
ตารางที่ 4.1 ตัวอย่างของการบันทึกข้อมูล Profile ของผู้ใช้ใน userdetails	26
ตารางที่ 4.2 ตัวอย่างการอัปโหลดข้อมูลการออกกำลังกายของผู้ใช้จาก strava	27
ตารางที่ 5.2 ปัญหาและแนวทางการแก้ไขปัญหา	28

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบันนี้ ผู้คนเริ่มหันมาให้ความสนใจเกี่ยวกับการออกกำลังกายเพิ่มมากขึ้น ไม่ว่าจะเป็นกีฬากลางแจ้ง หรือแม้กระทั่งกีฬาที่สามารถออกกำลังกายได้ภายในฟิตเนส โดยส่วนมากแล้ว หลังจากการที่ผู้ใช้ได้ออกกำลังกายเสร็จ อุปกรณ์กีฬาต่างๆ มักจะมีการแจ้งค่าหรือข้อมูลที่ได้จากการออกกำลังกายในครั้งนั้นผ่านทางหน้าจอ แต่กลับไม่ได้มีการบันทึกข้อมูลเก็บไว้เพื่อให้ผู้ใช้นั้นสามารถนำไปใช้หรือวิเคราะห์ในด้านต่างๆ ได้

จากปัญหาดังนั้น ทางผู้จัดทำจึงคิดที่จะจัดทำเครื่องมือที่ทำหน้าที่เป็นศูนย์กลางในการเชื่อมต่อข้อมูลที่ได้จากการออกกำลังกายของผู้ใช้ในรูปแบบต่างๆ ให้สามารถนำไปเก็บไว้บนเว็บไซต์ที่เกี่ยวกับการบันทึกผลทางด้านการออกกำลังกาย ซึ่งจะใช้วิธีการรับ-ส่ง ข้อมูลผ่านทาง API (Application Programming Interface) และยังเป็นตัวกลางในการนำข้อมูลของผู้ใช้จากเว็บไซต์ต่างๆ ที่ใช้ในการเก็บข้อมูลไปประยุกต์ในการใช้งานด้านอื่นๆ อีกด้วย

ทางผู้จัดทำได้เล็งเห็นว่าโครงการนี้ สามารถนำไปใช้ในการเชื่อมต่อข้อมูลต่างๆ ที่เกี่ยวกับการออกกำลังกายในแง่ของการบันทึกและการนำข้อมูลไปใช้ได้อย่างหลากหลาย ซึ่งทำให้ผู้ใช้งานมีความสะดวกในการบันทึกข้อมูลต่างๆ ได้ดียิ่งขึ้น ไม่ต้องนำข้อมูลที่ได้จากการออกกำลังกายไปบันทึกลงตามเว็บไซต์ต่างๆ ที่ต้องใช้อุปกรณ์คอมพิวเตอร์หรืออุปกรณ์อื่นๆ เป็นตัวช่วย และเป็นประโยชน์ต่อการนำไปพัฒนาแอปพลิเคชันต่างๆ ที่มีความเกี่ยวข้องกับการออกกำลังกายได้อีกด้วย

### 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1. พัฒนาระบบในการเชื่อมต่อข้อมูลระหว่างอุปกรณ์กีฬากับเว็บไซต์ที่เกี่ยวกับการบันทึกผลทางด้านการออกกำลังกาย(Strava)
- 1.2.2. พัฒนาระบบให้ง่ายต่อการใช้งานของ user ในการจัดเก็บข้อมูลและ developers ในการพัฒนา application ต่างๆ
- 1.2.3. แก้ไขปัญหาในเรื่องของการเก็บข้อมูลหลังจากที่ user ได้ทำการออกกำลังกายเสร็จให้สามารถจัดเก็บข้อมูลได้สะดวกเพิ่มมากขึ้น

## 1.3 ขอบเขตของโครงการ

- 1.3.1 สร้างเครื่องมือเพื่อทำการเชื่อมต่อ Social API เข้ากับ Strava API
- 1.3.2 สร้างเครื่องมือเพื่อทำการเชื่อมต่อระหว่างเครื่องออกกำลังกายกับ Social API (Bike API)
- 1.3.3 ใช้ mongoDB ในการจัดการกับฐานข้อมูล พัฒนาเครื่องมือที่ใช้ในการเชื่อมต่อโดย Node.js

## 1.4. ขั้นตอนการทำงานและระยะเวลาการดำเนินงาน

### ขั้นตอนการทำงาน

1. การวางแผนโครงการ
  - 1.1. การดำเนินงานโครงการ
  - 1.2. ศึกษา รวบรวมข้อมูลและเลือกโครงการที่สนใจ
  - 1.3. ปรึกษาโครงการกับอาจารย์ที่ปรึกษา
  - 1.4. กำหนดขอบเขตของโครงการ
  - 1.5. จัดทำเอกสารโครงสร้งรายงาน
  - 1.6. จัดทำเอกสารนำเสนอโครงสร้งรายงาน
  - 1.7. จัดทำเอกสารนำเสนอโครงร่างรายงานครั้งที่ 1 (draft project#1)
  - 1.8. จัดทำเอกสารนำเสนอโครงร่างรายงานครั้งที่ 2 (draft project#2)
  - 1.9. จัดทำเอกสารนำเสนอโครงร่างรายงานครั้งที่ 3 (draft project#3)
2. การดำเนินงานโครงการ
  - 2.1. ศึกษา รวบรวมข้อมูลที่ต้องใช้ในการทำโครงการ
  - 2.2. ศึกษา API ของ เว็บไซต์ Strava
  - 2.3. ศึกษาการทำงานและโครงสร้างของ API
  - 2.4. ศึกษาวิธีการสร้าง API ในส่วนต่างๆ
  - 2.5. ทำการสร้าง API ในส่วนของ Social API
  - 2.6. ทำการเชื่อมต่อระหว่าง API Strava กับ API Social
  - 2.7. ทำการทดลองเรียกข้อมูลผ่านทาง Social API ไปยัง Strava และแก้ไขระบบ
  - 2.8. ทำการสร้าง API ในส่วนของ Bike API

### 3. การรวบรวมโครงงาน

3.1. ทำการเชื่อมต่อระหว่าง Social API และ Bike API

3.2. ทำการทดลองเรียกข้อมูลผ่านทาง Bike API ไปยัง Strava และแก้ไขระบบ

3.3. ทำการทดลองการอัปโหลดข้อมูลจาก Bike ขึ้นไปยัง เว็บไซต์ Strava และแก้ไขระบบ

3.4. ทดสอบการทำงานร่วมกันของระบบ

3.5. จัดทำเอกสารรายงานโครงงาน

3.6. จัดทำเอกสารนำเสนอโครงงาน

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

โครงการชิ้นนี้เป็นการทำงานในการเชื่อมต่อข้อมูลที่เกี่ยวข้องกับการออกกำลังกาย เพื่อนำไปบันทึกไว้บนเว็บไซต์ที่ให้บริการเกี่ยวกับการบันทึกผลที่ได้จากการออกกำลังกาย โดยผู้ใช้เพียงระบุความต้องการที่เกี่ยวกับข้อมูลว่าต้องการที่จะทำอะไร เช่น อัฟโพลด์ ระบบจะทำการเชื่อมต่อข้อมูลต่างๆของผู้ใช้ในลักษณะต่างๆตามความต้องการของผู้ใช้ และในการที่ข้อมูลในการออกกำลังกายนั้นถูกบันทึกไว้บนเว็บไซต์ที่ให้บริการ จะทำให้ง่ายต่อการที่นักพัฒนาแอปพลิเคชันต่างๆ จะนำข้อมูลดังกล่าวไปใช้ประโยชน์ทางด้านอื่นๆต่อไป โดยได้รับความยินยอมจากตัวของผู้ใช้เอง

จากที่ผู้จัดทำได้ศึกษาถึงเว็บไซต์ที่ให้บริการในการเก็บข้อมูลทางด้านการออกกำลังกายนั้น มีเว็บไซต์ที่ให้บริการในการเก็บข้อมูลทางด้านการออกกำลังกาย คือ [www.strava.com](http://www.strava.com) โดยทางเว็บไซต์ strava นั้น ได้มีการพัฒนา API ให้สำหรับนักพัฒนานำไปใช้งานได้อีกด้วย แต่โดยปกติแล้วทางผู้พัฒนานั้นจะต้องเชื่อมต่อกับทาง [www.strava.com](http://www.strava.com) โดยตรง แต่โครงการชิ้นนี้จะทำหน้าที่เป็นศูนย์กลางในการเชื่อมต่อระหว่าง [www.strava.com](http://www.strava.com) และ แอปพลิเคชันต่างๆ ได้สะดวกมากขึ้น โดยแอปพลิเคชันอื่นๆที่ต้องการใช้ข้อมูลประเภทนี้ จะไม่ต้องไปเชื่อมต่อกับทาง [www.strava.com](http://www.strava.com) โดยตรง และการบันทึกข้อมูลที่ได้จากการออกกำลังกายไว้บนเว็บไซต์นั้น ทาง [www.strava.com](http://www.strava.com) ไม่ได้ทำการเก็บไฟล์ข้อมูลต้นฉบับและเมื่ออัฟโพลด์ไฟล์ข้อมูลไปเก็บไว้บนเว็บไซต์แล้ว ข้อมูลจะถูกประมวลผลให้กลายเป็นค่าเฉลี่ย ทางผู้จัดทำจึงได้สร้างฐานข้อมูลขึ้นมาเพื่อทำการเก็บไฟล์ต้นฉบับเอาไว้ เพราะไฟล์ต้นฉบับนั้นมีความละเอียดมากกว่าข้อมูลที่แสดงผลอยู่บนเว็บไซต์ strava โดยในบทนี้จะกล่าวถึง ระบบจักรยานเสมือน MixVR ที่มีความเกี่ยวข้องกับโครงการในส่วนของการเชื่อมต่อระหว่างระบบกับแอปพลิเคชัน เว็บไซต์ที่ให้บริการในการเชื่อมต่อในการบันทึกข้อมูลและอุปกรณ์ต่างๆ รวมไปถึงเครื่องมือและเทคโนโลยีที่ใช้ในการพัฒนาโครงการนี้ด้วย

#### 2.1 ระบบจักรยานเสมือน MixVR

เนื่องจากการปั่นจักรยานนั้น มีความได้เปรียบทางด้านความคล่องตัวในการใช้งาน ที่จอดจักรยานที่ไม่ต้องใช้พื้นที่มาก ไม่เปลืองน้ำมันที่ต้องใช้เป็นเชื้อเพลิงในพาหนะชนิดอื่นๆ จักรยานนั้นจึงได้รับความนิยมอย่างมากในปัจจุบัน แต่การปั่นจักรยานนั้นก็ยังมีอุปสรรคหลายอย่าง เช่น ความไม่สมบูรณ์ของเส้นทางในการปั่น การเกิดอุบัติเหตุต่างๆ สภาพอากาศ และการปั่นที่อาจจะไปไม่ถึงปลายทางอันเนื่องมาจากระยะทางหรือความพร้อมของตัวผู้ปั่นเอง

ดังนั้น MixVR จึงเป็นทางเลือกของการปั่นจักรยานเสมือน ที่เน้นไปทางด้านการออกกำลังกายรวมไปถึงการเล่นเกมด้วย โดยผู้ใช้นั้นสามารถที่จะปั่นจักรยานเสมือนจริงได้ในร่ม โดยจักรยาน

เสมือนนั้นจะมีการเชื่อมต่อกับอุปกรณ์ โปรโตคอลต่างๆ โดยการป้อนนั้น จะเป็นการจำลองมาจากค่าที่ได้จาก GPS, Activity tracker และกล้องวิดีโอ เมื่อผู้ใช้ได้ปั่นจักรยานแล้วจะได้รับความรู้สึกเหมือนกับได้ไปปั่นตามสถานที่นั้นจริงๆ เป็นทางเลือกที่เหมาะสมกับผู้ที่มีความชื่นชอบในการปั่นจักรยานที่ดีอีกทางหนึ่ง

แต่ในปัจจุบันระบบจักรยานเสมือน หรืออุปกรณ์กีฬาต่างๆยังไม่สามารถที่จะนำข้อมูลที่ได้จากการออกกำลังกายไปใช้งานหรือเก็บบันทึกไว้ในที่ต่างๆได้โดยตรง จะต้องจดบันทึกข้อมูลที่ได้แล้วไปบันทึกลงในเว็บไซต์ที่ให้บริการทางด้านการบันทึกข้อมูลที่ได้จากการออกกำลังกายด้วยตนเอง ทางผู้จัดทำจึงเล็งเห็นว่า หากผู้ออกกำลังกายได้ทำการออกกำลังกายเสร็จแล้ว สามารถอัปโหลดข้อมูลหรือบันทึกข้อมูลที่ได้จากการออกกำลังกายในครั้งนั้นได้โดยตรงคือ สามารถอัปโหลดข้อมูลหรือบันทึกข้อมูลผ่านทางโปรแกรมที่ถูกติดตั้งไว้กับเครื่องออกกำลังกาย ไปยังเว็บไซต์ที่ให้บริการในการบันทึกข้อมูลได้ทันที น่าจะทำให้ผู้ใช้มีความสะดวกสบายในการใช้งานมากยิ่งขึ้น



รูปที่ 2.1 Virtual cycling MixVR



## 2.2 การบริการเชื่อมต่อในการบันทึกข้อมูลและอุปกรณ์ Activity trackers

### 2.2.1 <https://www.strava.com>

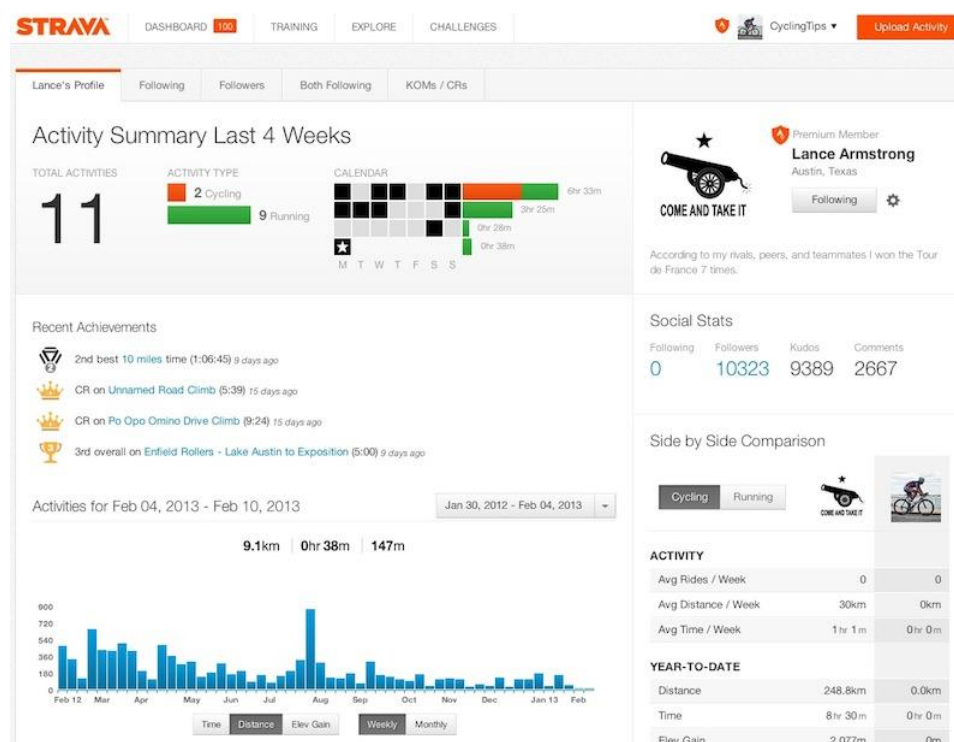
Strava เป็นเว็บไซต์ที่รวบรวมผู้ที่มีความชื่นชอบในการออกกำลังกายและนักกีฬาเป็นจำนวนมาก โดยเว็บไซต์นี้มีบริการให้กับการบันทึกข้อมูลในที่ได้จากการออกกำลังกายในรูปแบบต่างๆ โดยสามารถบันทึกชนิดของการออกกำลังกายได้มากถึง 29 ชนิดดังตารางที่ 2.1

ตารางที่ 2.1 ชนิดของการออกกำลังกายที่รองรับ

Ride	Canoeing	Kite surf	Snowboard	Yoga
Run	Cross fit	Nordic Ski	Snowshoe	Backcountry Ski
Swim	E-Bike Ride	Rock Climbing	Stair Stepper	Kayaking
Hike	Elliptical	Roller Ski	Standup Paddling	Virtual Ride
Walk	Ice-skate	Rowing	Weight Training	Workout
Alpine Ski	Inline Skate	Surfing	Windsurf	

[ที่มา: <http://strava.github.io/api/v3/activities/>]

จากการที่ [www.strava.com](http://www.strava.com) รองรับการใช้งานในหลายชนิดกีฬาและมีรูปแบบของหน้าเว็บไซต์ที่น่าสนใจ ตามรูปที่ 2.2 จึงทำให้ได้รับความนิยมเป็นจำนวนมาก



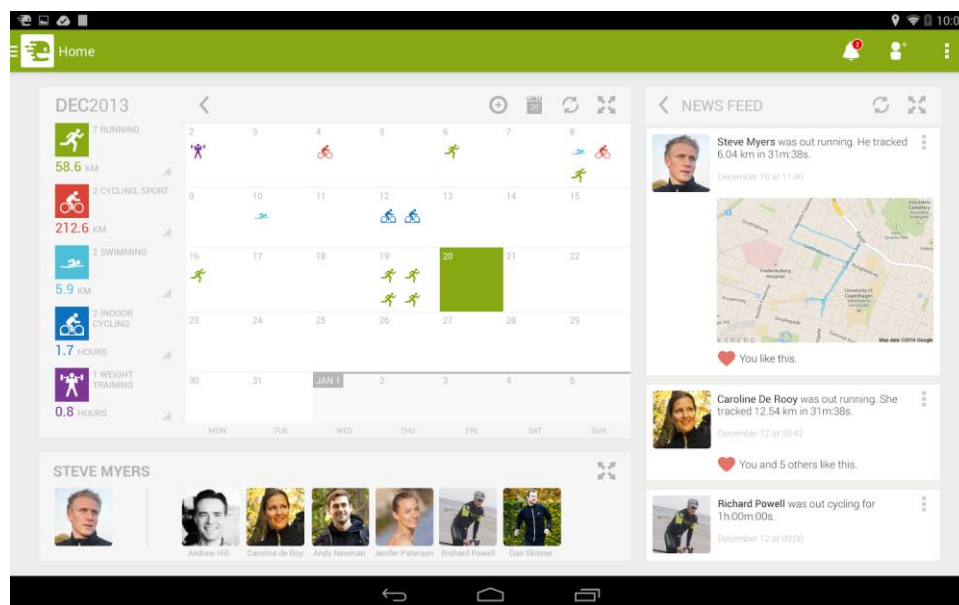
รูปที่ 2.2 รูปแบบหน้า <http://www.strava.com>

[ที่มา: <http://cyclingtips.com.au/2013/02/lance-armstrong-on-strava/>]

โดยทาง [www.strava.com](http://www.strava.com) นั้น ได้มีการเปิดให้ใช้ Strava V3 API ซึ่งเป็นอินเตอร์เฟซที่เปิดเผยต่อสาธารณชน ที่ช่วยให้นักพัฒนาสามารถที่จะเข้าถึงชุดข้อมูลของ Strava โดยอินเตอร์เฟซนี้รองรับทั้งการพัฒนาในส่วนของ web application และ mobile application ซึ่งไลบรารี รองรับหลากหลายภาษาที่ต้องการใช้ในการพัฒนา เช่น Java, Perl และ PHP เป็นต้น

#### 2.2.2 <https://www.endomondo.com>

Endomondo เป็นเว็บไซต์ที่ถูกออกแบบมาเพื่อติดตามการออกกำลังกาย โดยจะมีการให้คำแนะนำต่างๆเกี่ยวกับการออกกำลังกาย มีการฝึกอบรม วิเคราะห์การออกกำลังกายของผู้ใช้ โดยทาง endomondo นั้น ได้มีการพัฒนาอุปกรณ์ ต่างๆเป็นของตนเองด้วย

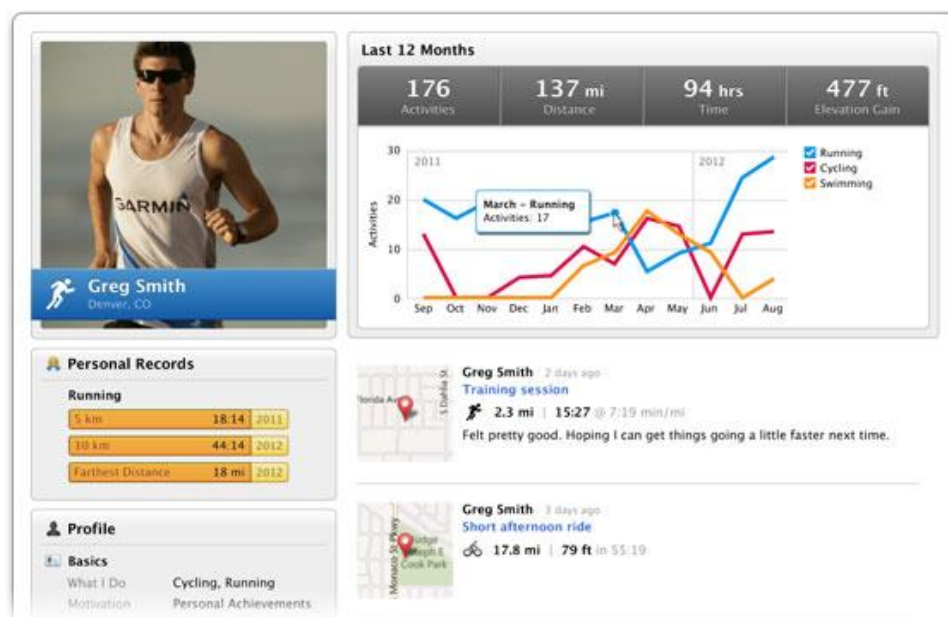


รูปที่ 2.3 รูปแบบหน้า <http://www.endomondo.com>

[ที่มา: <http://mob-core.com/th/programmy-dlja-android/3163-endomondo.html>]

#### 2.2.3 <https://connect.garmin.com>

Garmin connect เป็นเครื่องมือในการฝึกอบรมเรื่องต่างๆในการออกกำลังกายออนไลน์ โดยสามารถวิเคราะห์การออกกำลังกาย และมีการแบ่งปันข้อมูลต่างๆให้กับผู้ที่ใช้ garmin connect เหมือนกัน โดย garmin connect นั้น จะมีการเชื่อมต่อกับอุปกรณ์ของ garmin เอง ในการบันทึกผลต่างๆ



รูปที่ 2.4 รูปแบบหน้า <https://connect.garmin.com>

[ที่มา: <https://connect.garmin.com/features/share?ticket=ST-0768574-6VAQjXc35DgexLO0fdb5-cas>]

#### 2.2.4 Comparison features

จากการศึกษาเว็บไซต์ที่ให้บริการทางด้านการเชื่อมต่อข้อมูลนั้น สรุปได้ตารางที่

2.2

ตารางที่ 2.2 Comparison features

Features	Website		
	Strava	Endomondo	Garmin
1 บริการการจัดเก็บข้อมูล	✓	✓	✓
2 อุปกรณ์ที่รองรับ			
a. CATEYE	✓		
b. Fitbit	✓	✓	
c. Garmin	✓	✓	✓
d. Microsoft Band	✓		
e. Soleus	✓		
f. Suunto	✓		
g. Timex	✓	✓	
h. TomTom	✓		
i. iPhone	✓		

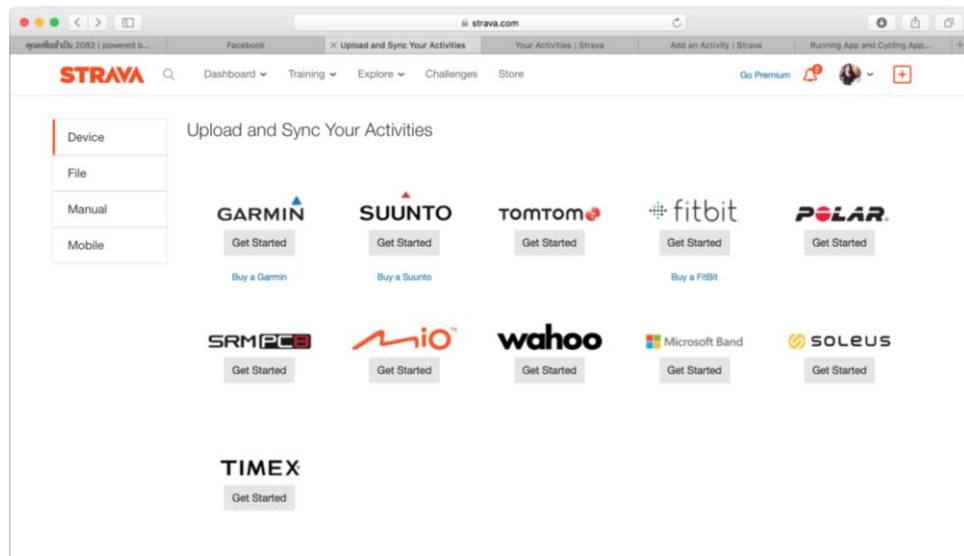
ตารางที่ 2.2 Comparison features (ต่อ)

Features	Website		
	Strava	Endomondo	Garmin
j. Android	✓		
k. Jabra		✓	
l. Myfitnesspal		✓	
m. Polar		✓	
n. Wahoo		✓	
3 ไฟล์ข้อมูลที่รองรับ			
a. .tcx	✓	✓	✓
b. .fit	✓	✓	
c. .gpx	✓	✓	✓
4 Development	✓	✓	✓

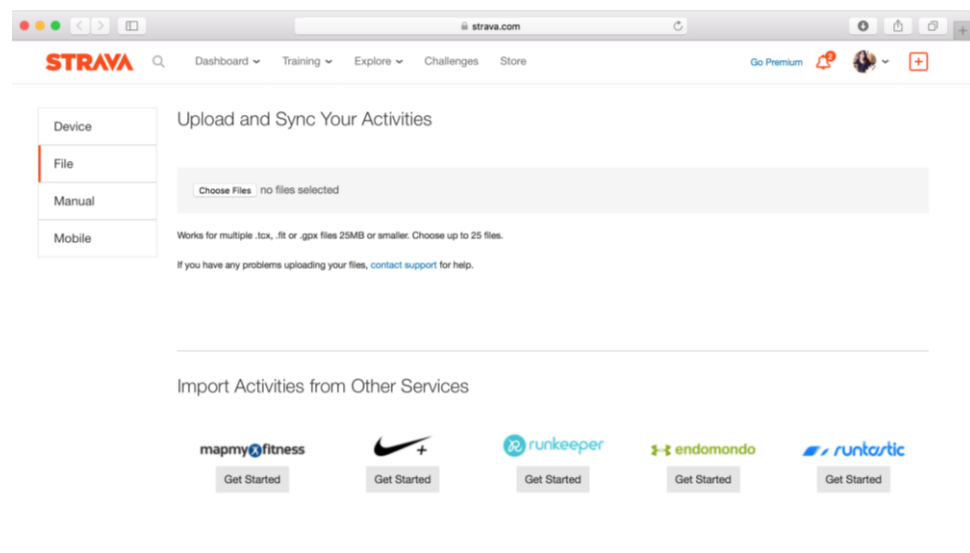
จากตาราง จะเห็นได้ว่า ในส่วนของการบริการการจัดเก็บข้อมูลนั้น สามารถทำได้ทั้ง 3 เว็บไซต์ ในส่วนของอุปกรณ์ที่รองรับนั้น strava สามารถรองรับอุปกรณ์ได้ทั้งหมด 10 ชนิด ซึ่งมากกว่า endomondo และ garmin ที่สามารถรองรับได้ 7 และ 1 ชนิดตามลำดับ ในส่วนของไฟล์ข้อมูลที่รองรับนั้น strava และ endomondo สามารถรองรับไฟล์ข้อมูลได้ทั้งหมด คือ .tcx, .fit และ .gpx แต่ garmin สามารถรองรับได้เพียง 2 ประเภท คือ .tcx และ .gpx และสุดท้ายในส่วนของการพัฒนาทุกเว็บไซต์ มีการเปิดให้นักพัฒนาสามารถเข้าไปพัฒนาระบบที่มีความเกี่ยวข้องได้ จากข้อมูลทั้งหมดนี้ ผู้จัดทำ จึงได้เลือก strava เป็นเว็บไซต์ที่ใช้ในการเก็บบันทึกข้อมูลที่ได้จากการออกกำลังกายในครั้งนี้

แต่ [www.strava.com](http://www.strava.com) ก็ยังมีข้อจำกัดในหลายด้าน เช่น การเชื่อมต่อแอปพลิเคชันต่างๆ เข้ากับ [www.strava.com](http://www.strava.com) เพราะ Strava จำกัดให้ 1 Account ของนักพัฒนาแอปพลิเคชันสามารถสร้างแอปพลิเคชันได้เพียง 1 แอปพลิเคชันเท่านั้น และในส่วนของการอัปโหลดข้อมูลที่ได้จากการออกกำลังกายนั้น ถึงแม้ [www.strava.com](http://www.strava.com) จะมีทางเลือกในการอัปโหลดข้อมูลที่ได้จากการออกกำลังกายอย่างหลากหลาย คือ 1. อัปโหลดผ่านทาง Device ดังรูปที่ 2.5 ซึ่งการอัปโหลดข้อมูลประเภทนี้ ผู้ใช้จำเป็นต้องมี Device ที่เกี่ยวข้องกับการบันทึกผลข้อมูลที่ได้จากการออกกำลังกาย ซึ่งเป็นอุปกรณ์ที่สามารถเชื่อมต่อกับเว็บไซต์ที่มีบริการทางด้านการเก็บบันทึกข้อมูลได้มักมีราคาสูง และในขั้นตอนการอัปโหลดนั้นก็ยังจำเป็นที่จะต้องใช้คอมพิวเตอร์ในการทำการอัปโหลดข้อมูล 2. การอัปโหลดข้อมูลผ่านการอัปโหลดไฟล์ข้อมูลโดยตรง ดังรูปที่ 2.6 โดยวิธีนี้ผู้ใช้จะต้องทำการเซฟไฟล์ที่ได้จากการออกกำลังกายมาก่อน แล้วค่อยมาเลือกไฟล์ที่ต้องการจะอัปโหลดในภายหลังผ่านทาง

คอมพิวเตอร์ 3. การอัปโหลดข้อมูลแบบ Manual ดังรูปที่ 2.7 การอัปโหลดข้อมูลในรูปแบบนี้ จะมีความยุ่งยากคือ ผู้ใช้จะต้องทำการกรอกข้อมูลที่ได้จากการออกกำลังกายทั้งหมดด้วยตนเอง ทำให้เกิดความล่าช้าและความไม่สะดวกขึ้น และรูปแบบสุดท้ายคือ การอัปโหลดข้อมูลผ่านทางแอปพลิเคชันของ strava ดังรูปที่ 2.8 แต่การอัปโหลดข้อมูลในรูปแบบนี้จะได้ข้อมูลที่ครบถ้วน ทำให้ไม่สามารถบันทึกข้อมูลที่ได้จากการออกกำลังกายไว้ได้ทั้งหมด



รูปที่ 2.5 การอัปโหลดข้อมูลที่ได้จากการออกกำลังกายผ่านทาง Device  
[ที่มา: <https://www.strava.com/upload/device>]



รูปที่ 2.6 การอัปโหลดข้อมูลที่ได้จากการออกกำลังกายผ่านทางไฟล์  
[ที่มา: <https://www.strava.com/upload/select>]

The screenshot shows the Strava 'Manual Entry' page. On the left is a sidebar with 'Device', 'File', 'Manual' (highlighted), and 'Mobile'. The main form has the following fields:

- Distance:** A text input followed by a dropdown menu set to 'kilometers'.
- Duration:** A time picker set to 01 hr, 00 min, and 00 s.
- Sport:** A dropdown menu set to 'Ride'.
- Date & Time:** Two input fields showing '05/24/2016' and '10:00 PM'.
- Title:** A text input containing 'Night Ride'.
- Type:** A dropdown menu.
- Tags:** Two buttons labeled 'Commute' and 'Trainer'.
- Bike:** A dropdown menu set to 'No Bike' with a '+ New Bike' link.
- Notes:** A text area with the placeholder text 'How did it go? Were you tired or rested? How was the weather?'.

รูปที่ 2.7 การอัปโหลดข้อมูลที่ได้จากการออกกำลังกายแบบ Manual

[ที่มา: <https://www.strava.com/upload/manual>]

The screenshot shows the Strava mobile app's 'Activity' screen. At the top are 'Cancel', 'Activity', and 'Save' buttons. Below are three tabs: 'Time', 'Distance', and 'Speed'. The main form has the following fields:

- Name:** 'Evening Ride'.
- Add a photo:** A button with a photo icon.
- Sport:** 'Ride'.
- Tag:** 'Select Tag'.
- When:** 'วันนี้ 20:23'.
- Description:** A text input field.
- Share your activity data with Health:** A button with a heart icon and a right arrow.

รูปที่ 2.8 การอัปโหลดข้อมูลที่ได้จากการออกกำลังกายผ่านทางแอปพลิเคชัน

[ที่มา: <https://www.strava.com/mobile>]

## 2.3 เครื่องมือและเทคโนโลยีที่ใช้ในการพัฒนา

### 2.3.1 Node.js

โครงการนี้ ผู้จัดทำได้เลือกใช้ Node.js ซึ่งมี Syntax ที่ใช้ในการเขียนคือ JavaScript เป็นภาษาที่มีการทำงานอยู่บนฝั่งของ Server โดยเป็นภาษาที่ถูกออกแบบมาเพื่อให้ทำงานแบบ Event Driven และการทำงานแบบ Asynchronous ซึ่งจะส่งผลในสามารถทำงานได้เร็ว

ขึ้นเพราะไม่ต้องรอให้คำสั่งก่อนหน้านี้ทำงานเสร็จก่อนจึงจะทำงานในขั้นต่อไปได้  
ตัวอย่าง Node.js ที่นำมาใช้ในโครงงานดังรูปที่ 2.5

```
app.get('/',function(req, res){
  var code = req.query.code;

  strava.oauth.getToken(code, function(err, result){
    console.log(err, result);           //get Profile
    //getListActivities(result.access_token); //get Activities
    //uploadFile(result.access_token);      //upload file
    //res.send(result);
    res.send('Hey!!');
  });

});

app.post("/gpx", multipath(), function(req, res){
  //console.log(req.files);
  var gpx = req.files.gpx;
  uploadFile("cbdd17216d4870b7f848336c97c1fcb1cb20aa95", gpx.path, req.body.name);
  res.send(req.body);
});
```

ภาพที่ 2.9 ตัวอย่างของ Node.js ที่นำมาใช้งาน

### 2.3.2 RESTful Web Service

REST เป็นการสร้าง web service อย่างเรียบง่าย โดยจะมีการเรียกใช้ผ่านทาง HTTP โดย method GET, POST, PUT และ DELETE และมีการส่งข้อมูลออกมาในรูปแบบของ XML

ตารางที่ 2.3 แสดงลักษณะการทำงานของ Method ใน RESTful Web Service

Method	ลักษณะการทำงาน
GET	เรียกข้อมูลมาแสดงแบบหลายรายการ
	เรียกข้อมูลมาแสดงแบบทีละรายการ
POST	ส่งข้อมูลจากฟอร์มหรือโปรแกรม เพื่อเพิ่มข้อมูล
PUT	ส่งข้อมูลจากฟอร์มหรือโปรแกรม เพื่อแก้ไขข้อมูล
DELETE	ส่งข้อมูลจากโปรแกรม เพื่อลบข้อมูล

[ที่มา: <http://www.softmelt.com/article.php?id=134>]

### 2.3.3 OAuth

#### 2.3.3 OAuth

OAuth คือ open authentication ซึ่งเป็นมาตรฐานที่ใช้สำหรับการกำหนดสิทธิ์ให้แอปพลิเคชัน ให้สามารถร้องขอข้อมูลของผู้ใช้จากแอปพลิเคชันหนึ่งได้โดยที่ไม่จำเป็นต้องทราบรหัสผ่านของผู้ใช้

### 2.3.4 ฐานข้อมูล MongoDB

ข้อมูลในการออกกำลังกายที่ได้จากผู้ใช้งาน เป็นข้อมูลที่ประกอบด้วยส่วนของ ประวัติของผู้ใช้งาน เช่น User ID, ชื่อ, ที่อยู่ เป็นต้น และผลที่ได้จากการออกกำลังกายของผู้ใช้งานแต่ละประเภทของการออกกำลังกาย ซึ่งข้อมูลเหล่านี้ล้วนเป็นข้อมูลเชิงเอกสารและมีปริมาณของข้อมูลเป็นจำนวนมาก ดังนั้นระบบของฐานข้อมูลที่เราเลือกมาใช้จึงมีความจำเป็นที่ต้องสอดคล้องกับการใช้งานในการทำโครงการ ทางผู้จัดทำจึงเลือกใช้ฐานข้อมูล MongoDB มาใช้ในการเก็บข้อมูลทั้งหมด

ฐานข้อมูล MongoDB เป็น open-source ซึ่งเป็นฐานข้อมูลที่เหมาะสมกับข้อมูลเชิงเอกสาร รองรับข้อมูลขนาดใหญ่ โดยใช้เวลาในการโหลดข้อมูลต่างๆ ได้เร็วกว่าเมื่อเทียบกับ MySQL ดังข้อมูลที่แสดงในตารางที่ 2.4

ตารางที่ 2.4 เปรียบเทียบระยะเวลาที่ใช้ในการทำงานระหว่าง MySQL และ MongoDB

การทำรายการจำนวน 10,000 ข้อมูล	MySQL(ms)	MongoDB(ms)
เวลาที่ใช้ในการโหลดข้อมูล	16,327	1,912
เวลาที่ใช้ในการบันทึกข้อมูล	31,151	6,484

[ที่มา: [http://www.scalability.com/blog/mongodb\\_vs\\_mysql](http://www.scalability.com/blog/mongodb_vs_mysql)] 2.3.6 MongoChef

### 2.3.5 Digitalocean

Digital Ocean เป็นผู้ให้บริการ Cloud Platform สำหรับ Developer โดย Digital Ocean มีบริการ Server ให้เลือกใช้ในหลายเขตพื้นที่ และมีประเทศ Singapore ให้บริการ ซึ่งถ้าเว็บของเราให้บริการกับคนไทยเป็นหลักก็เลือกวางที่ Singapore ซึ่งเหมาะกับผู้ที่ต้องการทำ Web Application แบบ Realtime และรองรับการพัฒนา Application ด้วยภาษา Node.js

Record Type	Name	Value	Actions
A	api	128.199.102.205	Save Remove
NS	ns1.digitalocean.com.		Save Remove
NS	ns2.digitalocean.com.		Save Remove

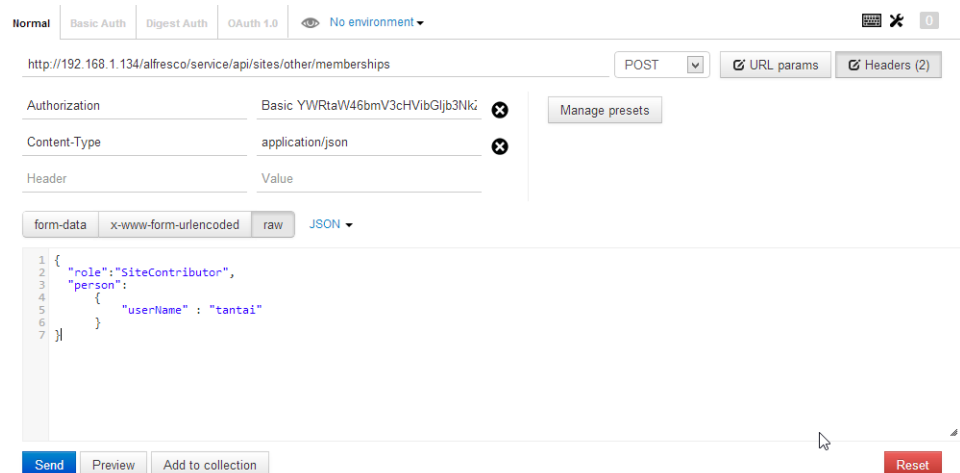
รูปที่ 2.10 ตัวอย่างของ Digitalocean



### 2.3.6 Postman

Postman คือ เครื่องมือที่ใช้ในการจำลองการ request คำสั่งต่างๆเข้ามาที่ระบบที่มีลักษณะเป็น Web API โดย Postman นั้นเป็นเครื่องมือที่สามารถใช้ได้ฟรีเพราะเป็น Google Chrome Extension และสามารถใช้งานได้ง่าย

ความสามารถของ Postman ในด้านการเก็บ History นั้น ในทุกครั้งที่เราทำการจำลองการ request คำสั่งต่างๆ เราสามารถย้อนกลับไปดู History ก่อนหน้าเพื่อทำการเปรียบเทียบข้อมูลกันได้โดยง่าย ส่วนการเก็บ Collection เมื่อมีคำสั่ง request ที่ต้องใช้งานเป็นประจำ ก็สามารถที่จะบันทึกลงใน collection เพื่อแบ่งหมวดหมู่การใช้งานและเป็นการทำให้สะดวกในการใช้งานในครั้งถัดไป และสุดท้ายคือการ Export ข้อมูล โดย Postman นั้นสามารถทำการส่งออกข้อมูลได้โดยง่าย โดยข้อมูลที่ส่งออกนั้นจะออกมาในรูปแบบของ file.json เพื่อให้ง่ายต่อการ import ไปใช้งานในครั้งต่อไป



รูปที่ 2.11 ตัวอย่างของ Postman

### 2.3.7 Bitbucket

Bitbucket เป็นเว็บไซต์ที่ให้บริการ Hosting สำหรับโครงการที่ต้องการฝากไฟล์ไปยัง Server โดยสามารถใช้งานในประเภท Private ได้ฟรีซึ่งแตกต่างจาก Github ที่ถ้าต้องการใช้งานแบบ Private ต้องเสียค่าใช้จ่ายในการใช้งาน โดย Bitbucket นั้นสามารถทำงานพร้อมกันได้หลายคนโดยไม่เกิดปัญหา และมีความง่ายและสะดวกต่อการใช้งาน

## บทที่ 3

### วิธีการดำเนินโครงการ

โครงการนี้เป็นโครงการที่ทำขึ้นเพื่อเชื่อมต่อข้อมูลที่เกี่ยวข้องกับการออกกำลังกาย โดยการพัฒนาต้องใช้คอมพิวเตอร์และเครื่องมือต่างๆในการพัฒนาโครงการ เพื่อให้โครงการมีประสิทธิภาพและสำเร็จ อุปกรณ์และซอฟต์แวร์ที่ใช้มีดังนี้

#### 3.1 อุปกรณ์ที่ใช้

##### คอมพิวเตอร์

- คอมพิวเตอร์แบบพกพา MacBook Pro รุ่น 13 นิ้ว

Processor 2.7 GHz Intel Core i5

Memory 8 GB 1867 MHz DDR3

Startup Disk Macintosh HD

- คอมพิวเตอร์แบบพกพา Lenovo IdeaPad Y480

CPU Intel Core i7-3630QM

Memory 8 GB DDR3

Operating System Windows 8

#### 3.2 ภาษาและชุดเครื่องมือที่ช่วยในการพัฒนา

##### 3.2.1 ภาษาที่ใช้ในการพัฒนา

1. Node.js
2. HTML5
3. JavaScript

##### 3.2.2 เครื่องมือที่ใช้ในการพัฒนา

1. Atom
2. Postman

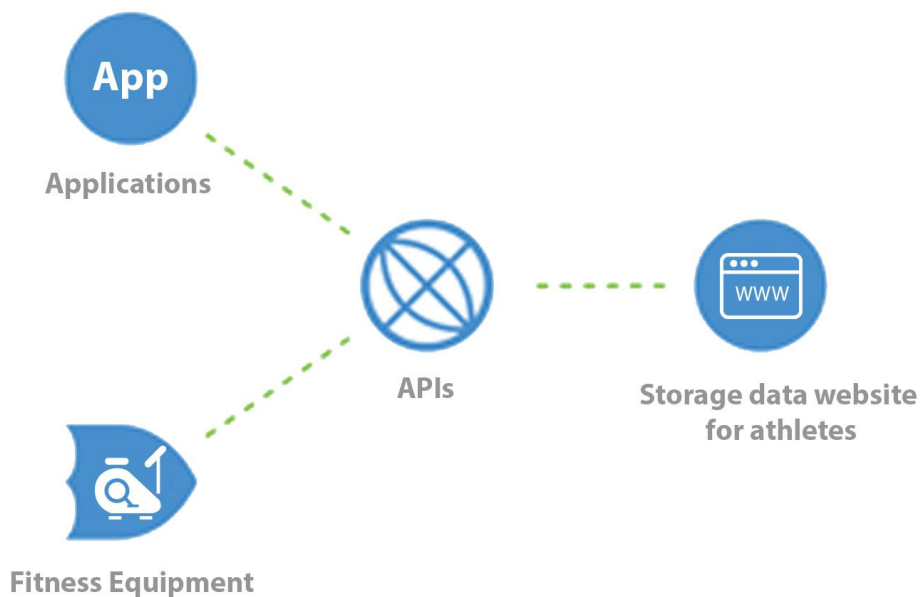
### 3.3 ขั้นตอนการดำเนินโครงการ

เพื่อให้งานที่ออกมามีประสิทธิภาพและตรงตามเวลาที่กำหนด ทางผู้จัดทำจึงได้แบ่งงานออกเป็น 4 ขั้นตอนคือ

- 3.2.1 ศึกษาข้อมูลที่จำเป็นจากแหล่งข้อมูลต่างๆ
- 3.2.2 วิเคราะห์และออกแบบโครงสร้างการทำงานโดยรวม
- 3.2.3 เขียนโปรแกรมจากที่ได้วิเคราะห์ไว้ ให้โปรแกรมสามารถทำการเชื่อมต่อข้อมูล
- 3.2.4 ทดสอบและพัฒนาโปรแกรม รวมถึงแก้ไขปัญหาที่เกิดขึ้น

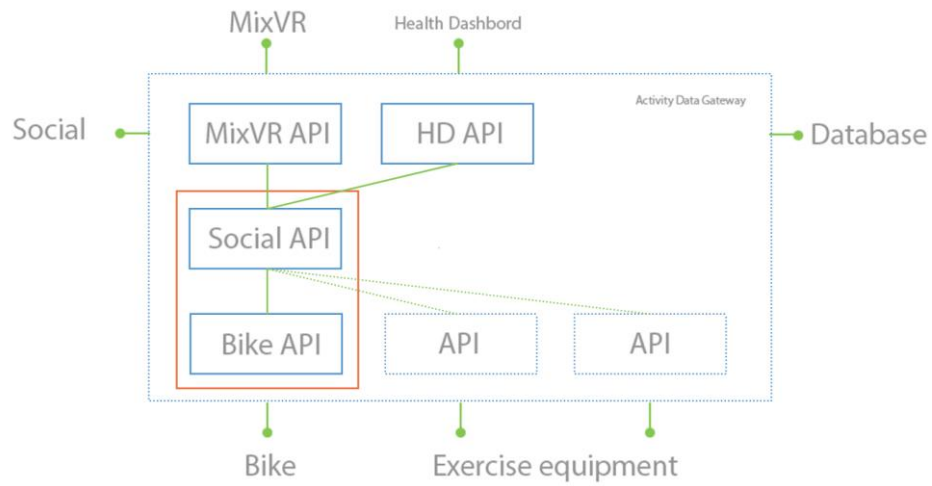
### 3.4 การออกแบบและพัฒนาโปรแกรมสำหรับเชื่อมต่อข้อมูล

โครงการนี้ จะมีส่วนของผู้ใช้ 2 รูปแบบ คือ ผู้ใช้ที่ใช้อุปกรณ์ออกกำลังกายที่ต้องการเชื่อมต่อข้อมูล และ ผู้ใช้ที่ต้องการเชื่อมต่อข้อมูลผ่านทางแอปพลิเคชันต่างๆ โดยผู้ใช้นั้นจะเชื่อมต่อข้อมูลผ่านทาง API ต่างๆที่สร้างขึ้นมาเพื่อรองรับการทำงานในการเชื่อมต่อข้อมูลทั้งหมดรวมทั้งเว็บไซต์ที่ให้บริการทางด้านการเก็บข้อมูลในการออกกำลังกายเข้าด้วยกันดังรูปที่ 3.1



รูปที่ 3.1 การทำงานในภาพรวมของโครงการ

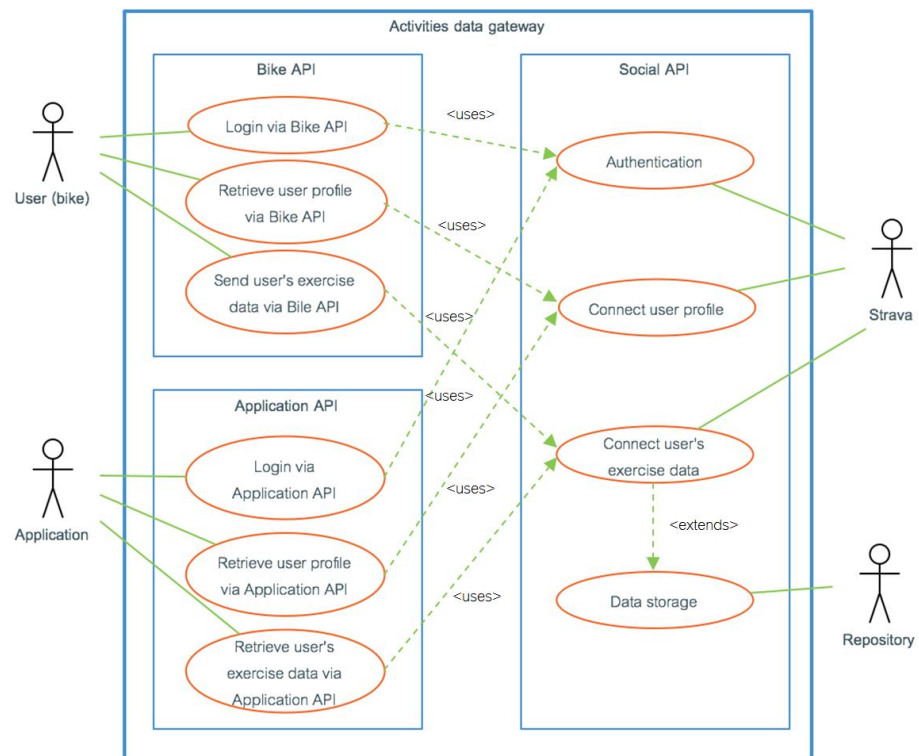
โดยในส่วนของ Fitness equipment จะรองรับทั้งหมด 2 อุปกรณ์คือ จักรยานอยู่กับที่และลู่วิ่งเท่านั้น และเว็บไซต์ที่ใช้ในการเก็บบันทึกข้อมูลที่ได้จากการออกกำลังกายของผู้ใช้ คือ [www.strava.com](http://www.strava.com) โดย ในส่วนของ Activity data gateway จะประกอบไปด้วย API ในส่วนต่างๆ ดังรูปที่ 3.2 โดยโครงการนี้จะทำในส่วนของ Social API และ Bike API



รูปที่ 3.2 ภาพรวมของระบบ API ในการทำงาน

### 3.4.1 Use Case diagram

จากรูปที่ 3.3 ในโครงงานนี้จะมีหน้าที่ในการทำ API ทั้งหมด 2 ส่วน คือ Bike API และ Social API



รูปที่ 3.3 Use case diagram

- Bike API

Bike API เป็น API ที่ทำหน้าที่ในการเชื่อมต่อระหว่าง ผู้ที่ใช้อุปกรณ์ออกกำลังกายที่ต้องการเชื่อมต่อข้อมูลกับ Social API โดยจะมีฟังก์ชันการทำงานดังต่อไปนี้

- User Authentication

เมื่อผู้ใช้ต้องการที่จะทำการเชื่อมต่อข้อมูล ผู้ใช้จะต้องทำการล็อกอินเพื่อทำการยืนยันในการให้ระบบเข้าถึงข้อมูลของผู้ใช้ในส่วนของบริษัทส่วนตัวและข้อมูลในการออกกำลังกายของผู้ใช้ที่มีอยู่ใน [www.strava.com](http://www.strava.com) ผ่านทาง [https://www.strava.com/oauth/authorize?client\\_id=8663&response\\_type=code&redirect\\_uri=http://139.59.238.160&scope=write&state=mystate&approval\\_prompt=force](https://www.strava.com/oauth/authorize?client_id=8663&response_type=code&redirect_uri=http://139.59.238.160&scope=write&state=mystate&approval_prompt=force)

- getProfile

เป็นการเรียกข้อมูลส่วนตัว, ประวัติของผู้ใช้จาก [www.strava.com](http://www.strava.com) ผ่านทาง url: [http://139.59.238.160/getProfile?id=\(id ของผู้ใช้งาน\)](http://139.59.238.160/getProfile?id=(id ของผู้ใช้งาน))

- uploadActivity

สำหรับผู้ใช้ที่ได้ออกกำลังกายเสร็จเรียบร้อยแล้ว และต้องการที่จะส่งข้อมูลการออกกำลังกายในครั้งนั้นไปเก็บไว้ที่ [www.strava.com](http://www.strava.com) เพื่อเป็นบันทึกประวัติและข้อมูลที่ได้ออกกำลังกาย ผ่านทาง url: <http://139.59.238.160/uploadActivity>

- Social API

Social API เป็น API ที่ทำหน้าที่ในการเชื่อมต่อระหว่าง Bike API กับ Strava API ของ [www.strava.com](http://www.strava.com) โดยจะมีฟังก์ชันการทำงานดังต่อไปนี้

- User Authentication

เมื่อมีการร้องขอในการ login เพื่อทำการเชื่อมต่อข้อมูล จะมีการให้ผู้นั้นทำการยินยอมให้ระบบเข้าถึงข้อมูลส่วนตัวที่อยู่ใน [www.strava.com](http://www.strava.com) เพื่อที่จะทำการเชื่อมต่อข้อมูล ผ่านทาง [https://www.strava.com/oauth/authorize?client\\_id=8663&response\\_type=code&redirect\\_uri=http://139.59.238.160&scope=write&state=mystate&approval\\_prompt=force](https://www.strava.com/oauth/authorize?client_id=8663&response_type=code&redirect_uri=http://139.59.238.160&scope=write&state=mystate&approval_prompt=force)

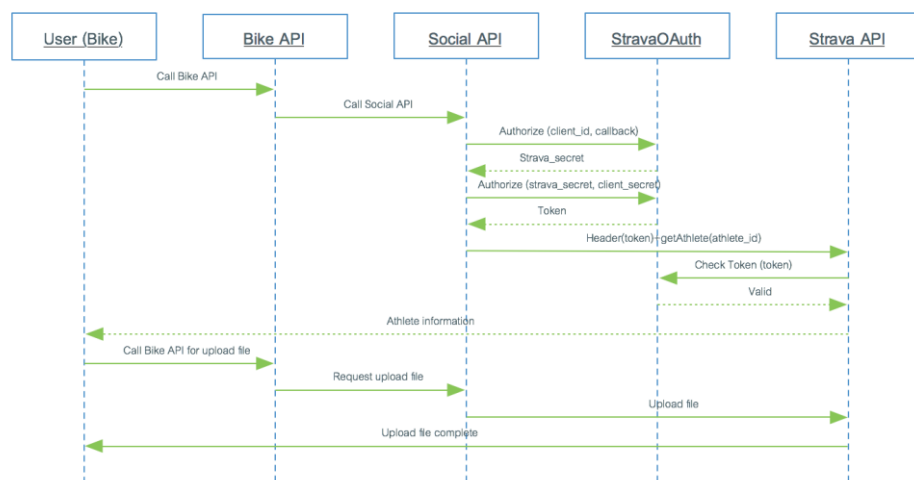
- getProfile

เป็นการเรียกข้อมูลส่วนตัว, ประวัติของผู้ใช้จาก [www.strava.com](http://www.strava.com) ผ่านทาง url: [http://139.59.238.160/getProfile?id=\(id ของผู้ใช้งาน\)](http://139.59.238.160/getProfile?id=(id ของผู้ใช้งาน))

### - getActivity

เป็นการเชื่อมต่อข้อมูลในส่วนของผู้ใช้ผ่านทางแอปพลิเคชันต่างๆ  
 getActivity จะทำหน้าที่ในการเรียกข้อมูลในการออกกำลังกายของผู้ใช้นั้นๆ  
 จากwww.strava.com ผ่านทาง url: [http://139.59.238.160/getActivity?id=\(idของผู้ใช้งาน\)](http://139.59.238.160/getActivity?id=(idของผู้ใช้งาน))

### 3.4.2 Sequence diagram



รูปที่ 3.4 Sequence diagram

### 3.4.3 การเชื่อมต่อในการอัปโหลดไฟล์ข้อมูล

ในการอัปโหลดไฟล์นั้น ผู้ที่ต้องการจะอัปโหลดไฟล์ จะต้องทำการส่งข้อมูล ผ่าน  
 ทาง url: [139.56.238.160/uploadActivity](http://139.56.238.160/uploadActivity) โครงานนี้จะรองรับไฟล์ที่เกี่ยวกับการออกกำลังกาย  
 ทั้งหมด 3 ประเภทคือ .fit, .tcx และ .gpx โดยมี Parameters ในการรับส่งข้อมูลดังนี้

ตารางที่ 3.1 Parameters ที่ใช้ในการรับส่งข้อมูลในการอัปโหลดไฟล์

Parameters	Details
id	รหัสของผู้ใช้งาน
activity_type	ประเภทของการออกกำลังกาย
name	ชื่อไฟล์
description	คำบรรยายกิจกรรม

ตารางที่ 3.1 Parameters ที่ใช้ในการรับส่งข้อมูลในการอัปโหลดไฟล์(ต่อ)

Paremeters	Details
private	ค่าความเป็นส่วนตัวในการเข้าถึงกิจกรรม
trainer	ผู้ฝึกสอน
possible values	ประเภทของไฟล์
external_id	รหัสไฟล์
file	ไฟล์ข้อมูลที่ต้องการอัปโหลด

### 3.5 การเก็บข้อมูล

หลังจากที่ผู้ใช้อัปโหลดข้อมูล หรือ มีการยืนยันเพื่อให้สามารถเข้าถึงข้อมูลของตัวผู้ใช้นั้นแล้ว ข้อมูลและรายละเอียดต่างๆจะมีการเก็บข้อมูลลงในฐานข้อมูลที่มีชื่อว่า strava ซึ่งฐานข้อมูลจะแบ่งประเภทของข้อมูลคือ userdetails และ originalfile ซึ่งแต่ละฐานข้อมูลจะเก็บข้อมูลแตกต่างกันออกไป ตามตารางคุณลักษณะต่างๆต่อไปนี้

#### 3.5.1 userdetails

userdetails จะเก็บข้อมูลในส่วนของ keys ต่างๆที่จำเป็นในการเข้าถึงข้อมูลเพื่อดึงข้อมูลในส่วนของ Profile, Activity รวมไปถึง original file ของผู้ใช้งาน ดังตารางที่ 3.1

ตารางที่ 3.2 แสดงคุณลักษณะต่างๆของฐานข้อมูลที่เป็นประเภท userdetails

Attributes	Details
id	รหัสของผู้ใช้งาน
access_token	รหัสในการเข้าใช้งานของผู้ใช้งาน

#### 3.5.2 originalfile

Originalfile จะเก็บข้อมูลในส่วน access\_token ที่ได้ทำการอัปโหลดข้อมูลการออกกำลังกายของผู้ใช้งาน และเก็บ original file การออกกำลังกาย ของผู้ใช้งาน ดังตารางที่ 3.2

ตารางที่ 3.3 แสดงคุณลักษณะต่างๆของฐานข้อมูลที่เป็นประเภท originalfile

Attributes	Details
access_token	รหัสในการเข้าใช้งานของผู้ใช้งาน
file	ไฟล์ข้อมูลการออกกำลังกายของผู้ใช้งาน

## บทที่ 4

### ผลการทดลองและการอภิปรายผลการทดลอง

หลังจากที่ได้มีการศึกษาถึงข้อมูลและวิธีการในการทำงานในส่วนต่างๆแล้ว ผู้จัดทำได้รวบรวมข้อมูลเหล่านั้นมาวิเคราะห์ เพื่อที่จะสร้างโครงสร้างการทำงานที่ครอบคลุมระบบ

#### 4.1 วิเคราะห์และออกแบบโครงสร้างในการทำงาน

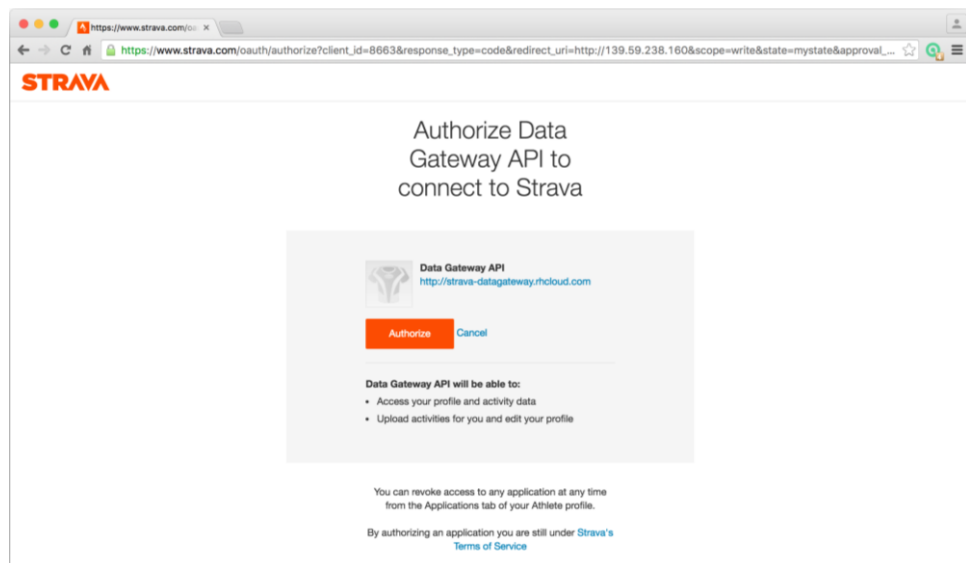
กระบวนการทำงานของระบบนั้น แบ่งได้ออกเป็น 2 ส่วน ส่วนแรกคือส่วนของการ ทำงาน ซึ่งเชื่อมต่อกับ MixVR โดยในส่วนนี้ระบบจะเริ่มจากเมื่อผู้ใช้ได้เริ่มต้นการใช้งานเครื่องออกกำลังกาย โดยผู้ใช้นั้นจะต้องทำการ Login เข้าสู่ระบบโดยผ่าน บัญชีผู้ใช้ของ Strava ในการ Login ระบบจะให้ผู้ใช้ทำการยืนยันสิทธิ์ในการเข้าถึงข้อมูลส่วนตัวต่างๆ โดยระบบจำทำการเรียกใช้ Bike API และเชื่อมต่อไปยัง Social API ว่าต้องการตรวจสอบบัญชีผู้ใช้ จะมีการ Request ไปยัง Strava เพื่อทำการตรวจสอบบัญชี เมื่อผู้ใช้ทำการ Login เสร็จ จะได้ access\_token ออกมา เพื่อใช้ในการดึงข้อมูลในส่วนต่างๆ โดยในขั้นแรกจะสามารถแสดงข้อมูลในส่วนของประวัติส่วนตัวของผู้ใช้ และเมื่อผู้ใช้ออกกำลังกาย(ปั่นจักรยาน)เสร็จเรียบร้อยแล้ว ผู้ใช้จะสามารถทำการอัปโหลดไฟล์ข้อมูลที่ได้จากการออกกำลังกายในครั้งนั้น เพื่อไปบันทึกไว้ที่ Strava และบันทึกไฟล์ต้นฉบับไว้ที่ฐานข้อมูลได้ในทันที ในส่วนที่ 2 คือส่วนที่เชื่อมต่อกับ แอปพลิเคชัน Dashboard โดยในส่วนนี้ จะให้ผู้ใช้ทำการ Login เข้าสู่ระบบผ่านบัญชีผู้ใช้ของ Strava เช่นเดียวกัน แต่นอกจากข้อมูลในส่วนของ Profile ของผู้ใช้นั้น ระบบจะสามารถทำการส่งข้อมูลในส่วนของ Activity ของผู้ใช้ แต่ในส่วนที่ 2 นี้จะไม่มี การอัปโหลดไฟล์ข้อมูลไปเก็บไว้ที่ Strava



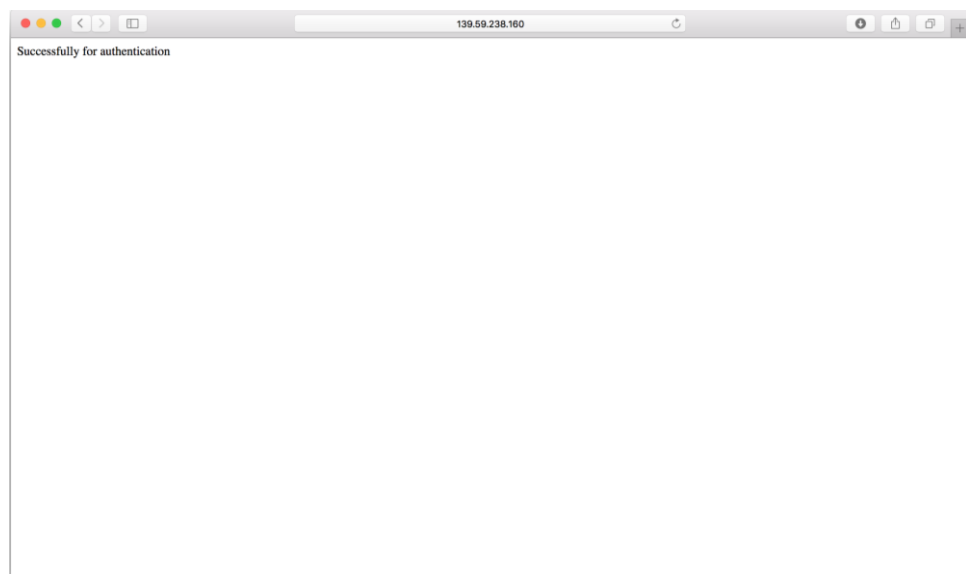
## 4.2 ผลจากการให้ผู้ใช้ทำการยินยอมในการเข้าถึงข้อมูล

จากการให้ผู้ใช้ทำการยินยอมการเข้าถึงข้อมูลนั้น ผู้ใช้หรือแอปพลิเคชันที่ต้องการจะเรียกใช้งานผ่านทาง url คือ `https://www.strava.com/oauth/authorize?client_id=8663&response_type=code&redirect_uri=http://139.59.238.160&scope=write&state=mystate&approval_prompt=force` ดังรูปที่ 4.1

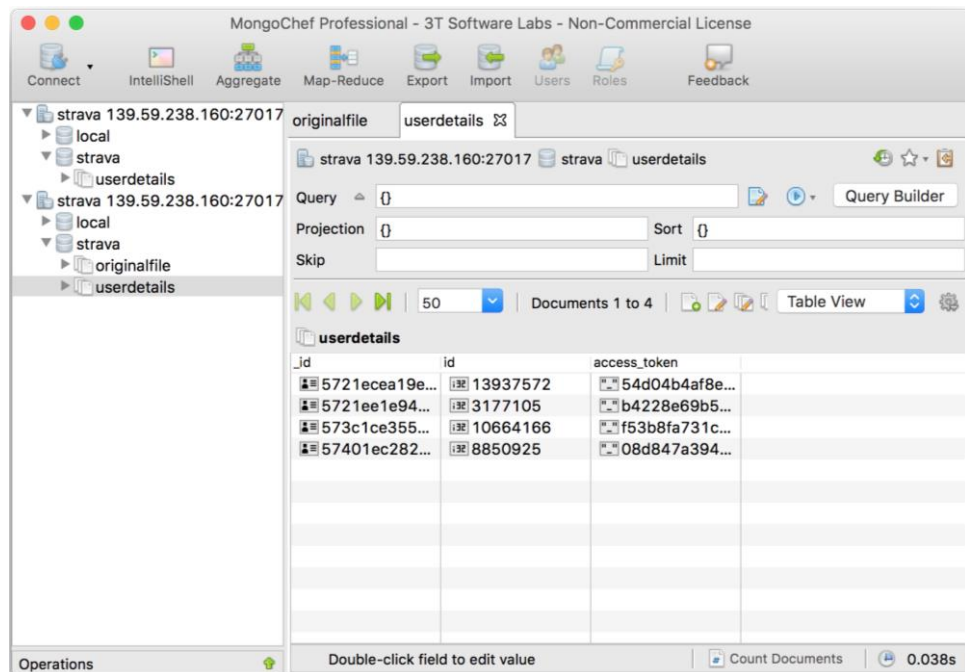
โดยผลลัพธ์เมื่อผู้ใช้ทำการยินยอมในการเข้าถึงข้อมูลนั้น จะแสดงดังรูปที่ 4.2 และระบบจะทำการบันทึก id และ access\_token ของผู้ใช้ไว้ในฐานข้อมูล ดังรูปที่ 4.3



รูปที่ 4.1 ตัวอย่างหน้าที่ใช้ในการ Authentication



รูปที่ 4.2 ตัวอย่างผลการทำการ Authentication



รูปที่ 4.3 ตัวอย่างผลการบันทึกข้อมูลของผู้ใช้ลงในฐานข้อมูล

## 4.3 ผลจากการดึงข้อมูล การอัปโหลดไฟล์และการบันทึกข้อมูล

### 4.3.1 ผลการดึงข้อมูล Profile

จากการดึงข้อมูลในส่วนของ Profile ของผู้ใช้นั้น ผู้ใช้หรือแอปพลิเคชันที่ต้องการจะต้องเรียกใช้งานผ่านทาง URL คือ 139.59.238.160/getProfile?id=(id ของผู้ใช้งาน)

โดยผลลัพธ์เมื่อมีการเรียกข้อมูลในส่วนของ Profile จะแสดงได้ดังรูปที่ 4.4 ซึ่งมีข้อมูลภายในที่ถูกเก็บไว้



รูปที่ 4.4 ตัวอย่างผลการดึงข้อมูล Profile ของผู้ใช้งานจาก Strava

#### 4.3.2 ผลการดึงข้อมูล Activity

จากการดึงข้อมูลในส่วนของ Activity ของผู้ใช้นั้น ผู้ใช้หรือแอปพลิเคชันที่ต้องการจะต้องเรียกใช้งานผ่านทาง URL คือ 139.59.238.160/getActivity?id=(id ของผู้ใช้งาน) โดยระบบจะไม่เก็บข้อมูลในส่วนของ activities เอาไว้

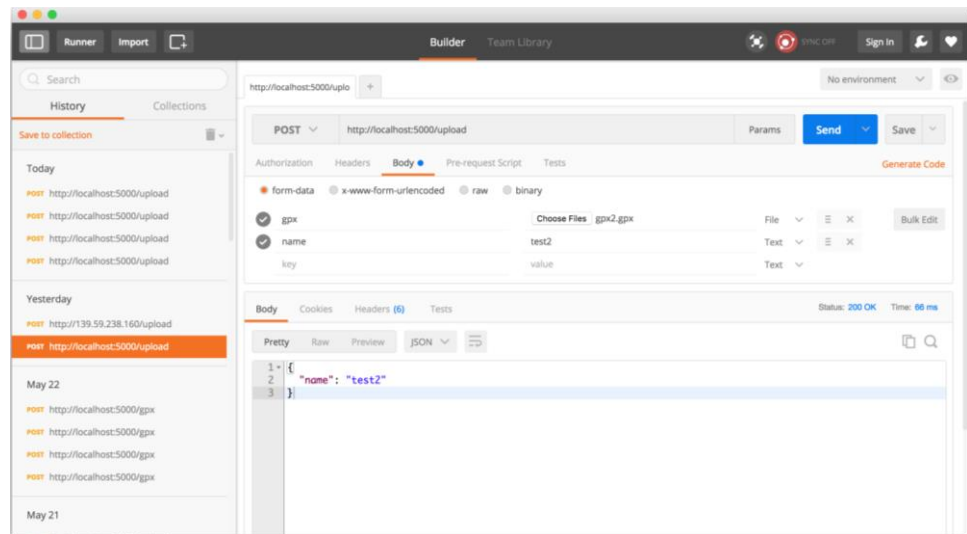
โดยเมื่อผลลัพธ์เมื่อมีการเรียกข้อมูลในส่วนของ Activity จะแสดงได้ดังรูปที่ 4.5



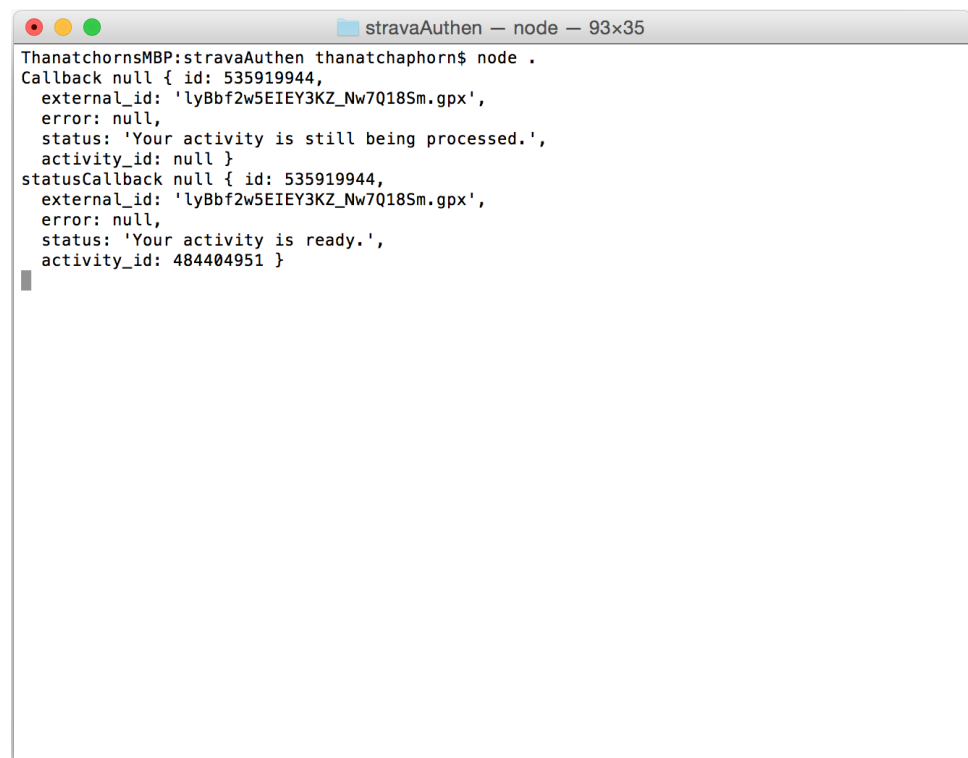
รูปที่ 4.5 ตัวอย่างผลการดึงข้อมูล Activity ของผู้ใช้งาน Strava

#### 4.3.3 ผลการอัปโหลดข้อมูล

ในการอัปโหลดข้อมูลไปเก็บไว้ที่ Strava นั้น สามารถอัปโหลดไฟล์ได้ทั้งหมด 3 ประเภท คือ .tcx, .gpx และ .fit โดยการอัปโหลดไฟล์นั้นจะต้องเรียกใช้งานผ่านทาง URL คือ 139.59.238.160/uploadActivity?id=(id ของผู้ใช้งาน) ผู้ใช้สามารถเปลี่ยนแปลงชื่อไฟล์ที่ต้องการอัปโหลดได้โดยไม่จำเป็นต้องเป็นชื่อไฟล์เดียวกันกับไฟล์ต้นฉบับ โดยสามารถตรวจสอบการอัปโหลดไฟล์ได้ผลดังรูปที่ 4.6

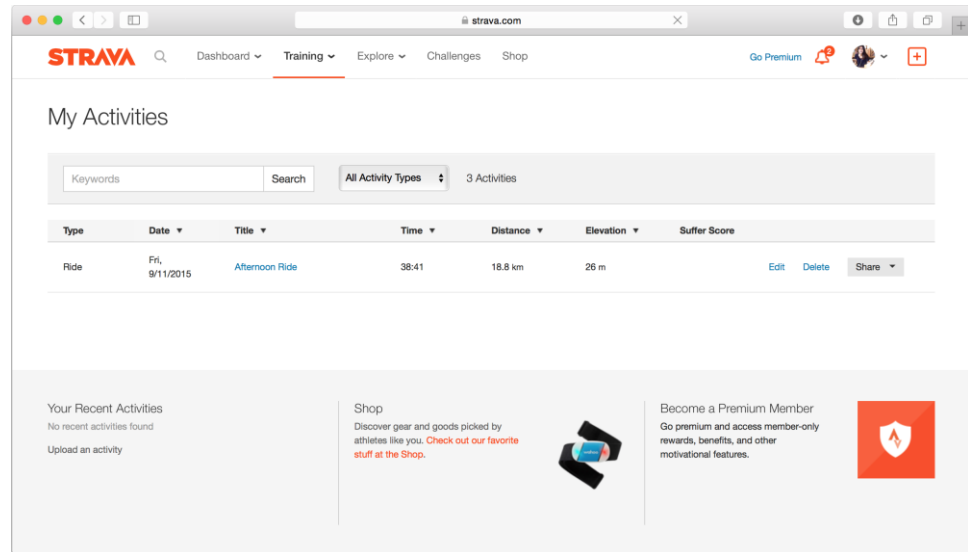


รูปที่ 4.6 การทดสอบการอัปโหลดไฟล์โดย Postman

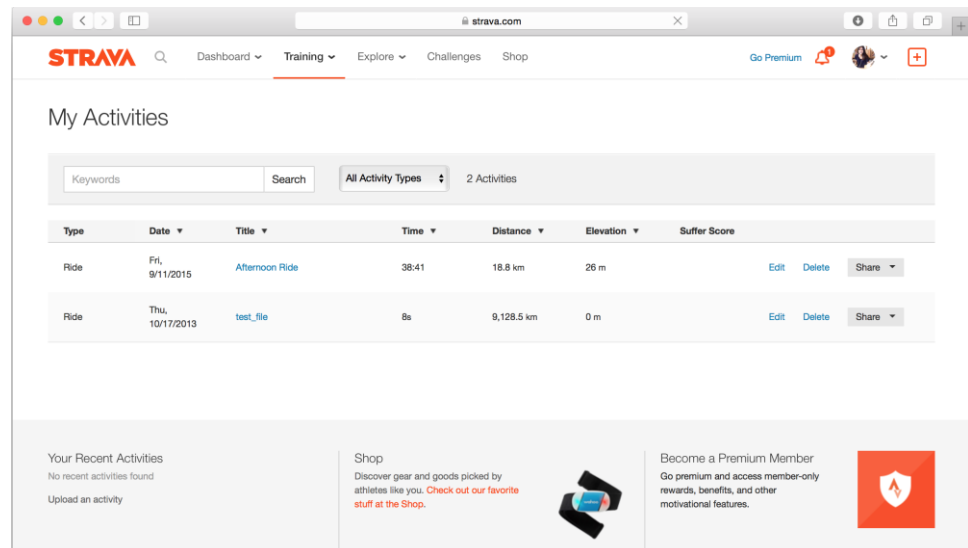


รูปที่ 4.7 แสดงผลจากการอัปโหลดไฟล์ของผู้ใช้ไปยัง Strava

เมื่อผู้ใช้ทำการอัปโหลดไฟล์แล้ว จะพบว่าไฟล์ที่ถูกอัปโหลด ได้ถูกเก็บที่ Strava ดังรูปที่ 4.8 และรูปที่ 4.9



รูปที่ 4.8 แสดงไฟล์ข้อมูลของผู้ใช้ก่อนการอัปโหลด



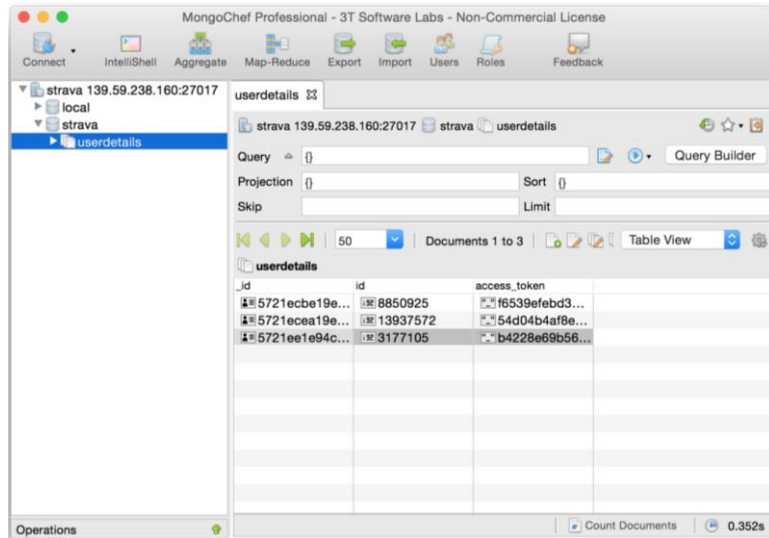
รูปที่ 4.9 แสดงไฟล์ข้อมูลของผู้ใช้หลังการอัปโหลด

#### 4.3.4 ผลการบันทึกข้อมูลลงฐานข้อมูล

การดึงข้อมูลในส่วนของ Profile ของผู้ใช้ ข้อมูลจะถูกบันทึกลงใน userdetails โดยใน userdetails ข้อมูลจะถูกจัดเก็บใน Attributes ต่างๆ ตัวอย่างตามตารางที่ 4.1

ตารางที่ 4.1 ตัวอย่างผลการบันทึกข้อมูล Profile ของผู้ใช้ใน userdetails

Attributes	Details
id	8850925
access_token	f6539efebd342263c14d3b03227dfa706b2b303e

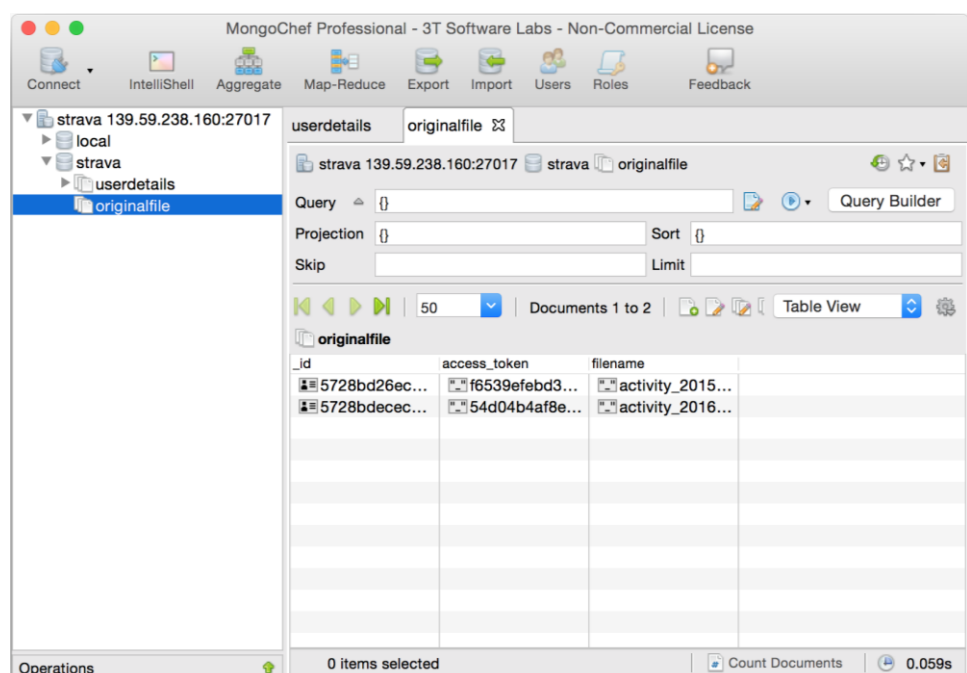


รูปที่ 4.10 แสดงผลการบันทึกข้อมูลในส่วนของ userdetails

การอัปเดตข้อมูลการออกกำลังกายของผู้ใช้ ไฟล์ข้อมูลการออกกำลังกายจะถูกบันทึกลงใน originalfile โดยใน originalfile ข้อมูลจะถูกจัดเก็บใน Attributes ต่างๆ ตัวอย่างตามตารางที่ 4.2

ตารางที่ 4.2 ตัวอย่างการอัปเดตข้อมูลการออกกำลังกายของผู้ใช้จาก Strava

Attributes	Details
access_token	f6539efebd342263c14d3b03227dfa706b2b303e
filename	activity_2015-07-09_00-58-28.tcx



รูปที่ 4.11 แสดงผลการบันทึกข้อมูลในส่วนของ originalfile

## บทที่ 5

### สรุปผลการดำเนินงานและข้อเสนอแนะ

จากการทำโครงงาน ระบบสามารถดึงข้อมูลของผู้ใช้ในส่วนของ Profile และ Activity จาก [www.strava.com](http://www.strava.com) ได้อย่างครบถ้วน สามารถทำการอัปโหลดไฟล์ข้อมูลในการออกกำลังกายของผู้ใช้งานไปเก็บไว้บน [www.strava.com](http://www.strava.com) และบันทึกไฟล์ลงในฐานข้อมูล MongoDB ได้ เป็นไปตามเป้าหมายและขอบเขตของโครงงานที่ได้กำหนดไว้

#### 5.1 สรุปผลการทดลอง

จากการทดลองในการดึงข้อมูลในส่วนของ Profile และ Activity ของผู้ใช้งานผ่านระบบนั้น เมื่อผู้ใช้งานหรือแอปพลิเคชันต่างๆต้องการเรียนข้อมูล โดยการส่งค่า id ผ่านมาทาง url พบว่าระบบสามารถดึงข้อมูลได้อย่างครบถ้วน รวมไปถึงการอัปโหลดไฟล์ข้อมูลการออกกำลังกาย โดย [www.strava.com](http://www.strava.com) จะรองรับไฟล์ทั้งหมด 3 ประเภทคือ .tcx, .gpx และ .fit ไฟล์ที่เกี่ยวกับการออกกำลังกายประเภทอื่นจะไม่สามารถทำการอัปโหลดไฟล์ข้อมูลไปเก็บไว้บน [www.strava.com](http://www.strava.com) ได้ ในการทดลองนั้น ได้ทดลองอัปโหลดไฟล์ทั้ง 3 ประเภท ผลคือสามารถอัปโหลดไฟล์ข้อมูลผ่านระบบเพื่อไปเก็บไว้บน [www.strava.com](http://www.strava.com) ได้ และสามารถบันทึกไฟล์ข้อมูลการออกกำลังกายต้นฉบับเก็บไว้ในฐานข้อมูลของระบบได้

#### 5.2 ปัญหาและอุปสรรคในการทดลอง

ตารางที่ 5.2 ปัญหาและแนวทางการแก้ไขปัญหา

ปัญหา	แนวทางการแก้ไขปัญหา
1. ไม่มีประสบการณ์และพื้นฐานในการทำ API	พยายามสอบถามผู้ที่มีความรู้ทางด้านการทำ API และสืบค้นหาข้อมูลที่เกี่ยวข้องจากอินเทอร์เน็ต
2. ภาษาที่ใช้ในการพัฒนาระบบ เป็นภาษาที่ทางผู้จัดทำไม่เคยใช้งานมาก่อน	พยายามสอบถามผู้ที่มีความรู้ในการเขียน node.js และหาตัวอย่างรวมถึงวิธีการเขียนจากอินเทอร์เน็ต
3. ไม่สามารถทดสอบระบบบน localhost ได้ เนื่องจากทาง <a href="http://www.strava.com">www.strava.com</a> ไม่สามารถทำการ callback กลับมายัง localhost ได้	ทำการอัปโหลดไฟล์ที่เกี่ยวข้องในการทำงานทุกอย่างไปบน Server เพื่อให้ <a href="http://www.strava.com">www.strava.com</a> สามารถทำการ callback กลับมาได้

ตารางที่ 5.2 ปัญหาและแนวทางการแก้ไขปัญหา(ต่อ)

ปัญหา	แนวทางการแก้ไขปัญหา
4. ในประเทศไทยยังไม่ค่อยมี Server ที่รองรับการทำงานในภาษา node.js จึงทำให้ไม่สามารถทำประประมวลผลการทำงานได้ ถึงจะมีบาง Server ที่รองรับ node.js แต่ก็ยังไม่สามารถรองรับการทำงานของระบบได้ทั้งหมด	ทำการหาข้อมูล Server ที่รองรับการทำงานในภาษา node.js และครอบคลุมการทำงานของโครงการ และทำการเช่า Server ต่างประเทศที่พบว่ามีความสมบัติครบและเพียงพอตามความต้องการในการทำโครงการ

### 5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

5.3.1 อุปกรณ์ในการออกกำลังกายที่โครงการนี้รองรับ มีเพียงแค่ 2 ชนิดเท่านั้น คือ จักรยานอยู่กับที่และลู่วิ่ง ระบบจะมีความสมบูรณ์มากขึ้นหากสามารถรองรับอุปกรณ์ในการออกกำลังกายได้อย่างครอบคลุม

5.3.2 เว็บไซต์ที่ใช้ในการเก็บข้อมูล ([www.strava.com](http://www.strava.com)) ยังมีข้อจำกัดในหลายด้าน เช่น ไม่สามารถเก็บข้อมูลที่เป็นต้นฉบับ เก็บเฉพาะค่าเฉลี่ยของข้อมูลที่ได้จากการออกกำลังกาย หากสามารถพัฒนาเว็บไซต์ประเภทนี้ได้เอง จะทำให้ระบบมีความสมบูรณ์ยิ่งขึ้น



## บรรณานุกรม

1. 2558. *What is Node.js?*[ออนไลน์], เข้าถึงได้จาก: [http://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](http://www.tutorialspoint.com/nodejs/nodejs_introduction.htm). 12 กันยายน 2558
2. Chai Phonbopit. 2558. *Node.js คืออะไร ? + เริ่มต้นใช้งาน Node.js* [ออนไลน์], เข้าถึงได้จาก: <http://devahoy.com/posts/getting-started-with-nodejs/>. 10 ตุลาคม 2558
3. *The MongoDB 3.2 Manual*[ออนไลน์], เข้าถึงได้จาก: <https://docs.mongodb.org/manual/>. 18 ธันวาคม 2558
4. *Node.js MongoDB Driver*[ออนไลน์], เข้าถึงได้จาก: <https://docs.mongodb.org/ecosystem/drivers/node-js/>. 20 ธันวาคม 2558
5. 2559. *Node.js MongoDB Driver API* [ออนไลน์], เข้าถึงได้จาก: <http://mongodb.github.io/node-mongodb-native/2.1/api/>. 4 มกราคม 2559
6. William Ritson. 2559. *What is bitbucket?*[ออนไลน์], เข้าถึงได้จาก: <https://www.quora.com/What-is-bitbucket>. 5 มกราคม 2559
7. Devrim Yasar. 2559. *Using Bitbucket On Koding* [ออนไลน์], เข้าถึงได้จาก: <http://www.koding.com/docs/using-bitbucket-on-koding>. 5 มกราคม 2559
8. Suthat R. 2559. *API คืออะไร?* [ออนไลน์], เข้าถึงได้จาก: <https://lengm0.wordpress.com/2008/11/04/webmasterorth-canlendar/>. 6 มกราคม 2559
9. Dusan Radojevic. 2558. *How to Use JSON files in Node.js* [ออนไลน์], เข้าถึงได้จาก: <https://www.codementor.io/nodejs/tutorial/how-to-use-json-files-in-node-js>. 20 พฤศจิกายน 2558
10. Peter Mortensen. 2559. *How to get GET (query string) variables in Express.js on Node.js* [ออนไลน์], เข้าถึงได้จาก: <http://stackoverflow.com/questions/6912584/how-to-get-get-query-string-variables-in-express-js-on-node-js>. 12 มีนาคม 2559
11. 2558. *การเขียนอ้างอิงเว็บไซต์ที่ถูกต้อง*[ออนไลน์], เข้าถึงได้จาก: <http://cstproject.exteen.com/20100926/entry>. 22 ตุลาคม 2558