

ฟาร์มหลังบ้าน : การออกแบบและพัฒนาระบบเบื้องหลังสำหรับโลกเสมือนที่มีผู้เล่นจำนวนมาก

The Backyard Farm : Designing and Developing a back-end system for a massive multiplayer online virtual world



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Backyard Farm : Designing and Developing a back-end
system for a massive multiplayer online virtual world

The seal of King Mongkut's Institute of Technology Ladkrabang is a circular emblem. It features a central sunburst with rays emanating from a central point. Below the sunburst are three tiered, ornate structures resembling traditional Thai stupas or pagodas, each supported by a decorative base. The entire emblem is surrounded by a circular border containing Thai text. The text at the top of the border reads 'สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง' (King Mongkut's Institute of Technology Ladkrabang).

Kidssana Chinsoi

Ratchakorn Pattaranavik

A SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT

FOR THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER
SCIENCE) DEPARTMENT OF, FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ

ฟาร์มหลังบ้าน : การออกแบบและพัฒนาระบบเบื้องหลังสำหรับ
โลกเสมือนที่มีผู้เล่นจำนวนมาก

The Backyard Farm : Designing and Developing a back-
end system for a massive multiplayer online virtual
world

ชื่อนักศึกษา

นาย กฤษณะ ชินสร้อย

รหัสนักศึกษา 58050206

นาย รัชกร ภัทรนาวิก

รหัสนักศึกษา 58050364

ปริญญา

วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชา

วิทยาศาสตรคอมพิวเตอร์

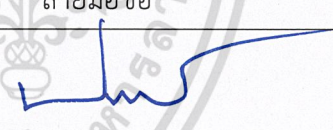

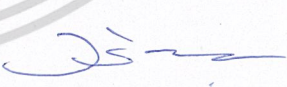
ปีการศึกษา

2561

อาจารย์ที่ปรึกษา

ดร.วิษณุ ต่ดวงศ์ไพชยนต์

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้
ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (ชื่อหลักสูตร)
ประจำปีการศึกษา 2561

คณะกรรมการสอบ	ลายมือชื่อ
ดร.ประพนธ์ ศรีนิวัดติวงศ์ ประธานกรรมการ	
ผศ.ธีระ ศิริธีรากล กรรมการ	
ดร.วิษณุ ต่ดวงศ์ไพชยนต์ อาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ ฟาร์มหลังบ้าน : การออกแบบและพัฒนาระบบเบื้องหลังสำหรับโลกเสมือนที่มีผู้เล่นจำนวนมาก

The Backyard Farm : Designing and Developing a back-end system for a massive multiplayer online virtual world

ชื่อนักศึกษา	นาย กฤษณะ ชินสร้อย	รหัสนักศึกษา 58050206
	นาย รัชกร ภัทธนาวิก	รหัสนักศึกษา 58050364
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)	
ภาควิชา	วิทยาการคอมพิวเตอร์	
คณะ	วิทยาศาสตร์	
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)	
ปีการศึกษา	2561	
อาจารย์ที่ปรึกษา	ดร.วิษุฒะ ต่ดวงศ์ไพชยนต์	

บทคัดย่อ

ทางทีมพัฒนาได้ร่วมงานกับทีมงานที่ต้องการจะลดช่องว่างระหว่างผู้คนที่อยู่ภายในเมืองที่สนใจการทำกาเกษตรและเกษตรกรเข้าด้วยกัน โดยมีปัญหาอยู่ว่าในช่วงเวลาของการเพาะปลูกอาจจะใช้เวลาค่อนข้างยาวนาน จึงทำให้การเชื่อมต่อระหว่างกลุ่มบุคคลสองกลุ่มนี้ไม่สามารถเป็นไปได้อย่างยาวนานจนกระทั่งจบกระบวนการ ทางทีมพัฒนาจึงมีแนวคิดที่จะนำหลักการออกแบบเกมเข้ามาร่วมประยุกต์ใช้ โดยสร้างโลกเสมือนขึ้นมาเพื่อนำเสนอการลงทุนโดยจะมีฟาร์มจำลองเป็นพื้นที่หลักในการเชื่อมต่อระหว่างพื้นที่เพาะปลูกของเกษตรกรกับผู้ใช้งานที่อยู่อาศัยภายในเมือง นอกจากนี้ผู้ใช้งานยังสามารถเข้ามาใช้งานโลกเสมือนเป็นสถานที่สำหรับพักผ่อนหย่อนใจในฟาร์มของโลกเสมือนเพื่อให้เกิดความสบายใจ และยังมีกรจำลองพื้นที่ในส่วนของเมืองเพื่อให้ผู้ใช้งานหลายคนสามารถมาพบปะพูดคุยกันเป็นเครือข่ายสังคมออนไลน์บนโลกเสมือน รวมถึงมีองค์ประกอบและกิจกรรมต่าง ๆ ภายในโลกเสมือนเพื่อดึงดูดผู้ใช้งานให้ทำการใช้งานโลกเสมือนระหว่างรอผลผลิตทางการเกษตร

โดยทางกลุ่มการทำงานเบื้องหลังได้มีการจัดทำระบบฐานข้อมูลสำหรับเก็บข้อมูลที่มีความจำเป็นต่อการทำงานและการใช้งานเบื้องต้นของระบบโลกเสมือน และได้มีการสร้างเอพีไอขึ้นมาเพื่อช่วยลดขั้นตอนการทำงานให้กับกลุ่มการทำงานเบื้องหน้าระหว่างการพัฒนาโลกเสมือน รวมถึงการทดสอบการทำงานของระบบโลกเสมือนหลังจากที่ได้ไฮสตร์บบลงบนเซิร์ฟเวอร์ Google Cloud

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Platform จากผู้ทดลองใช้งานจำนวน 20 คน ซึ่งมีผู้เข้าใช้งานพร้อมกันสูงสุด 17 คน โดยมีการทดสอบจำนวน 3 ครั้ง

ผลการทดสอบจากการสังเกตการณ์การทำงานของระบบโลกเสมือนเมื่อมีผู้ใช้งานจำนวนหลายคนพบว่าระบบสามารถทำงานได้อย่างปกติในช่วงที่มีผู้ใช้งานระหว่าง 1 – 13 คน แต่ไม่สามารถทำงานด้วยประสิทธิภาพสูงสุดได้ในช่วงผู้ใช้งานตั้งแต่ 14 คน ขึ้นไป ซึ่งต่ำกว่าเกณฑ์ที่คาดการณ์ไว้ซึ่งเกิดมาจากการใช้หน่วยประมวลผลกลางของคอมพิวเตอร์ที่ต่ำเกินไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title The Backyard Farm : Designing and Developing a back-end system for a massive multiplayer online virtual world

Students Mr. Kidssana Chinsoi Student ID 58050206
Mr. Ratchakorn Pattaranavik Student ID 58050364

Degree Bachelor of Science (Computer Science)

Department Science

University King Mongkut's Institute of Technology Ladkrabang (KMITL)

Academic Year 2018

Advisor Dr. Witchaya Towongpaichayont

Abstract

This special problem is a collaboration between our team and another team who intends to bring urban people who are interested in cultivating and actual agriculturists together. Due to a long timeframe in each cultivating process, there are some possibilities that the processes could be dismissed incompletely. This work's main purpose is to use game design concepts to develop a virtual world that can facilitate engagement between these groups of people on an ongoing basis and providing representations of the real-world agricultural. This virtual world is also aimed to be used as users' social and relaxing place with town area that allows users to communicate with each other. Various events and activities can be included in this virtual world later to keep the users engaged throughout the real-world cultivating processes.

The back-end development team are responsible for preparing and configuring the database for the virtual world. We have also developed APIs to simplify database access for front-end development team. We also perform a series of server tolerance tests of the system on Google Cloud Platform with 20 users 3 times. The highest number of users which logged in simultaneously was 17 users.

After observing on a series of tests, the result reveals that server can operate optimally while there are not more than 13 users logging in simultaneously.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The performance gradually dropped as the user number is increased. This number is lower than the threshold of the back-end development team had initially predicted which expected to be the result of the performance of the CPU of server.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการจัดทำปัญหาพิเศษเรื่องฟาร์มหลังบ้านนั้นสามารถประสบผลสำเร็จด้วยความอนุเคราะห์ของบุคคลหลายฝ่าย ทางผู้พัฒนาจึงขอกราบขอบพระคุณในด้านของการให้ความช่วยเหลือและคำปรึกษาดังต่อไปนี้

ขอขอบพระคุณ ดร.วิษณุ ต๋องศรีไพชยนต์ อาจารย์ที่ปรึกษาในการจัดทำปัญหาพิเศษนี้ที่ช่วยให้คำปรึกษา ให้คำแนะนำ ตรวจสอบและแก้ไขข้อบกพร่องในรูปแบบต่าง ๆ มาโดยตลอดจนสามารถสำเร็จลุล่วงมาได้ด้วยดี

ขอขอบพระคุณ บิดา มารดา และสมาชิกในครอบครัวทุกท่าน ที่ช่วยให้กำลังใจและสนับสนุนการศึกษาตลอดมา

ขอขอบพระคุณ นาย ธีรบุคัลย์ โพธิ์เขียว (พี่เค้ก) ที่ช่วยให้คำแนะนำ ให้คำปรึกษา และชี้แนะแนวทางในการเขียนเล่มวิทยานิพนธ์เล่มนี้ได้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ บุคลากรจากกิลด์ **คอร์गी-แอตแทค** (Corgi Attack, คอร์กี แอทแทค) จากเกม Granblue Fantasy ที่ช่วยในการทดสอบการทำงานของเซิร์ฟเวอร์เบื้องต้น

ขอขอบพระคุณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่มอบโอกาส และพื้นที่ทางการศึกษาให้

สุดท้ายนี้ผู้จัดทำขอขอบพระคุณคณาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ที่มอบความรู้ตลอดระยะเวลาระยะเวลาที่ผ่านมา ทำให้ผู้จัดทำได้มีความรู้ความสามารถในการพัฒนาโครงการปัญหาพิเศษได้สำเร็จ

กฤษณะ ชินสร้อย

รัชกร ภัทรนาวิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ฅ
สารบัญตาราง.....	จ
สารบัญรูป.....	ฉ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขต.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 API.....	4
2.2 การรับส่งข้อมูลระหว่าง API กับ Web Server.....	7
2.2.1 เว็บเซิร์ฟเวอร์ (WEB Server).....	7
2.2.2 RESTful Web service.....	7
บทที่ 3 การวิเคราะห์และออกแบบระบบฐานข้อมูล.....	10
3.1 สถาปัตยกรรมของระบบงานโดยรวม.....	10
3.2 การวิเคราะห์และออกแบบฐานข้อมูล.....	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.3 API.....	32
3.4 การเลือกใช้เซิร์ฟเวอร์ที่ใช้ในการโฮสต์ระบบ	34
บทที่ 4 ผลการวิจัยและการอภิปรายผล	35
4.1 ผลการทดสอบการทำงานของระบบโลกเสมือน	35
4.2 ผลการใช้งานของ API ที่มีต่อกลุ่มการทำงานเบื้องหน้า.....	37
4.3 ความต้องการขั้นต่ำของระบบ.....	40
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	41
5.1 สรุปผลการดำเนินการ.....	41
5.1.1 การทำงานของระบบโลกเสมือน	41
5.1.2 ผลการใช้งานของ API ที่มีต่อกลุ่มการทำงานเบื้องหน้า.....	42
5.2 ข้อเสนอแนะ	42
5.3 การพัฒนาในอนาคต.....	42
เอกสารอ้างอิง	43
ภาคผนวก.....	44
ภาคผนวก ก.	44
ภาคผนวก ข.....	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 Use case description “Record Investment History”	14
3.2 โครงสร้างตาราง Account	14
3.3 โครงสร้างตาราง Character	14
3.4 โครงสร้างตาราง Decoration	14
3.5 โครงสร้างตาราง Decoration_Type	14
3.6 โครงสร้างตาราง Default_Value	14
3.7 โครงสร้างตาราง Farm.....	14
3.8 โครงสร้างตาราง Farm_Species	14
3.9 โครงสร้างตาราง Invest	14
3.10 โครงสร้างตาราง Items.....	14
3.11 โครงสร้างตาราง Item_type	14
3.12 โครงสร้างตาราง Plant.....	14
3.13 โครงสร้างตาราง RealFarm	14
3.14 โครงสร้างตาราง Server_list.....	14
3.15 โครงสร้างตาราง Species.....	14
1x API ลำดับที่ 1	47
2x API ลำดับที่ 2	47
3x API ลำดับที่ 3	48
4x API ลำดับที่ 4	48
5x API ลำดับที่ 5	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง(ต่อ)

6ข API ลำดับที่ 6.....	50
7ข API ลำดับที่ 7.....	50
8ข API ลำดับที่ 8.....	51
9ข API ลำดับที่ 9.....	52
10ข API ลำดับที่ 10.....	52
11ข API ลำดับที่ 11.....	53
12ข API ลำดับที่ 12.....	53
13ข API ลำดับที่ 13.....	54
14ข API ลำดับที่ 14.....	54
15ข API ลำดับที่ 15.....	55
16ข API ลำดับที่ 16.....	55
17ข API ลำดับที่ 17.....	56
18ข API ลำดับที่ 18.....	57
19ข API ลำดับที่ 19.....	57
20ข API ลำดับที่ 20.....	58
21ข API ลำดับที่ 21.....	58
22ข API ลำดับที่ 22.....	59
23ข API ลำดับที่ 23.....	59
24ข API ลำดับที่ 24.....	60
25ข API ลำดับที่ 25.....	60
26ข API ลำดับที่ 26.....	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง(ต่อ)

27ข API ลำดับที่ 27	61
28ข API ลำดับที่ 28	62
29ข API ลำดับที่ 29	62
30ข API ลำดับที่ 30	63
31ข API ลำดับที่ 31	63
32ข API ลำดับที่ 32	64
33ข API ลำดับที่ 33	64
34ข API ลำดับที่ 34	65
35ข API ลำดับที่ 35	65
36ข API ลำดับที่ 36	66
37ข API ลำดับที่ 37	66
38ข API ลำดับที่ 38	67
39ข API ลำดับที่ 39	67
40ข API ลำดับที่ 40	68
41ข API ลำดับที่ 41	68
42ข API ลำดับที่ 42	69
43ข API ลำดับที่ 43	69
44ข API ลำดับที่ 44	70
45ข API ลำดับที่ 45	70
46ข API ลำดับที่ 46	71
47ข API ลำดับที่ 47	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง(ต่อ)

48ข API ลำดับที่ 48	72
49ข API ลำดับที่ 49	72
50ข API ลำดับที่ 50	73
51ข API ลำดับที่ 51	73
52ข API ลำดับที่ 52	74
53ข API ลำดับที่ 53	74
54ข API ลำดับที่ 54	75
55ข API ลำดับที่ 55	75
56ข API ลำดับที่ 56	76
57ข API ลำดับที่ 57	76
58ข API ลำดับที่ 58	77
59ข API ลำดับที่ 59	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 รับคำร้องขอจากไคลแอนท์.....	5
2.2 ติดต่อส่งคำร้องขอไปยังระบบ.....	5
2.3 รับคำตอบรับคำขอจากระบบ.....	6
2.4 ส่งคำตอบรับคำขอไปยังไคลแอนท์.....	6
2.5 HTTP Request Format.....	8
2.6 HTTP Response Format.....	9
3.1 สถาปัตยกรรมการทำงานของระบบโดยรวม.....	11
3.2 สถาปัตยกรรมของเซิร์ฟเวอร์.....	12
3.3 Main Use Case Diagram.....	13
3.4 Subsystem Use Case : Request.....	15
3.5 Subsystem Use Case : Request 2.....	16
3.6 Subsystem Use Case : Request 3.....	17
3.7 Subsystem Use Case : Request 4.....	18
3.8 Subsystem Use Case : Parse.....	19
3.9 Subsystem Use Case : Json Object Manage 1.....	20
3.10 Subsystem Use Case : Json Object Manage 2.....	21
3.11 ER-Diagram ของฐานข้อมูล.....	22
3.12 สถาปัตยกรรมการทำงานของ Back End API โดยรวม.....	32
3.13 สถาปัตยกรรมการทำงานของ API.....	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.14 Google Cloud Platform.....	34
4.1 ภาพตัวอย่างช่วงเวลาการทำงานของ CPU จำนวน 1 คอร์ บนเซิร์ฟเวอร์ที่ติดตั้งระบบโลก เสมือน ระหว่างการทำสอบการใช้งานโลกเสมือนโดยผู้ใช้งานหลายคน.....	36
4.2 ภาพตัวอย่างผู้เข้าร่วมทดสอบที่กำลังเข้าใช้งานระบบ	36
4.3 ตัวอย่างฟังก์ชันบลูพริ้นท์ไลบรารีภายใน Unreal Engine 4 สำหรับเรียกใช้งานข้อมูลที่กลุ่ม การทำงานเบื้องหลังได้สร้างขึ้น	37
4.4 เปรียบเทียบระหว่างกลุ่มการทำงานเบื้องหน้าเขียนฟังก์ชันเรียกใช้งาน Item ด้วยตนเอง	39
1ก Unreal Engine.....	44
2ก Visual Studio	45
3ก AppServ	45
4ก FileZilla	45
5ก Google Cloud Platform.....	46
6ก MySQL-Front และ MySQL Workbench.....	46

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในการทำการเกษตรจำเป็นต้องใช้พื้นที่ในการเพาะปลูกมีการใช้เงินลงทุนสูงและระยะเวลาในการเพาะปลูก แต่ละครั้งค่อนข้างใช้เวลานาน รวมถึงปัญหาหลังจากเก็บเกี่ยวผลผลิต เช่น เกิดสินค้าล้นตลาดและทำให้สินค้ามีราคาที่ถูกลงอย่างมาก และยังมีปัญหาที่ตามมาคือการกู้เงินมาเพื่อทำการเพาะปลูกรอบถัดไปเพื่อชดเชยเงินที่ขาดทุนไปเมื่อครั้งก่อนเป็นวงจรมีปัญหาไม่มีสิ้นสุด ในทางกลับกันผู้ที่อาศัยอยู่ใจกลางเมืองหลายคนมีความคิดอยากไปทำการเกษตร และมีเงินทุนมากพอที่จะใช้ลงทุนกลับพบปัญหาในอีกมุมหนึ่ง คือ ไม่มีพื้นที่เพาะปลูก และไม่คุ้มค่าเวลาที่เสียไป ทีมพัฒนาได้ร่วมงานกับทีมงานที่สามารถลดช่องว่างระหว่างคนสองกลุ่มได้ แต่อย่างไรก็ตามการเพาะปลูกก็ยังคงใช้เวลาค่อนข้างนาน ซึ่งอาจทำให้การเชื่อมต่อระหว่างคนสองกลุ่มไม่สามารถเป็นไปได้ อย่างยาวนานจนกระทั่งจบกระบวนการ ทีมพัฒนาจึงมีแนวคิดที่จะนำหลักการออกแบบเกมเข้ามาร่วมประยุกต์ใช้ โดยสร้างโลกเสมือนขึ้นมาเพื่อทำการเพาะปลูก แล้วยังมีองค์ประกอบและกิจกรรมต่าง ๆ ภายในโลกเสมือนเพื่อดึงดูดผู้ใช้งานให้ใช้งานโลกเสมือนนี้ระหว่างรอผลผลิต ทีมพัฒนาเห็นถึงความเป็นไปได้ที่จะขยายประสบการณ์ผู้ใช้ที่ตีมากขึ้นจากระบบการเชื่อมต่อระหว่างเกษตรกรและคนเมืองที่มีอยู่แล้ว โดยการสร้างโลกเสมือนมาเพื่ออำนวยความสะดวกการใช้งานของคนในเมือง และ เป็นการสร้างความรู้สึกในการใช้ชีวิตที่ใกล้เคียงกับเกษตรกรมากขึ้น

โลกเสมือนจะสามารถเข้าถึงได้ง่ายผ่านคอมพิวเตอร์ โดยจะมีฟาร์มจำลองเป็นพื้นที่หลักในการเชื่อมต่อระหว่างพื้นที่เพาะปลูกของเกษตรกรกับผู้เล่นที่เป็นคนในเมือง มีการสอดแทรกความเป็นเกมลงไปเพื่อให้ผู้ใช้งานสามารถเข้ามาใช้เป็นที่สำหรับพักผ่อนหย่อนใจภายในฟาร์มของโลกเสมือนให้เกิดความสบายใจ นอกจากนี้ยังมีการจำลองพื้นที่ในส่วนของเมืองเพื่อให้ผู้เล่นหลายคนมาพบปะพูดคุยกันเป็นเครือข่ายสังคมออนไลน์บนโลกเสมือน และพื้นที่ซื้อขายผลผลิตสำหรับผู้ใช้งานที่ต้องการจำหน่ายผลผลิตของตนเองที่ได้จากการทำการลงทุนกับเกษตรกร โดยทางกลุ่มได้จัดทำระบบฐานข้อมูล (Database) มาเพื่อรองรับการทำงานและเก็บข้อมูลของกลุ่มการทำงานเบื้องหลัง (Back end) ให้กับกลุ่มการทำงานเบื้องหน้า (Front end) และได้สร้าง API ขึ้นมาให้ทางกลุ่มผู้ดูแลการทำงานเบื้องหน้าเรียกใช้งานข้อมูลได้อย่างสะดวก รวมทั้งติดตั้งระบบฐานข้อมูลและดูแลการเชื่อมต่อของเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อพัฒนาฐานข้อมูล (Database) ให้รองรับกับระบบการเก็บและเรียกใช้ข้อมูลทั้งหมด สำหรับผู้เล่นหลายคน (เช่น Character, Inventory, Decorations และอื่นๆ)
- 2) สร้าง API ให้กับกลุ่มการทำงานเบื้องหน้า (Front end) เพื่อนำข้อมูลจากฐานข้อมูลไปใช้ได้ง่าย

1.3 ขอบเขตของงานวิจัย

- 1) การเก็บ Requirement เพื่อเป็นข้อมูลในการตัดสินใจสร้างฐานข้อมูล เช่น
 - ข้อมูลผู้ใช้งาน ฟาร์ม และที่ดินที่ใช้ปลูก
 - ข้อมูลของเมล็ดพืช และผลผลิต
 - ข้อมูลช่องทางการปลูก และ เกษตรกร
- 2) การนำฐานข้อมูลที่ถูกสร้างขึ้นมาใช้ทดลอง เช่น
 - การเรียกวัตถุดิบแสดงภายในโลกเสมือน
 - การเคลื่อนย้ายวัตถุภายในห้องสัมภาระ
- 3) สร้าง API ให้กับกลุ่มการทำงานเบื้องหน้าเพื่อเรียกใช้ข้อมูลหรือแก้ไขข้อมูลจากฐานข้อมูล เช่น
 - เรียกใช้งานข้อมูลจากฐานข้อมูล สามารถเพิ่มหรือแก้ไขในส่วนนั้นๆได้ เช่นข้อมูลเมล็ดพันธุ์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) สามารถนำแนวคิดวิธีการวางระบบฐานข้อมูลไปปรับปรุงเพื่อใช้กับเกมแนวอื่นๆได้
- 2) สามารถนำแนวคิดจากการใช้งาน API ในการเชื่อมต่อระหว่างระบบไปปรับปรุงเพื่อใช้งานได้

1.5 ขั้นตอนการดำเนินงาน

- 1) ศึกษาการรับส่งข้อมูลของ API
- 2) ออกแบบฐานข้อมูลตามข้อมูลที่ได้รับมาจากกลุ่มการทำงานเบื้องหน้า
- 3) ศึกษาสิ่งที่จำเป็นต่อการเชื่อมต่อเซิร์ฟเวอร์และฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) ดำเนินการสร้าง API สำหรับกลุ่มการทำงานเบื้องหน้า
- 5) ดำเนินการเชื่อมต่อระบบโลกเสมือนเข้าด้วยกัน
- 6) ทดสอบกาทำงานของโลกเสมือน
- 7) สรุปผลการดำเนินงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

จากปัญหาและขอบเขตการวิจัยทางผู้พัฒนาได้นำความรู้เกี่ยวกับ API และการรับส่งข้อมูลระหว่าง API กับ Web Server มาร่วมใช้งานกับ เซิร์ฟเวอร์-ไคลแอนท์ (Server-Client) เพื่อใช้เป็นตัวกลางการสื่อสารในการเก็บและแลกเปลี่ยนข้อมูลระหว่างเซิร์ฟเวอร์และไคลแอนท์ เช่น ชุดข้อมูลคำสั่งที่ได้รับเข้ามาจากผู้ใช้งาน ชุดข้อมูลคำสั่งการลงทุน โดยจะมีการใช้งานระบบเว็บเซอร์วิสที่ชื่อ Restful ในการติดต่อระหว่างเซิร์ฟเวอร์และฐานข้อมูล

2.1 API [1]

เอพีไอ (API, Application Programming Interface) หรือส่วนต่อประสานโปรแกรมประยุกต์ เป็นซอฟต์แวร์ตัวกลางที่ทำให้แอปพลิเคชันสองอย่างสามารถติดต่อสื่อสารกันได้ โดยสามารถเป็นได้ทั้งระบบปฏิบัติการ(Operating System) ไลบรารี (Library) หรือบริการ (Service) อื่น ๆ

หลักการทํางานของ API

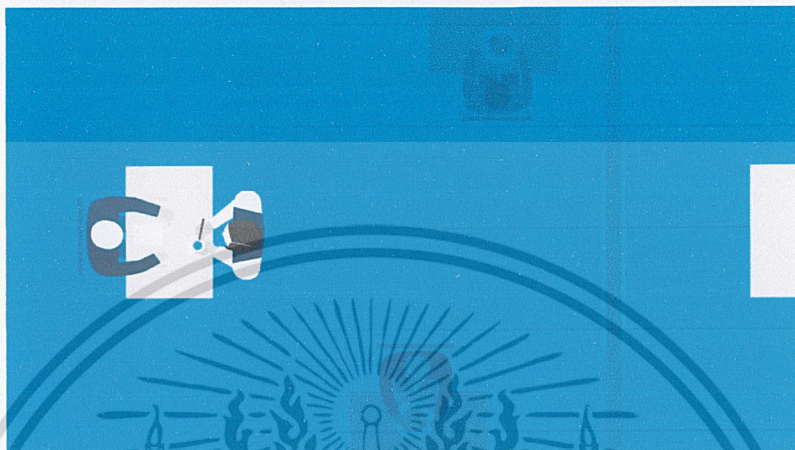
เอพีไอจะเป็นตัวกลางในการรับส่งข้อมูล โดยจะมีการรับคำร้องขอจากผู้ใช้งานหรือไคลแอนท์ (Client) ที่ต้องการเรียกใช้ข้อมูลจากระบบ และนำคำร้องขอข้อมูล (Request) นั้นไปส่งให้กับระบบ (System) และรอคำตอบรับคำขอ (Respond) จากระบบ เมื่อได้รับคำตอบรับแล้วเอพีไอจะนำคำตอบรับคำขอมาส่งให้กับไคลแอนท์เพื่อใช้งานข้อมูลที่ต้องการ

ยกตัวอย่างเช่น

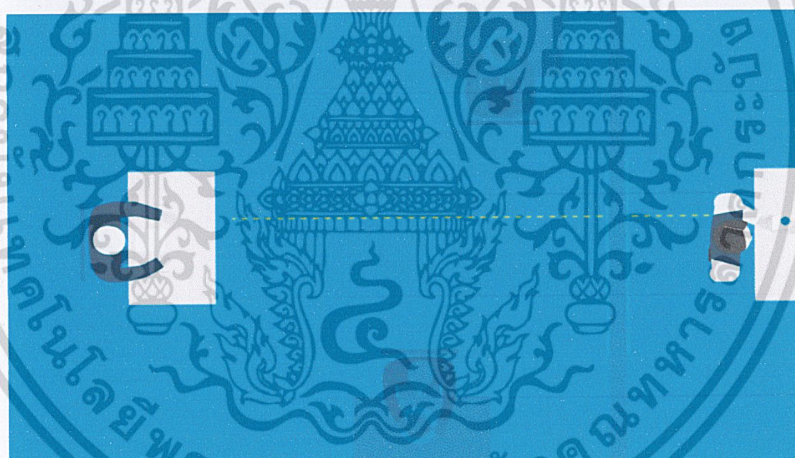
ผู้ใช้งานเปรียบเสมือนลูกค้าที่มาใช้บริการในร้านอาหารและต้องการจะสั่งอาหารเปรียบเสมือนไคลแอนท์ (Client) โดยห้องครัวที่จะมีการจัดเตรียมอาหารนั้นเปรียบเสมือนระบบการที่สองส่วนนี้จะสามารถติดต่อกันได้นั้นจะต้องผ่านบริกรที่เปรียบเสมือนเอพีไอที่เป็นสื่อกลางระหว่างบุคคลสองกลุ่มนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยบริกรจะเป็นผู้รับคำสั่งอาหาร (Request) จากลูกค้า (Client) และนำคำสั่งอาหารไปยังห้องครัว (System) เพื่อเตรียมการตอบรับคำร้องขอ ดังรูป 2.1 และ 2.2



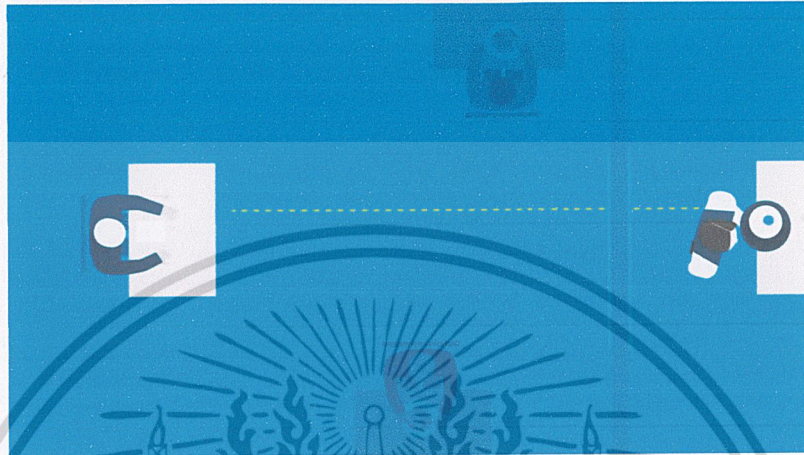
รูปที่ 2.1 รับคำร้องขอจากไคลเอนท์[2]



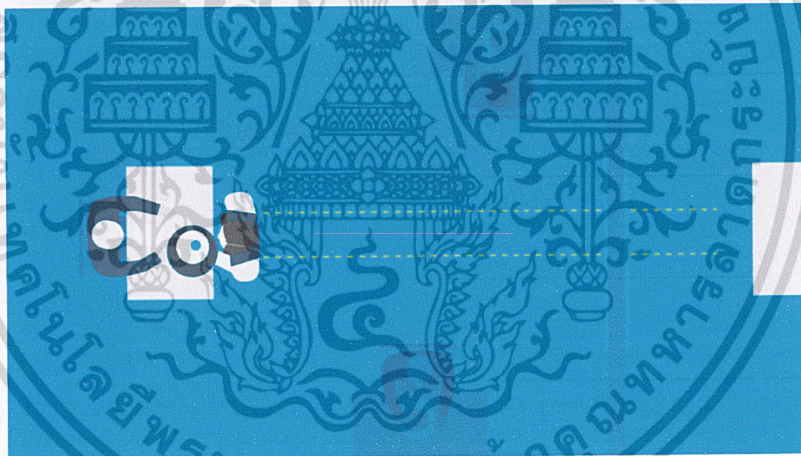
รูปที่ 2.2 ติดต่อส่งคำร้องขอไปยังระบบ[2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากห้องครัวได้ทำการเตรียมอาหารจะทำการส่งคำตอบรับคำร้องขอกลับมา (Respond) ในที่นี้คืออาหารที่สั่ง โดยบริการ (API) จะรับคำตอบรับและส่งกลับไปยังลูกค้า ดังรูป 2.3 และ 2.4



รูป 2.3 รับคำตอบรับคำขอจากระบบ[2]



รูปที่ 2.4 ส่งคำตอบรับคำขอไปยังไคลเอนท์[2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การรับส่งข้อมูลระหว่าง API กับ Web Server

ในระบบโลกเสมือนทางฝั่งผู้ใช้งานเรียกใช้ฟังก์ชันภายในระบบที่กลุ่มการทำงานเบื้องหน้าทำไว้ และทางแอปพลิเคชันจะมีการติดต่อสื่อสารกับระบบฐานข้อมูลภายในเซิร์ฟเวอร์ได้โดยผ่าน เว็บเซิร์ฟเวอร์ด้วยเว็บเซอร์วิสที่เรียกว่า RESTful หรือ REST (Representational state transfer)

2.2.1.เว็บเซิร์ฟเวอร์ (WEB Server)

การทำงานของตัวเว็บเซิร์ฟเวอร์จะทำหน้าที่จัดเก็บข้อมูลและประมวลผลจัดส่งเว็บเพจให้กับผู้ใช้งาน โดยการติดต่อสื่อสารกันระหว่างเซิร์ฟเวอร์และผู้ใช้งานจะผ่านตัวโปรโตคอลที่เรียกว่า Hypertext Transfer Protocol (HTTP) ข้อมูลที่ใช้งานในการรับส่งส่วนใหญ่จะเป็น Hypertext Markup Language (HTML) นอกจากนี้ยังมีรูปภาพ, รูปแบบหน้าเว็บและ สคริปต์

ฝั่งผู้ใช้จะใช้เว็บเบราว์เซอร์ (Web Browser) ในการติดต่อโดยการส่งคำขอข้อมูลทรัพยากรผ่านทาง HTTP เมื่อเซิร์ฟเวอร์ตอบสนองจะส่งข้อมูลรายละเอียดของข้อมูลบนเซิร์ฟเวอร์ก็จะทำการแสดงผลข้อมูลบนเว็บเบราว์เซอร์ (Web Browser)

การทำงานของตัวเว็บเซิร์ฟเวอร์จะเป็นการให้บริการเนื้อหาของข้อมูลการดำเนินการอื่นของ HTTP ยังมีการรับข้อมูลจากผู้ใช้งานด้วยเช่น คำตอบจากแบบฟอร์ม, การอัปโหลดไฟล์, ฯลฯ.

เว็บเซิร์ฟเวอร์ส่วนมากจะมีการสนับสนุนการใช้งานสคริปต์ฝั่งเซิร์ฟเวอร์ เช่น Active Server Pages(ASP), PHP (Hypertext Preprocessor) และ สคริปต์ภาษาอื่นอีกในการใช้งาน[3]

2.2.2 RESTful Web service

RESTful เป็นเว็บเซอร์วิสประเภทหนึ่งที่สร้างมาจากการออกแบบโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่ชื่อว่า REST (Representational State Transfer) ซึ่งตัว RESTful ทำหน้าที่ให้บริการในการทำงานร่วมกันของคอมพิวเตอร์บนอินเทอร์เน็ต โดยที่ RESTful จะเป็นตัวแทนในการเข้าถึงและจัดการทรัพยากรบนเว็บ ในการดำเนินงานรูปแบบ Stateless

ทรัพยากรบนเว็บ เป็นได้ทั้งเอกสารและไฟล์ต่างๆ ที่ใช้ข้อมูลที่อยู่ของเว็บหรือ URL เมื่อตัว RESTful มีคำขอเข้าถึงทรัพยากรกับ URI จะทำให้เกิดการตอบสนองกับข้อมูลที่มีการจัดรูปแบบแล้ว เช่น HTML, JSON, XML หรือรูปแบบอื่นๆ และยังสามารถยืนยันการเปลี่ยนแปลงของทรัพยากร ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การดำเนินการในการทำงานของ RESTful นั้นจะมีคำสั่งใช้ในการดำเนินการ คือ GET, POST, PUT, DELETE และ คำสั่งอื่นๆ [4] [5]

HTTP Request

การส่งค่าข้อมูลบน HTTP มีรูปแบบในการส่งเป็นดังรูปที่ 2.5

<Verb> POST	<Path> /request.php	<HTTP VERSION> HTTP/1.1
<Request Header> Host : https://www.backyardfarm.in.th/Content-Type: text/xml; charset=utf-8Content-Length: 123		
<Request Body> <?xml version="1.0" encoding="utf-8"?> <Player> <ID>1</ID> <Name>Bright</Name> <Gold>5000</Gold> <Cash>5000</Cash> <Position_x>141.0215</Position_x> <Position_y>15.0215</Position_y> <Position_z>12.0212</Position_z> </Player>		

รูปที่ 2.5 HTTP Request Format

โดยในรูปแบบจะแบ่งเป็นแต่ละส่วนมีดังนี้

<VERB> เป็น HTTP verbs ประกอบไปด้วย GET, POST, PUT, DELETE, OPTIONS, ฯลฯ

<PATH> เป็น ลิงค์หรือที่อยู่ของข้อมูลที่ใช้ในการประมวลผล

<HTTP Version> เป็นเวอร์ชันของ HTTP, โดยทั่วไปจะเป็น "http v1.1"

<Request Header> เป็นส่วนที่ไว้ระบุการจัดรูปแบบของ Request Body, เก็บการตั้งค่า

การตอบกลับข้อมูล

HTTP Response

การตอบกลับข้อมูลบน HTTP มีรูปแบบในการตอบกลับเป็นดังรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<p><HTTP VERSION> HTTP/1.1</p>	<p><Response Code> 200 OK</p>
<p><Response Header> Date: Sat, 24 Sep 2018 12:28:53 GMT Server: Apache/2.2.14 Last-Modified: Fri, 23 Sep 2018 19:15:56 GMT Content-Length: 88 Content-Type: text/xml Connection: Closed</p>	
<p><Response Body> <?xml version="1.0" encoding="utf-8"?> <Item> <ID>1</ID> <Name>Shovel</Name> <Type>Tools</Type> </Item></p>	

รูปที่ 2.6 HTTP Response Format

โดยในรูปแบบจะแบ่งเป็นแต่ละส่วนมีดังนี้

<HTTP Version> เป็นเวอร์ชันของ HTTP, โดยทั่วไปจะเป็น "http v1.1"

<Response Code> เป็นเลขที่ใช้บอกความพยายามที่จะเข้าใจและตอบกลับคำขอเป็นเลขรหัสสามหลักโดยหลักๆจะแบ่งตามหลักแรกเป็นดังนี้

1xx เป็นข่าวสาร เช่น ได้รับคำขอแล้วและกำลังอยู่ระหว่างการทำงาน

2xx เป็นผลสำเร็จ เช่น ได้รับคำขอเสร็จแล้ว, เข้าใจและยอมรับเรียบร้อยแล้ว

3xx เป็นการเปลี่ยนทิศทาง เช่น ต้องดำเนินการเพิ่มเติมเพื่อให้คำขอสมบูรณ์

4xx เป็นข้อผิดพลาดของผู้ใช้ เช่น คำขอมิไวยากรณ์ที่ไม่ถูกต้องหรือไม่สามารถทำได้[6]

<Response Header> เป็นส่วนของการจัดการ การตั้งค่าของข้อความของข้อมูลที่ตอบกลับมา

<Response Body> เป็น ส่วนการแสดงผล ถ้าคำขอสำเร็จ[7]

ทางกลุ่มการทำงานเบื้องหลังได้ใช้บล็อกอินของโปรแกรม Unreal Engine ในการใช้งาน เว็บเซิร์ฟเวอร์ RESTful หรือ REST ที่ชื่อว่า VaRest

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และออกแบบระบบฐานข้อมูล

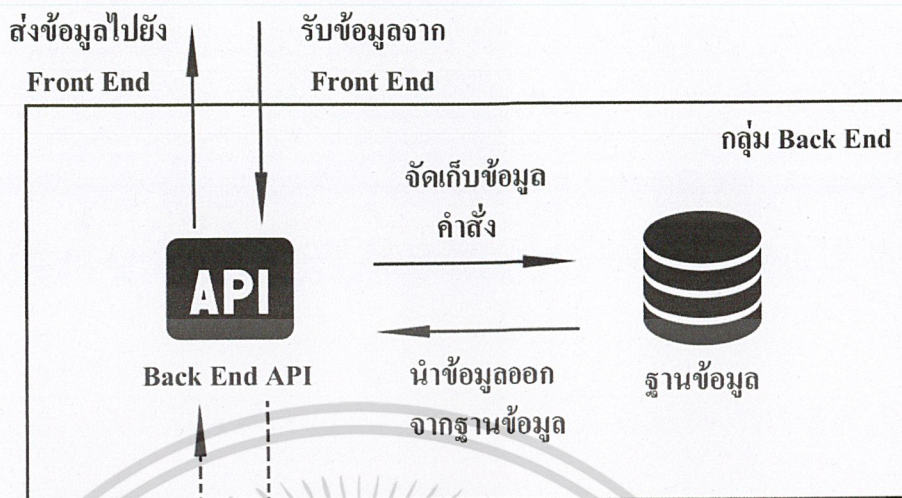
กลุ่มฟาร์มหลังบ้านได้มีการแบ่งกลุ่มงานออกเป็นสามกลุ่มย่อย ๆ ประกอบด้วย กลุ่มการทำงานเบื้องหน้า, กลุ่มการทำงานเบื้องหลัง และ กลุ่มออกแบบโดยทางกลุ่มการทำงานเบื้องหลัง จะมีการสร้างระบบฐานข้อมูลให้สามารถรองรับการทำงานตามที่กลุ่มออกแบบต้องการ และได้มีการสร้าง API ขึ้นมาเพื่อให้กลุ่มการทำงานเบื้องหน้า สามารถนำข้อมูลจากฐานข้อมูลไปใช้งาน, แก้ไขและบันทึกข้อมูลชุดใหม่ลงบนฐานข้อมูลได้ตามที่ต้องการได้อย่างสะดวก

3.1 สถาปัตยกรรมของระบบงานโดยรวม

ระบบการทำงานของโลกเสมือน จะมีการรับคำสั่งการใช้งานของผู้อาศัยภายในเมืองที่มีความสนใจในการลงทุนเพื่อเพาะปลูกส่งไปให้กับเกษตรกรจริงในชนบทได้ทำการเพาะปลูก โดยในการพัฒนาต่อในอนาคตจะมีส่วนของคำสั่งที่จะถูกเก็บค่าแล้วส่งไปให้กับเกษตรกรผ่าน GAP API ที่ใช้เพื่อให้เกษตรกรและผู้ใช้งานสามารถติดต่อรายงานผลความคืบหน้าเกี่ยวกับการเพาะปลูก โดย GAP API จะเป็นส่วนที่รับมาจากภายนอก ตัวอย่างชุดคำสั่งมีดังนี้ เช่น ขอข้อมูลอัปเดตของข้อมูลการเก็บเกี่ยวกับข้อมูลทางการเกษตร และผลสรุปการลงทุน ส่วนคำสั่งเกี่ยวกับพื้นที่ฟาร์มส่วนตัวในโลกเสมือนจะไม่ถูกส่งไปให้กับเกษตรกรแต่จะปรับใช้กับโลกเสมือนของผู้ใช้งานคนนั้นเท่านั้น

ทางกลุ่มการทำงานเบื้องหลังได้มีการออกแบบและสร้าง API ให้กับกลุ่มทำงานเบื้องหน้าให้สามารถเรียกใช้งานข้อมูลที่ถูกจัดเก็บจากฐานข้อมูลและสามารถบันทึกข้อมูลชุดใหม่เข้าไปยังฐานข้อมูล เช่น ข้อมูลใช้งาน, ตัวละครผู้เล่น, ช่องเก็บของ, ผลผลิตทางการเกษตร, ตัวฟาร์ม, สถานที่ต่างๆ, ชุดคำสั่งให้แก่เกษตรกร, การเรียกพิกัดตำแหน่งของวัตถุที่ถูกบันทึกตำแหน่งเอาไว้ในฐานข้อมูล และการบันทึกพิกัดตำแหน่งของวัตถุชิ้นใหม่ที่ผู้ใช้งานได้ทำการวางเพิ่มเข้ามาในตัวโลกเสมือนบนเซิร์ฟเวอร์ เป็นต้น แสดงได้ดังรูป 3.1

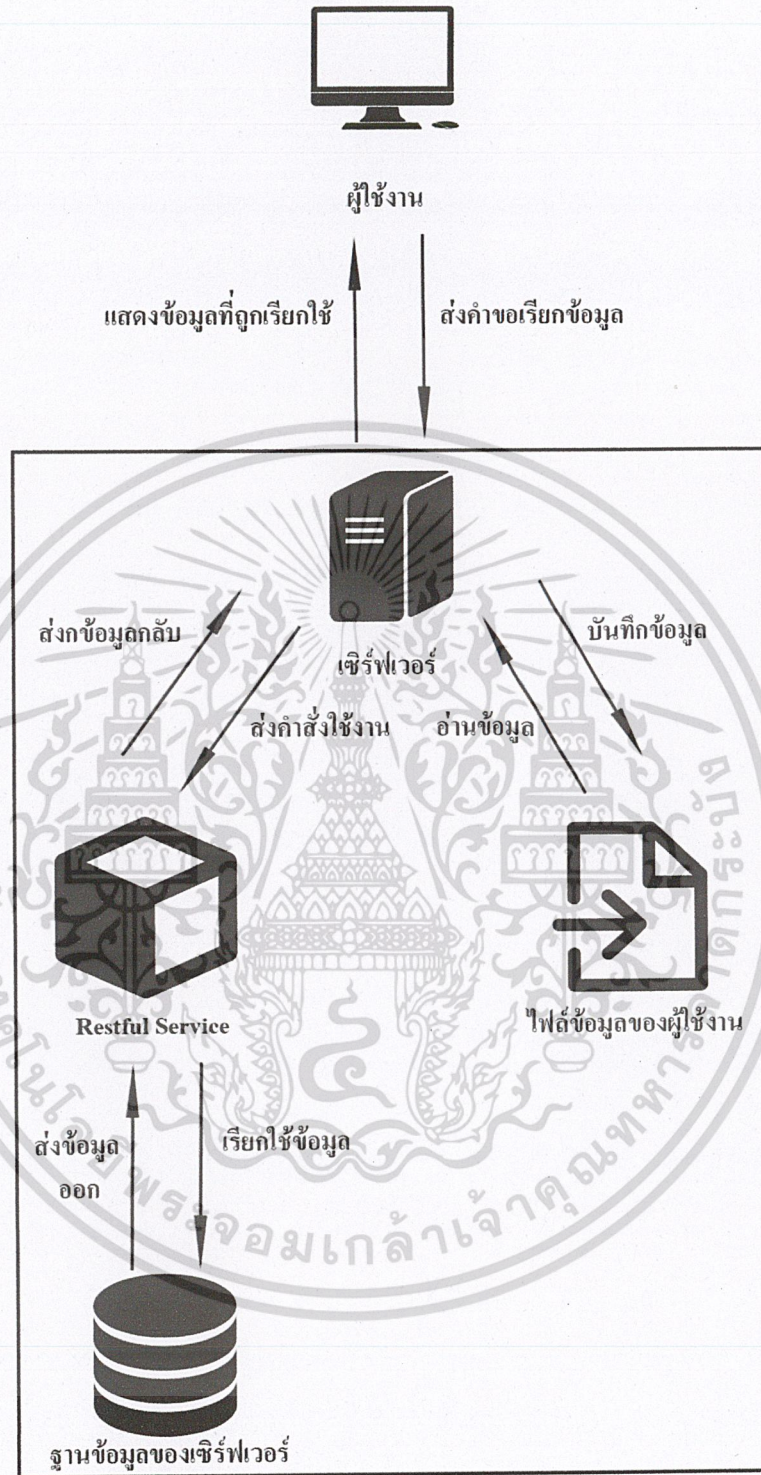
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.1 สถาปัตยกรรมการทำงานของระบบโดยรวม

ระหว่างผู้ใช้งานได้ทำการใช้งานระบบ ภายในระบบนั้นจะมีการติดต่อกับทางเซิร์ฟเวอร์โดย การส่งคำขอเรียกข้อมูลในการใช้แสดงผลหรือเพื่อการที่จะบันทึกข้อมูล โดยคำสั่งจะถูกส่งให้กับทาง เซิร์ฟเวอร์และเซิร์ฟเวอร์จะมีการอ่านหรือบันทึกข้อมูลในไฟล์ของผู้ใช้งาน (Save File) หรือทำการ ติดต่อกับทางเว็บเซิร์ฟเวอร์โดยผ่านระบบเซอร์วิสที่ชื่อ Restful เพื่อสื่อสารกันระหว่างเซิร์ฟเวอร์ของ ระบบ และ เว็บเซิร์ฟเวอร์ เมื่อทางเว็บเซิร์ฟเวอร์ได้รับข้อมูลมาแล้วจะทำการติดต่อกับเซิร์ฟเวอร์ของ ฐานข้อมูลเพื่อขอข้อมูลที่ต้องการและทำการส่งข้อมูลนั้นกลับไปยังเซิร์ฟเวอร์ของระบบ และเมื่อทาง เซิร์ฟเวอร์ของระบบได้รับข้อมูลที่ทางระบบของผู้ใช้งานต้องการก็จะทำการส่งข้อมูลกลับไปยัง ผู้ใช้งานเพื่อแสดงผล แสดงได้ดังรูป 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



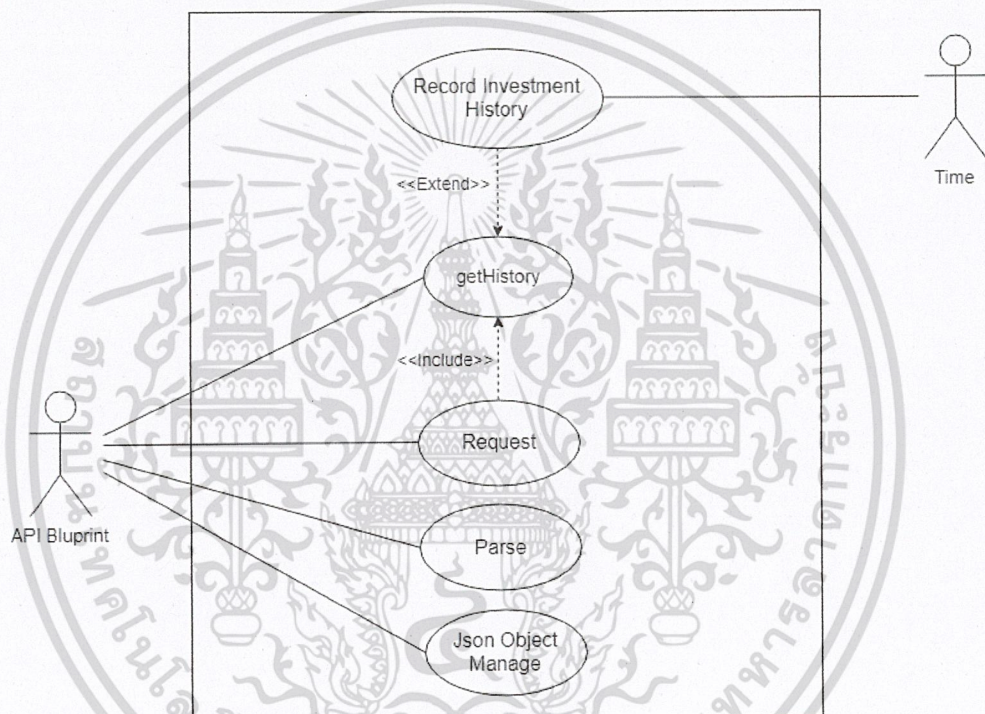
รูป 3.2 สถาปัตยกรรมของเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การวิเคราะห์และออกแบบฐานข้อมูล

Use Case Diagram

ในการออกแบบและสร้าง API ทางกลุ่มการทำงานเบื้องหลังได้มีการแบ่งการทำงานของ API ออกเป็น 3 ประเภทได้แก่ Request Parse และ Json Object Manage เพื่อจัดการกับข้อมูลจากฐานข้อมูล สามารถอธิบายได้ดังรูป 3.3



รูปที่ 3.3 Main Use Case Diagram

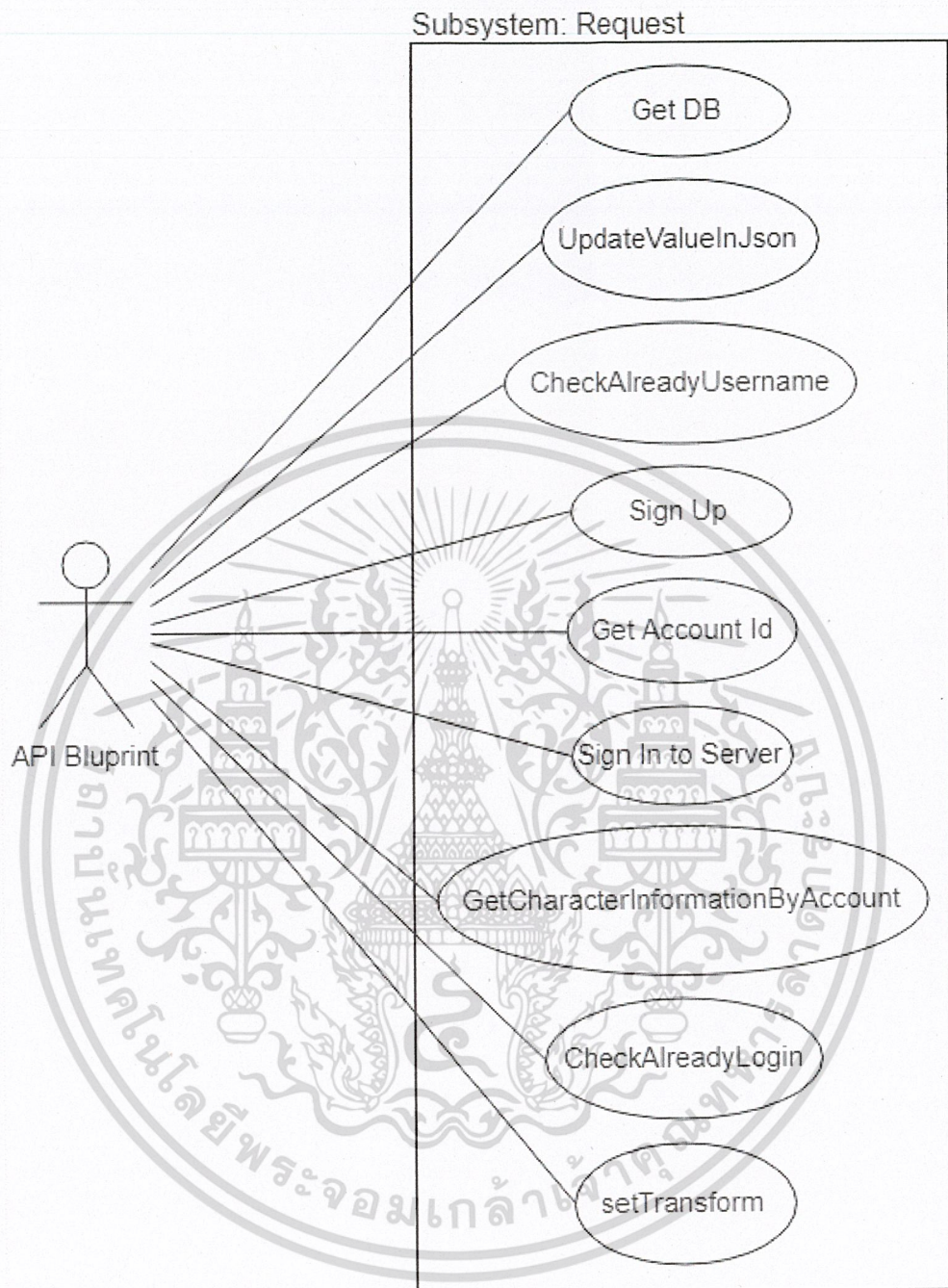
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.1 Use case description “Record Investment History”

Use Case Name	Record Investment History
Actor	Time
Purpose	เพื่อบันทึกข้อมูลประวัติการลงทุน
Overview	ระบบจัดเก็บข้อมูลรายงานประวัติการลงทุน
Preconditions	ระบบเรียกใช้ getHistory API
Postconditions	ระบบได้จัดเก็บข้อมูลการลงทุน
Main Flow	<ol style="list-style-type: none"> 1. ระบบส่งคำขอเรียกใช้ getHistory API 2. ระบบทำการบันทึกรายงานประวัติการลงทุน

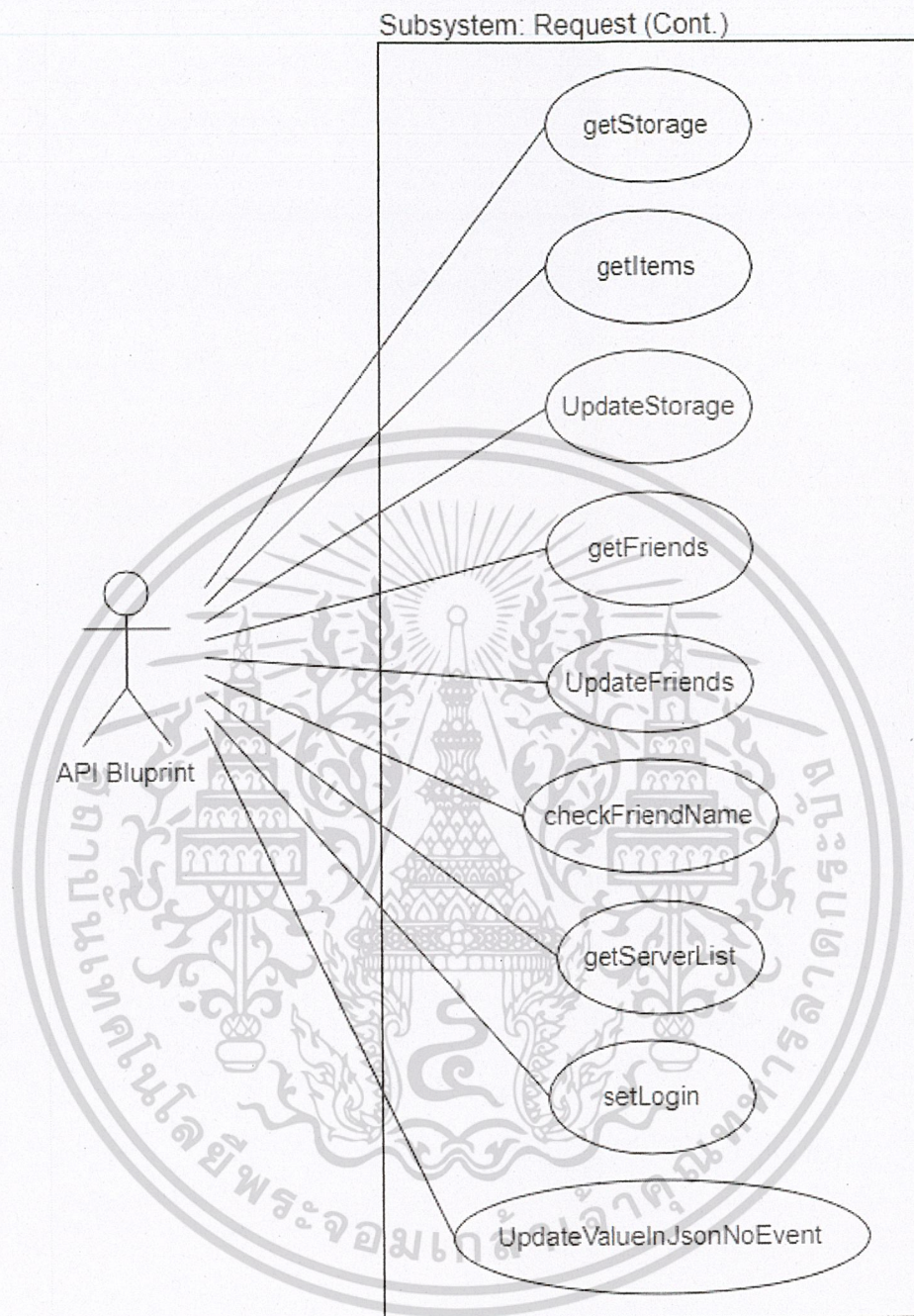
รายละเอียด Subsystem ของ Request Parse และ Json Object Manage สามารถอธิบายได้ดังรูปที่ 3.4 ถึง 3.10 รายละเอียดและข้อมูลเพิ่มเติมทั้งหมดของ API อธิบายอยู่ในภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



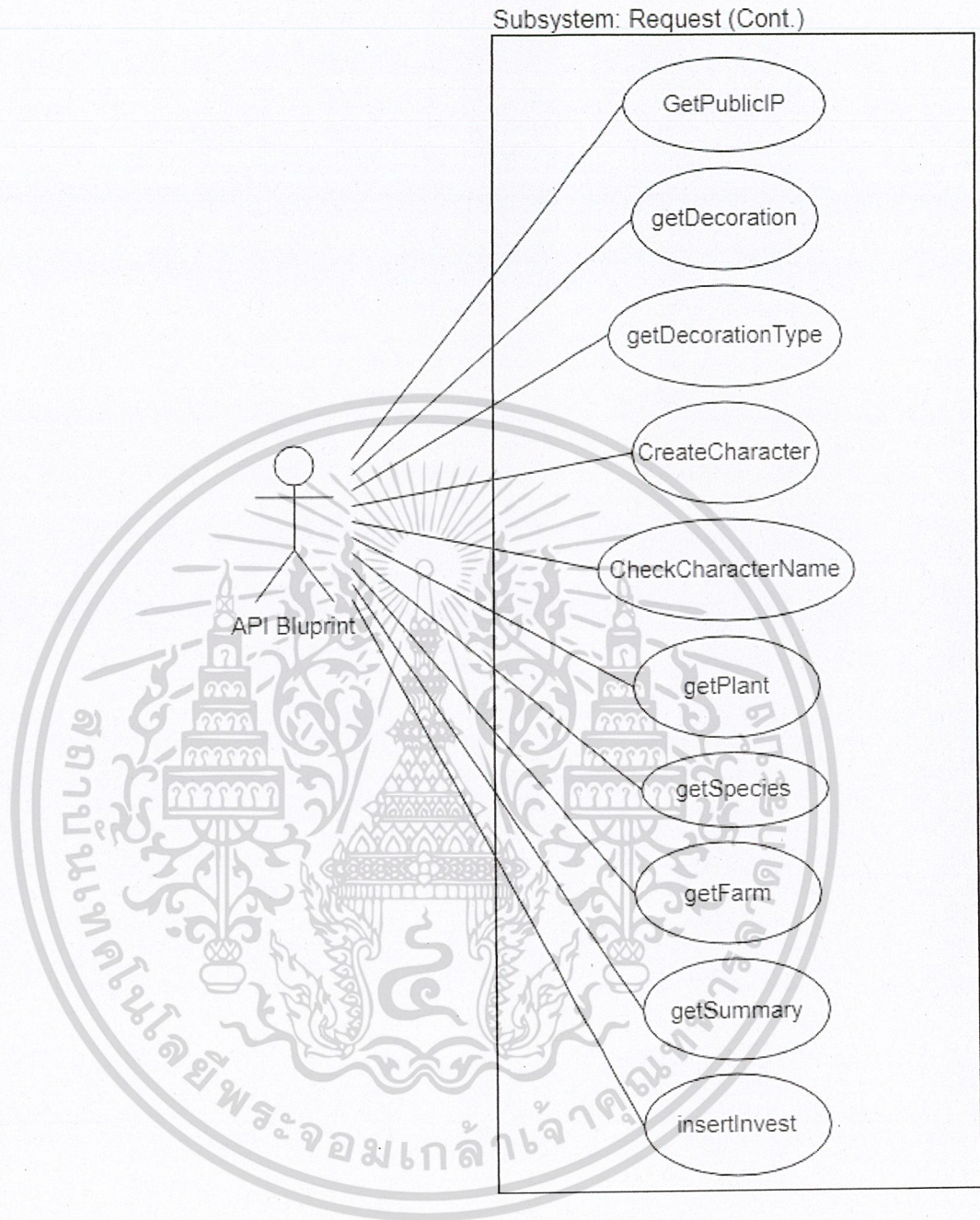
รูปที่ 3.4 Subsystem Use Case : Request 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



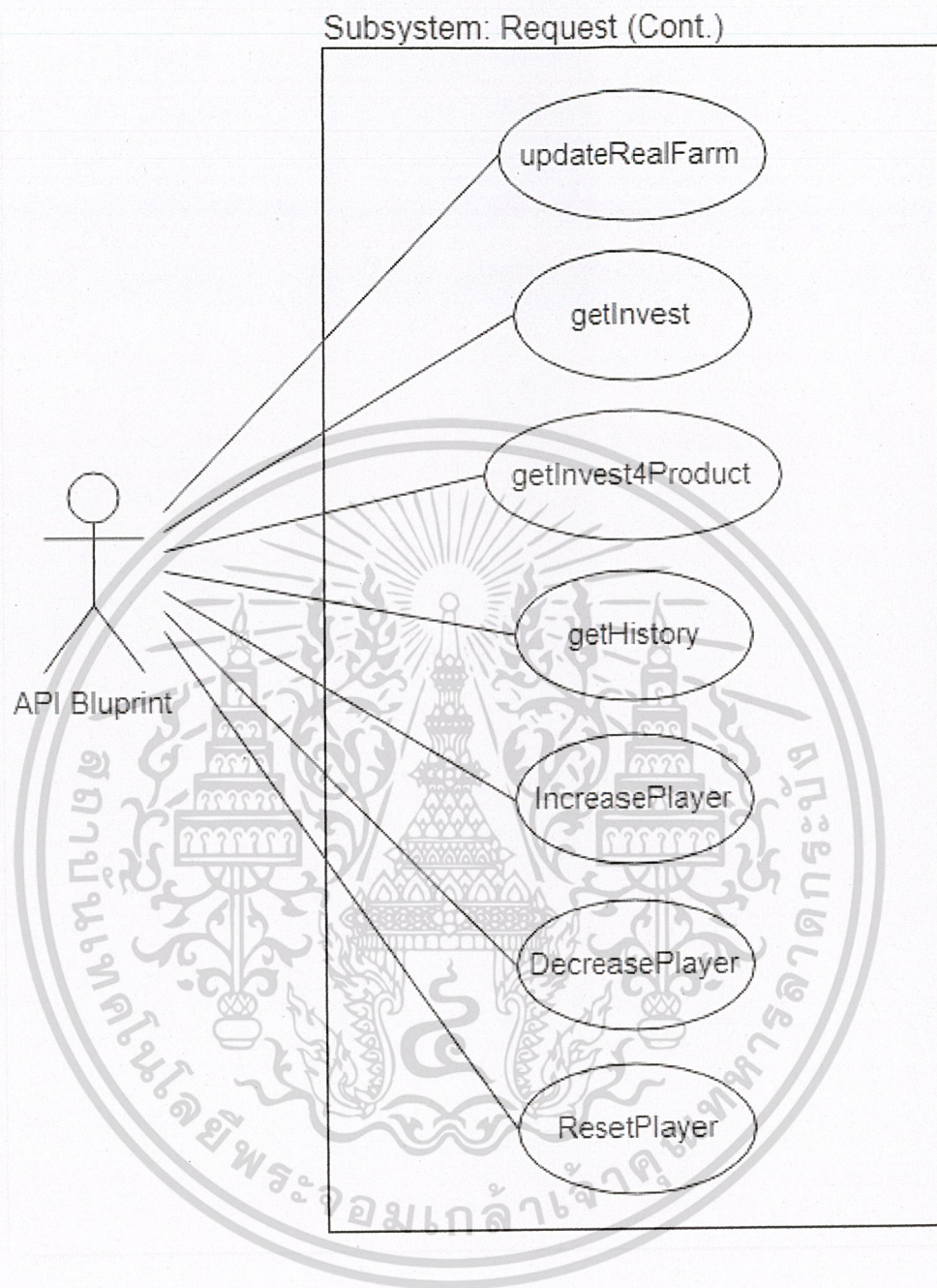
รูปที่ 3.5 Subsystem Use Case : Request 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



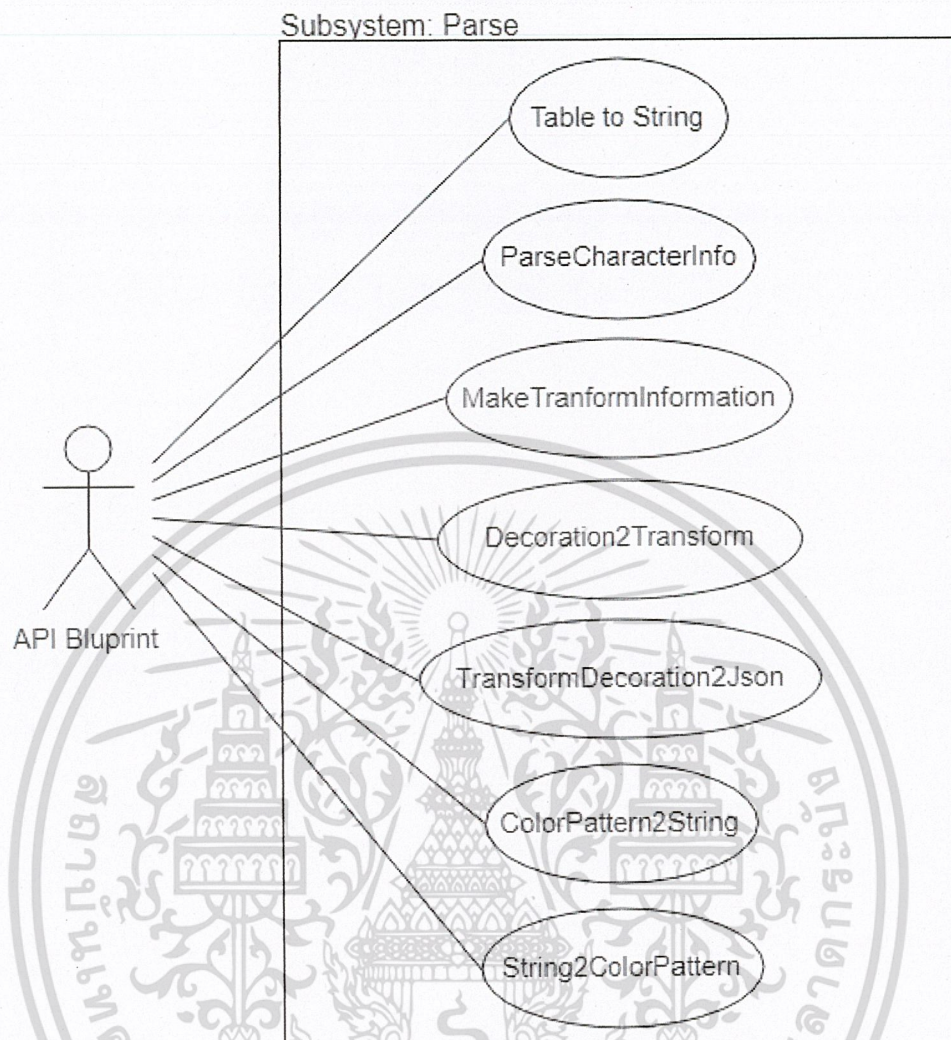
รูปที่ 3.6 Subsystem Use Case : Request 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



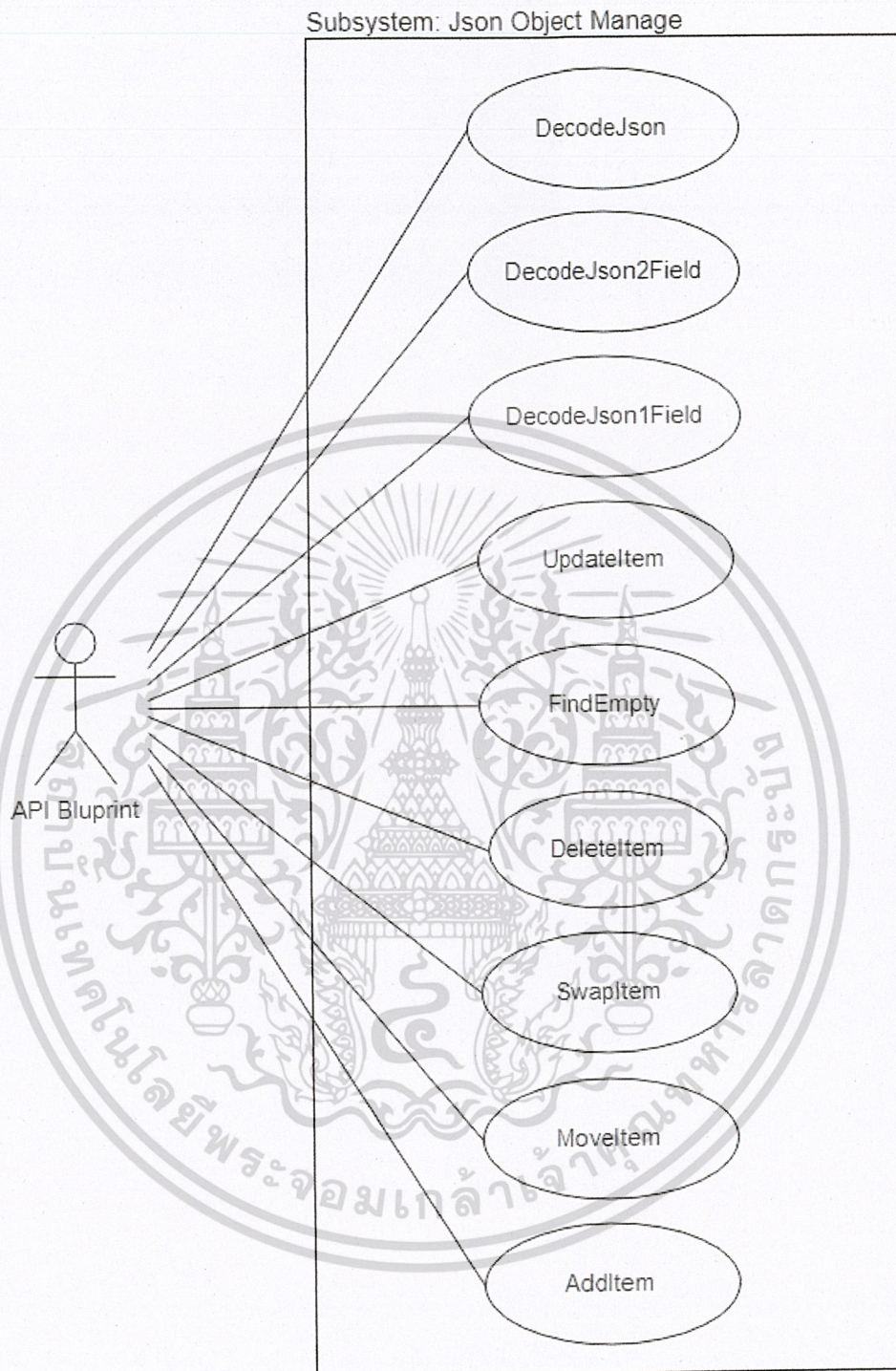
รูปที่ 3.7 Subsystem Use Case : Request 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



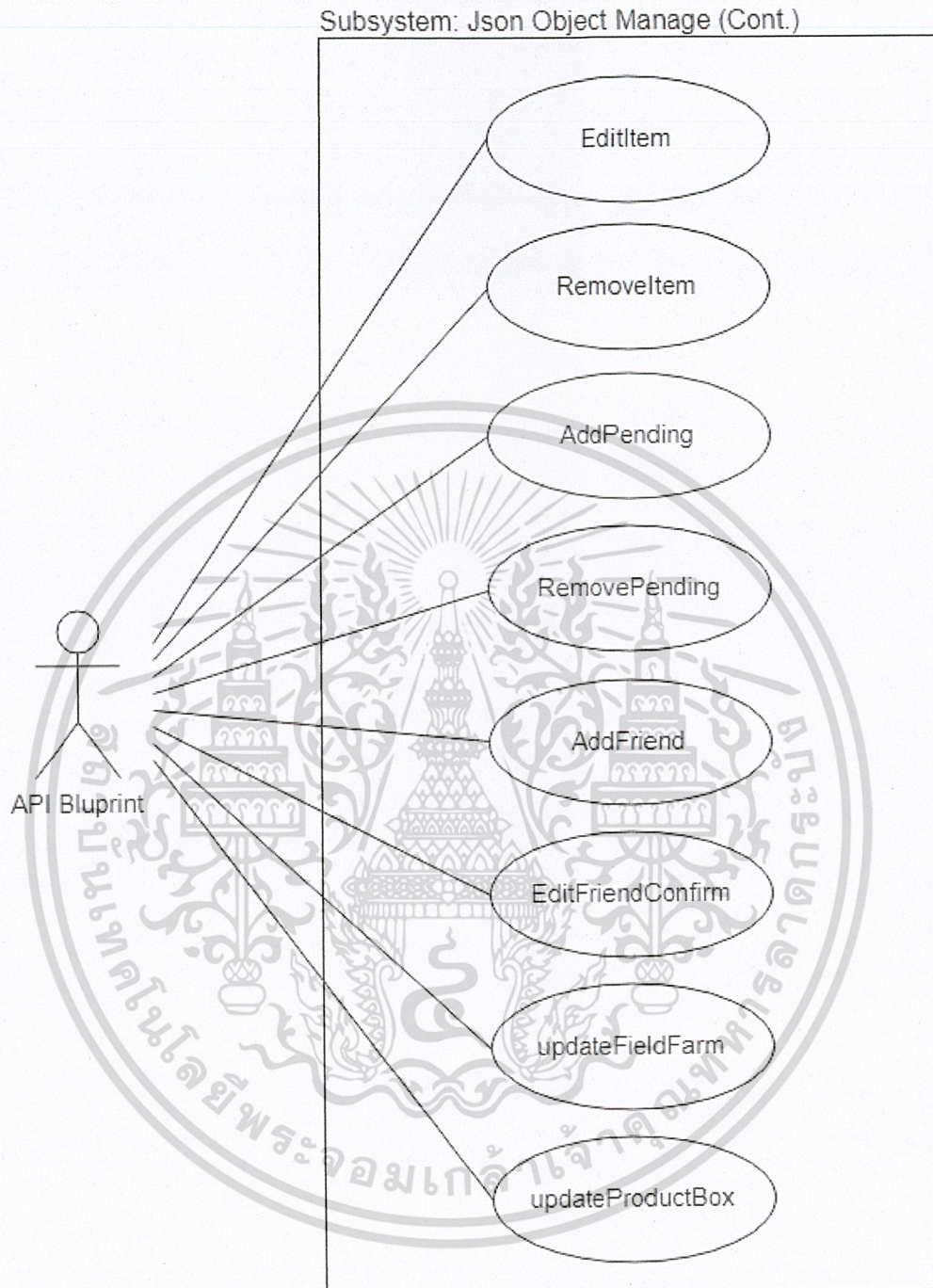
รูปที่ 3.8 Subsystem Use Case : Parse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 Subsystem Use Case : Json Object Manage 1

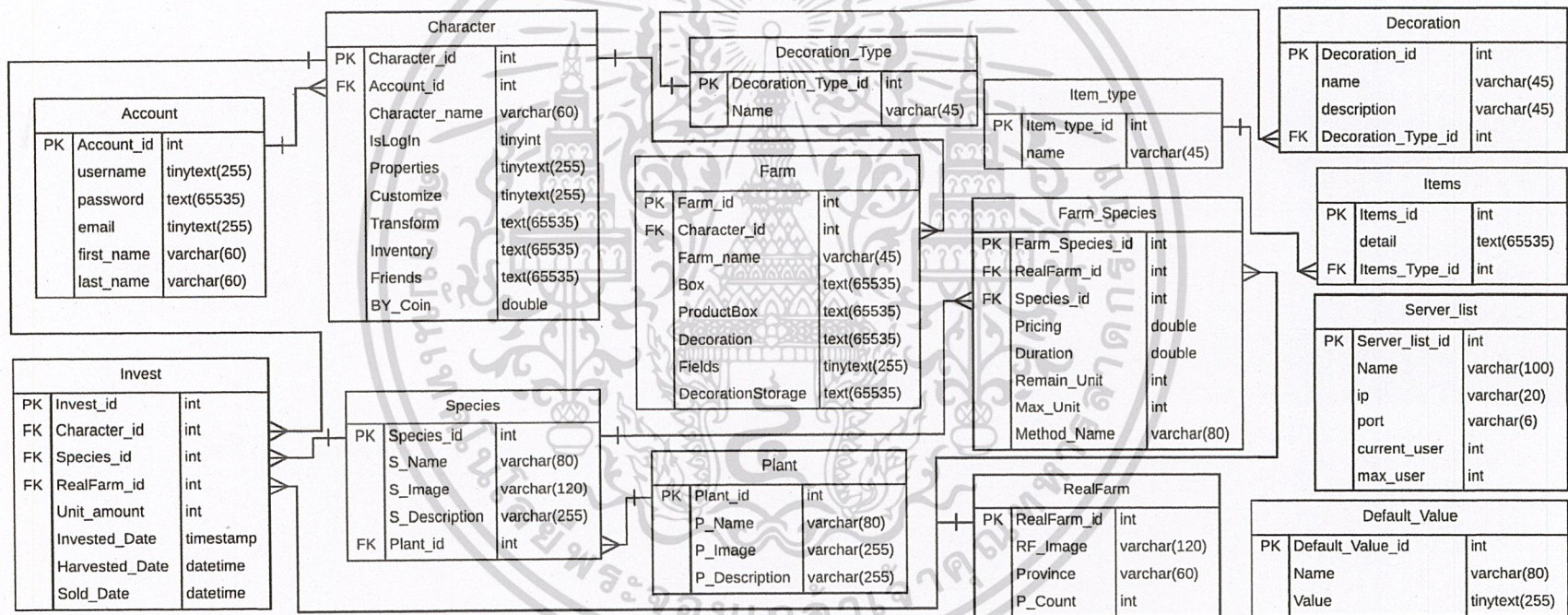
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Subsystem Use Case : Json Object Manage 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ER-Diagram



รูปที่ 3.11 ER-Diagram ของฐานข้อมูล

ตาราง Account ใช้สำหรับเก็บข้อมูลของผู้ใช้งานที่ได้ทำการสมัครสมาชิกเพื่อเข้าใช้งานระบบ
อธิบายได้ตามตาราง 3.2

ตาราง 3.2 โครงสร้างตาราง Account

No	Attribute	Datatype	Constraints	Description
01	account_id	int	NOT NULL	รหัสลำดับ ประจำตัวผู้ใช้งาน ในระบบของแต่ละ ผู้ใช้งาน
02	username	tinytext(255)	NOT NULL	ชื่อบัญชีผู้ใช้งาน
03	password	text(65535)	NOT NULL	รหัสผ่านของบัญชี ผู้ใช้งาน
04	email	tinytext(255)	NOT NULL	อีเมลของผู้ใช้งาน
05	first_name	varchar(60)	NOT NULL	ชื่อของผู้ใช้งาน
06	last_name	varchar(60)	NOT NULL	นามสกุลของ ผู้ใช้งาน

ตาราง Character ใช้สำหรับเก็บข้อมูลตัวละครของ ช่องสัมภาระ ไอเทม และตำแหน่งยืนล่าสุดของ
ผู้ใช้งานที่จะถูกนำมาเรียกใช้ในโลกเสมือน อธิบายได้ตามตาราง 3.3

ตาราง 3.3 โครงสร้างตาราง Character

No	Attribute	Datatype	Constraints	Description
01	Character_id	int	NOT NULL	รหัสลำดับ ประจำตัวของตัว ละคร
02	Account_id	int	NOT NULL	ใช้สำหรับอ้างอิง ว่าตัวละครตัวนี้มี บัญชีไหนเป็น เจ้าของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No	Attribute	Datatype	Constraints	Description
03	Character_name	varchar(60)	NOT NULL	ชื่อของตัวละคร ของผู้ใช้งาน
04	IsLogin	tinyint	NOT NULL	ใช้สำหรับ ตรวจสอบว่า ผู้ใช้งานเข้าสู่ ระบบอยู่หรือไม่
05	Properties	tinytext(255)	NOT NULL	ใช้สำหรับบันทึก ข้อความในช่อง About Me ของ ผู้ใช้งาน
06	Customize	tinytext(255)	NOT NULL	ใช้สำหรับบันทึก ค่าการตกแต่งตัว ละครของผู้ใช้งาน
07	Transform	text(65535)	NOT NULL	ใช้สำหรับบันทึก ค่าตำแหน่งยืน ล่าสุดของ ผู้ใช้งาน
08	Inventory	text(65535)	NOT NULL	ใช้สำหรับบันทึก ข้อมูลไอเทมที่ ผู้ใช้งาน ครอบครอง จำนวน และ ตำแหน่งภายใน ช่อง
09	Friends	text(65535)	NOT NULL	ใช้สำหรับบันทึก สถานะความเป็น เพื่อนของ ผู้ใช้งานแต่ละคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No	Attribute	Datatype	Constraints	Description
10	BY_Coin	double	NOT NULL	ใช้สำหรับเก็บสกุลเงินสำหรับใช้จ่ายภายในโลกเสมือน (ชั่วคราว)

ตาราง Decoration ใช้สำหรับเก็บข้อมูลสิ่งของตกแต่งฟาร์ม รวมถึงชื่อ และคำอธิบายของสิ่งของนั้น อธิบายได้ตามตาราง 3.4

ตาราง 3.4 โครงสร้างตาราง Decoration

No	Attribute	Datatype	Constraints	Description
01	Decoration_id	int	NOT NULL	รหัสลำดับของสิ่งของตกแต่ง
02	name	varchar(45)	NOT NULL	ชื่อของสิ่งของตกแต่ง
03	description	Varchar(45)	NOT NULL	คำอธิบายของสิ่งของตกแต่ง
04	Decoration_Type_id	int	NOT NULL	รหัสสำหรับใช้อ้างอิงชนิดของสิ่งของตกแต่ง

ตาราง Decoration_Type ใช้สำหรับเก็บชนิดของสิ่งของตกแต่งเพื่ออ้างอิงการเรียกใช้งานภายในโลกเสมือน อธิบายได้ตามตาราง 3.5

ตาราง 3.5 โครงสร้างตาราง Decoration_Type

No	Attribute	Datatype	Constraints	Description
01	Decoration_Type_id	int	NOT NULL	รหัสชนิดของสิ่งของตกแต่ง
02	name	Varchar(45)	NOT NULL	ชื่อชนิดสิ่งของตกแต่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง Default_Value ใช้สำหรับเก็บค่าเริ่มต้นสำหรับข้อมูลในตารางเมื่อเรียกใช้งานครั้งแรกและ
ยังไม่มีข้อมูล อธิบายได้ตามตาราง 3.6

ตาราง 3.6 โครงสร้างตาราง Default_Value

No	Attribute	Datatype	Constraints	Description
01	Default_Value_id	int	NOT NULL	รหัสลำดับของ ชนิดค่าเริ่มต้น
02	Name	varchar(80)	NOT NULL	ชื่อของค่าเริ่มต้น
03	Value	varchar(255)	NOT NULL	ใช้สำหรับเก็บข้อ มูลค่าเริ่มต้น

ตาราง Farm ใช้สำหรับเก็บข้อมูลสิ่งของตกแต่งฟาร์มที่ผู้ใช้งานได้เป็นเจ้าของ อธิบายได้ตามตาราง
3.7

ตาราง 3.7 โครงสร้างตาราง Farm

No	Attribute	Datatype	Constraints	Description
01	Farm_id	int	NOT NULL	รหัสลำดับของ ฟาร์ม
02	Character_id	int	NOT NULL	ใช้สำหรับอ้างอิง เจ้าของฟาร์ม
03	Farm_name	Varchar(45)	NOT NULL	ชื่อของฟาร์มที่ใช้ แสดงในโลก เสมือน
04	Box	text(65535)	NOT NULL	กล่องเก็บของ ของผู้ใช้งานที่ ตั้งอยู่ภายใน ฟาร์ม
05	ProductBox	text(65535)	NOT NULL	กล่องเก็บผลผลิต ที่แสดงผลผลิตที่ พร้อมเก็บเกี่ยว จากการลงทุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No	Attribute	Datatype	Constrain	Description
06	Decoration	text(65535)	NOT NULL	บันทึกตำแหน่งของสิ่งของตกแต่งที่ถูกจัดวางภายในฟาร์ม
07	Fields	tinytext(255)	NOT NULL	ตำแหน่งของที่ดินที่ถูกจัดแสดงภายในฟาร์ม และแสดงถึงสถานะการลงทุน
08	DecorationStorage	text(65535)	Not NULL	ใช้สำหรับเก็บของตกแต่งที่ผู้ใช้งานมีอยู่ในครอบครอง

ตาราง Farm_Species ใช้สำหรับเก็บข้อมูลของฟาร์มจริงที่รับเพาะปลูกพืชทางการเกษตรว่าเป็นพันธุ์อะไร มีวิธีการเพาะปลูกอย่างไร และราคาค่าใช้จ่าย อธิบายได้ตามตาราง 3.8

ตาราง 3.8 โครงสร้างตาราง Farm_Species

No	Attribute	Datatype	Constrains	Description
01	Farm_Species_id	int	NOT NULL	รหัสลำดับของสายพันธุ์ที่ฟาร์มรับปลูก
02	RealFarm_id	int	NOT NULL	รหัสของฟาร์มจริง
03	Species_id	varchar(255)	NOT NULL	รหัสของสายพันธุ์
04	Pricing	double	NOT NULL	ราคาค่าใช้จ่าย
05	Duration	double	NOT NULL	จำนวนวันเพาะปลูกที่คาดการณ์ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No	Attribute	Datatype	Constraints	Description
06	Remain_unit	int	NOT NULL	จำนวนคงเหลือ ที่สามารถลงทุน ได้
07	Max_Unit	int	NOT NULL	จำนวนสูงสุดที่ สามารถลงทุนได้
08	Method_Name	varchar(80)	NOT NULL	ชื่อวิธีการ เพาะปลูกที่ใช้ สำหรับสายพันธุ์ นี้

ตาราง Invest ใช้สำหรับเก็บข้อมูลการลงทุนที่ผู้ใช้งานโลกเสมือนได้ทำการลงทุนไป อธิบายได้ตาม
ตาราง 3.9

ตาราง 3.9 โครงสร้างตาราง Invest

No	Attribute	Datatype	Constraints	Description
01	Invest_id	int	NOT NULL	รหัสลำดับการ ลงทุนที่ผู้ใช้งาน ได้ลงทุนไป
02	Character_id	int	NOT NULL	รหัสลำดับของตัว ละครที่ได้ลงทุน ไป
03	Species_id	int	NOT NULL	รหัสสายพันธุ์พืช ทางการเกษตร
04	RealFarm_id	int	NOT NULL	รหัสของฟาร์ม จริง
05	Unit_amount	int	NOT NULL	จำนวนที่ได้ลงทุน
06	Invested_Date	timestamp	NOT NULL	วันที่ลงทุน
07	Harvested_Date	datetime	NOT NULL	วันที่เก็บเกี่ยว ผลผลิต
08	Sold_Date	datetime	NOT NULL	วันที่ขายผลผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง Items ใช้สำหรับเก็บข้อมูลและคำอธิบายของสิ่งของภายในโลกเสมือน อธิบายได้ตามตาราง

3.10

ตาราง 3.10 โครงสร้างตาราง Items

No	Attribute	Datatype	Constraints	Description
01	Items_id	int	NOT NULL	รหัสของสิ่งของ
02	detail	text(65535)	NOT NULL	คำอธิบายของ สิ่งของ
03	Items_Type_id	int	NOT NULL	รหัสชนิดของ สิ่งของ

ตาราง Item_type ใช้สำหรับเก็บข้อมูลชนิดของสิ่งของภายในโลกเสมือน อธิบายได้ตามตาราง

3.11

ตาราง 3.11 โครงสร้างตาราง Item_type

No	Attribute	Datatype	Constraints	Description
01	Item_type_id	int	NOT NULL	รหัสชนิดของ สิ่งของ
02	name	varchar(45)	NOT NULL	ชื่อชนิดของ สิ่งของ

ตาราง Plant ใช้สำหรับเก็บข้อมูลและรูปของพืชทางการเกษตรที่สามารถใช้ลงทุนได้ อธิบายได้ตาม

ตาราง 3.12

ตาราง 3.12 โครงสร้างตาราง Plant

No	Attribute	Datatype	Constraints	Description
01	Plant_id	int	NOT NULL	รหัสของพืชทาง การเกษตร
02	P_Name	varchar(80)	NOT NULL	ชื่อของพืชทาง การเกษตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No	Attribute	Datatype	Constraints	Description
03	P_Image	varchar(255)	NOT NULL	รูปภาพของพืช ทางการเกษตร
04	P_description	varchar(255)	NOT NULL	คำอธิบายของพืช ทางการเกษตร

ตาราง RealFarm ใช้สำหรับเก็บข้อมูล และรายละเอียดของฟาร์มจริงเพื่อนำมาแสดงในหน้าต่าง
ลงทุน อธิบายได้ตามตาราง 3.13

ตาราง 3.13 โครงสร้างตาราง RealFarm

No	Attribute	Datatype	Constraints	Description
01	RealFarm_id	int	NOT NULL	รหัสลำดับของ ฟาร์มจริง
02	RF_Image	varchar(120)	NOT NULL	รูปภาพของฟาร์ม จริงที่ใช้แสดง
03	Province	varchar(60)	NOT NULL	จังหวัดที่ตั้งของ ฟาร์มจริง
04	P_Count	int	NOT NULL	ลำดับของฟาร์มที่ อยู่ในจังหวัด เดียวกัน

ตาราง Server_list ใช้สำหรับเก็บข้อมูลและวิธีการเข้าใช้เซิร์ฟเวอร์ในหน้าเลือกเซิร์ฟเวอร์ อธิบายได้
ตามตาราง 3.14

ตาราง 3.14 โครงสร้างตาราง Server_list

No	Attribute	Datatype	Constraints	Description
01	Server_list_id	int	NOT NULL	รหัสลำดับของ เซิร์ฟเวอร์
02	Name	varchar(100)	NOT NULL	ชื่อของเซิร์ฟเวอร์
03	ip	varchar(20)	NOT NULL	ip ที่ใช้ติดต่อเข้า เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No	Attribute	Datatype	Constraints	Description
04	port	varchar(6)	NOT NULL	พอร์ตที่ใช้ติดต่อ กับเซิร์ฟเวอร์
05	current_user	int	NOT NULL	จำนวนผู้เข้าใช้ งานปัจจุบัน
06	max_user	int	NOT NULL	จำนวนผู้ใช้งาน สูงสุด

ตาราง Species ใช้สำหรับเก็บข้อมูลชนิดของสายพันธุ์พืชทางการเกษตรสำหรับเรียกใช้ในหน้าต่าง
ลงทุนภายในโลกเสมือนอธิบายได้ตามตาราง 3.15

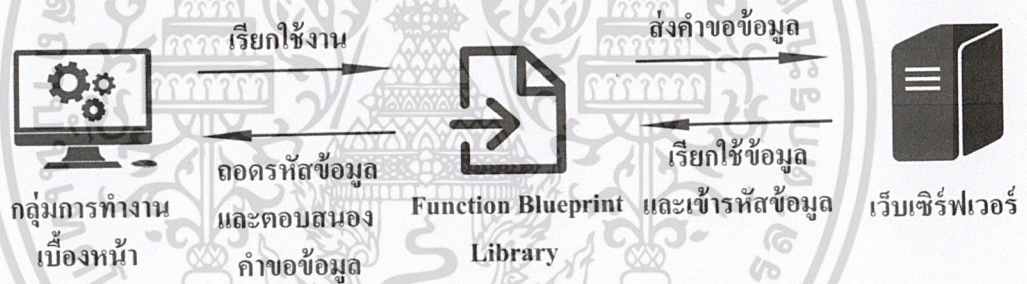
ตาราง 3.15 โครงสร้างตาราง Species

No	Attribute	Datatype	Constraints	Description
01	Species_id	int	NOT NULL	รหัสของสายพันธุ์ พืชทางการ เกษตร
02	S_Name	varchar(80)	NOT NULL	ชื่อสายพันธุ์
03	S_Image	varchar(120)	NOT NULL	รูปภาพของพืช สายพันธุ์นี้
04	S_Description	varchar(255)	NOT NULL	คำอธิบายของพืช สายพันธุ์นี้
05	Plant_id	int	NOT NULL	รหัสของพืชทาง การเกษตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

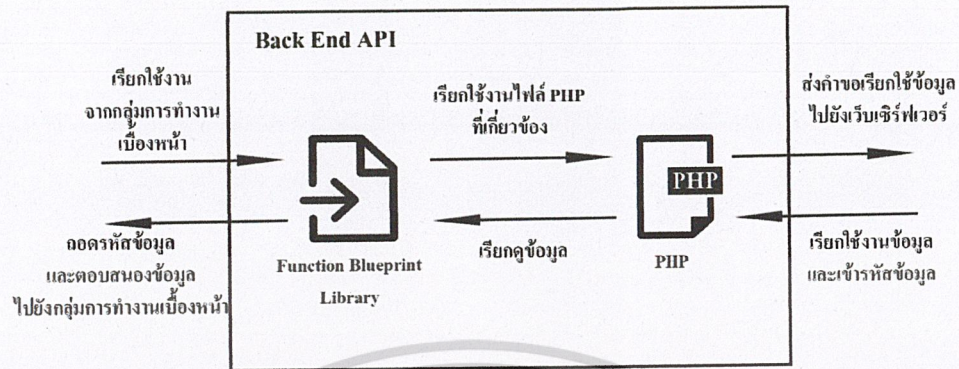
3.3 API

ทางกลุ่มการทำงานเบื้องหลังได้มีการสร้าง API สำหรับกลุ่มการทำงานเบื้องหน้าเพื่อที่จะสามารถเรียกใช้งานข้อมูลและแก้ไขข้อมูลจากฐานข้อมูลได้สะดวก ทำให้การใช้งานเป็นไปได้ง่ายขึ้น และลดเวลาการทำงานของระบบเมื่อเรียกใช้ข้อมูลจากเซิร์ฟเวอร์เพื่อนำมาเรียกใช้งานฟังก์ชันในโลกเสมือน โดยจะมีแยกเป็น 2 ส่วนคือ ส่วนของ PHP ที่เรียกใช้งานข้อมูลจากฐานข้อมูลแล้วทำการแปลงข้อมูล (Encode) ให้กลายเป็น JSON String เมื่อมีการเรียกใช้งานโดย Function Blueprint Library และส่วนของ Function Blueprint Library ที่ถูกสร้างเป็น Blueprint ภายใน Unreal Engine 4 เพื่อใช้เป็นตัวถอดรหัส (Decoder) สำหรับแปลง JSON String ที่รับมาจากส่วนของ PHP ให้กลายเป็น JSON Object เพื่อทำการเรียกใช้งานข้อมูลภายในโลกเสมือน อธิบายได้ดังรูปที่ 3.5 และ 3.6



รูป 3.12 สถาปัตยกรรมการทำงานของ Back End API โดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.13 สถาปัตยกรรมการทำงานของ API

โดย API ที่กลุ่มการทำงานเบื้องหลังได้สร้างขึ้นมีทั้งหมด 59 ตัวและมีการแบ่งการทำงานออกเป็น 3 ประเภทคือ

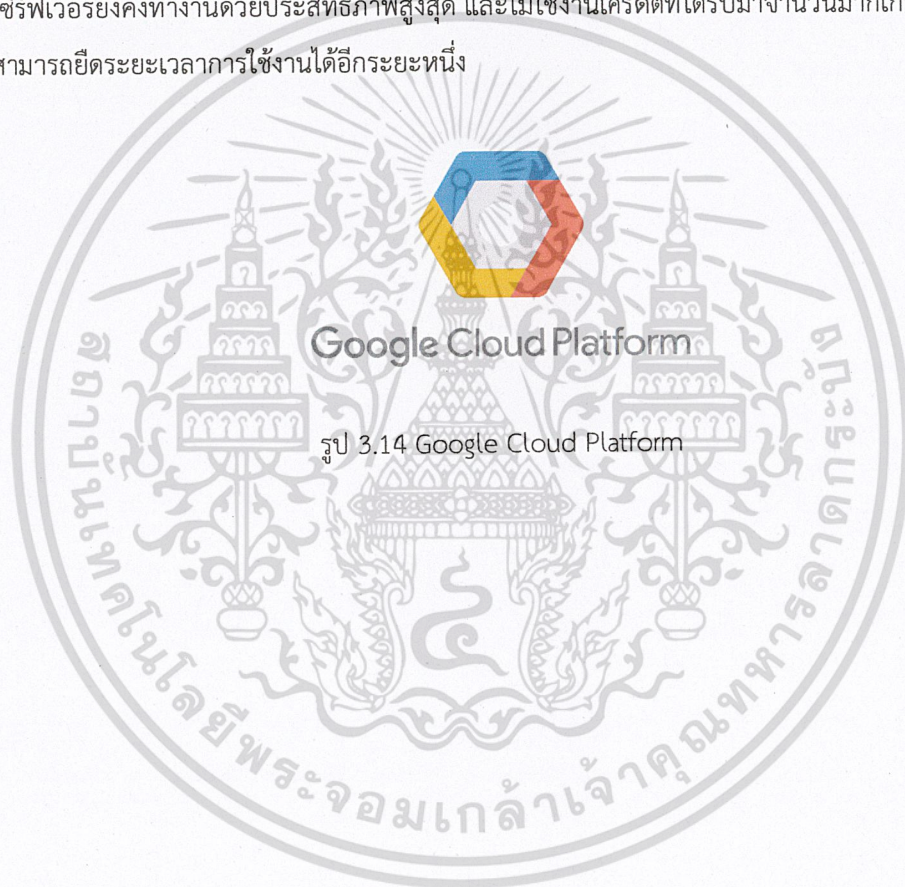
1. API กลุ่มประเภท Request จะมีการส่งคำขอเพื่อทำการขอข้อมูล เพิ่มข้อมูล ลบข้อมูล ตรวจสอบข้อมูล และแก้ไขข้อมูล เช่น Get DB, UpdateValueInJson, CheckAlreadyUsername
2. API กลุ่มประเภท Phrase จะมีการแปลงค่า รวมถึงการจัดการตัวแปรต่าง ๆ ก่อนที่จะนำข้อมูลไปใช้ต่อ เช่น Table to String, ParseCharacterInfo, MakeTranformInformation
3. API กลุ่มประเภท Json Object Manage จะมีการจัดการข้อมูลภายใน Json Object เช่น DecodeJson, UpdateItem, SwapItem

รายละเอียดและข้อมูลเพิ่มเติมทั้งหมดของ API อธิบายอยู่ในภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การเลือกใช้เซิร์ฟเวอร์ที่ใช้ในการโฮสต์ระบบ

ทางกลุ่มการทำงานเบื้องหลังได้เลือกใช้งาน Google Cloud Platform เป็นเซิร์ฟเวอร์ในการโฮสต์ระบบของโลกเสมือนสำหรับการทดลองการใช้งานเบื้องต้น เนื่องจากในด้านของ Google ได้มีการออกค่าใช้จ่ายในช่วงทดลองใช้เป็นจำนวน 300 เหรียญสหรัฐเครดิต และในด้านการใช้งานกับจำนวนผู้ใช้งานทางกลุ่มการทำงานเบื้องหลังได้คาดการณ์ไว้ว่า การเลือกทรัพยากรของเซิร์ฟเวอร์เป็น n1-standard-1 (1 vCPU, 3.75 GB memory) us-central1-a with 50 GB boot disk นั้นน่าจะเพียงพอสำหรับการทดสอบที่มีผู้เข้าใช้งานจำนวนหลายคนพร้อมกันเป็นจำนวน 20 คน โดยที่เซิร์ฟเวอร์ยังคงทำงานด้วยประสิทธิภาพสูงสุด และไม่ใช้งานเครดิตที่ได้รับมาจำนวนมากเกินไปทำให้สามารถยืดระยะเวลาการใช้งานได้อีกระยะหนึ่ง



รูป 3.14 Google Cloud Platform

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

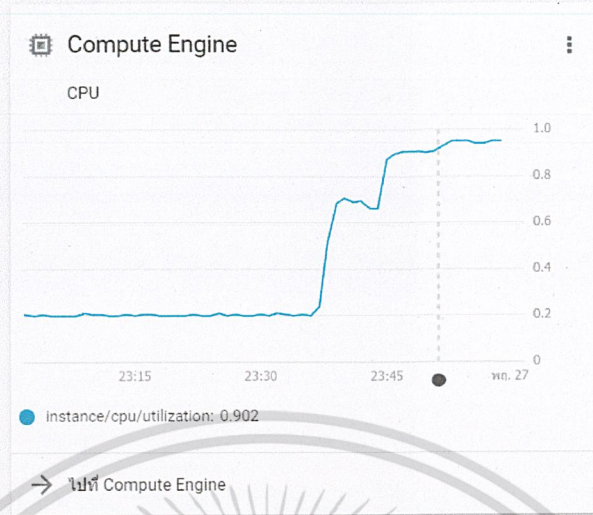
ผลการวิจัยและการอภิปรายผล

ทางทีมผู้พัฒนาได้นำระบบโลกเสมือนไปทดลองกับกลุ่มผู้ใช้งาน โดยมีการสังเกตการและเก็บข้อมูลการทำงานของเซิร์ฟเวอร์เมื่อมีผู้ใช้งานหลายคนเข้าใช้งานพร้อมกัน โดยมีผู้เข้าร่วมทดสอบพร้อมกันสูงสุด 17 คน จากผู้ทดสอบทั้งหมด 20 คน และได้มีการเก็บข้อมูลผลการใช้งาน API จากกลุ่มการทำงานเบื้องต้นที่ได้มีการเรียกใช้งาน API ที่กลุ่มการทำงานเบื้องต้นได้ทำการจัดเตรียมไว้ให้ระหว่างการพัฒนา

4.1 ผลการทดสอบการทำงานของระบบโลกเสมือน

ทางกลุ่มการทำงานเบื้องต้นได้เลือกใช้งาน Google Cloud Platform เป็นเซิร์ฟเวอร์สำหรับระบบการทำงานโลกเสมือนโดยเลือกทรัพยากรการใช้งานเป็น n1-standard-1 (1 vCPU, 3.75 GB memory) us-central1-a with 50 GB boot disk คือเซิร์ฟเวอร์ที่มี CPU จำนวน 1 คอร์แรม 3.75 GB ถูกโฮสต์ในอเมริกากลางโซน us-central1-a และพื้นที่ว่างบนฮาร์ดดิสก์จำนวน 50 GB โดยทางกลุ่มการทำงานเบื้องต้นได้มีการคาดการณ์ไว้ว่าเซิร์ฟเวอร์อาจสามารถรองรับผู้ใช้งานได้พร้อมกันสูงสุดถึง 20 คนโดยที่ยังคงมีประสิทธิภาพการทำงานสูงสุด แต่ผลที่ออกมาเมื่ออยู่ในช่วงเวลาที่ผู้ใช้งานได้เข้าสู่ระบบจำนวน 14 – 15 คน การทำงานของระบบนั้นมีประสิทธิภาพลดลงเล็กน้อยในส่วนของการสังเกตพบการเกิดอาการแล็ก (Lagging, การทำงานที่ล่าช้าและไม่สามารถแสดงผลได้ติดต่อกันกับผู้อื่น) ยกตัวอย่าง เช่น เมื่อผู้ใช้งานเดินจากจุดหนึ่งไปยังอีกจุดหนึ่งผู้ใช้งานคนอื่นจะเห็นผู้ใช้งานที่กำลังเดินอยู่หายไป และไปอยู่ที่อีกจุดหนึ่งในทิศทางที่ผู้ใช้งานคนนั้นกำลังเดินไป โดยปัญหานี้ทางกลุ่มการทำงานเบื้องต้นได้เห็นว่าทางเลือกใช้เซิร์ฟเวอร์ที่มี CPU จำนวน 1 คอร์นั้นมีข้อจำกัดอยู่ในเรื่องของประสิทธิภาพการทำงานสูงสุดในระบบเมื่อมีผู้ใช้งานพร้อมกันมากกว่า 15 คน ทางกลุ่มการทำงานเบื้องต้นมีวิธีการแก้ปัญหาได้ด้วยการเพิ่มคอร์ของ CPU เพื่อรองรับการใช้งานของผู้ใช้งานระบบที่มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 ภาพตัวอย่างช่วงเวลาการทำงานของ CPU จำนวน 1 คอร์ บนเซิร์ฟเวอร์ที่ติดตั้งระบบโลกเสมือน ระหว่างการทดสอบการใช้งานโลกเสมือนโดยผู้ใช้งานหลายคน



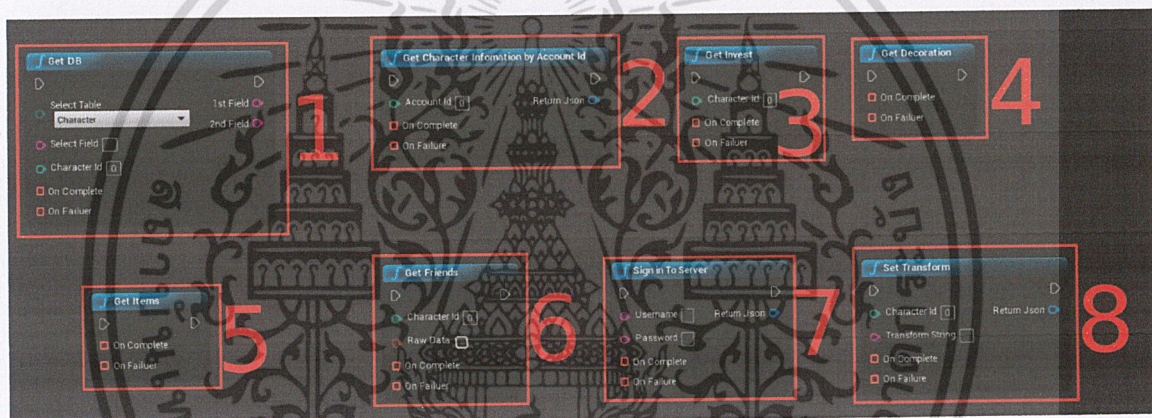
รูปที่ 4.2 ภาพตัวอย่างผู้เข้าร่วมทดสอบที่กำลังเข้าใช้งานระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการใช้งานของ API ที่มีต่อกลุ่มการทำงานเบื้องหน้า

จากการสังเกตและสอบถามการใช้งานจากกลุ่มการทำงานเบื้องหน้าที่ได้เรียกใช้งาน API ระหว่างการพัฒนาในระบบภายในโลกเสมือนได้ผลตอบกลับมาว่า API ได้ช่วยลดขั้นตอนการทำงาน และทำให้ใช้งานได้สะดวกขึ้นในส่วนของการเรียกใช้ฟังก์ชันข้อมูลชุดเดิม ทำให้ไม่ต้องสร้างฟังก์ชันบลูพริ้นท์ไลบรารี (Function Blueprint Library) เพื่อเรียกใช้งานข้อมูลเองทุกครั้งที่ต้องการเรียกใช้ข้อมูล

ตัวอย่าง API บางส่วนและการเรียกใช้งาน API ภายใน Unreal Engine 4



รูป 4.3 ตัวอย่างฟังก์ชันบลูพริ้นท์ไลบรารีภายใน Unreal Engine 4 สำหรับเรียกใช้งานข้อมูลในกลุ่มการทำงานเบื้องหลังได้สร้างขึ้น

จากรูปที่ 4.3 มีตัวอย่างฟังก์ชันบลูพริ้นท์ไลบรารีจำนวน 8 ตัวอย่างจากทั้งหมด 59 ไลบรารี ไลบรารีตามหลายเลขสามารถอธิบายได้ดังนี้

1. Library Get DB เป็นไลบรารีสำหรับเรียกใช้ข้อมูลในตารางในฐานข้อมูล โดยจำเป็นต้องระบุชื่อตาราง เช่น ตารางตัวละคร ตารางฟาร์ม
2. Library Get Character information by Account id เป็นไลบรารีสำหรับเรียกดูข้อมูลตัวละคร เช่น ข้อมูลการตกแต่งตัวละคร ข้อมูลตำแหน่งยืนล่าสุด ข้อมูลช่องสัมภาระ รายชื่อเพื่อน เป็นต้น โดยระบุ Account id (เลขรหัสลำดับของบัญชี)

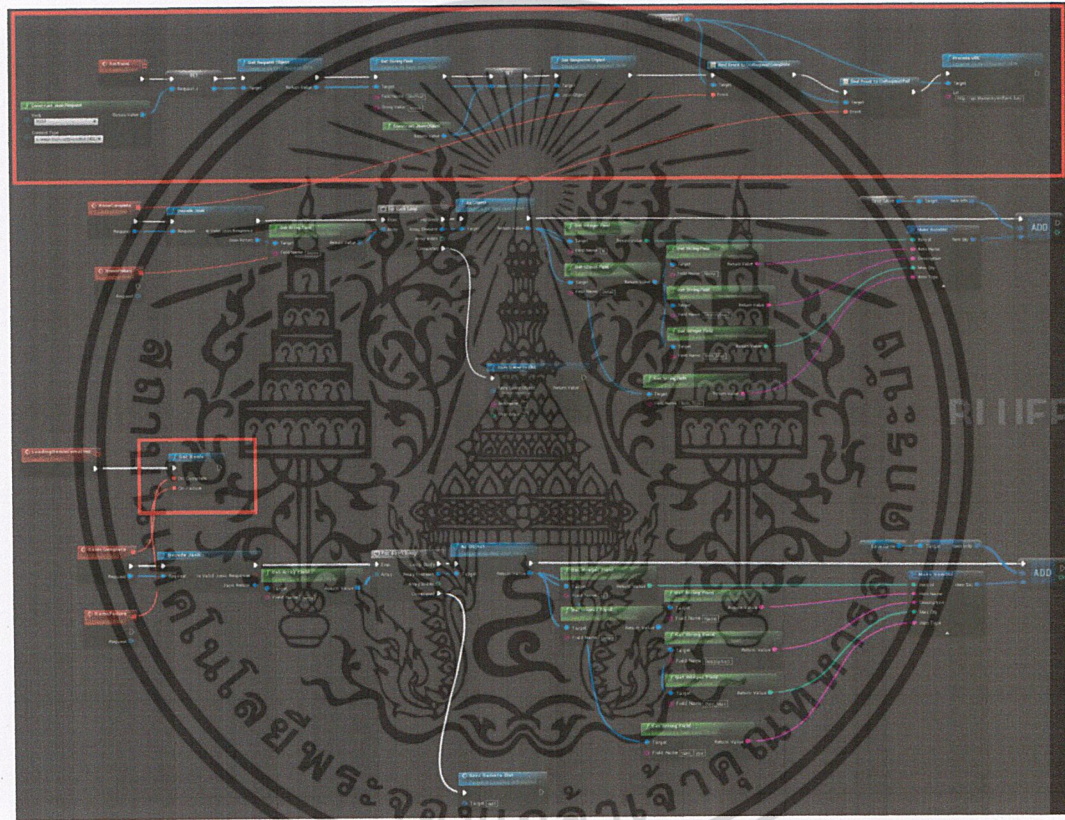
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Library Get Invest เป็นไลบรารีสำหรับเรียกดูข้อมูลการลงทุนที่ถูกลงทุนไปโดยตัวละครของ
ผู้ใช้งาน โดยต้องระบุ Character id (เลขรหัสลำดับของตัวละคร)
4. Library Get Decoration เป็นไลบรารีสำหรับเรียกดูของตกแต่งทุกอย่างที่อยู่ภายในระบบ
และสามารถนำไปเรียกใช้ในฟังก์ชันอื่น
5. Library Get Items เป็นไลบรารีสำหรับเรียกดูสิ่งของ (Items) ทุกอย่างที่มีอยู่ในระบบ และ
สามารถนำไปเรียกใช้ในฟังก์ชันอื่น
6. Library Get Friends เป็นไลบรารีสำหรับเรียกดูรายชื่อเพื่อนทั้งหมดที่มีจากตัวละครของ
ผู้ใช้งาน โดยระบุจาก Character Id
7. Library Sign in to Server เป็นไลบรารีสำหรับเรียกใช้งานตัวละคร ชื่อตัวละคร ข้อมูลการ
ตกแต่งตัวละคร ข้อมูลตำแหน่งยืนล่าสุด ข้อมูลช่องสัมภาระ ข้อมูลเพื่อนเป็นต้น โดยจะ
ทำงานเมื่อใส่ Username (ชื่อบัญชีผู้ใช้งาน) และ Password (รหัสผ่าน) ถูกต้องตามที่ได้
ลงทะเบียนกับระบบไว้
8. Library Transform เป็นไลบรารีสำหรับเรียกใช้งานตำแหน่งยืนล่าสุดของตัวละครก่อนที่จะ
ออกจากระบบเพื่อเอาไว้ใช้สำหรับเป็นตำแหน่งยืนในการเข้าสู่ระบบครั้งต่อไป

การเรียกใช้งานไลบรารีที่กลุ่มการทำงานเบื้องหลังได้จัดเตรียมไว้ให้สามารถช่วยลดขั้นตอน
การทำงานในการเรียกใช้ข้อมูลในส่วนที่กลุ่มการทำงานเบื้องหน้าต้องการในกรณีที่ต้องการเรียกใช้
ข้อมูลชุดเดิมในการทำงานอื่นๆภายใน Unreal Engine 4 สามารถเปรียบเทียบความแตกต่างได้ดังรูป

4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 เปรียบเทียบระหว่างกลุ่มการทำงานเบื้องหน้าเขียนฟังก์ชันเรียกใช้งาน Item ด้วยตนเอง (รูปส่วนบน) กับ กลุ่มการทำงานเบื้องหน้าเรียกใช้งาน

ไลบรารี Get Items ที่ทางกลุ่มการทำงานเบื้องหลังเตรียมไว้ให้ (รูปส่วนล่าง)

4.3 ความต้องการขั้นต่ำของระบบ

ในการพัฒนาระบบฐานข้อมูลสำหรับโลกเสมือนและโครงสร้างการทำงานของโลกเสมือนเป็นแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการ Windows และมีความจำเป็นในการใช้อินเทอร์เน็ตเพื่อเข้าถึงเซิร์ฟเวอร์

เซิร์ฟเวอร์ที่ทำการโฮสต์ระบบของโลกเสมือนจำเป็นต้องมีความต้องการขั้นต่ำของระบบดังนี้

Operating System (OS) : Windows

Processor : Any CPU with 1 Core (or more)

Memory : 3.75 GB of RAM (or more)

Network : Broadband Internet connection (จำเป็นต้องที่
การเชื่อมต่ออินเทอร์เน็ต)

Storage : 5 GB available space (50 GB default Google
Cloud Platform)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

การทดสอบการทำงานของเซิร์ฟเวอร์เมื่อมีผู้ใช้งานหลายคนพร้อมกันที่เป็นบุคคลภายนอกจำนวน 20 คน โดยให้ทำการดาวน์โหลดตัวโคลนแอนท์ของโลกลเสมือนฟาร์มหลังบ้านและเข้าใช้งานระบบพร้อมกัน โดยมีการเฝ้าดูการทำงาน CPU เซิร์ฟเวอร์ในแดชบอร์ด (Dashboard) บน Google Cloud Platform Console และการทำงานของระบบของโลกลเสมือนภายในโคลนแอนท์ว่าสามารถดำเนินไปได้อย่างราบรื่นหรือไม่ นอกจากนี้ได้มีการสอบถามความคิดเห็นและผลจากการใช้งาน API ที่กลุ่มการทำงานเบื้องหลังสร้างขึ้นจากกลุ่มการทำงานเบื้องหน้า หลังจากได้ข้อมูลของทั้งสองส่วนแล้วจะนำมาบางแยกเพื่อสรุปถึงปัญหาที่เกิดขึ้นต่าง ๆ ของหัวข้อเพื่อนำมาแก้ไขปัญหาต่างๆ ต่อไป

5.1 สรุปผลการดำเนินการ

5.1.1. การทำงานของระบบโลกลเสมือน

ในการทดสอบการทำงานของระบบโลกลเสมือนที่อยู่บนเซิร์ฟเวอร์ที่ทางกลุ่มการทำงานเบื้องหลังได้ใช้ทดลองกับผู้ใช้งานที่เป็นบุคคลภายนอกจำนวน 20 คน โดยมีผู้เข้าใช้งานพร้อมกันสูงสุด 17 คน จำนวน 3 ครั้ง จากการสังเกตการณ์ระหว่างจำนวนผู้ใช้งานและการทำงานของ CPU บน Dashboard ของ Google Cloud Platform มีผลออกมาว่าการทำงานทั่วไปของระบบเป็นไปได้ อย่างปกติภายในช่วงเวลาที่จำนวนของผู้เข้าทดสอบได้เข้าใช้งานระบบระหว่าง 1 - 13 คน แต่ภายในช่วงผู้เข้าใช้งานระบบตั้งแต่ 14 คนขึ้นไปจะเกิดอาการแล็คชั่นเล็กน้อย ซึ่งต่ำกว่าเกณฑ์ที่กลุ่มการทำงานเบื้องหลังได้คาดการณ์ไว้ว่าเหตุการณ์นี้อาจเกิดขึ้นเมื่อมีผู้ใช้งานเข้าใช้งานพร้อมกันถึง 20 คน โดยผู้ใช้งานจะเห็นตัวละครของผู้ใช้งานคนอื่นเวลาเดินเกิดการขาดตอนและหายไปและกลับเข้ามาในจอภาพอีกครั้งในตำแหน่งทางที่กำลังเดิน นอกจากนี้ยังลองให้ผู้ใช้ 3 คนจาก 17 คนจากที่ผู้เข้าใช้งานระบบพร้อมกันทำการออกจากระบบ มีผลออกมาว่ามีเป็นไปได้ที่ระบบจะกลับมาทำงานปกติในช่วงที่ผู้ใช้งานทั้งหมดอยู่ในระบบเป็น 15 คน จาก 2 ใน 3 ครั้งของการทดสอบ ซึ่งปัญหานี้เกิดขึ้นมาจากการเลือกใช้เซิร์ฟเวอร์ที่มี CPU คอร์การประมวลผลที่น้อย การแก้ปัญหาคือต้องมีการเพิ่มจำนวนคอร์ของ CPU ที่เลือกใช้ใน Google Cloud Platform Console ซึ่งจะทำให้มีค่าใช้จ่ายเพิ่มขึ้นตามมาด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2. ผลการใช้งานของ API ที่มีแต่กลุ่มการทำงานเบื้องหน้า

ในระหว่างการพัฒนากระบวนการทำงานเบื้องหลังได้ทำ API ที่เป็น Function Blueprint Library มาให้กับกลุ่มการทำงานเบื้องหน้าได้ใช้เพื่อช่วยในการลดเวลาการเรียกใช้งานข้อมูลเมื่อมีการเรียกใช้ข้อมูลชุดเดิมซ้ำ ๆ และลดขั้นตอนในการทำงานในการเขียนฟังก์ชัน Blueprint ใน Unreal Engine 4 ระหว่างพัฒนาระบบส่วนหน้า โดยมีผลตอบกลับมาว่าช่วยทำให้การทำงานสะดวกขึ้นและช่วยให้ประหยัดเวลาส่วนที่ต้องไปเขียนฟังก์ชันเรียกใช้งานข้อมูลด้วยตนเองทุกครั้ง

5.2 ข้อเสนอแนะ

การแก้ปัญหาในส่วนของประสิทธิภาพการทำงานของระบบบนเซิร์ฟเวอร์แก้ได้ด้วยการเพิ่มคอร์การทำงานบน CPU ในการเลือกตั้งเซิร์ฟเวอร์จะช่วยให้การทำงานของระบบเมื่อมีผู้ใช้งานพร้อมกันหลายคนได้ดียิ่งขึ้นซึ่งอาจทำให้เกิดปัญหาใหม่ตามมาในเรื่องของค่าใช้จ่ายที่เพิ่มขึ้น

ในส่วนของ API หากมีฟังก์ชันใหม่ได้ถูกเพิ่มเข้ามาในภายหลังและฟังก์ชันส่วนนั้นอยู่นอกเหนือความครอบคลุมจาก API ที่ทางกลุ่มการทำงานเบื้องหลังได้สร้างไว้จะไม่สามารถเรียกใช้งาน API เพื่อติดต่อรับข้อมูลจากฟังก์ชันส่วนใหม่ได้ แต่ยังสามารถเรียกใช้ข้อมูลจากฟังก์ชันส่วนเก่าไปใช้งานในฟังก์ชันส่วนใหม่ได้ ซึ่งอาจจะต้องมีการเขียนและเพิ่มฟังก์ชัน API ใหม่ที่เป็น Function Blueprint Library เข้าไปเองและรวมถึง PHP ในการเรียกข้อมูล

5.3 การพัฒนาในอนาคต

จากการพัฒนาในปัจจุบันทางทีมพัฒนาได้มีการพัฒนาระบบโลกเสมือน ให้สามารถรองรับผู้เข้าใช้งานหลายคนสามารถเดินทางไปยังโลกเสมือนในส่วนของเกมเพื่อพบปะสังสรรค์กับผู้ใช้งานอื่น และในส่วนของเกมส่วนตัวของผู้ใช้งาน สำหรับการพัฒนาต่อในอนาคตในส่วนของเกมต่างการลงทุนจะมีการเชื่อมต่อกับ GAP API เพื่อรองรับการส่งข้อมูลการลงทุนทางการเกษตรไปยังเกษตรกร ส่วนของการแจ้งความคืบหน้าทางการเกษตร ส่วนของสรุปผลการลงทุน เช่น ค่าใช้จ่าย และระบบการชำระเงินจากการลงทุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] MuleSoft, LLC (n.d.). "What is an API? (Application Programming Interface)". สืบค้นจาก <https://www.mulesoft.com/resources/api/what-is-an-api>
- [2] MuleSoft, LLC (2015) *What is API* [วิดีโอ] นาทีที่ 1:13 ถึง 1:45. สืบค้นจาก <https://www.mulesoft.com/resources/api/what-is-an-api> หรือ <https://youtu.be/s7wmiS2mSXY>
- [3] Web Developers Notes (n.d.). "What is web server – a computer OR a program?". สืบค้นจาก <https://www.webdevelopersnotes.com/what-is-web-server>
- [4] Roy Thomas Fielding (2000). "Representational State Transfer (REST)". สืบค้นจาก https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [5] REST API Tutorial (n.d.). "What is REST". สืบค้นจาก <https://restfulapi.net/>
- [6] R. F. Edward & J. R. Edward (June 2014). "Response Status Code", *Internet Engineering Task Force(IETF)*, 46. สืบค้นจาก <https://tools.ietf.org/html/rfc7231#page-46>
- [7] M. Vaqqas (2014). "RESTful Web Services: A tutorial" สืบค้นจาก <http://www.drdoobs.com/web-development/restful-web-services-a-tutorial/240169069>

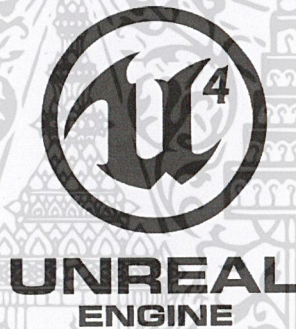
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ก. โปรแกรมและเครื่องมือที่ใช้ในการพัฒนา

1. Unreal Engine 4

เป็นเครื่องมือพัฒนาเกมที่ถูกพัฒนาขึ้นโดยบริษัท Epic Games การพัฒนาเกมโดย Unreal Engine สามารถพัฒนาได้สองรูปแบบคือ การพัฒนาโดยการเขียนโค้ดด้วยตนเอง โดยใช้ภาษา C++ ในการเขียน และอีกวิธีหนึ่งคือการใช้งาน Blueprint ซึ่งเป็น Visual Code หรือ โค้ดที่แสดงออกเป็นแบบภาพหรือกราฟแทนที่จะเป็นโค้ดจริง ๆ ในรูปแบบนี้สามารถเขียนและเพิ่มฟังก์ชันเข้าไปเองได้ โดยการทำงานทั้งสองรูปแบบนี้จำเป็นจะต้องใช้ Visual Studio ในการรองรับการทำงานของโค้ด ซึ่งทางกลุ่มการพัฒนาโดยรวมได้เลือกใช้วิธีการเขียนโค้ดในรูปแบบของ Blueprint



รูปที่ 1ก Unreal Engine

2. Visual Studio 2019

เป็นเครื่องมือสำหรับพัฒนาโปรแกรมบนระบบปฏิบัติการต่าง ๆ ที่ถูกพัฒนาขึ้นโดยบริษัท Microsoft สำหรับนักพัฒนาโปรแกรมสามารถใช้พัฒนาโปรแกรมต่าง ๆ ของตนได้อย่างอิสระ โดยภายในโปรแกรมนี้อาจมีระบบช่วยคาดเดาคำโค้ดที่กำลังจะเขียนลงไปทำให้ช่วยในการลดขั้นตอนการทำงานและประหยัดเวลาได้ในระดับหนึ่ง โดย Visual Studio นั้นมีการรองรับการเขียนโค้ดในหลายภาษาและสามารถดาวน์โหลด Plug in ภาษาเพิ่มเติมลงมาในโปรแกรมให้รองรับได้อีกด้วย



รูปที่ 2ก Visual Studio

3. AppServ

เป็นเครื่องมือชุดโปรแกรมการสร้างเว็บเซิร์ฟเวอร์สำเร็จรูปที่สามารถใช้งานบนระบบปฏิบัติการ Windows โดยทางโปรแกรมได้มีการรวบรวมโปรแกรมจำนวน 4ตัวในการสร้างเว็บเซิร์ฟเวอร์ได้แก่ Apache HTTP Server, PHP, MySQL และ phpMyAdmin เข้าด้วยกัน



รูปที่ 3ก AppServ

4. FileZilla

เป็นเครื่องมือซอฟต์แวร์แอปพลิเคชันสำหรับการโอนถ่ายข้อมูลผ่านโพรโทคอล (FTP, File Transfer Protocol) เพื่อใช้สำหรับโอนถ่ายข้อมูลระหว่างไคลแอนท์และเซิร์ฟเวอร์ ตัวไคลแอนท์ของไฟล์ซึลล่ารองรับระบบปฏิบัติการ Windows, macOS และ Linux ส่วนตัวการทำงานบนฝั่งเซิร์ฟเวอร์นั้นรองรับการทำงานเพียงแค่ระบบปฏิบัติการ Windows เท่านั้น



รูปที่ 4ก FileZilla

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Google Cloud Platform

เป็นคลาวด์ แพลตฟอร์ม (Cloud Platform) ที่ให้บริการในรูปแบบของเว็บเซิร์ฟเวอร์ที่ถูกพัฒนาขึ้นโดยบริษัท Google โดยมีความสามารถในการวิเคราะห์และจัดการข้อมูลและให้ผู้ใช้งานสามารถเลือกจัดสรรทรัพยากรเพื่อตั้งเซิร์ฟเวอร์ของตนเองได้ตามความต้องการ รวมถึงให้ทดลองใช้งานฟรีด้วยเครดิตจำนวน 300 เหรียญสหรัฐ



Google Cloud Platform

รูปที่ 5ก Google Cloud Platform

6. MySQL-Front และ MySQL Workbench

เป็นเครื่องมือซอฟต์แวร์แอปพลิเคชันที่ช่วยอำนวยความสะดวกในการสร้างและการออกแบบฐานข้อมูล แก้ไขข้อมูลในตารางข้อมูล ปรับปรุงและดูแลระบบฐานข้อมูล รวมถึงสามารถแสดงข้อมูลออกมาในรูปแบบของตาราง หรือกราฟและไดอะแกรมเพื่อให้ง่ายต่อความเข้าใจ



รูปที่ 6ก MySQL-Front และ MySQL Workbench

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข. ตารางอธิบายข้อมูลของ API ทั้งหมดที่กลุ่มการทำงานเบื้องหลังได้สร้างขึ้น

ตารางที่ 1x API ลำดับที่ 1

ชื่อ API	Get DB
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลเฉพาะเจาะจงหรือทั้งหมดของ ฟาร์ม หรือ ตัวละคร จากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อต้องการอัปเดตข้อมูลในการแสดงผลของตัวละครหรือฟาร์ม
มีโปรแกรมอะไรบ้าง	นำพารามิเตอร์ ไปใช้เชื่อมโยงในการขอข้อมูลเฉพาะเจาะจงหรือทั้งหมดของ ตัวละครหรือฟาร์ม
พารามิเตอร์	ระบุข้อมูลที่ต้องการระหว่างข้อมูลตัวละคร หรือ ฟาร์ม, ระบุค่าเพื่อเลือกว่าทั้งหมดหรือเฉพาะเจาะจง (ทั้งหมดคือ All เฉพาะเจาะจงคือ ชื่อฟิลในฐานข้อมูล)
ข้อมูลที่ได้รับ	ได้รับข้อมูลตัวละครหรือฟาร์ม จากฐานข้อมูลในรูปแบบ Json Object

ตารางที่ 2x API ลำดับที่ 2

ชื่อ API	Table to String
ประเภทกลุ่ม	Phrase
ใช้ทำอะไร	แปลงค่าจาก Enum TableInDatabase เป็น String เพื่อนำไปใช้ในการต่อ String บนฐานข้อมูล
ใช้เมื่อไหร่	ถูกเรียกใช้ใน Get DB, Update
มีโปรแกรมอะไรบ้าง	รับค่าจากที่ระบุ Enum TableInDatabase จาก Get DB มาแล้ว ทำการแปลงเป็น String แล้วนำไปใช้ส่งคำขอ
พารามิเตอร์	รับ Enum TableInDatabase
ข้อมูลที่ได้รับ	String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3 ข API ลำดับที่ 3

ชื่อ API	DecodeJson
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้แปลงค่า Json Object ที่รับมาจาก Response ให้เป็น Json Object สำหรับใช้งาน
ใช้เมื่อไหร่	ใช้เมื่อมี Response มาจากเซิร์ฟเวอร์ ใน Event OnRequestComplete / OnRequestFailure หลังจากทำการส่งคำขอ
มีโปรแกรมอะไรบ้าง	มีการเช็คค่า Json Object ที่รับมาว่ามีการกำหนดค่าหรือไม่(IsValid) แล้ว แปลงค่า(ตัวแปลงของVarestPlugin)จาก Json Object ที่ได้ จาก Respond เป็น Json Object ที่สามารถใช้งานได้ แล้วคืนค่า Json Object ให้
พารามิเตอร์	Json Object Respond
ข้อมูลที่ได้รับ	Json Object Available

ตารางที่ 4 ข API ลำดับที่ 4

ชื่อ API	DecodeJson2Field
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้แปลงค่า Json Object ที่รับมาจาก Response ให้ เป็น Json Object สำหรับใช้งาน โดยจะเข้าใช้งานฟิลด์ของ JsonObject ที่กำหนดมาทั้ง 1st Field และ 2nd Field เพื่อให้สามารถเรียกใช้งานได้ทันที
ใช้เมื่อไหร่	ใช้เมื่อมี Response มาจากเซิร์ฟเวอร์ ใน Event OnRequestComplete / OnRequestFailure หลังจากทำการส่งคำขอ
มีโปรแกรมอะไรบ้าง	มีการเช็คค่า Json Object ที่รับมาว่ามีการกำหนดค่าหรือไม่(IsValid) แล้ว แปลงค่า(ตัวแปลงของVarestPlugin)จาก Json Object ที่ได้ จาก Respond เป็น Json Object ที่สามารถใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	งานได้ แล้วคืนค่า Json Object ให้ โดยเริ่มจากฟิลตามที่กำหนด
พารามิเตอร์	Json Object Respond
ข้อมูลที่ได้รับ	Json Object Available

ตารางที่ 5x API ลำดับที่ 5

ชื่อ API	UpdateValueInJson
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอในการอัปเดตข้อมูลบน ฐานข้อมูลที่อยู่ในตาราง Character หรือ Farm โดยการกำหนดฟิลที่ต้องการอัปเดต, CharacterId, JsonStringValue
ใช้เมื่อไหร่	เมื่อต้องการอัปเดตข้อมูลในฟิลของดาต้าเบส เช่นหลังจากย้ายไอเทมในคลัง, หลังจากกด Save ใน DecorationMode
มีโปรแกรมอะไรบ้าง	ส่งคำขอโดยระบุ Character/Farm , กำหนด Field, Character Id และ ค่า Json Object ที่แปลงเป็น String แล้ว
พารามิเตอร์	Select Table (Character,Farm), Select Field, Character Id, Json String Value
ข้อมูลที่ได้รับ	Respond Result(True/False)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6x API ลำดับที่ 6

ชื่อ API	DecodeJson1Field
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้แปลงค่า Json Object ที่รับมาจาก Response ให้ เป็น Json Object สำหรับใช้งาน โดยจะเข้าใช้งานฟิลด์ของ JsonObject ที่กำหนด มาทั้ง 1st Field เพื่อให้สามารถเรียกใช้งานได้ทันที
ใช้เมื่อไหร่	ใช้เมื่อมี Response มาจากเซิร์ฟเวอร์ ใน Event OnRequestComplete / OnRequestFailure หลังจากทำการส่งคำขอ
มีโปรแกรมอะไรบ้าง	มีการเช็คค่า Json Object ที่รับมาว่ามีการกำหนดค่าหรือไม่ (IsValid) แล้วแปลงค่า(ตัวแปลงของ VarestPlugin) จาก Json Object ที่ได้จาก Respond เป็น Json Object ที่สามารถใช้งานได้ แล้วคืนค่า Json Object ให้ โดยเริ่มจาก ฟิลด์ตามที่กำหนด
พารามิเตอร์	Json Object Respond
ข้อมูลที่ได้รับ	Json Object Available

ตารางที่ 7x API ลำดับที่ 7

ชื่อ API	UpdateItem
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้แก้ไขค่าใน Json Object Array ตาม ฟิลด์ที่เป็น id (Integer Only)
ใช้เมื่อไหร่	ต้องการแก้ไขค่าใน Json Object Array มีการเรียกใช้ในการจัดการใน Item's Storage(แก้ไขจำนวนของไอเทมในคลัง)
มีโปรแกรมอะไรบ้าง	มีการค้นหา Json Object ที่อยู่ใน Json Object Array และทำการกำหนดค่าใหม่ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์	Target(Json Object), Array Field Name, Integer Use To ID, Equals (Integer) , Integer Use To Set, Set Value (Integer)
ข้อมูลที่ได้รับ	-

ตารางที่ 8 ข API ลำดับที่ 8

ชื่อ API	FindEmpty
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้หาที่อยู่ใน Json Object Array ตาม ฟิลด์ที่เป็น id (Integer Only) เพื่อเช็คว่ามี id นี้ อยู่ใน Json Object Array หรือไม่
ใช้เมื่อไหร่	ต้องการทราบ ฟิลด์ที่เป็น id (Integer Only) ใน Json Object Array ว่ามีค่า ตาม id ฟิลด์ที่กำหนดหรือไม่ มีการเรียกใช้ในการจัดการใน Item's Storage(เช็คค่าช่องของไอเทมในคลังมีไอเทมอยู่หรือไม่)
มีโปรแกรมอะไรบ้าง	มีการค้นหา Json Object ที่อยู่ใน Json Object Array เมื่อเจอจะทำการคืนค่า True(Boolean)
พารามิเตอร์	Target(Json Object), Array Field Name, Integer Use To ID, Equals (Integer)
ข้อมูลที่ได้รับ	Return Value(Boolean)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 9x API ลำดับที่ 9

ชื่อ API	CheckAlreadyUsername
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ใช้ตรวจสอบข้อมูลว่า Username หรือ Email มีอยู่ในระบบหรือไม่
ใช้เมื่อไหร่	เมื่อต้องการทราบว่า Username หรือ Email ที่ผู้ใช้กรอกในขั้นตอนการสมัครสมาชิกนั้นเคยถูกใช้ในระบบแล้วหรือไม่
มีโปรแกรมอะไรบ้าง	ส่งคำขอโดยระบุ Username / Email (ไม่ตรวจสอบอันไหนให้เว้นว่าง) แล้วจะได้รับผลกลับเป็น True/False(Boolean)
พารามิเตอร์	Username , Email (เลือกใส่อย่างใดอย่างหนึ่ง)
ข้อมูลที่ได้รับ	Respond Result(True/False)

ตารางที่ 10x API ลำดับที่ 10

ชื่อ API	Sign Up
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอเพื่อเพิ่มข้อมูลผู้ใช้งานคนใหม่ลงในฐานข้อมูล
ใช้เมื่อไหร่	ใช้เมื่อผู้ใช้ทำการคลิกสมัครสมาชิก
มีโปรแกรมอะไรบ้าง	นำข้อมูลที่ผู้ใช้งานกรอกไว้ในหน้าสมัครสมาชิกส่งคำขอไปเพื่อทำการสมัครสมาชิก
พารามิเตอร์	username, password (hash password in web server), email, fname, lname
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 11x API ลำดับที่ 11

ชื่อ API	Get Account Id
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอไอดีของตัวละคร จากฐานข้อมูล
ใช้เมื่อไหร่	ใช้เมื่อผู้ใช้ทำการสมัครสมาชิกสำเร็จ แล้วเลือก ว่าจะเข้าใช้งานเพื่อสร้างตัวละครต่อ
มีโปรแกรมอะไรบ้าง	ส่งคำขอโดยระบุ Username แล้วจะได้รับผล รับเป็น Account ID
พารามิเตอร์	Username
ข้อมูลที่ได้รับ	Json Object Available

ตารางที่ 12x API ลำดับที่ 12

ชื่อ API	Sign In to Server
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอเพื่อตรวจสอบ username กับ password กับฐานข้อมูล
ใช้เมื่อไหร่	ใช้เมื่อผู้ใช้ทำการคลิกเข้าสู่ระบบ
มีโปรแกรมอะไรบ้าง	ส่งคำขอโดยระบุ Username / Password แล้ว จะได้รับผลรับเป็น True/False(Boolean)
พารามิเตอร์	username, password
ข้อมูลที่ได้รับ	Respond Result(True/False) , ถ้า Result = True จะได้ account id ใน Json Object Available

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 13x API ลำดับที่ 13

ชื่อ API	GetCharacterInformationByAccount
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลของตัวละคร จากฐานข้อมูล
ใช้เมื่อไหร่	ใช้เมื่อผู้ใช้งานทำการเข้าสู่ระบบสำเร็จ(ตรวจสอบว่าสร้างตัวละครไปแล้วหรือไม่), ใช้เมื่อผู้ใช้งานสร้างตัวละครสำเร็จ
มีโปรแกรมอะไรบ้าง	ส่งคำขอโดยระบุ AccountId แล้วจะได้รับผลรับเป็น CharacterInformation
พารามิเตอร์	aid (account id)
ข้อมูลที่ได้รับ	Json Object Available

ตารางที่ 14x API ลำดับที่ 14

ชื่อ API	CheckAlreadyLogin
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลการเข้าใช้งานของตัวละครสำหรับตรวจว่าได้ใช้งานอยู่หรือไม่ จากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อผู้ใช้งานทำการเข้าใช้งานโลกเสมือน
มีโปรแกรมอะไรบ้าง	เมื่อผู้ใช้งานทำการเข้าใช้งานโลกเสมือนจะตรวจสอบว่ามีการใช้งานอยู่ในโลกเสมือนอยู่ตอนนี้หรือไม่
พารามิเตอร์	cid (Character Id)
ข้อมูลที่ได้รับ	Respond Result(True/False)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 15 ข API ลำดับที่ 15

ชื่อ API	ParseCharacterInfo
ประเภทกลุ่ม	Parse
ใช้ทำอะไร	แปลงค่าจาก Json Object เป็น CharacterInfo เพื่อนำค่าข้อมูลตัวละครมาเรียกใช้บนโลกเสมือน
ใช้เมื่อไหร่	เมื่อต้องการดึงข้อมูลของผู้ใช้งาน เช่น เมื่อผู้ใช้เข้าใช้งานโลกเสมือน(ใช้เพื่อนำค่ามาเก็บไว้ใช้ในระหว่างใช้งานโลกเสมือน)
มีโปรแกรมอะไรบ้าง	จำเป็นต้องมีการดึงข้อมูลของตัวละครมาก่อน เช่น Get DB(Character) แล้วใช้ Json Object (Available) แล้วตัว API จะทำการดึงค่าจาก Json Object ออกมาแล้วคืนค่าให้
พารามิเตอร์	Character Json(Json Object)
ข้อมูลที่ได้รับ	AccountId, CharacterId, Name, Farm Name, About Me, Gender, Eyes, Skin, Boots, Hair, Shirt, Pants, Map(Current Map), Transform, BY Coin

ตารางที่ 16 ข API ลำดับที่ 16

ชื่อ API	SetTransform
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่ออัปเดตข้อมูลตำแหน่งของตัวละคร ในฐานข้อมูล
ใช้เมื่อไหร่	ใช้เมื่อผู้ใช้ในงานอยู่ในโลกเสมือน
มีโปรแกรมอะไรบ้าง	ส่งคำขอโดยจะอัปเดตฟิลด์ Transform ของตาราง Character ในฐานข้อมูลตามพารามิเตอร์ที่รับเข้ามา
พารามิเตอร์	cid (Character Id), Transform (Parsed By MakeTranformInformation)
ข้อมูลที่ได้รับ	Respond Result(True/False)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 17 ข API ลำดับที่ 17

ชื่อ API	MakeTranformlInformation
ประเภทกลุ่ม	Parse
ใช้ทำอะไร	จัดการตัวแปรประเภท Transformation ของตัวละครเพื่อแปลงค่าเป็น String (Json String) โดยมีการจัดรูปแบบ สำหรับเก็บลงบน ฐานข้อมูล เพื่อให้เวลาดึงมาใช้งานแล้วสามารถ เรียกใช้ได้ง่าย
ใช้เมื่อไหร่	เมื่อต้องการทำการแปลงค่าตำแหน่งเป็น String(Json String)
มีโปรแกรมอะไรบ้าง	จะรับค่าตัว Map(Current_Map) และ Transform(Translation: X=?? Y=?? Z=?? Rotation: P=?? Y=?? R=?? Scale: X=?? Y=?? Z=??) แล้วมาจัดรูปแบบใหม่ให้เป็น Json Object แล้วทำการ Encode เป็น String
พารามิเตอร์	Map , Transform
ข้อมูลที่ได้รับ	String(JsonString) ({"Map": "Farm", "Transform": {"Position": {"X": "???.???", "Y": "???.???", "Z": "???.???"}, "Rotation": {"X": "???.???", "Y": "???.???", "Z": "???.???"}, "Scale": {"X": "???.???", "Y": "???.???", "Z": "???.???"}}})

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 18ข API ลำดับที่ 18

ชื่อ API	getStore
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลรายการสิ่งของในกล่องเก็บของ และกระเป๋าเก็บของ จากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อมีการเรียกใช้ข้อมูลของคลังเก็บของ มาแสดงผลในระบบโลกเสมือน
มีโปรแกรมอะไรบ้าง	ส่งคำโดยจะเรียกข้อมูลตามที่ระบุค่าของตัวแปรตอนเรียกใช้ Storage State(ประเภทคลัง) ของตัวละครใด Character Id ได้รับข้อมูลที่อยู่ในรูปแบบ Json Object
พารามิเตอร์	Character Id, Storage State
ข้อมูลที่ได้รับ	Respond (Json Object Storage Information)

ตารางที่ 19ข API ลำดับที่ 19

ชื่อ API	getItem
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลของไอเทมในโลกเสมือนในฐานข้อมูล
ใช้เมื่อไหร่	ตอนเข้าใช้งานโลกเสมือนจะมีการเรียกใช้งานเพื่อ ตรวจสอบข้อมูลของตัว SaveGame ว่ามีข้อมูลของไอเทมครบหรือไม่
มีโปรแกรมอะไรบ้าง	ส่งคำขอเพื่อขอข้อมูล
พารามิเตอร์	ไม่มีการรับพารามิเตอร์
ข้อมูลที่ได้รับ	Respond (Json Object Item Information)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 20x API ลำดับที่ 20

ชื่อ API	DeleteItem
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	เพื่อลบ Json Object ที่อยู่ใน Json Object Array ตาม 필และidที่กำหนด
ใช้เมื่อไหร่	เมื่อต้องการลบไอเทมจากไอเทมที่มีอยู่ หรือ กรณีที่เป็นการย้ายไอเทมข้ามคลัง
มีโปรแกรมอะไรบ้าง	มีการค้นหา Id ตาม 필ที่ระบุ ที่อยู่ใน Json Object Array เมื่อไม่เจอจะทำการเพิ่ม Json Object ลงใน Json Object Array ที่ทำหน้าที่ เป็น Buffer
พารามิเตอร์	Json(Json Object), Field Name(Name's Json Object Array), Id(Id's Integer Field in Json Object Array), Id Field(Name's Integer in Json Object Array)
ข้อมูลที่ได้รับ	Buffer(New Json Object Array ที่จะไม่ มี Json Object ที่ลบ)

ตารางที่ 21x API ลำดับที่ 21

ชื่อ API	SwapItem
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	เพื่อสลับตำแหน่งของไอเทมระหว่างสองชนิดใน คลัง
ใช้เมื่อไหร่	เมื่อผู้ใช้ทำการย้ายไอเทมไปยังตำแหน่งที่มี ไอเทมอยู่แล้วภายในคลัง
มีโปรแกรมอะไรบ้าง	ค้นหาตำแหน่งไอเทมที่อยู่ใน Json Object Array โดยอ้างอิงจาก ตำแหน่งในคลัง (Slot) ทั้งสองแล้ว ทำการสลับค่า ItemId, Qty ของทั้งสอง
พารามิเตอร์	Json 1 (Json Object), Json 2 (Json Object), Slot 1, Slot 2
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 22ข API ลำดับที่ 22

ชื่อ API	Moveltem
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	เพื่อย้ายตำแหน่งของไอเทมไปยังที่ว่าง
ใช้เมื่อไหร่	เมื่อผู้ใช้ทำการย้ายไอเทมไปยังตำแหน่งที่ไม่มีไอเทมอยู่ภายในคลัง
มีโปรแกรมอะไรบ้าง	ค้นหาตำแหน่งของไอเทมที่ก่อนจะย้ายตำแหน่งเพื่อหาตำแหน่ง Index ของ Json Object Array และทำการตรวจสอบว่าตัวข้อมูลของ Json Object Array มาจากคลังเดียวกันหรือไม่ ถ้าเป็นคลังเดียวกันจะนำค่าของไอเทมไปเปลี่ยนเฉพาะตำแหน่ง ถ้าไม่ใช่คลังเดียวกันจะทำการบันทึกข้อมูลไอเทมก่อนย้ายตำแหน่งแล้วทำสำเนาไปยังคลังเป้าหมายและเปลี่ยนข้อมูลของตำแหน่งไอเทมเป็นตำแหน่งที่ผู้ใช้เลือกวางแล้วจะทำการลบข้อมูลของตัวไอเทมก่อนย้ายตำแหน่ง
พารามิเตอร์	Json 1 (Json Object), Json 2 (Json Object), Slot 1, Slot 2
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

ตารางที่ 23ข API ลำดับที่ 23

ชื่อ API	AddItem
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	เพื่อเพิ่มไอเทมลงในข้อมูลไอเทม(Json Object's Storage)
ใช้เมื่อไหร่	เมื่อมีการเพิ่มไอเทมหรือการย้ายไอเทมระหว่างคลังต่างชนิดกัน
มีโปรแกรมอะไรบ้าง	นำ Json Object ที่ได้ เก็บค่า Json Object Array ของ ItemList และสร้าง Buffer แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	เพิ่ม ข้อมูลไอเทมใหม่เข้าไปใน Json Object Array
พารามิเตอร์	Storage Json, Item Id, Slot No, Qty
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

ตารางที่ 24x API ลำดับที่ 24

ชื่อ API	EditItem
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	เพื่อแก้ไขข้อมูลไอเทมในตำแหน่ง ตามที่กำหนด
ใช้เมื่อไหร่	เมื่อมีการแก้ไขข้อมูลของไอเทม
มีโปรแกรมอะไรบ้าง	นำ Json Object ที่ได้ ใช้ค่า Json Object Array ของ ItemList แล้วแก้ไข ข้อมูล ItemId และ Qty ของไอเทม ตาม Slot No ที่ ระบุ
พารามิเตอร์	Json1 (Json Object), ItemId, Slot No, Qty
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

ตารางที่ 25x API ลำดับที่ 25

ชื่อ API	UpdateStorage
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่ออัปเดตข้อมูลรายการสิ่งของใน กล่องเก็บของ และกระเป๋าเก็บของ ในฐานข้อมูล
ใช้เมื่อไหร่	เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับตัวคลังเก็บของ
มีโปรแกรมอะไรบ้าง	รับค่าจากที่ระบุ Enum TableInDatabase จาก UpdateStorage มาแล้ว ทำการแปลงเป็น String แล้วนำไปใช้ส่งคำขอ แล้วนำ Json String Value ไปอัปเดตข้อมูลคลังในฐานข้อมูล ตาม Select Storage,CharacterId
พารามิเตอร์	Select Storage(Enum StorageState), CharacterId, Json String Value
ข้อมูลที่ได้รับ	Respond Result(True/False)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 26x API ลำดับที่ 26

ชื่อ API	Removeltem
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	เพื่อลบไอเทมที่อยู่ในข้อมูลไอเทม(Json Object's Storage) ตาม Slot ที่ระบุ
ใช้เมื่อไหร่	เมื่อต้องการลบไอเทมจากไอเทมที่มีอยู่ หรือกรณีที่เป็นการย้ายไอเทมข้ามคลัง
มีโปรแกรมอะไรบ้าง	มีการค้นหาฟิล Slot_No ที่อยู่ใน ItemList (Json Object Array) เมื่อเจอจะทำการลบ Json Object ที่เจอ Slot_No นั้น
พารามิเตอร์	Json 1(Json Object), Slot 1(ตัว Slot ของไอเทมที่จะลบ)
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

ตารางที่ 27x API ลำดับที่ 27

ชื่อ API	getFriends
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลรายชื่อเพื่อน ของตัวละครนั้นๆ จากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อมีการเรียกดูข้อมูลรายชื่อเพื่อนในการแสดงผลของโลกเสมือน
มีโปรแกรมอะไรบ้าง	ส่งคำขอ โดยกำหนด Character Id (Integer), Raw Data(Boolean)(เป็นการกำหนดว่าต้องการข้อมูลแบบไหน True เป็นข้อมูลดิบใช้สำหรับวิเคราะห์คำนวณระบบ False เป็นข้อมูลที่ผ่านการจัดรูปแบบสำหรับการแสดงผล)
พารามิเตอร์	Character Id(Integer), Raw Data(Boolean)
ข้อมูลที่ได้รับ	Json Object Available

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 28ข API ลำดับที่ 28

ชื่อ API	UpdateFriends
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่ออัปเดตข้อมูลรายชื่อเพื่อน ในฐานข้อมูล
ใช้เมื่อไหร่	เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับรายชื่อเพื่อนของตัวละครนั้นๆ
มีโปรแกรมอะไรบ้าง	ส่งคำขอ โดยกำหนด Character Id (Integer), Json String Value(String ที่ผ่านการแปลงข้อมูลของ Json Object มาแล้ว)
พารามิเตอร์	Character Id(Integer), Json String Value (String)
ข้อมูลที่ได้รับ	Respond Result(True/False)

ตารางที่ 29ข API ลำดับที่ 29

ชื่อ API	AddPending
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้แก้ไขค่าใน Json Object Array โดยการเพิ่มไอดีของตัวละครที่ส่งคำขอเป็นเพื่อน ในรายชื่อคำขอเป็นเพื่อนของตัวละครที่ถูกขอเป็นเพื่อน
ใช้เมื่อไหร่	เมื่อผู้ใช้ทำการส่งคำขอเพิ่มเพื่อน
มีโปรแกรมอะไรบ้าง	เพิ่ม id ของตัวละครที่ขอใน Json Object (ของตัวละครที่ถูกขอเป็นเพื่อน)
พารามิเตอร์	Friends Json(Json Object),Number(Character id)
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 30x API ลำดับที่ 30

ชื่อ API	RemovePending
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้แก้ไขค่าใน Json Object Array โดยการลบไอดีของตัวละครที่อยู่ในรายการคำขอเป็นเพื่อน
ใช้เมื่อไหร่	เมื่อผู้ใช้ทำการปฏิเสธคำขอเพิ่มเพื่อน
มีโปรแกรมอะไรบ้าง	ลบ id ของตัวละครที่เลือกปฏิเสธใน Json Object (ของตัวละครที่ถูกขอเป็นเพื่อน)
พารามิเตอร์	Friends Json(Json Object),Number(Character id)
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

ตารางที่ 31x API ลำดับที่ 31

ชื่อ API	AddFriend
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้เพิ่มไอดีตัวละครลงใน Json Object Array ของรายชื่อเพื่อน
ใช้เมื่อไหร่	เมื่อผู้ใช้ทำการส่งคำขอเพิ่มเพื่อน
มีโปรแกรมอะไรบ้าง	เพิ่ม id ของตัวละครจะขอใน Json Object โดยที่ตัว is Confirm จะเป็น False ต้องรอรับถึงจะค่อยเปลี่ยนเป็น True
พารามิเตอร์	Friends Json(Json Object),Character id(Integer), Is Confirm(Boolean)
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 32x API ลำดับที่ 32

ชื่อ API	EditFriendConfirm
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้แก้ไขค่าใน Json Object Array โดยตรวจสอบว่ามีการยืนยันการเป็นเพื่อนกันหรือไม่
ใช้เมื่อไหร่	เมื่อผู้ใช้ทำการยอมรับคำขอเพิ่มเพื่อน
มีโปรแกรมอะไรบ้าง	แก้ไขข้อมูลของเพื่อนตาม CharacterId ที่ต้องการ ใน Friends Json
พารามิเตอร์	Friends Json(Json Object), Character id(Integer), Is Confirm(Boolean)
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

ตารางที่ 33x API ลำดับที่ 33

ชื่อ API	checkFriendName
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลรายชื่อเพื่อนสำหรับการตรวจสอบว่ามีชื่อของตัวละครนี้ จากฐานข้อมูลหรือไม่
ใช้เมื่อไหร่	เมื่อผู้ใช้ทำการส่งคำขอเพิ่มเพื่อน
มีโปรแกรมอะไรบ้าง	นำชื่อที่ส่งคำขอไปตรวจว่ามีชื่อนี้ในระบบหรือไม่
พารามิเตอร์	Friend Name(String)
ข้อมูลที่ได้รับ	Json Object Available

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 34 ข API ลำดับที่ 34

ชื่อ API	getServerList
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลเกี่ยวกับเซิร์ฟเวอร์ ให้ผู้ใช้เลือกเชื่อมต่อ
ใช้เมื่อไหร่	เมื่อผู้ใช้ ทำการลงชื่อเข้าใช้สำเร็จ จะถูกนำมาแสดงผลให้ผู้ใช้สามารถเลือกเชื่อมต่อ
มีโปรแกรมอะไรบ้าง	ส่งคำขอไปยังเว็บเซิร์ฟเวอร์เพื่อขอข้อมูล
พารามิเตอร์	ไม่มีการรับพารามิเตอร์
ข้อมูลที่ได้รับ	Respond (JSon Object Server Information)

ตารางที่ 35 ข API ลำดับที่ 35

ชื่อ API	setLogin
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่ออัปเดตข้อมูลการเข้าใช้งานของตัวละคร
ใช้เมื่อไหร่	เมื่อผู้ใช้งานเข้าใช้งาน หรือ ออกจากระบบ
มีโปรแกรมอะไรบ้าง	ส่งคำขอเพื่อแก้ไขข้อมูล IsLogin ในตารางตัวละคร
พารามิเตอร์	Character (Integer), Login Status (Enum IsLogin [0:False,1:True])
ข้อมูลที่ได้รับ	Respond Status(Success/Failure)(String)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 36 ข API ลำดับที่ 36

ชื่อ API	UpdateValueInJsonNoEvent
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอในการอัปเดตข้อมูลบนฐานข้อมูลที่อยู่ในตาราง Character หรือ Farm โดยการกำหนดฟิลที่ต้องการอัปเดต, CharacterId, JsonStringValue แบบทันทีสำหรับข้อมูลที่ไม่ต้องรอ Respond
ใช้เมื่อไหร่	เมื่อต้องการอัปเดตข้อมูลในฟิลของดาต้าเบส เช่นหลังจากย้าย ไอเทมในคลัง,หลังจากกด Save ใน DecorationMode
มีโปรแกรมอะไรบ้าง	ส่งคำขอโดยระบุ Character/Farm, กำหนด Field, Character Id และ ค่า Json Object ที่แปลงเป็น String แล้ว
พารามิเตอร์	Select Table (Character, Farm), Select Field, Character Id, Json String Value
ข้อมูลที่ได้รับ	Respond Result(True/False)

ตารางที่ 37 ข API ลำดับที่ 37

ชื่อ API	GetPublicIP
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอ PublicIP ของเครื่อง
ใช้เมื่อไหร่	เมื่อมีการ เพิ่มลดรีเซ็ตข้อมูลตัวละครของ เซิร์ฟเวอร์ (57.IncreasePlayer, 58.DecreasePlayer, 59.ResetPlayer)
มีโปรแกรมอะไรบ้าง	ส่งคำขอ โดยมีการกำหนดให้เลือก IP Version(IPV4 / IPV6)
พารามิเตอร์	IP Version (Enum)
ข้อมูลที่ได้รับ	Respond (JSoN Object IP Information)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 38ข API ลำดับที่ 38

ชื่อ API	getDecoration
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลสิ่งขอตกลงส่วนตัวละคร จากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อผู้ใช้ทำการเข้าใช้งานโลกเสมือน
มีโปรแกรมอะไรบ้าง	ส่งคำขอเพื่อขอข้อมูล
พารามิเตอร์	ไม่มีการรับพารามิเตอร์
ข้อมูลที่ได้รับ	Respond (JSoN Object Decoration Information)

ตารางที่ 39ข API ลำดับที่ 39

ชื่อ API	getDecorationType
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลประเภทของสิ่งขอตกลง จากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อผู้ใช้ทำการเข้าใช้งานส่วนระบบตกแต่ง
มีโปรแกรมอะไรบ้าง	ส่งคำขอเพื่อขอข้อมูล
พารามิเตอร์	ไม่มีการรับพารามิเตอร์
ข้อมูลที่ได้รับ	Respond (JSoN Object DecorationType Information)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 40x API ลำดับที่ 40

ชื่อ API	Decoration2Transform
ประเภทกลุ่ม	Parse
ใช้ทำอะไร	จัดการตัวแปรประเภท Float เพื่อแปลงค่าเป็นตัวแปร Transformation โดยมีการจัดรูปแบบ "(Location(x,y,z), Rotation(x,y,z), Scale(x,y,z))"
ใช้เมื่อไหร่	เมื่อต้องการแปลงค่า Float เป็น Transform
มีโปรแกรมอะไรบ้าง	รับค่าตัวแปรที่เป็น Float เข้ามาใส่ลงในพารามิเตอร์ x,y,z ของ Location, Rotation, Scale แล้วนำทั้งสามตัวนี้แปลงเป็น Transform
พารามิเตอร์	Target(Json Object)
ข้อมูลที่ได้รับ	Id(Decoration Id), Transform

ตารางที่ 41x API ลำดับที่ 41

ชื่อ API	TransformDecoration2Json
ประเภทกลุ่ม	Parse
ใช้ทำอะไร	จัดการตัวแปรประเภท Transform ของสิ่งของตกแต่งเพื่อแปลงค่าเป็นตัวแปร Json โดยมีการจัดรูปแบบ "(Location(x,y,z), Rotation(x,y,z), Scale(x,y,z))"
ใช้เมื่อไหร่	เมื่อต้องการแปลงค่า Decoration Actor (เฉพาะ id, Transform) เป็น Json
มีโปรแกรมอะไรบ้าง	นำค่า id, Transform ของ Decoration จัดให้อยู่ในรูปแบบ Json Object
พารามิเตอร์	Decoration Json (Decoration Destroyable Actor)
ข้อมูลที่ได้รับ	Json Object Available

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 42x API ลำดับที่ 42

ชื่อ API	CreateCharacter
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อสร้างข้อมูลของตัวละครตามที่ได้เลือกไว้
ใช้เมื่อไหร่	ใช้เมื่อผู้ใช้ทำการคลิกสร้างตัวละครสำเร็จ
มีโปรแกรมอะไรบ้าง	ส่งคำขอเพื่อสร้างตัวละคร โดยมีการกำหนดโครงสร้างที่ใช้ในการเก็บลง ฐานข้อมูล และมีการใช้ ColorPattern2String ในการแปลงค่าของ Hair, Shirt, Pants เป็น String เพื่อให้ง่ายสำหรับเรียกใช้
พารามิเตอร์	Character Name, Gender, Eyes, Skin, Boots, Hair, Shirt, Pants, Account Id
ข้อมูลที่ได้รับ	Respond Result(True/False)

ตารางที่ 43x API ลำดับที่ 43

ชื่อ API	ColorPattern2String
ประเภทกลุ่ม	Parse
ใช้ทำอะไร	จัดการตัวแปรประเภท ColorPattern เพื่อแปลงค่าเป็นตัวแปร String โดยมีการจัดรูปแบบ "(?,?,-?)"
ใช้เมื่อไหร่	เมื่อต้องการแปลงค่ากลับเป็น String แล้วนำไปอัปเดตข้อมูลใน ฐานข้อมูล
มีโปรแกรมอะไรบ้าง	นำค่าที่ได้จากตัวแปรประเภท ColorPattern มาใส่ลง ตัวแปรประเภท String ในรูปแบบ "(?,?,-?)"
พารามิเตอร์	Color Pattern
ข้อมูลที่ได้รับ	String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 44x API ลำดับที่ 44

ชื่อ API	CheckCharacterName
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อตรวจสอบชื่อของตัวละครที่มีการถูกเรียกใช้งานหรือไม่ จากฐานข้อมูล
ใช้เมื่อไหร่	ใช้เมื่อผู้ใช้ทำการคลิกสร้างตัวละครสำเร็จ
มีโปรแกรมอะไรบ้าง	ส่งคำขอเพื่อสร้างตัวละคร โดยมีการกำหนด Player Name
พารามิเตอร์	Player Name (String)
ข้อมูลที่ได้รับ	Respond Result(True/False)

ตารางที่ 45x API ลำดับที่ 45

ชื่อ API	String2ColorPattern
ประเภทกลุ่ม	Parse
ใช้ทำอะไร	จัดการตัวแปรประเภท String ที่ถูกจัดรูปแบบไว้แล้ว "(?,-?)" เพื่อนำค่าจากในข้อความไปอยู่ในตัวแปร ColorPattern
ใช้เมื่อไหร่	เมื่อต้องการค่าสีชุดของตัวละคร
มีโปรแกรมอะไรบ้าง	นำค่าที่ได้จากตัวแปรประเภท String ที่อยู่ในรูปแบบ "(?,-?)" แปลงและจัดลงในตัวแปร ColorPattern
พารามิเตอร์	String
ข้อมูลที่ได้รับ	Color Pattern

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 46x API ลำดับที่ 46

ชื่อ API	getPlant
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลพืชแต่ละชนิด จากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อเรียกใช้ฟังก์ชันเลือกแผนการลงทุน
มีโปรแกรมอะไรบ้าง	ส่งคำขอเพื่อขอข้อมูลทั้งหมดของพืช
พารามิเตอร์	ไม่มีพารามิเตอร์
ข้อมูลที่ได้รับ	ได้รับข้อมูลพืชทั้งหมด จากฐานข้อมูลในรูปแบบ Json Object

ตารางที่ 47x API ลำดับที่ 47

ชื่อ API	getSpecies
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลสายพันธุ์ของพืชแต่ละชนิด จากฐานข้อมูล
ใช้เมื่อไหร่	หลังจากเลือกชนิดของพืชในแผนการลงทุน
มีโปรแกรมอะไรบ้าง	นำพารามิเตอร์ ไปเช็คเงื่อนไขในการขอข้อมูลทั้งหมดของสายพันธุ์พืช
พารามิเตอร์	ระบุ ข้อมูลพืชที่ต้องการทราบสายพันธุ์
ข้อมูลที่ได้รับ	ได้รับข้อมูลสายพันธุ์พืชของพืชที่เลือก จากฐานข้อมูลในรูปแบบ Json Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 48x API ลำดับที่ 48

ชื่อ API	getFarm
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลฟาร์มที่สามารถปลูกพืชสายพันธุ์ที่เลือกได้ จากฐานข้อมูล
ใช้เมื่อไหร่	หลังจากเลือกสายพันธุ์ของพืชในแผนการลงทุน
มีโปรแกรมอะไรบ้าง	นำพารามิเตอร์ ไปเช็คเงื่อนไขในการขอข้อมูลทั้งหมดของฟาร์ม
พารามิเตอร์	ระบุ ข้อมูลสายพันธุ์พืชที่ต้องการทราบฟาร์มที่รับปลูกพืช
ข้อมูลที่ได้รับ	ได้รับข้อมูลฟาร์มของสายพันธุ์พืชที่เลือก จากฐานข้อมูลในรูปแบบ Json Object

ตารางที่ 49x API ลำดับที่ 49

ชื่อ API	getSummary
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลผลสรุปแผนการลงทุนจากฐานข้อมูล
ใช้เมื่อไหร่	หลังจากเลือกฟาร์มและจำนวนยูนิตที่ต้องการในแผนการลงทุน
มีโปรแกรมอะไรบ้าง	นำพารามิเตอร์ ไปเช็คเงื่อนไขในการขอข้อมูลสรุปแผนการลงทุน
พารามิเตอร์	ระบุ ข้อมูลสายพันธุ์พืช และข้อมูลฟาร์มที่เลือก
ข้อมูลที่ได้รับ	ได้รับข้อมูลสรุปแผนการลงทุนที่เลือก จากฐานข้อมูลในรูปแบบ Json Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 50x API ลำดับที่ 50

ชื่อ API	updateFieldFarm
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้แก้ไขค่า Invest id ใน Json Object Array ตาม ฟิลที่เป็น id (Integer Only)
ใช้เมื่อไหร่	เมื่อยืนยันการชำระเงิน และสามารถชำระเงินได้สำเร็จ
มีโปรแกรมอะไรบ้าง	มีการเพิ่มไอดีแผนการลงทุนลงใน Json Object และนำไปเพิ่มลงใน Json Object Array ที่เป็น buffer
พารามิเตอร์	ระบุ ข้อมูลไอดีตัวละคร และไอดีแผนการลงทุน
ข้อมูลที่ได้รับ	Buffer(New Json Object Array โดยมีการเปลี่ยนแปลงข้อมูลแผนการลงทุนในแปลง)

ตารางที่ 51x API ลำดับที่ 51

ชื่อ API	insertInvest
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อเพิ่มข้อมูลแผนการลงทุน ลงในฐานข้อมูล
ใช้เมื่อไหร่	เมื่อยืนยันการชำระเงิน และสามารถชำระเงินได้สำเร็จ
มีโปรแกรมอะไรบ้าง	นำพารามิเตอร์ ที่ใส่มาส่งค่าไปยังฐานข้อมูล
พารามิเตอร์	ระบุ ข้อมูลสายพันธุ์พืช ข้อมูลฟาร์ม ไอดีตัวละคร และจำนวนยูนิต
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 52x API ลำดับที่ 52

ชื่อ API	updateRealFarm
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่ออัปเดตข้อมูลจำนวนแปลงที่เหลืออยู่ในฟาร์มจริง ในฐานข้อมูล
ใช้เมื่อไหร่	เมื่อยืนยันการชำระเงิน และสามารถชำระเงินได้สำเร็จ
มีโปรแกรมอะไรบ้าง	นำพารามิเตอร์ ที่ใส่มาส่งค่าไปยังฐานข้อมูล
พารามิเตอร์	ระบุ ข้อมูลของฟาร์มที่ต้องการ และจำนวนยูนิตที่ต้องการเปลี่ยนแปลง
ข้อมูลที่ได้รับ	ไม่ได้รับข้อมูล

ตารางที่ 53x API ลำดับที่ 53

ชื่อ API	getInvest
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลแผนการลงทุนจากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อต้องการข้อมูลของการลงทุนสำหรับแสดงผลในแปลงจำลอง
มีโปรแกรมอะไรบ้าง	นำพารามิเตอร์ ไปเช็คเงื่อนไขในการขอข้อมูลแผนการลงทุนทั้งหมด
พารามิเตอร์	ระบุ ข้อมูลไอดีตัวละคร
ข้อมูลที่ได้รับ	ได้รับข้อมูลแผนการลงทุนของตัวละครที่เลือกจากฐานข้อมูลในรูปแบบ Json Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 54x API ลำดับที่ 54

ชื่อ API	updateProductBox
ประเภทกลุ่ม	Json Object Manage
ใช้ทำอะไร	ใช้เพิ่มค่า Invest id ใน Json Object Array ตาม ฟิลที่เป็น id (Integer Only)
ใช้เมื่อไหร่	หลังจากเก็บเกี่ยวผลผลิตที่โตเต็มที่
มีโปรแกรมอะไรบ้าง	มีการเพิ่มไอดีแผนการลงทุนลงใน Json Object และนำไปเพิ่มลงใน Json Object Array ที่เป็น buffer
พารามิเตอร์	ระบุ ข้อมูลไอดีตัวละคร และไอดีแผนการลงทุน
ข้อมูลที่ได้รับ	Buffer(New Json Object Array ที่เพิ่มไอดีแผนการลงทุนใหม่)

ตารางที่ 55x API ลำดับที่ 55

ชื่อ API	getInvest4Product
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลแผนการลงทุนในคลังเก็บผลผลิต จากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อต้องการข้อมูลของผลผลิตที่เก็บเกี่ยวแล้ว สำหรับแสดงผลในคลังเก็บผลผลิต
มีโปรแกรมอะไรบ้าง	นำพารามิเตอร์ ไปเช็คเงื่อนไขในการขอข้อมูลผลผลิตในคลังเก็บผลผลิตทั้งหมด
พารามิเตอร์	ระบุ ข้อมูลไอดีตัวละคร
ข้อมูลที่ได้รับ	ได้รับข้อมูลผลผลิตในคลังเก็บผลผลิตของตัวละครที่เลือก จากฐานข้อมูลในรูปแบบ Json Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 56x API ลำดับที่ 56

ชื่อ API	getHistory
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ส่งคำขอ เพื่อขอข้อมูลประวัติการลงทุน จากฐานข้อมูล
ใช้เมื่อไหร่	เมื่อเปิดหน้าต่างโปรไฟล์ เพื่อแสดงผลประวัติการลงทุนคร่าวๆ
มีโปรแกรมอะไรบ้าง	นำพารามิเตอร์ ไปเช็คเงื่อนไขในการขอข้อมูลประวัติการลงทุนทั้งหมด
พารามิเตอร์	ระบุ ข้อมูลไอดีตัวละคร
ข้อมูลที่ได้รับ	ได้รับข้อมูลประวัติการลงทุนของตัวละครที่เลือก จากฐานข้อมูลในรูปแบบ Json Object

ตารางที่ 57x API ลำดับที่ 57

ชื่อ API	IncreasePlayer
ประเภทกลุ่ม	Request
ใช้ทำอะไร	เพิ่มจำนวนผู้ใช้ของเซิร์ฟเวอร์บน ฐานข้อมูล
ใช้เมื่อไหร่	เมื่อมีผู้ใช้ เข้ามาใช้งานในเซิร์ฟเวอร์นั้นๆ
มีโปรแกรมอะไรบ้าง	แก้ไขข้อมูลของ current_user(int) เพิ่มจากเดิม 1 ในตาราง Server_list บนฐานข้อมูล โดยอ้างอิงจาก ip ของเซิร์ฟเวอร์ที่ผู้ใช้ เข้าใช้งาน
พารามิเตอร์	ipaddress
ข้อมูลที่ได้รับ	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 58ข API ลำดับที่ 58

ชื่อ API	DecreasePlayer
ประเภทกลุ่ม	Request
ใช้ทำอะไร	ลดจำนวนผู้ใช้ของเซิร์ฟเวอร์บน ฐานข้อมูล
ใช้เมื่อไหร่	เมื่อมีผู้ใช้ ออกจากโลกเสมือนในเซิร์ฟเวอร์นั้นๆ
มีโปรแกรมอะไรบ้าง	แก้ไขข้อมูลของ current_user(int) ลดจากเดิม 1 ในตาราง Server_list บนฐานข้อมูล โดยอ้างอิงจาก ip ของเซิร์ฟเวอร์ที่ผู้ใช้ เข้าใช้งาน
พารามิเตอร์	ipaddress
ข้อมูลที่ได้รับ	-

ตารางที่ 59ข API ลำดับที่ 59

ชื่อ API	ResetPlayer
ประเภทกลุ่ม	Request
ใช้ทำอะไร	เซตจำนวนผู้ใช้ของเซิร์ฟเวอร์บนฐานข้อมูล เป็น 0
ใช้เมื่อไหร่	เมื่อเซิร์ฟเวอร์เริ่มต้นทำงาน
มีโปรแกรมอะไรบ้าง	แก้ไขข้อมูลของ current_user(int) เป็น 0 ในตาราง Server_list บนฐานข้อมูล โดยอ้างอิงจาก ip ของเซิร์ฟเวอร์ ที่รันอยู่
พารามิเตอร์	ipaddress
ข้อมูลที่ได้รับ	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้