**ระบบสกัดข้อมูลจากเอกสารแบบอัตโนมัติ**

**Automated Documents Extraction System**

**โดย**

**จิรวัฒน์ บุญกำเนิด**

**Jirawat Boonkumnerd**

อาจารย์ที่ปรึกษา

ดร. **กิติ์สุชาต พสุภา**

**Kitsuchart Pasupa**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต**

**สาขาวิชาเทคโนโลยีสารสนเทศ**

**คณะเทคโนโล**ยี**สารสนเทศ**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ภาคเรียนที่ 2 ปีการศึกษา 2560**

**Automated Documents Extraction System**

**Jirawat Boonkumnerd**

**Advisors**

**Kitsuchart Pasupa**

**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT**

**OF THE REQUIREMENT FOR THE DEGREE OF**

**BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT’S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2/2017**

**COPYRIGHT 2018**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT’S INSTITUTE OF TECHNOLOGY LADKRABANG**

**ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2560**

**คณะเทคโนโลยีสารสนเทศสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**เรื่อง ระบบสกัดข้อมูลจากเอกสารแบบอัตโนมัติ**

**AUTOMATED DOCUMENTS EXTRACTION SYSTEM**

**ผู้จัดทำ**

1. **นายจิรวัฒน์ บุญกำเหนิด รหัสนักศึกษา รหัสนักศึกษา 57070019**

**...................................................................... อาจารย์ที่ปรึกษา**

**(ผู้ช่วยศาสตราจารย์ ดร. กิติ์สุชาต พสุภา)**

**ใบรับรองโครงงาน (PROJECT)**

**เรื่อง**

**ระบบสกัดข้อมูลจากเอกสารแบบอัตโนมัติ**

**Automated Documents Extraction System**

**นาย จิรวัฒน์ บุญกำเหนิด** รหัสประจำตัว 5**7070019**

**ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด**

**รายงานฉบับนี้ได้รับการตรวจสอบเเละอนุมัติให้เป็นส่วนหนึ่งของการศึกษาวิชาโครงงาน**

**หลักสูตรวิทยาศาสตรบัณฑิต (เทคโนโลยีสารสนเทศ)**

**ภาคเรียนที่ 2 ปีการศึกษา 2560**

**...................................................................**

**(นายจิรวัฒน์ บุญกำเหนิด)**

**หัวข้อโครงงาน**  ระบบจัดการเอกสารเเบบอัตโนมัติ

**นักศึกษา** นายจิรวัฒน์ บุญกำเหนิด

**รหัสนักศึกษา** 57070019

**ปริญญา** วิทยาศาสตรบัณฑิต

**สาขาวิชา** เทคโนโลยีสารสนเทศ

**ปีการศึกษา** 2560

**อาจารย์ที่ปรึกษา ผศ.** ดร. กิติ์สุชาต พสุภา

บทคัดย่อ

เป็นระบบ Cloud Application ที่ใช้ในการจัดเก็บเอกสารเข้าสู่ระบบ มีฟังก์ชัน ในการใช้งานหลากหลายเพื่อเพิ่มประสิทธิภาพในการจัดเก็บ เเละเพิ่มความสะดวกสบาย ในการจัดเก็บหรือค้นหาเอกสารในระบบ ระบบนี้สามารถสกัดข้อมูลจากเอกสารด้วยเทคโนโลยี Optical character recognition ได้ สามารถจำแนกกลุ่มของเอกสารแบบอัตโนมัติ ด้วยการทำ Classification ด้วยการทดสอบหาวิธีที่เหมาะสมที่สุดในการทำ Classification สำหรับระบบ และระบบนี้ยังสามารถจำกัดสิทธิ์การเข้าถึงของแต่ละผู้ใช้ ระบบนี้เหมาะสำหรับเป็นระบบที่ใช้ภายในองค์กรที่ต้องการจัดเก็บเอกสารรุปเเบบดิจิทัลต่างๆ เช่น รูปภาพ หรือไฟล์ PDF โดยต้องการแยกหมวดหมู่ของเอกสารชัดเจน ต้องการความสะดวกสบายและความง่ายดายในการจัดเก็บ และเพื่อลดต้นทุนการจัดเก็บแทนการจัดเก็บในรูปเเบบสิ่งพิมพ์

**Project Title** Automated Documents Extraction System

**Student** Jirawat Boonkumnerd Student ID 57070019

**Degree** Bachelor of Science

**Program** Information Technology

**Academic Year** 2017

**Advisor** Assistant Professor Kitsuchart Pasupa

ABSTRACT

Document storage system that have many function for increase performance of storage and make more comfortable for storage or search documents in system. In addition it can extract data from documents. This system is for internal systems suitable for organization that need to storage document in digital format by category and reduce cost for storage document.

# กิตติกรรมประกาศ

โครงงานฉบับนี้จะไม่สามารถบรรลุผลสำเร็จได้ หากไม่ได้รับความกรุณาจากอาจารย์ที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ ดร.กิติ์สุชาต พสุภา ซึ่งเป็นผู้ที่ให้คำปรึกษา คำแนะนำ แนวทางและคอยช่วยเหลือชี้แนะและตอบข้อสงสัย ทั้งนี้อาจารย์ยังคอยให้กำลังใจและดูแลผู้จัดทำโครงงานตลอดมา และกราบขอบพระคุณอาจารย์ทุกท่านที่ช่วยประสาทวิชาความรู้ อีกทั้งช่วยเป็นคณะกรรมการผู้ช่วยตรวจสอบความสำเร็จของโครงงาน และขอบคุณเพื่อนๆ พี่ๆ น้องทุกคน รวมทั้งคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ณ ที่นี้ด้วย

จิรวัฒน์ บุญกำเหนิด

**สารบัญ**

[กิตติกรรมประกาศ iii](#_Toc513821177)

[บทนำ 1](#_Toc513821178)

[1.1 สาระสําคัญของโครงการ 1](#_Toc513821179)

[1.2 หลักการและเหตุผล 1](#_Toc513821180)

[1.3 วัตถุประสงค์ 2](#_Toc513821181)

[1.4 เป้าหมายและขอบเขตของโครงการ 2](#_Toc513821182)

[การทบทวนวรรณกรรมที่เกี่ยวข้อง 3](#_Toc513821183)

[2.1 ทฤษฏีที่เกี่ยวข้อง 3](#_Toc513821184)

[Optical Character Recognition (OCR) 3](#_Toc513821185)

[Support vector machine (SVM) 3](#_Toc513821186)

[Gradient Descent 4](#_Toc513821187)

[Stochastic Gradient Descent (SGD) 5](#_Toc513821188)

[NoSQL 5](#_Toc513821189)

[2.2 เทคโนโลยีที่เกี่ยวข้อง 6](#_Toc513821190)

[Flask Web framework 6](#_Toc513821191)

[MongoDb 6](#_Toc513821192)

[Vue.js 6](#_Toc513821193)

[OpenCV 6](#_Toc513821194)

[Tesseract OCR 6](#_Toc513821195)

[ImageMagick 6](#_Toc513821196)

[SciPy Toolkits 7](#_Toc513821197)

[การวิเคราะห์และออกแบบระบบ 8](#_Toc513821198)

[3.1 ขั้นตอนการปฏิบัติ 8](#_Toc513821199)

[3.2 ขอบเขตและข้อจำกัด 9](#_Toc513821200)

[3.3 โครงสร้างของซอฟต์แวร์ 9](#_Toc513821201)

[3.4 Database Schema 10](#_Toc513821202)

[3.5 Flow การทำงานของระบบ 11](#_Toc513821203)

[3.6 เเบบจำลองหน้าจอ 12](#_Toc513821204)

[3.9 Use Case Diagram 15](#_Toc513821205)

[Use Case Diagram for System 15](#_Toc513821206)

[Use Case Diagram for Parser Role 16](#_Toc513821207)

[ผลการดำเนินงาน 17](#_Toc513821208)

[4.1 Web Application User Interfaces 17](#_Toc513821209)

[4.7 Application programming interface (API) 23](#_Toc513821210)

[สรุปผลการดำเนินงาน 27](#_Toc513821211)

[5.1 สรุปผลภาคเรียนที่ 2 27](#_Toc513821212)

[5.2 ข้อเสนอเเนะเเละเเนวทางการพัฒนาระบบในอนาคต 27](#_Toc513821213)

[5.3 ปัญหาเเละอุปสรรคที่พบ 27](#_Toc513821214)

[ภาคผนวก ก 28](#_Toc513821215)

[คู่มือการติดตั้งอย่างละเอียด 28](#_Toc513821216)

[Python 3.5 or above 29](#_Toc513821217)

[Install Python Library 29](#_Toc513821218)

[Install ImageMagick 29](#_Toc513821219)

[Install Node.js 30](#_Toc513821220)

[Install Node package 30](#_Toc513821221)

[Install MongoDB 30](#_Toc513821222)

[ติดตั้ง Tesseract 4.0 30](#_Toc513821223)

[บรรณานุกรม 31](#_Toc513821224)

[ประวัติผู้เขียน 32](#_Toc513821225)

**สารบัญรูปภาพ**

[บทนำ 1](#_Toc513821273)

[การทบทวนวรรณกรรมที่เกี่ยวข้อง 3](#_Toc513821274)

[รุปที่ 2.1 แสดงถึงกระบวนการทำ OCR 3](#_Toc513821275)

[รูปที่ 2.2 แสดงถึงการทำ Support vector machine (ที่มา https://docs.opencv.org/2.4/doc/tutorials/ml/introduction\_to\_svm) 4](#_Toc513821276)

[รูปที่ 2.3 แสดงการหา cost function ด้วย Gradient Descent 4](#_Toc513821277)

[(ที่มา https://tupleblog.github.io/gradient-descent-part1/) 4](#_Toc513821278)

[รูปที่ 2.4 แสดงถึงความแตกต่างของ SQL และ Non-SQL 5](#_Toc513821279)

[การวิเคราะห์และออกแบบระบบ 8](#_Toc513821280)

[รูปที่ 3.1 รูปภาพโครงสร้างการทำงานของระบบ 9](#_Toc513821281)

[รูปที่ 3.2 Database Schema 10](#_Toc513821282)

[รุปที่ 3.3 Flow ของระบบ 11](#_Toc513821283)

[รูปที่ 3.4 แบบจำลองหน้าจอการ import document 12](#_Toc513821284)

[จากรูปภาพที่ 1 ทำการ import documents ด้วยวิธีการใช้ File Explorer หรือลากไฟล์มาวาง สามารถ import เเบบหลายไฟล์ได้ 13](#_Toc513821285)

[รูปที่ 3.5 แบบจำลองหน้าจอ preview 14](#_Toc513821286)

[รูปที่ 3.6 Use Case Diagram for System 15](#_Toc513821287)

[รูปที่ 3.7 Use Case Diagram for Parser Role 16](#_Toc513821288)

[ผลการดำเนินงาน 17](#_Toc513821289)

[รูปที่ 4.1 หน้า Login 17](#_Toc513821290)

[รูปที่ 4.2 หน้าสมัครสมาชิก 17](#_Toc513821291)

[รูปที่ 4.4 เพิ่ม Parsers 18](#_Toc513821292)

[รูปที่ 4.5 upload document 19](#_Toc513821293)

[รูปที่ 4.6 document table 19](#_Toc513821294)

[รูปที่ 4.7 Parser 20](#_Toc513821295)

[รูปที่ 4.8 Add Parser Rule 21](#_Toc513821296)

[รูปที่ 4.9 Document Page 22](#_Toc513821297)

[รูปที่ 4.10 Register API 23](#_Toc513821298)

[รูปที่ 4.11 Login API 24](#_Toc513821299)

[รูปที่ 4.12 Add Parser API 24](#_Toc513821300)

[รูปที่ 4.13 Get Parsers by User API 25](#_Toc513821301)

[รูปที่ 4.14 Add document API 25](#_Toc513821302)

[สรุปผลการดำเนินงาน 27](#_Toc513821303)

[ภาคผนวก ก 28](#_Toc513821304)

[คู่มือการติดตั้งอย่างละเอียด 28](#_Toc513821305)

[รูปที่ ก.1 การติดตั้ง Python 29](#_Toc513821306)

[รูปที่ ก.2 การติดตั้ง Python Library 29](#_Toc513821307)

[รูปที่ ก.3 การติดตั้ง Node.js 30](#_Toc513821308)

[บรรณานุกรม 31](#_Toc513821309)

[ประวัติผู้เขียน 32](#_Toc513821310)

บทที่1

บทนำ

1.1 ความสำคัญและความเป็นมาของโครงงาน

หน่วยงานต่างๆในปัจจุบัน ยังคงใช้วิธีในการจัดเก็บเอกสารแบบสิ่งพิมพ์ ทำให้ มีปัญหาในการค้นหาอย่างยากลำบาก และเอกสารมีโอกาสที่จะสูญหายหรือชำรุด

ทำให้บางหน่วยงานในปัจจุบันเริ่มเปลี่ยนมาจัดเก็บเอกสารในรูปแบบดิจิทัลกันแล้ว เพื่อลดปัญหาข้อมูล สูญหายหรือชำรุด เพื่อเพิ่มความสะดวกในการค้นหา จึงทำให้บางบริษัทเริ่มมีการพัฒนา Cloud Application ที่จะช่วยในการจัดการกับเอกสารในรูปแบบดิจิทัลออกมาแล้ว เพื่อนำมาจำหน่ายบริการให้กับองค์กรต่างๆ

ด้วยเหตุผลประการนี้ดังนั้นเราจึงมีเเนวคิดที่จะพัฒนาระบบที่สามารถจัดเก็บเอกสารเป็นข้อมูลเเบบดิจิทัล ด้วยวิธีการแสกนเอกสารเป็นรูปภาพแล้วจึงจัดเก็บไว้บนฐานข้อมูล และยังมีฟีเจอร์การใช้งานต่างๆ เพื่อเพิ่มความสะดวกให้แก่ผู้ใช้งาน เช่น สามารถแนะนำผู้ใช้ว่าเอกสารนี้เป็นเอกสารชนิดไหนด้วยเทคนิคการจำแนก (Classification) รูปภาพของเอกสาร สามรถสกัดข้อมูลด้วยวิธีการ Optical character recognition (OCR) ออกมาเป็นรูปแบบ Text เพื่อนำไปใช้งานอย่างอื่นได้ สามารถ Export ข้อมูลเป็นรูปแบบ Json format หรือ CSV format ได้

1.2 วัตถุประสงค์

* เพื่อเป็นระบบสำหรับจัดเก็บเอกสารในรูปเเบบดิจิทัล
* เพื่อเพิ่มความสะดวกเเละรวดเร็วในการจัดเก็บข้อมูลเข้าสู่ฐานข้อมูล
* เพื่อลดความผิดพลาดในการป้อนข้อมูลเข้าสู่ฐานข้อมูล
* เพื่อทดสอบวิธีในการจำแนกเอกสาร
* เพื่อสกัดข้อมูลเอกสารและนำไปใช้งานรูปแบบอื่น
* เพื่อลดปัญหาเอกสารสูญหายหรือชำรุด
* เพื่อเพิ่มความสะดวกในการค้นหาเอกสาร
* เพื่อกำหนดสิทธิ์การเข้าถึงของกลุ่มของเอกสาร ของผู้ใช้งานแต่ละคน

1.3 เป้าหมายและขอบเขตของโครงการ

1. ออกแบบฐานข้อมูล

2. จัดทำ Web Server ในการจัดเก็บเอกสาร

3. จัดทำ Web Application

4. ทำ Feature ในการ Extract ข้อมูลจากเอกสาร

5. ทดสอบวิธีที่ใช้ในการจำแนกเอกสาร จากนั้นเลือกวิธีทื่เหมาะสมที่สุด

6. สามารถ Export ข้อมูลเป็นรูปแบบ Image, PDF, CSV, JSON

7. มีการยืนยันตัวตนก่อนเข้าใช้งานระบบ

8. สามารถกำหนดสิทธิ์การเข้าถึงของกลุ่มของเอกสาร

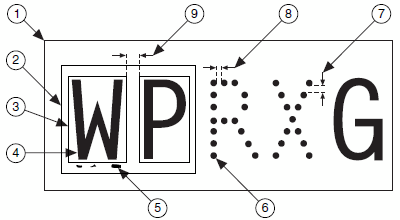
บทที่ 2

# การทบทวนวรรณกรรมที่เกี่ยวข้อง

2.1 ทฤษฏีที่เกี่ยวข้อง

Optical Character Recognition (OCR)

คือ กระบวนการจดจำตัวอักษรของคอมพิวเตอร์ ที่มากจากสิ่งพิมพ์หรือมาจาก การเขียนด้วยรายมือ สามารถสกัดตัวอักษรจากรูปภาพเเปลงเป็นข้อความในรูปเเบบที่ จะนำไปใช้งานอื่นๆในคอมพิวเตอร์ได้ง่ายขึ้น ในการทำ OCR สามารถใช้อัลกอริทึม หรือวิธีที่ต่างกันไป



รุปที่ 2.1 แสดงถึงกระบวนการทำ OCR

Support vector machine (SVM)

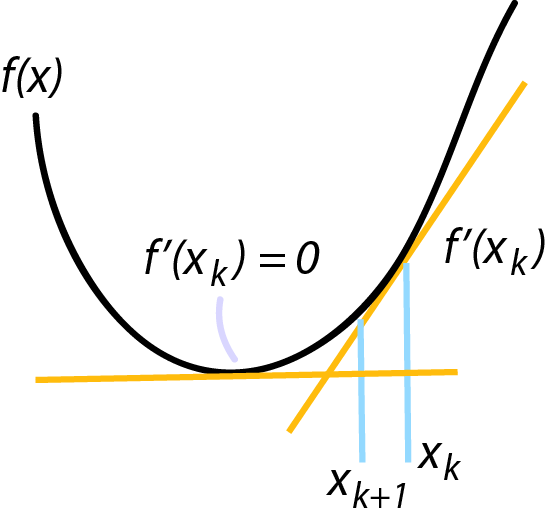
คือ อัลกอริทึมในการเรียนรู้ของคอมพิวเตอร์เเบบ supervised สามารถใช้เพื่อ classification หรือ regression โดยการกำหนดจุดของทุกๆข้อมูลเเล้วทำการหาเส้นเเบ่งของชนิดของข้อมูล เเละเมื่อทำการปรับเส้นเเบ่งอย่างดีเเล้วสามารถบอกได้ว่าข้อมูลใหม่ที่เข้ามานั้นเป็นข้อมูลชนิดไหน การจำเเนกสามารถเเบ่งได้ทั้งเเบบ One-vs-All หรือ One-vs-One โดยขึ้นอยู่กับความเหมาะสม ของข้อมูล



รูปที่ 2.2 แสดงถึงการทำ Support vector machine

Gradient Descent

เป็นอัลกอริทึมในการหา cost function หรือค่าที่เหมาะสมให้กับฟังก์ชัน โดยการวนหาค่าเพื่อหาที่ที่ต่ำที่สุด อาจใช้เพื่อหาสมการเส้นตรง



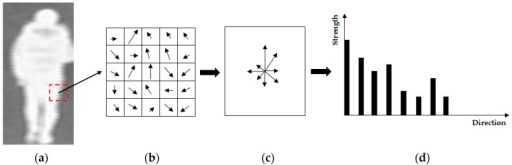
รูปที่ 2.3 แสดงการหา cost function ด้วย Gradient Descent

Stochastic Gradient Descent (SGD)

เป็นการหา cost function คล้ายกลับ Gradient Descent แต่ทุกรอบในการคำนวณ จะสุ่มข้อมูลเพียงบางส่วนเพื่ออัพเดตค่า ไม่ได้ใช้ข้อมูลทั้งหมดเหมือน Gradient Descent

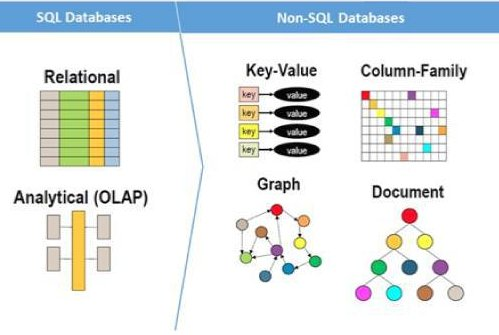
Histogram of oriented gradients

เป็นการหาทิศทางของรูปภาพโดยดูจากสีของรูปภาพ โดยแบ่งเป็น Block และแต่ละ Block แบ่งเป็นหลายๆ Cell



NoSQL

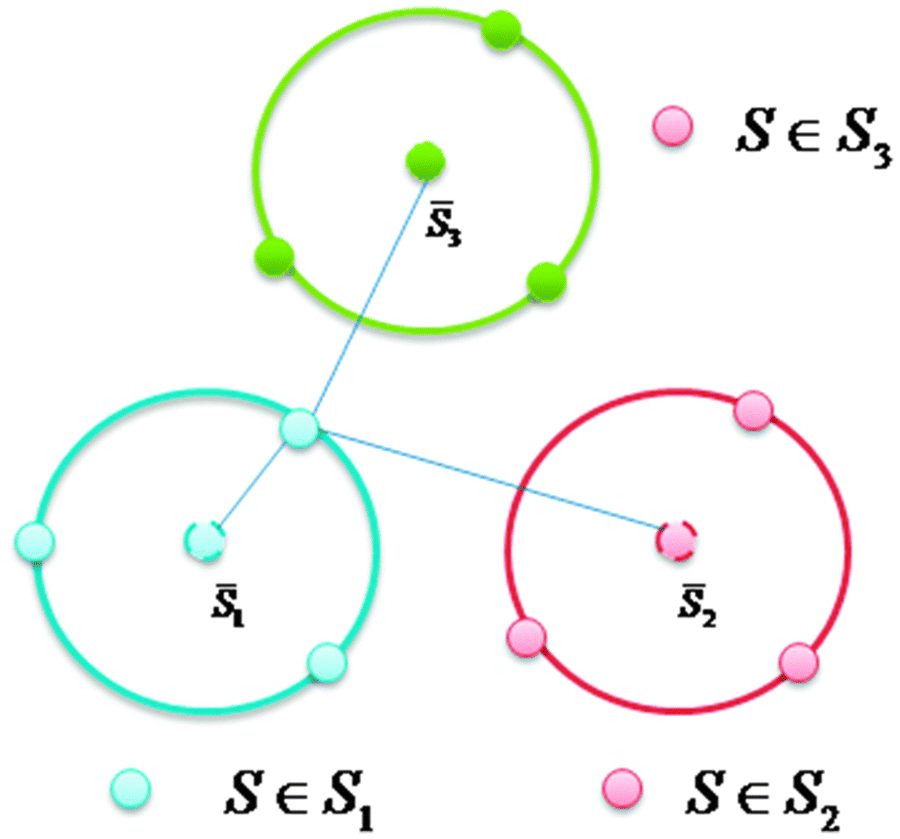
คือแนวทางสำหรับออกแบบและจัดเก็บฐานข้อมูล ที่เหมาะกับข้อมูลที่มีขนาดใหญ่ เเละไม่มีโครงสร้างที่ชัดเจน ทำให้รองรับข้อมูลที่มีรูปแบบหลากหลายและยืดหยุ่น และยังเหมาะสำหรับแอพลิเคชันที่มีการใช้งานแบบ Realtime



รูปที่ 2.4 แสดงถึงความแตกต่างของ SQL และ Non-SQL

Nearest centroid classifier

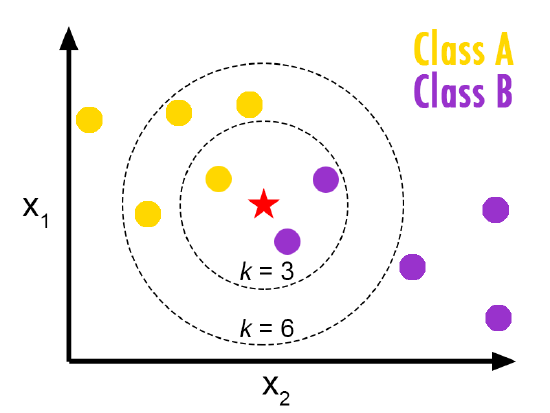
คือ อัลกอริทึมที่ใช้ในการ Classification โดยหาค่า Mean ของทุกๆจุดใน Class นั้น เรียกว่าจุด Centroid แล้วจึง Predict ค่าที่เข้ามาใหม่ด้วย Euclidean distance ดูว่าจุด Centroid ของ Class ไหนไกล้ที่สุด



รูปที่ 2.5 แสดงถึงการทำงานของ Nearest centroid classifier

k-nearest neighbors

คือ อัลกอริทึมที่ใช้ในการ Classification โดยการหาจุดที่ไกล้ที่สุดจำนวน k จุด เพื่อหาว่าในจุดจำนวน k จุดที่ไกล้ที่สุดนั้นเป็น Class ไหนมากที่สุด



รูปที่ 2.6 แสดงความแตกต่างระหว่างการหา KNN จำนวน 3 จุด กับ 6 จุด

2.2 เทคโนโลยีที่เกี่ยวข้อง

Flask Web framework

เว็บ Framework ที่เขียนด้วยภาษา Python มีต้นแบบมาจาก Werkzeug และ Jinja 2 จุดเดนตรงที่มีความใช้งานง่ายและหนักเบา

MongoDb

ระบบฐานข้อมูลแบบ NoSQL หรือ Non-Relational Database เป็นฐานข้อมูลที่เหมาะสำหรับจัดเก็บเอกสารที่มีความยืดหยุ่นของข้อมูลสูง และมีการเรียกข้อมูลได้เร็วกว่า SQL ทั่วไป

Vue.js

เป็น Web front-end framework ที่ทำงานบน Node.js สามารถ Render ไฟล์เว็บ สามารถทำ Routing URL ได้ และยังมีเครื่องมืออื่นๆให้ใช้อีกครบครัน

OpenCV

เป็น Library ที่ใช้สำหรับการทำ Image Processing และ Computer Vision มีเครื่องมือให้ใช้งานได้หลากหลาย Library ถูกเขียนด้วยภาษา C++

Tesseract OCR

เป็น Library ในการทำ OCR ที่ทาง Google ได้พัฒนาขึ้น มีความแม่นยำสูง และในรุ่นใหม่มีการเทรนโมเดลด้วย LSTM ทำให้มีความแม่นยำมากขึ้น

ImageMagick

เป็น Open-source สำหรับ การจัดการรูปภาพ แปลงไฟล์รูปภาพ แก้ไขรุปภาพ หรือแม้แต่บีบอัดรูปภาพ ก็สามารถทำได้

SciPy Toolkits

เป็นกลุ่มของ open source ซอฟแวร์ที่ใช้สำหรับในการทำการคำนวนทางวิทยาศาสตร์ต่างๆ มีซอฟแวร์ย่อยๆอีกมากมาย เช่น [NumPy](http://www.numpy.org/), [Pandas](http://pandas.pydata.org/), [Scikit-learn](http://scikit-learn.org/) , [Scikit-image](http://scikit-image.org/), [IPython](http://ipython.org/)

**บทที่ 3**

**การวิเคราะห์และออกแบบระบบ**

3.1 ขั้นตอนการปฏิบัติ

1. ทำการศึกษาวิธีการทำระบบ Web Application โดยเทคโนโลยีที่จะใช้มีดังนี้ Flask, Vue.js
2. ทำการศึกษาวิธีการจัดการฐานข้อมูล โดยชนิดของฐานข้อมูลที่เลือกใช้คือ MongoDb
3. ศึกษาวิธีการใช้งาน Python Library เช่น Numpy, Scipy, Keras, OpenCV เป็นต้น
4. ศึกษาวิธีเทคนิคการทำ Image Processing ด้วยวิธีต่างๆ เช่น Maximally Stable Extremal Regions (MSER), Text Detection Methods, HoughLines
5. ทดสอบ Histogram of Oriented Gradients Image Feature Extraction
6. ศึกษาวิธีการทำ Document Classification ด้วยวิธี
   * Stochastic gradient descent
   * Nearest centroid
   * k-nearest neighbors

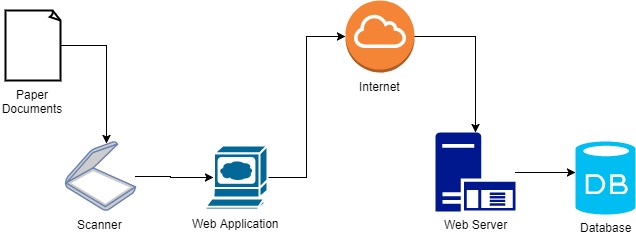
เเละวัดประสิทธืภาพเพื่อเปรียบเทียบผลของการจัดหมวดหมู่ของรูปเอกสาร แล้วจึงเลือกใช้วิธีที่ได้ผลดีที่สุด

1. ทำการออกแบบClass Diagram ของระบบ
2. ทำการออกแบบการทำงานของระบบและ Flow การทำงานต่างๆ
3. ทำการออกแบบตัวอย่างจำลอง (Mockup) ของWeb Application
4. ลงมือพัฒนาระบบ Web Application
5. ลงมือพัฒนาฟีเจอร์ในการสกัดข้อมูลจากเอกสารด้วยวิธีการ OCR
6. ลงมือพัฒนาฟีเจอร์ในการ Recommend Parser ด้วยวิธีการ Classification
7. ทำการทดสอบระบบ
8. ทำการ Deployment ระบบเพื่อสำหรับการใช้งานจริง
9. จัดทำGuide และวีดีโอสอนวิธีการใช้งาน
10. .จัดทำเอกสาร

3.2 ขอบเขตและข้อจำกัด

* ไม่สามารถใช้บน Device ที่ไม่มีการเชื่อมต่อกับอินเทอร์เน็ตได้
* สามารถใช้งานได้เฉพาะ Device ที่สามารถใช้งาน Web Browser ได้เท่านั้น

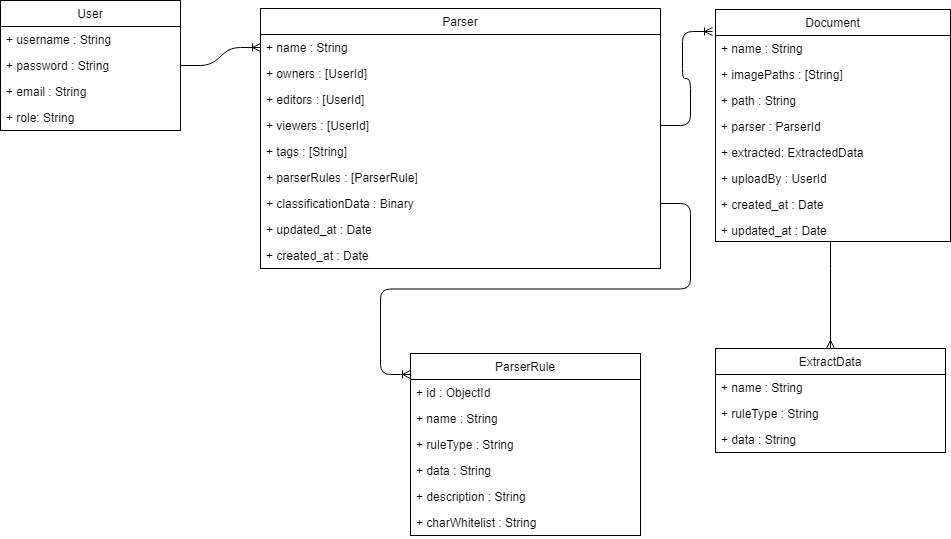
3.3 โครงสร้างของซอฟต์แวร์



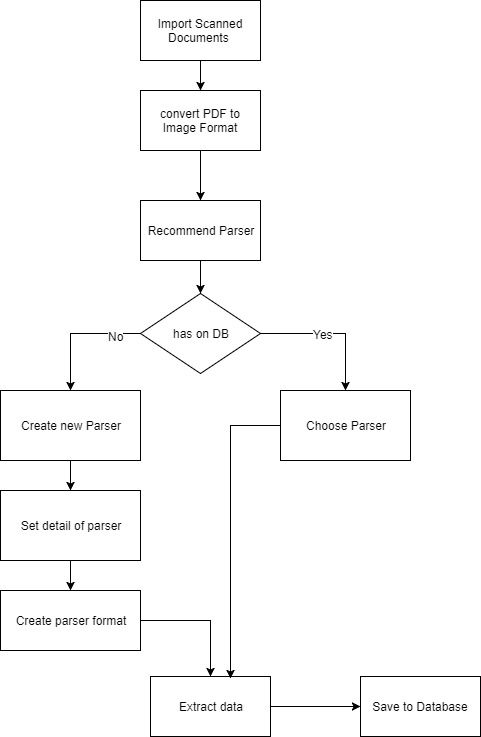
รูปที่ 3.1 รูปภาพโครงสร้างการทำงานของระบบ

ขั้นเเรกนำเอกสารรูปเเบบสิ่งพิมพ์ไปเเสกนเข้าคอมพิวเตอร์จากนั้นนำเปิดเว็บเเอพของเรา ผ่านBrowser แล้วจากนั้น import รูปภาพ หรือไฟล์ PDF ของเอกสารที่แสกนแล้วระบบ จะแนะนำหากมีเอกสารรูปแบบนี้มีอยู่ในระบบ หากไม่มีสามารถสร้าง Category ขึ้นเอง โดยกำหนด ชื่อ, Category และ รูปแบบของเอกสาร เช่น PO number, Date, Tax, Total Price เป็นต้น จากนั้นเอกสารจะถูกจัดเก็บลงบนฐานข้อมูลในหมวดหมู่ที่กำหนด

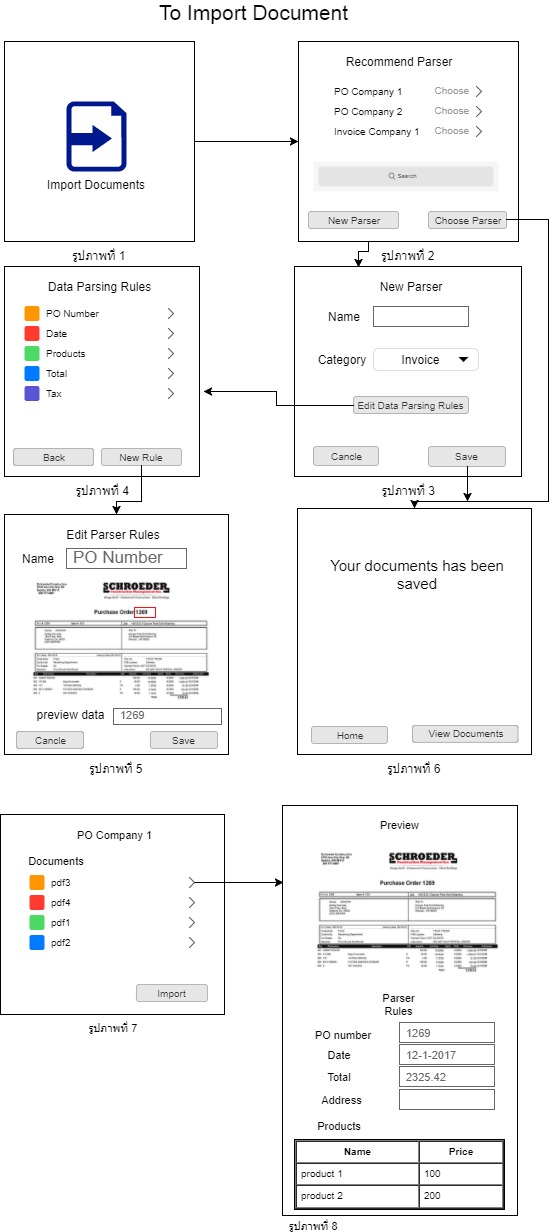
3.4 Database schema



รูปที่ 3.2 Database schema

3.5 Flow การทำงานของระบบ  
 

รุปที่ 3.3 Flow การทำงานของระบบ

3.6 เเบบจำลองหน้าจอ

รูปที่ 3.4 แบบจำลองหน้าจอการ import document

จากรูปภาพที่ 1 ทำการ import documents ด้วยวิธีการใช้ File Explorer หรือลากไฟล์มาวาง สามารถ import เเบบหลายไฟล์ได้

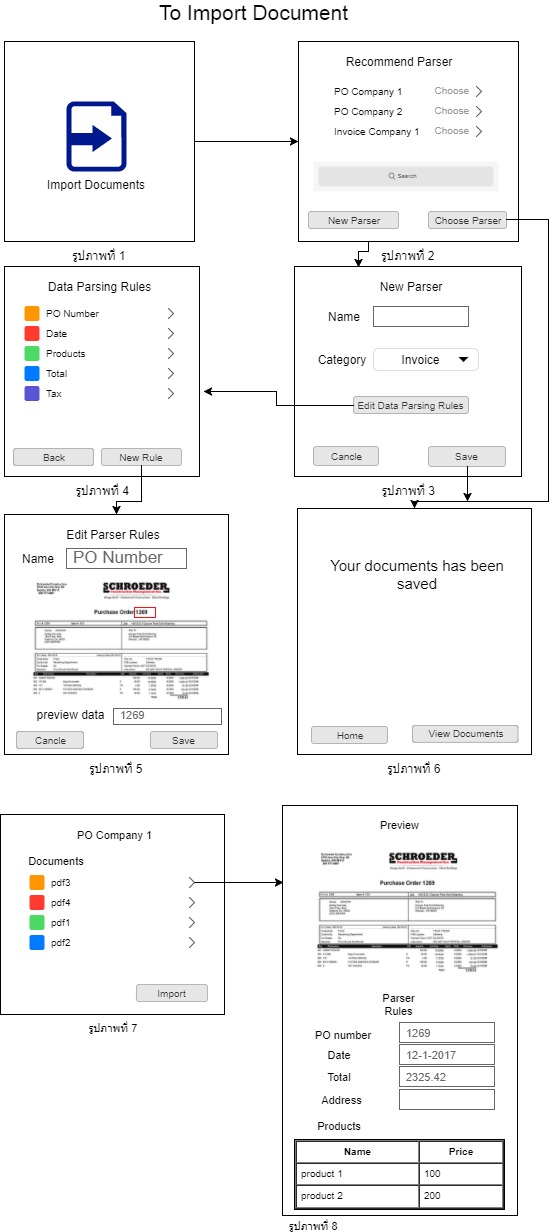
**จากรูปภาพที่ 2** ระบบจำทำการ Recommend Parser หาก Documents ที่ import มีลักษณะไกล้เคียงกับ Parser ที่มีอยู่ในระบบ หรือถ้าหากว่า Recommend ไม่ตรง ผู้ใช้ก็ยังสามารถค้นหา Parser ที่ตรงได้ เเต่ถ้าหากรูปเเบบเอกสารชนิดนี้ไม่มีอยู่ในระบบ ให้เลือก New Parser

**จากรูปภาพที่ 3** จะมีหน้าต่างให้กำหนดรายละเอียดของ Parser นั้นๆ อาธิเช่น ชื่อ, หมวดหมู่ เเละมีปุ่มให้ กด Edit Rules เพื่อกำหนดกฏต่างๆของ Parser นั้นๆ

**จากรูปภาพที่ 4** ในเเต่ละ Parser นั้นจะมีให้กำหนด Parsing Rules ต่างๆ อาจเป็นการสกัดข้อมูลด้วย OCR เช่น วันที่ สินค้า ราคารวม รหัสใบสั่งซื้อเป็นต้น ซึ่งในส่วนนี้ User จะเป็นคนกำหนดฟอร์มของ Parser เเต่ละชนิดเอง หรืออาจใช้รูปเเบบที่ทางเราจัดเตรียมไว้ให้เเล้วก็ได้

**จากรูปภาพที่ 5** ในเเต่ละ Parsing Rules จะมีหลายชนิดตัวอย่างเช่น การสกัด PO Number จะใช้วิธีการ OCR ภาพในกรอบ boundary ที่กำหนด หรือว่าหากข้อมูลเป็นตารางเช่นรายการสินค้า กับราคาสินค้า เราก็จะจัดเตรียมรูปเเบบในการสกัดข้อมูลออกมาเป็นตารางไว้ให้ เเละมี Preview สำหรับตรวจสอบว่าเราได้กำหนด Rule ได้ตรงหรือไม่

**จากรูปภาพที่ 6** ระบบจะเเสดงผลว่าผู้ใช้ได้ Import เอกสารเข้าไปในระบบเเล้ว



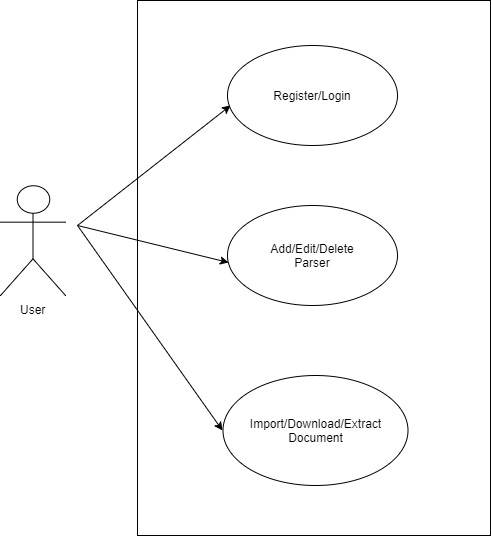
รูปที่ 3.5 แบบจำลองหน้าจอ preview

**จากรูปภาพที่ 7** ในเเต่ละ Parser จะมีเอกสารชนิดเดียวกันที่ถูกจัดเก็บในฐานข้อมูล ผู้ใช้จะสามารถ import เอกสารเพิ่มขึ้นอีกจากหน้านี้ ซึ่งไม่ต้องไปเลือกทีหลังจาก import ผู้ใช้สามารถเลือกดูเเต่ละเอกสารได้

นอกจากนี้ยังสามารถ Export All เพื่อนำอกเอกสารทั้งหมด สามารถ Export เป็น Pdf หรือจะ Export เป็น Json ซึ่งระบบจะทำการสกัดข้อมูลตาม Parser Rules ต่างๆด้วย OCR

**จากรูปภาพที่ 8** เมื่อคลิกเพื่อดูของเเต่ละเอกสาร จะมีปุ่มสำหรับ Export เฉพาะเอกสารนั้น เเละมี Preview ข้อมูลก่อนจะสกัดข้อมูลด้วย OCR

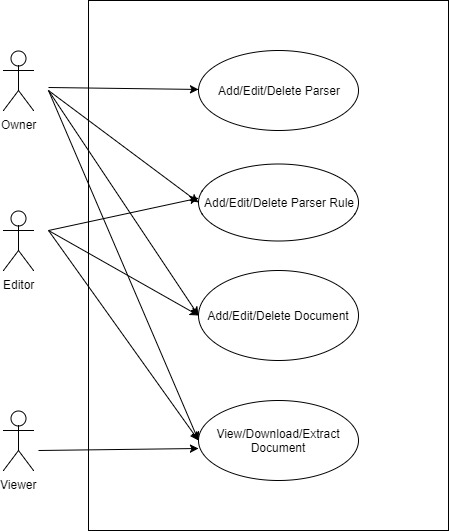
3.9 Use Case Diagram



รูปที่ 3.6 Use Case Diagram for System

Use Case Diagram for System

* บอกว่าผู้ใช้ปรกติสามารถทำอะไรกับระบบได้บ้าง
* ผู้ใช้สามารถสมัครสมาชิกและเข้าสู่ระบบได้
* ผู้ใช้สามารถเพิ่ม แก้ไข หรือลบ Category ในระบบได้
* ผู้ใช้สามารถเพิ่ม โหลด สกัดข้อมูล หรือลบเอกสารได้



รูปที่ 3.7 Use Case Diagram for Role

Use Case Diagram for Role

* บอกว่า role ของแต่ละ Category สามารถทำอะไรได้บ้าง
* Viewer สามารถดูเอการได้เพียงอย่างเดียว แต่ไม่สามารถแก้ไข หรือลบได้
* Editor สามารถ เพิ่ม แก้ไข หรือลบ เอกสารได้
* Owner สามารถ เพิ่ม แก้ไข หรือลบ Category ได้ สามารถให้สิทธิ์แก่ User อื่นได้

3.10 ขั้นตอนการทำ Document Classification

1. เก็บข้อมูลข้อมูลทดสอบทั้งหมด 2112 รูป และจำนวน class ทั้งหมด 24 แบบ จากหลายๆแห่ง ดังนี้ ghega genuine fpdf invoiced และ zoho

2.นำรูปทั้งหมดไปทำการ feature extraction ด้วยวิธีการ histogram of oriented gradients โดย resize รูปภาพให้เท่ากัน ที่ตั้งไว้คือ 700\*500 และกำหนดทิศทางของ HOG 8 ทิศทาง จำนวน Pixel ต่อ cell คือ 16 x 16 จะได้ฟีเจอร์ทั้งหมด 85,608 feature ต่อหนึ่งรูปภาพ

3.แบ่งชุดทดสอบเป็นชุด Train 70% และ ชุด Test 30% แบ่งตาม Scale ของทุกๆ Class

4.จากนั้นนำข้อมูล Train ไปทำ Cross Validation ด้วย k-Fold Cross-Validation แบ่งชุดทดสอบเป็น 5 ชุด สลับกัน Train และ Test

5.ทำการสร้าง Loop เพื่อปรับ Parameter ของแต่ละ Model ดังนี้

5.1. SGD Model ปรับ Learning rate 10^-5, 10^-4, 10^-3, 10^-2, 10^-1, 10^0

5.2 KNN Model ปรับค่าจำนวน k neighbor 1, 2, 3, 4, 5, 6

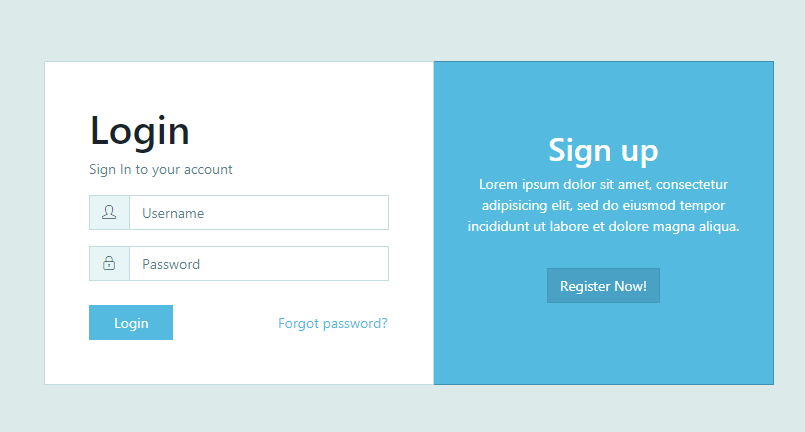
6.หลังจากที่ได้ Parameter ที่ดีที่สุดจากการทำ Cross Validation แล้ว จึงนำไปเทสกับข้อมูลเต็มที่แบ่งเป็น Train 70% Test 30% จากการ Random สลับตำแหน่งออกเป็น 10 ชุด

บทที่ 4

ผลการดำเนินงาน

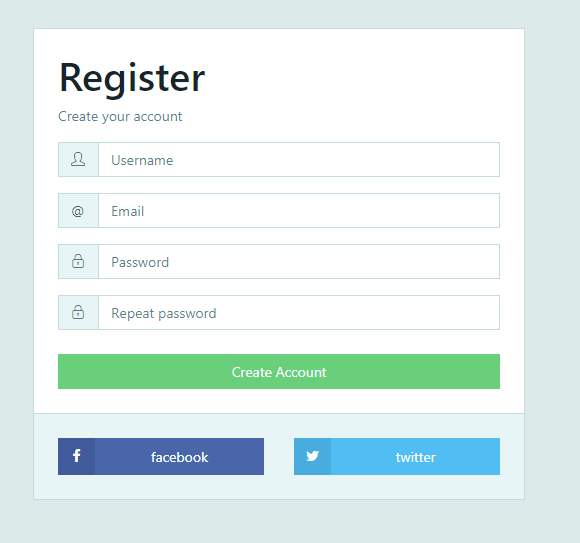
4.1 Web Application User Interfaces

* **Login**



รูปที่ 4.1 หน้า Login

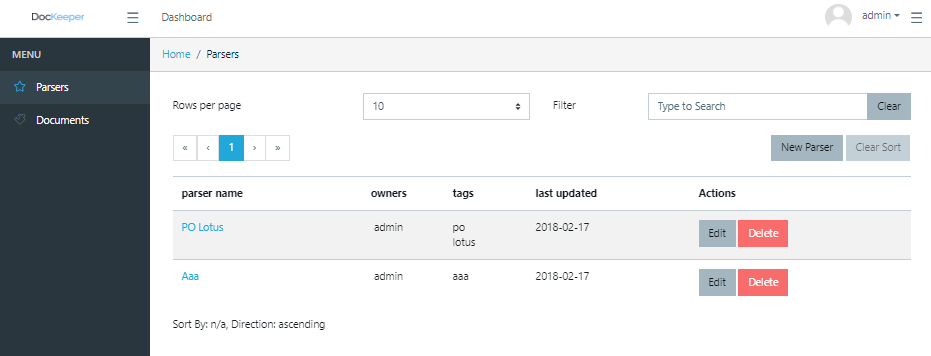
* **Register**

****

รูปที่ 4.2 หน้าสมัครสมาชิก

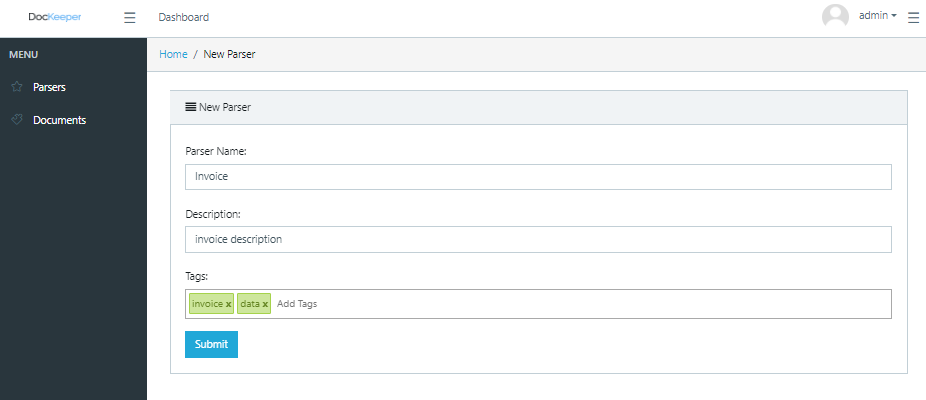
ผู้ใช้สามารถทำการเข้าสู่ระบบหรือสมัครสมาชิกได้จากหน้านี้ ระบบ Authentication เป็นแบบ JWT ที่จำนำรหัสไป hash ให้ตรงกับในฐานข้อมูล เพื่อรับ token มาใช้งาน token นั้นมีวันหมดอายุอยู่ 1 สัปดาห์

* **Parsers Table**

รูปที่ 4.3 ตาราง Parsers

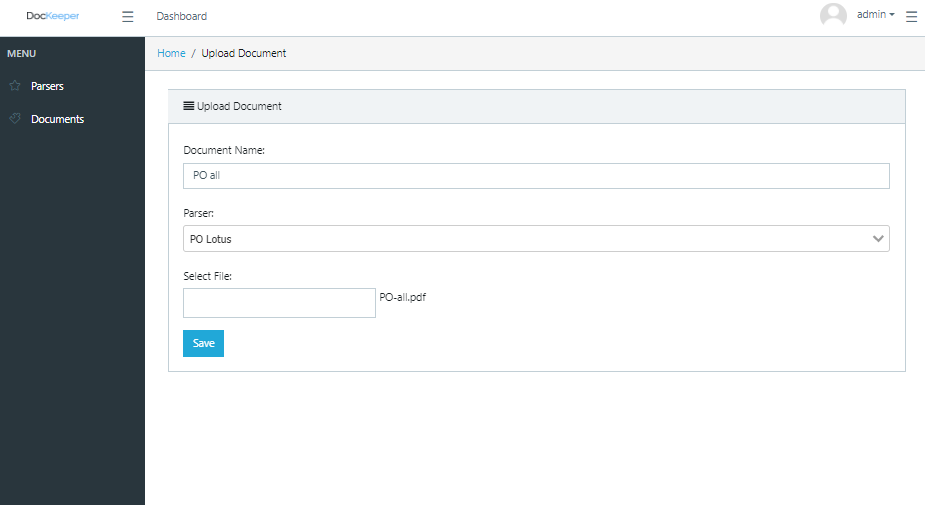
แสดงตารางของ Parser มีปุ่ม Add Edit และ Delete สามารถค้นหาจากช่อง Search ได้ และยังสามารถเรียงลำดับข้อมูลได้

* **New Parser**



รูปที่ 4.4 เพิ่ม Parsers

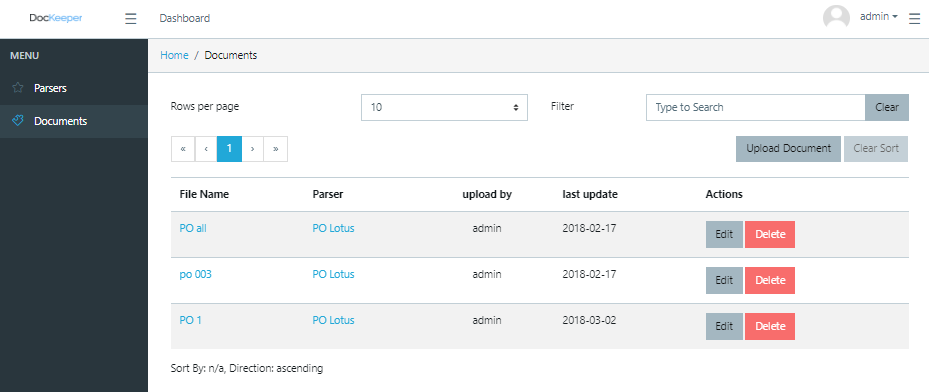
* **Upload Document**



รูปที่ 4.5 upload document

หน้า form การ Upload เอกสารเข้าสู่ระบบ

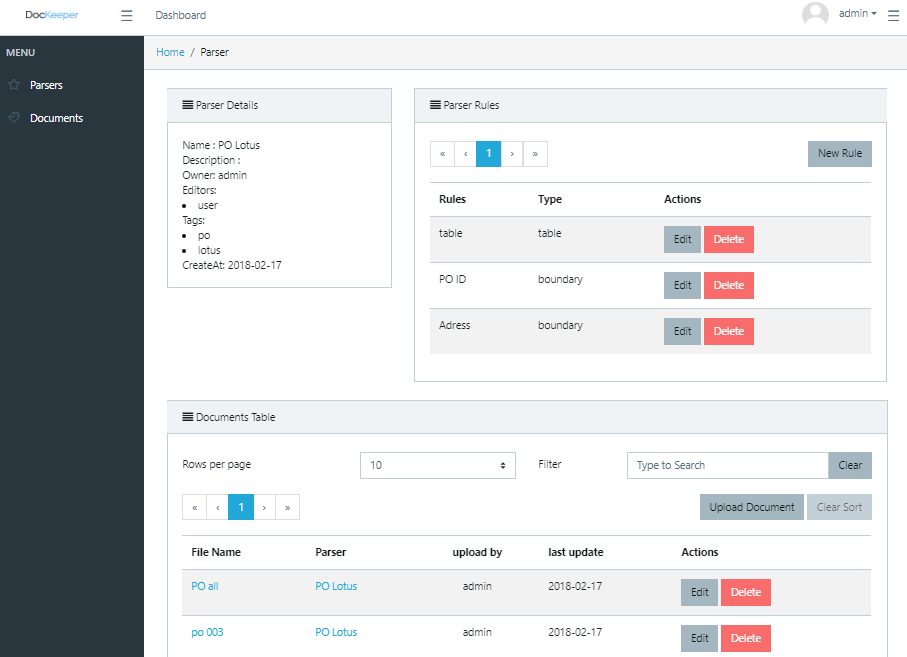
* **Documents Table**



รูปที่ 4.6 document table

หน้าแสดงเอกสาร ผู้ใช้สามารถ Edit Delete Add เอกสารได้จากหน้านี้ และสามารถค้นหาเอกสารได้

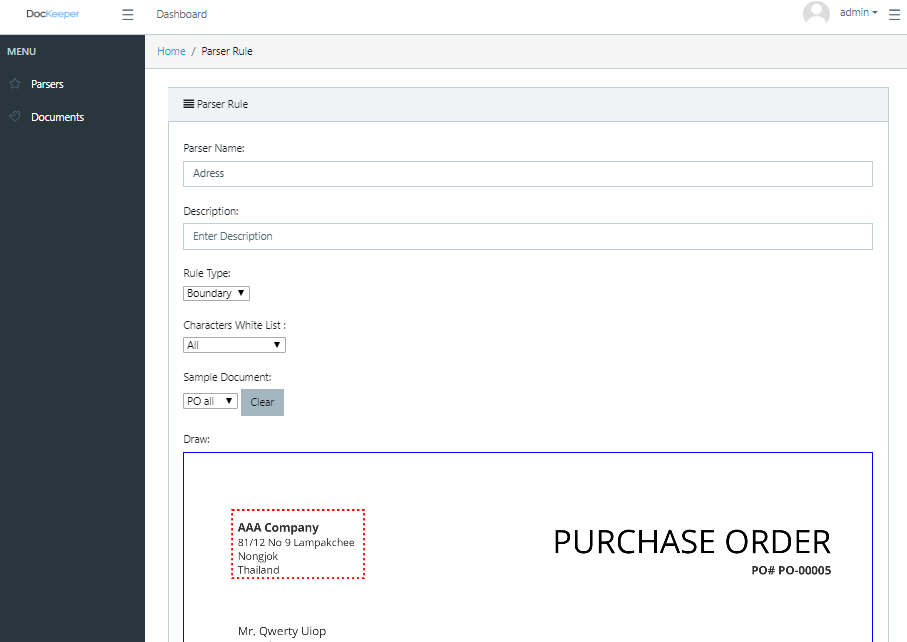
* **Parser**

****

รูปที่ 4.7 Parser

ในหน้า Parser จะมีตารางแสดง เอกสารที่ใช้รูปแบบ Parser นี้ และแสดงตารางของ Parser Rule

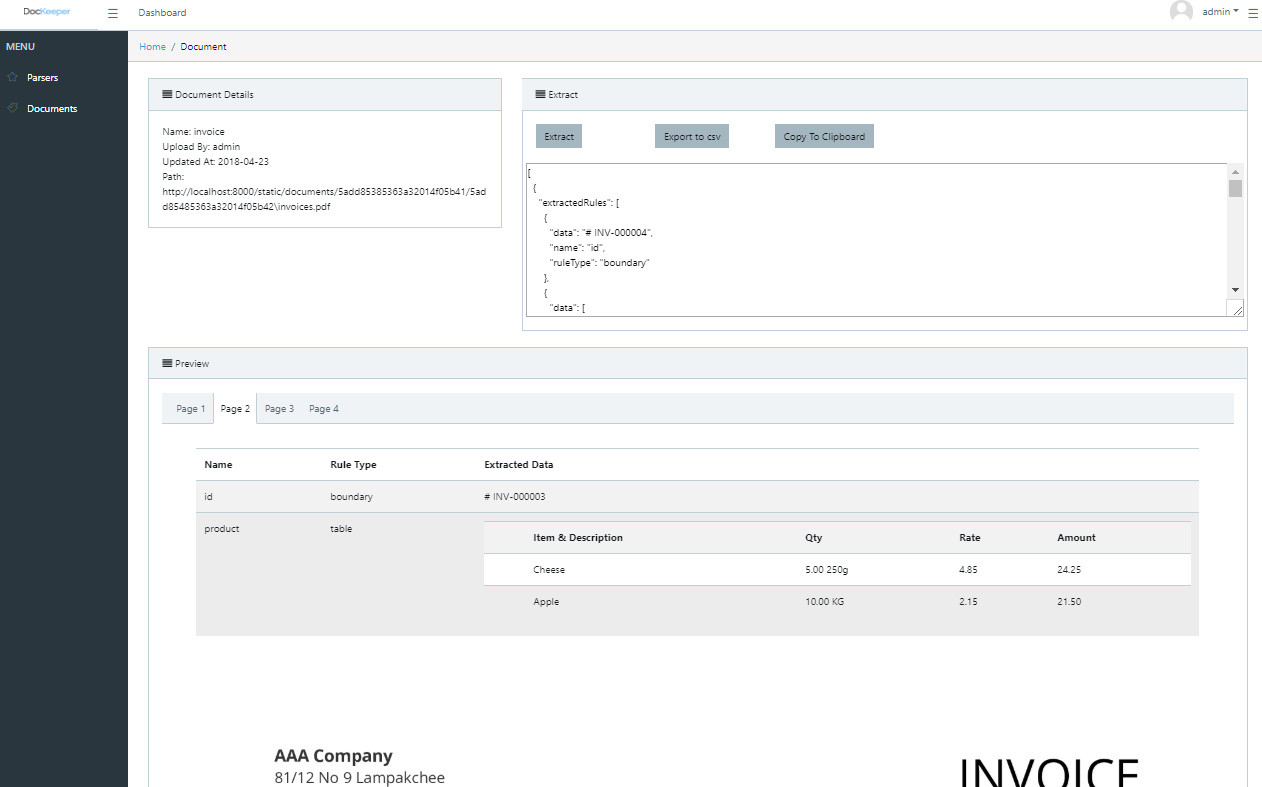
* **Add Parser Rule**

****

รูปที่ 4.8 Add Parser Rule

ผุ้ใช้สามารถเพิ่ม Parser Rule ได้เพื่อเพิ่มวิธีการ Extract ข้อมูล ในรูปเป็นการ Extract แบบเลือกเฉพาะใน Boundary

* **Document Page**



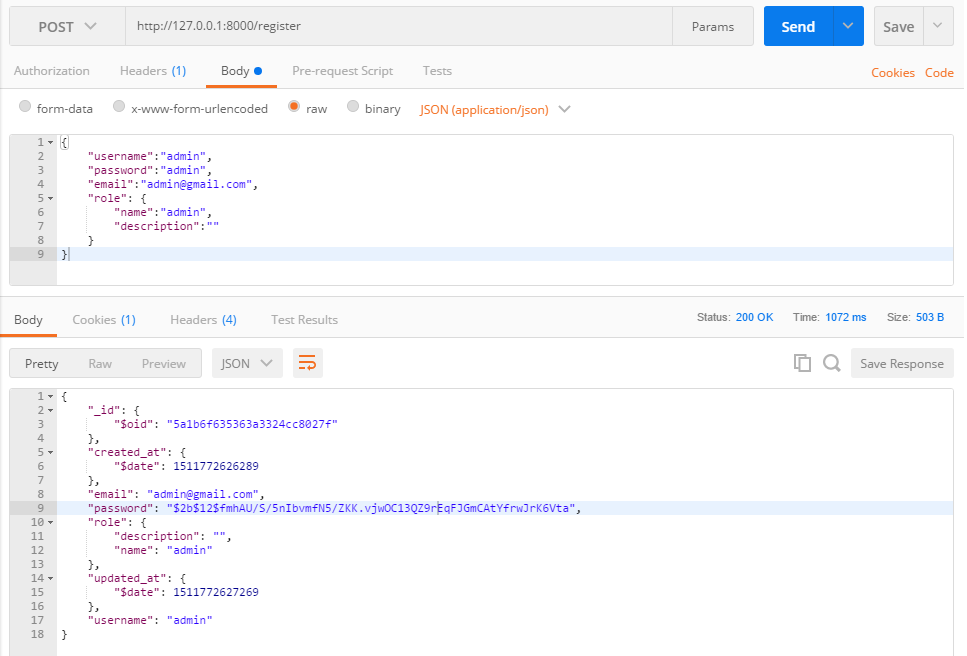
รูปที่ 4.9 Document Page

ในหน้านี้จะแสดงข้อมูลของเอกสาร และแสดง preview data ก่อน export ออกมา สามารถ Export ได้ทั้ง json และ csv และสามารถดาวโหลดเอกสารมาใช้งานได้

## 4.2 Application programming interface (API)

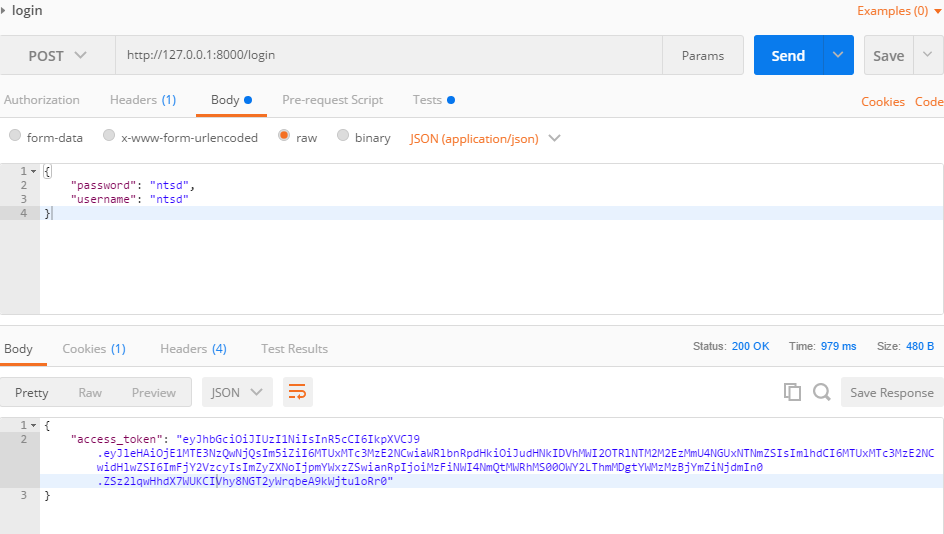
API ของระบบถูกเขียนด้วย Flask Web Framework โดยใช้เทคโนโลยี Restful API ซึ่งจะใช้การส่งข้อมูลในรูปแบบของ JSON ฟอร์เเมต โดยในรูปเราจะทดสอบ API ด้วย Software Postman ซึ่งเป็นการสร้าง Request แล้วรอรับ Response จากทางAPI

* **Register**

****

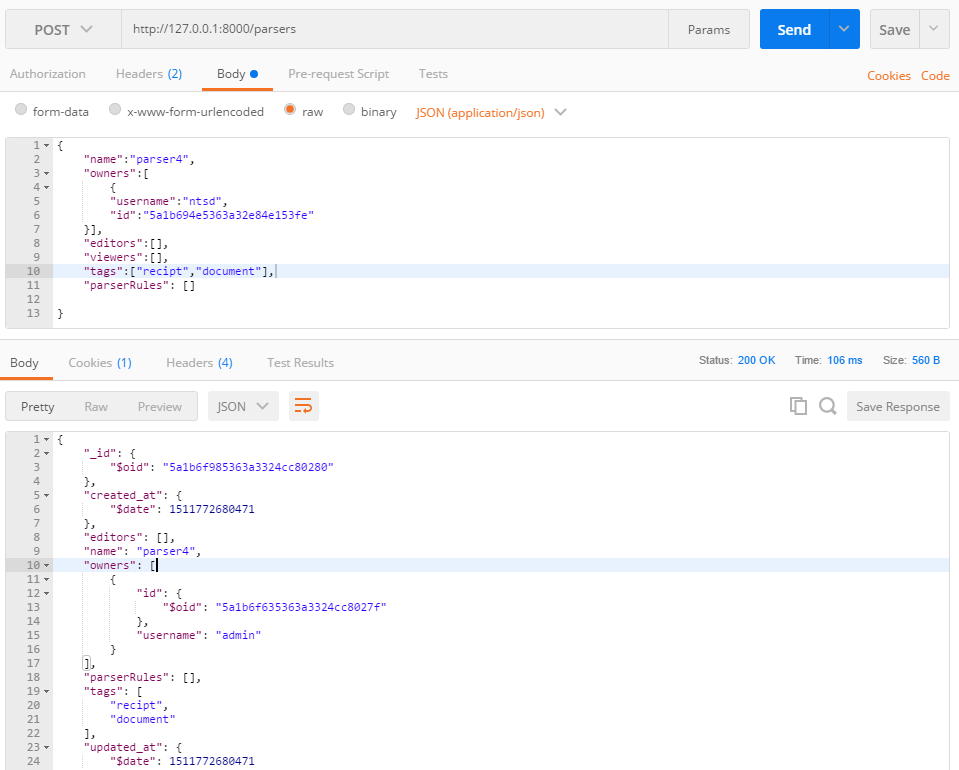
รูปที่ 4.10 Register API

* **Login**

****

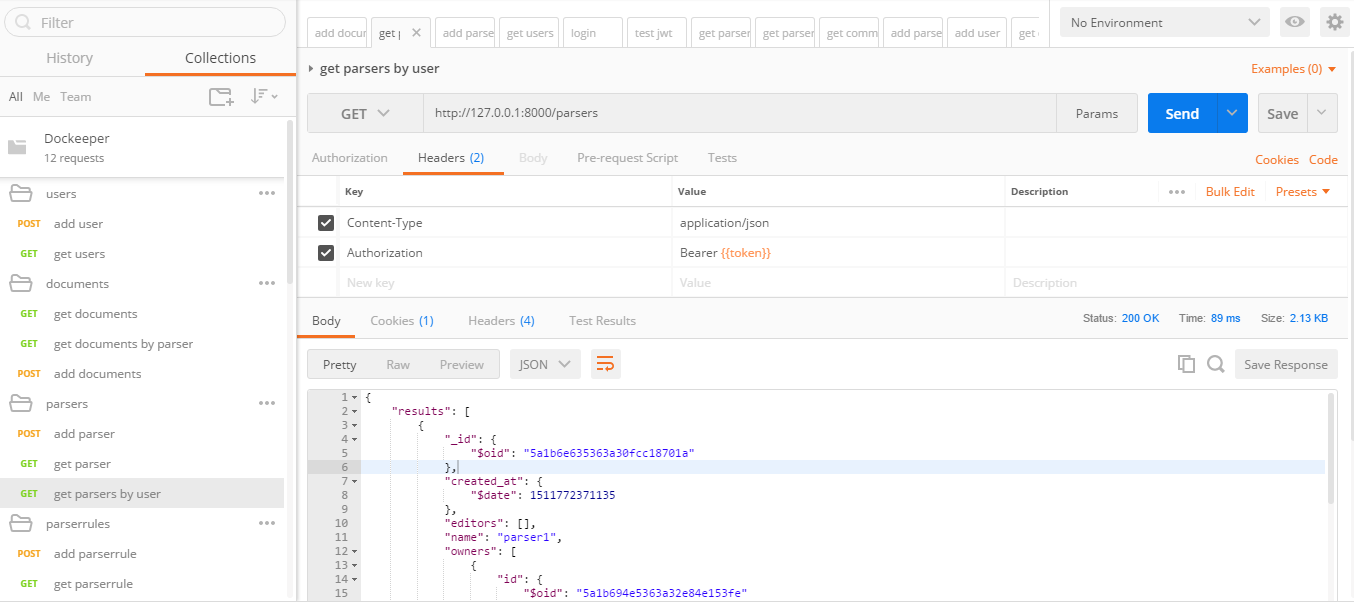
รูปที่ 4.11 Login API

* **Add Parser**

****

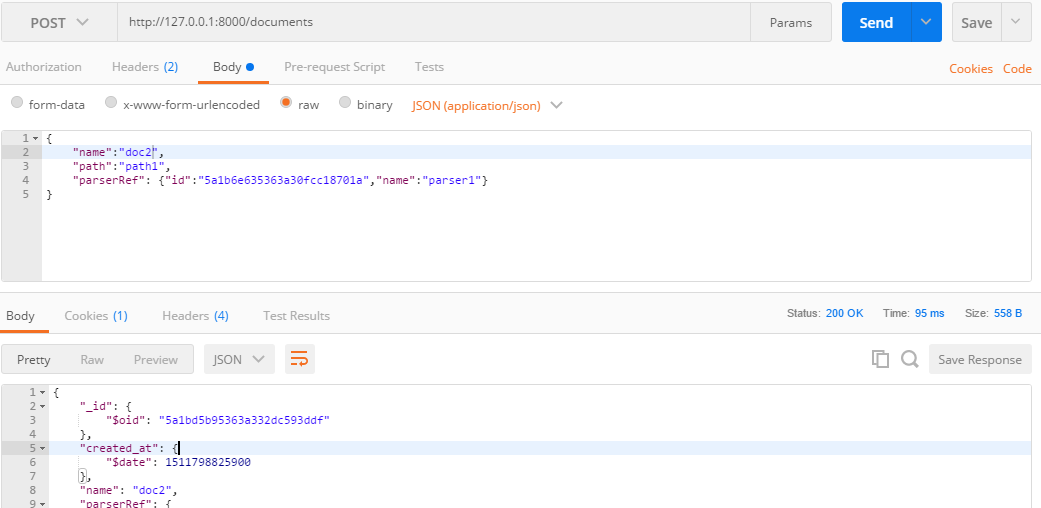
รูปที่ 4.12 Add Parser API

* **Get Parsers by User**

****

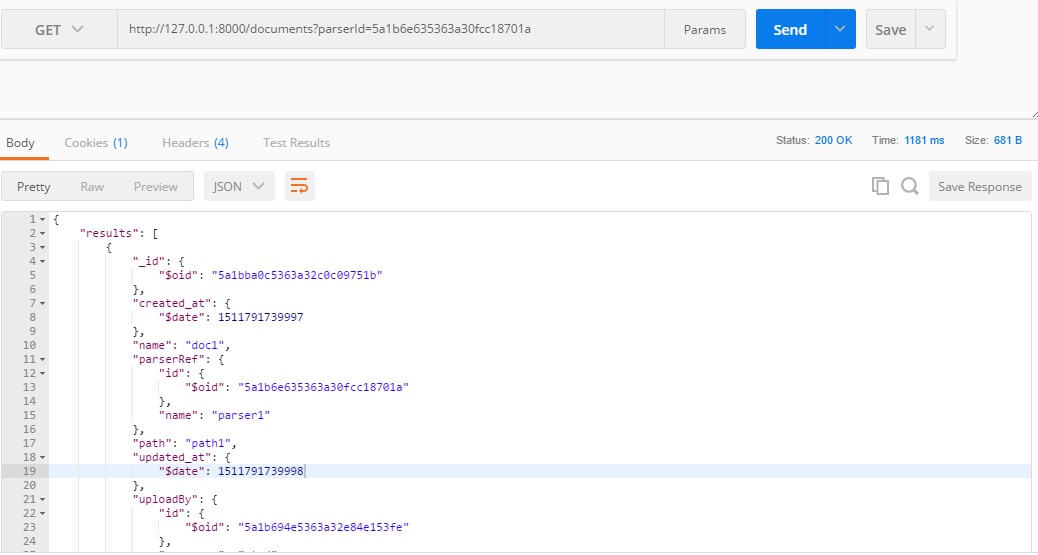
รูปที่ 4.13 Get Parsers by User API

* **Add Document**

****

รูปที่ 4.14 Add document API

* **Get Documents by Parser**

****

**รูปที่ 4.15 Get Documents by Parser API**

4.3 ผลการทำ Classification

4.3.1 ผลการปรับค่า Alpha ของ SGD

รูปที่ 4.16 กราฟแสดง F-score ของการปรับ Parameter ของ SGD

รูปที่ 4.17 กราฟแสดง F-score ของการปรับ Parameter ของ SGD โดยเฉลี่ย

4.3.2 ผลการปรับค่า neighbor ของ KNN

รูปที่ 4.18 กราฟแสดง F-score ของการปรับ Parameter ของ KNN

รูปที่ 4.19 กราฟแสดง F-score ของการปรับ Parameter ของ KNN โดยเฉลี่ย

4.3.3 ผล Mean ของ Nearest Centroid Classifier

รูปที่ 4.20 กราฟแสดง F-score ของ Nearest Centroid

4.3.4 ผล F-score ของการเทส 10 รอบ ของทุกๆ Method

รูปที่ 4.20 กราฟแสดง F-score ของทุกๆ Method จากการทดสอบ 10 รอบ

4.3.5 เวลาในการใช้เฉลี่ยของทุก Method

รูปที่ 4.20 กราฟแสดงเวลาเฉลี่ยของทุกๆ Method จากการทดสอบ 10 รอบ

**บทที่ 5**

สรุปผลการดำเนินงาน

## 5.1 สรุปผลภาคเรียนที่ 2

* เพิ่มปุ่ม Edit/Delete Parser
* เพิ่มปุ่ม Edit/Delete Document
* เพิ่มปุ่ม Edit/Delete Parser Rule
* สามารถอัพโหลด PDF เเละเเยกหน้าได้
* สามารถ Extract ข้อมูลเเบบตารางได้
* สามารถกำหนด Characters Whitelist เองได้
* สามารถเปลี่ยน Sample Document ได้
* ทำให้ข้อมูลไม่ต้องโหลดตลอดเวลาด้วย Vuex Store
* สามารถจำกัดสิทธ์การใช้งาน Owner สามารถเเก้ไข Parser ได้ Editors สามารถเเก้ไขเอกสารได้
* Export to csv
* Recommend feature
* อัลกอริทึมที่เลือกใช้ในระบบคือ Nearest Centroid เนื่องจากมีความเร็วมากที่สุด แต่ความแม่นยำไม่ได้น้อยมาก และเหมาะกับการใช้งานระบบที่ต้องทำการ Fit Model ตลอดเวลา

## 5.2 ข้อเสนอเเนะเเละเเนวทางการพัฒนาระบบในอนาคต

เราจะทำให้ระบบมีการพัฒนายิ่งขึ้นโดยการเพิ่มฟีเจอร์ต่างๆเข้าไป โดยเราได้กำหนดเอาไว้ก่อนเเล้ว บางส่วนได้มีการทดลองเเละออกเเบบเอาไว้เเล้ว เเต่ยังไม่ได้ลงมือนำส่วนนี้มาใช้ในระบบ โดยฟีเจอร์ต่างๆที่จะพัฒนาเพิ่มเติมมีดังนี

* ทำ Extract Queue เพราะการ OCR มีการใช้เวลาในการทำงาน

## 5.3 ปัญหาเเละอุปสรรคที่พบ

1. Library ที่เลือกใช้ในตอนเเรกไม่สามารถทำ Custom router ได้

2.OCR การทำงานช้า ต้องมีการทำ Queue สำหรับ Extract

3.การพัฒนาทำได้ช้ากว่าเวลาที่กำหนด

ภาคผนวก ก

คู่มือการติดตั้งอย่างละเอียด

Python 3.5 or above

1. Download Python ที่ <https://www.python.org/downloads/>

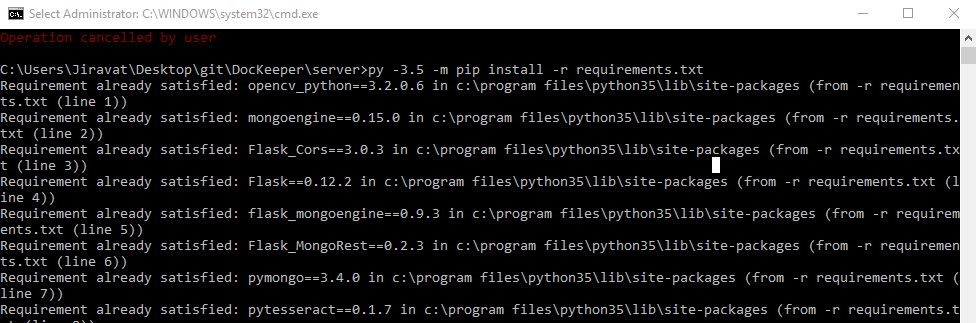
2. Run as administrator and Install (click on add Python 3.6 to PATH)

รูปที่ ก.1 การติดตั้ง Python

Install Python Library

ไปที่ folder Server

เปิด command line interface ใช้คำสั่ง ‘py -3 -m pip install -r requirements.txt’



รูปที่ ก.2 การติดตั้ง Python Library

Install ImageMagick

โหลดได้ที่ <https://www.imagemagick.org/script/download.php>

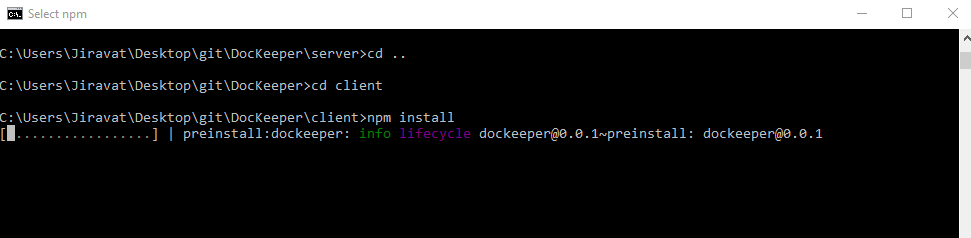
Install Node.js

โหลดได้ที่ <https://nodejs.org/en/download/>

Install Node package

ไปที่โฟลเดอร์ Client

พิมพ์คำสั่ง ‘npm install



รูปที่ ก.3 การติดตั้ง Node.js

Install MongoDB

สามารถดูวิธีการติดตั้งได้ที่นี่ <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>

สร้าง Database ชื่อ dockeeper-test

ติดตั้ง Tesseract 4.0

โหลดได้ที่  <https://github.com/tesseract-ocr/tesseract/wiki/Downloads>

บรรณานุกรม

[1] Andhra Pradesh. “OPTICAL CHARACTER RECOGNITION TECHNIQUE ALGORITHMS.” . 2016

[2] P. Forssen, D. Lowe, “S-H Wang.Region detectors.” . 2014

[3] Justin Johnson. “CS231n: Convolutional Neural Networks for Visual Recognition. “ [Online]. Available : <http://cs231n.github.io/convolutional-networks/> .2017

[4] Brandon Rohrer.”How do Convolutional Neural Networks work?.” [Online].

Available :<https://brohrer.github.io/how_convolutional_neural_networks_work.html>

[5] SUNIL RAY.”Understanding Support Vector Machine algorithm from examples (along with code). “[Online]. Available :<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/> .2017

[6] Unknow.” What Is A Non Relational Database” [Online]. Available:https://www.mongodb.com/scale/what-is-a-non-relational-database . 2016

[7] Vinay Patlolla.” How to make SGD Classifier perform as well as Logistic Regression using parfit” [Online]. Available: <https://towardsdatascience.com/how-to-make-sgd-classifier-perform-as-well-as-logistic-regression-using-parfit-cc10bca2d3c4> . 2017

[8] Unknow.”Scientific Computing Tools for Python” [Online]. Available: [https://www.scipy.org/about.html . 2018](https://www.scipy.org/about.html%20.%202018)

ประวัติผู้เขียน

ชื่อ-นามสกุล นายจิรวัฒน์ บุญกำเหนิด

วัน เดือน ปี เกิด 1เมษายน 2539

ที่อยู่ 81/12 หมู่ 9 ถนนฉลองกรุง แขวงลำผักชี เขตหนองจอก กรุงเทพ 15030

ประวัติการศึกษา วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง