บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ดาวเทียมไทยโชต (Thaichote) หรือ ดาวเทียมธีออส (THEOS) ดาวเทียมสำรวจทรัพยากรดวงแรกของไทยได้ทะยานขึ้นสู่อวกาศ ในวันพุธที่ 1 ตุลาคม 2551 ตามเวลาประเทศไทย 13:37:16 น. หรือ 6.37:16 น. ตามเวลามาตรฐานสากล (UTC) โดยจรวดนำส่ง "เนปเปอร์" (Dnepr) จากฐานส่งจรวดเมืองยาสนี (Yasny) ประเทศรัสเซีย ต่อมาพระบาทสมเด็จพระเจ้าอยู่หัวภูมิพลอดุลยเดช (รัชกาลที่ 9) ทรงพระกรุณาโปรดเกล้าฯ พระราชทานชื่อดาวเทียมสำรวจทรัพยากร THEOS ว่า “ดาวเทียมไทยโชต” ชื่อภาษาอังกฤษว่า “Thaichote” ซึ่งแปลว่า ดาวเทียมที่ทำให้ประเทศไทยรุ่งเรือง

ดาวเทียมไทยโชต ถูกออกแบบให้เป็นดาวเทียมขนาดเล็ก มีอายุการใช้งานอย่างน้อย 5 ปี ทำงานโดยอาศัย แหล่งพลังงาน จากดวงอาทิตย์ สามารถบันทึกภาพได้ครอบคลุมพื้นที่ทั่วโลก ติดตั้งอุปกรณ์ถ่ายภาพแบบ ออฟติคคอล (Optical Imagery) ทำให้สามารถบันทึกข้อมูลภาพ ในช่วงคลื่นแสงที่ตามองเห็น (Visible band) จนถึงช่วงคลื่นอินฟราเรดใกล้ (Near Infrared) ทั้งนี้เมื่อเปรียบเทียบช่วงคลื่นของดาวเทียมธีออสกับดาวเทียม อื่นๆ พบว่า 3 ช่วงคลื่นของดาวเทียมธีออส มีความคล้ายคลึงกับช่วงคลื่นของดาวเทียม SPOT ยกเว้นช่วงคลื่น สีน้ำเงิน ที่มีเพิ่มมากกว่าของดาวเทียม SPOT และมีความคล้ายคลึงกันกับช่วงคลื่นของดาวเทียม Landsat ระบบ TM

ในงานจัยและพัฒนาระบบการตรวจสอบสถานะไฟล์ควบคุมดาวเทียมไทยโชต ทางผู้วิจัยได้ทำการศึกษาข้อมูลและทฤษฎีที่เกี่ยวข้องประกอบไปด้วยภาษารูบี (Ruby), เว็บเซิฟเวอร์ (Web Server), รูบีออนเรลส์ (Ruby On Rails), บูตแสตรบ (Bootstrap), เอเอมคิวพี (AMQP:Advance Message Queuing Protocol), เว็บซ็อกเก็ต (WebSocket) เพื่อนำมาใช้สำหรับการพัฒนาระบบให้มีการทำงานที่ถูกต้องมากที่สุด

ภาษารูบี (Ruby)

ภาษารูบี (Ruby)[1] สร้างโดยโยชิฮิโระ มาซึโมโตะ (Yukihiro Matsumoto) หรือแมตซ์ (Matz) เป็นภาษาที่ถูกพัฒนาขึ้นมาบนพื้นฐานของโอเพนซอร์ส (Open Source) จึงทำให้เป็นภาษาที่ไม่เสียค่าใช้จ่ายและมีคนที่สนใจนำไปใช้งานเป็นจำนวนมากตั้งแต่รุ่นแรกที่ออกมาเมื่อปี 1995 และถูกพัฒนามาอย่างต่อเนื่องมาจนถึงปัจจุบัน จนเป็นภาษาที่มีความนิยมใช้งานเป็นอันดับต้น ๆ และการเขียนของภาษา Ruby เข้าใจง่ายไม่ซับซ้อนรูปแบบการเขียนให้เคียงภาษาซีและภาษาพีเอชพี

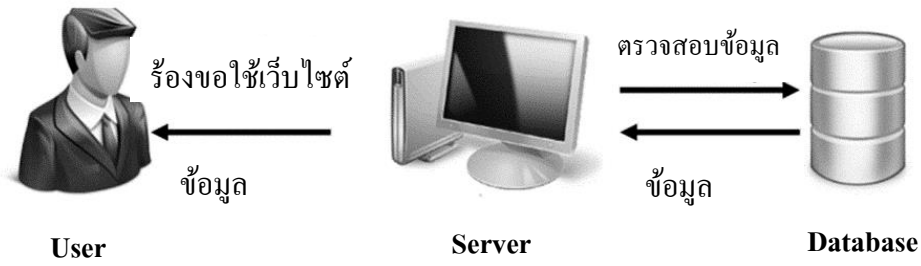
(PHP) ซึ่งหากคนที่มีพื้นฐานการเขียนโปรแกรมน่าจะสามารถเข้ามาศึกษาภาษารูบี (Ruby) ได้ไม่ยากซึ่งลักษณะเด่นของภาษารูบี (Ruby) มีดังนี้

1. ภาษารูบี (Ruby) เป็นภาษาที่มองทุกอย่างเป็นเชิงวัตถุ (Object) ทำให้ใช้งานง่ายต้อความเข้าใจ เพราะทุก ๆ วัตถุนั้นก็จะประกอบไปด้วยคุณสมบัติ (Properties) การแสดงผล (Action) หรือวัตถุและการทำงานขององค์ประกอบนั้น ๆ
2. ภาษารูบี (Ruby) เป็นโอเพนซอร์ส (Open source) ซึ่งไม่เสียค่าใช้จ่ายเมื่อนำภาษารูบี (Ruby) ไปใช้งาน
3. ภาษารูบี (Ruby) สามารถเขียนได้หลากหลายรูปแบบเพื่อแสดงผลลัพธ์ที่เหมือนกัน สามารถเข้าใจง่ายและทำงานได้กับทุกระบบปฏิบัติการ (Operation System)

1. ข้อจำกัดของการใช้งานภาษารูบี (Ruby) เมื่อเทียบกับภาษาอื่น[2]
   1.1. ภาษารูบี (Ruby) มีการชุมชนผู้ใช้งานและการสนับสนุนด้านเทคนิคน้อยกว่าภาษาอื่น
   1.2. มีการประมวลผลที่ช้ากว่าบ้างภาษา อาทิ ภาษาจาวา (Java)
   1.3. มีการพัฒนาและปรับปรุงเวอร์ชันที่ช้ากว่าภาษาพีเอชพี (PHP)
   1.4. ภาษารูบี (Ruby) ไม่ได้รับความนิยมในประเทศไทยอาจมีข้อจำกัดในการศึกษาและพัฒนาต่อยอด

การทำงานของเว็บแม่ข่าย (Web Server)

ระบบรายงานสถานะไฟล์ควบคุมดาวเทียมไทยโซต เมื่อผู้ใช้งานเข้าระบบ ซึ่งระบบจะส่งคำร้องขอมายังเครื่องแม่ข่าย และเครื่องมีการตรวจสอบทำงานรูปแบบตามเวลาจริง (Real-Time) เมื่อผู้ใช้งานสามารถบอกสถานการณ์ทำงานแบบเวลาจริงได้ ซึ่งข้อมูลทั้งหมดของการทำงานจะเก็บข้อมูลลงฐานข้อมูล (Database) ซึ่งแสดงโครงสร้างการทำงานตามภาพที่ 2.1 โครงสร้างการทำงานของเว็บแม่ข่าย



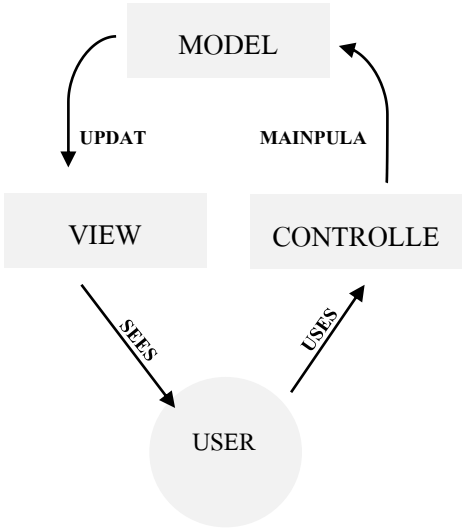
ภาพที่ 2.1 โครงสร้างการทำงานของเว็บแม่ข่าย

1. องค์ประกอบของการออกแบบเว็บแม่ข่าย
   1.1 ความเรียบง่าย (Simplicity) คือการกำหนดและจำกัดองค์ประกอบเสริมให้เหลือเฉพาะองค์ประกอบ เช่นการสื่อสารเนื้อหากับผู้ใช้งานเท่านั้น ถ้าหากใส่สีสันมากเกินไปอาจจะทำให้รบกวนสายตาและสร้างความรำคาญได้
   1.2 ความสม่ำเสมอ (Consistency) คือการสร้างเว็บแม่ข่ายที่สามารถใช้งานได้หลายเครื่องลูกข่ายโดยที่การทำงานไม่มีความผิดพลาดหรือข้อมูลที่ส่งให้กับผู้ใช้งานที่ไม่ผิดพลาด
   1.3 ความเป็นเอกลักษณ์ (Identity) คือการออกแบบเว็บแม่ข่าย ต้องคำนึกถึงลักษณะขององค์กรเป็นหลัก เนื่องจากการใช้งานของผู้ใช้งานจะสะท้อนเอกลักษณ์และลักษณะขององค์กร
   1.4 ความสะดวกของการใช้งานในสภาพต่าง ๆ (Compatibility) คือการใช้งานเว็บลูกข่ายนั้นสามารถเข้าถึงภายในองค์ได้สะดวกและไม่มีขอบจำกัดต่าง ๆ กล่าวคือต้องสามารถใช้งานได้ดีกับการเข้าถึงที่ง่าย
   1.5 ไม่ควรมีการติดตั้งโปรแกรมเสริมใด ๆ เพิ่มเติมและรองรับการการใช้งานเว็บบราวเซอร์ได้หลากหลาย เป็นต้น
2. กำหนดเป้าหมายของเว็บเซิร์ฟเวอร์

เป็นขั้นตอนแรกของการออกแบบเว็บเซิร์ฟเวอร์คือการกำหนดเป้าหมายฟังก์ชันการทำงานหน้าตาการใช้งาน หลักการทำงานของผู้ใช้งานให้ชัดเจนก่อน เพื่อจะได้ออกแบบการใช้งานได้ตรงกับเป้าหมายที่ตั้งเอาไว้มากที่สุด และสิ่งที่ผู้ใช้ต้องการจากการใช้งานเซิร์ฟเวอร์หลังจากที่ได้เป้าหมาย ฟังก์ชันการทำงาน หรือหน้าตาการใช้งาน ลำดับต่อมาคือการออกแบบเว็บเซิร์ฟเวอร์เพื่อให้การใช้งานสะดวกและเข้าใจง่ายกับผู้ใช้งาน ด้วยการใช้งานสามารถให้คนส่วนใหญ่เข้าใจ

รูบีออนเรียลส์ (Ruby on Rails)

รูบี เจม (Ruby Gem) หรือไลบรารี่ (Library) ที่ใช้ในการเขียนเว็บผ่านคอมแม้นไลน์อินเตอร์เฟซ(Command Line Interface: CLI) ซึ่งถูกสร้างโดยเดวิด เฮเนเมยร์ แฮนส์สัน (David Heinemeier Hansson)[3] ซึ่งสามารถเรียกสั้น ๆ ว่าเรียลส์ (Rails)



ภาพที่ 2.2 MVC Model

เรียลส์ (Rails) เองนั้นเป็นเจม (Gem) ตัวหนึ่งของภาษารูบี (Ruby) ซึ่งเป็นเจม (Gem) ที่มีความสามารถสูงมากในการพัฒนาเว็บไซต์ โดยเรียลส์ (Rails) เองมีแนวคิดในการสร้างเว็บแบบเอมวีซี (Model-View-Controller: MVC)[4] โดยโมเดล (Model) จะเป็นคลาส (Class) ของตารางบนฐานข้อมูลในส่วนของวิว (View) จะเป็นหน้าที่แสดงให้ผู้ใช้เห็นเป็นภาษาเอชทีเอมแอล (HTML) และคอนโทรเลอร์ (Controller) ก็จะเป็นคลาส (Class) ที่เป็นตัวเชื่อมระหว่างโมเดล (Model) กับวิว (View) นั้นเองซึ่งแบบเอมวีซี (MVC) แสดงดังภาพที่ 2.2 นอกจากเรียลส์ (Rails) จะสามารถพัฒนาเว็บไซต์ได้แล้วเรียลส์ (Rails) ยังสามารถเชื่อมต่อกับฐานข้อมูลชนิดใด ๆ ก็ได้ผ่านเจม (Gem) ชื่อแอคทีพ เรคคอร์ด (Active Record) จึงทำให้ผู้ที่เรียลส์ (Rails) สามารถเลือกใช้ฐานข้อมูลชนิดใดก็ได้ เช่น มายเอสคิลแอล (MySQL), โพสเกรเอสคิลแอล (PostgreSQL) หรือเอสคิลอแลไลต์ (SQL Lite) ได้อีกทั้งยังมีความสามารถในการจัดการฐานข้อมูลผ่านไมเกรชั่น (Migration) ได้เพื่อเปลี่ยนแปลงคอลั่ม (Column) ของตารางบนฐานข้อมูลได้อย่างง่าย ไม่ยุ่งยากในขั้นตอนการดีพลอย (Deploy) ซึ่งในระบบใช้เจม (Gem) หรือไลบรารี่ (Library) ของเรียลส์ (Rails) ที่ชื่อว่าเจม ลิสซึน (Gem listen) เป็นเจม (Gem) ที่ทำหน้าที่ตรวจจับเหตุการณ์ในไดเรคทอรี่ (Directory) ที่กำหนดไว้ เช่น ภายในไดเรคทอรี่ (Directory) มีการ เพิ่ม ลบ หรือแก้ไขไฟล์ในไดเรคทอรี่ (Directory)

บูตสแตรป (Bootstrap)

บูสแตรป (Bootstrap)[5] ถูกพัฒนาขึ้นโดยมาร์ค อ๊อตโตะ(MarkOtto) และเจคอบ ธอร์นตัน (Jacob Thornton) ทีมพัฒนาของบริษัททวิตเตอร์ (Twitter Inc.) ก่อนหน้านี้ใช้ชื่อว่าทวิตเตอร์ บลูป

ริ้นท์ (Twitter Blueprint) และเปิดให้นักพัฒนาสามารถนำไปใช้งานพัฒนาเว็บไซต์ได้แบบโอเพนซอร์ส(OpenSource) ในชื่อว่าบูสแตรป เฟรมเวิร์ค (BootstrapFramework)ซึ่งบูสแตรป (Bootstrap) คือชุดคำสั่งที่ประกอบด้วยภาษาซีเอสเอส CSS, เอชทีเอมแอล (HTML)และจาวาสคลิปท์ (JavaScript)เป็นชุดคำสั่งที่ถูกพัฒนาขึ้นมาเพื่อกำหนดกรอบหรือรูปแบบการพัฒนาเว็บไซต์ในส่วนของการปฏิสัมพันธ์กับผู้ใช้งานเว็บไซต์ (User Interface) เราจึงสามารถเรียกบูตสแตรป (Bootstrap) ว่าเป็นฟรอนท์เอ็น เฟรมเวิร์ค (Front-end Framework) คือใช้สำหรับ พัฒนาเว็บไซต์ส่วนการแสดงผลซึ่งแตกต่างจากภาษาประเภทสคลิปต์ฝั่งเซิฟเวอร์ (Server Side Script) อย่างภาษาพีเอชพี (PHP), ไพธอน (Python) หรือภาษาอื่น ๆ

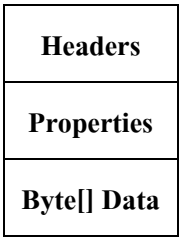
AMQP (Advance Message Queuing Protocol)

AMQP[6] เป็นโพรโตคอลสำหรับส่งข้อมูลโดยมีคุณลักษณะเด่นในด้านการจัดลำดับข้อความ (Queuing) และหาเส้นทาง (Routing) ที่ข้อมูลถูกกำหนดปลายทางให้ส่งไปยังผู้รับเพียงคนเดียวหรือการสื่อสารกันระหว่างเครื่องหลาย ๆ เครื่องที่จะกระจายข้อความไปยังผู้รับทั้งหมดที่อยู่ภายในระบบ อีกทั้งยังรองรับการเข้ารหัสข้อมูลช่วยเพิ่มความปลอดภัยให้กับข้อความมากขึ้น แสดงในตารางที่ 2.1 และ ภาพที่ 2.3 โครงสร้างของเอเอมคิลพี (AMQP) แสดงดังภาพที่ 2.3 ประกอบด้วย 3 ส่วนดังนี้

ตารางที่ 2.1 ตารางโครงสร้างข้อมูลของ Advance Message Queuing Protocol

ชนิดข้อมูล	คำอธิบาย
เฮดเดอร์(Header)	เป็นส่วนที่เก็บ Routing-key ซึ่งถูกกำหนดโดย AMQP ถูกใช้โดย Message Broker เพื่อเชื่อมโยงข้อความที่ส่งเข้ามาว่าข้อความนั้น ๆ เหมาะกับคิวหรือไม่โดยแต่ละคิวจะกำหนด Binding-Key เพื่อใช้สำหรับกำหนดลักษณะของข้อความที่คิวยอมรับซึ่งถ้า Routing-Key ของข้อความตรงกับ Binding-key ของคิวแล้วคิวนั้นก็จะรับข้อความนั้นเข้ามาเก็บไว้เพื่อส่งออกไปยังปลายทาง
คุณสมบัติ(Properties)	เป็นส่วนที่เก็บตัวตั้งค่าและข้อมูลเฉพาะของแอปพลิเคชันที่มีค่าต่างกันไปเนื่องจากแอปพลิเคชันนั้น ๆ จะเป็นตัวกำหนดตัวตั้งค่าเหล่านี้เอง
ข้อมูล (Data Byte[])	เป็นส่วนของข้อความที่ผู้ส่งข้อความต้องการสื่อสารไปยังผู้รับอื่น ๆ

RabbitMQ เป็น Application Server ที่สามารถการกระจายข้อมูลไปยังทุก Client ในระบบได้พร้อม ๆ กัน เช่น Line Server เมื่อส่งข้อความไปให้ผู้ใช้งานปลายทางต้อง Login เข้า Line Application ก่อน

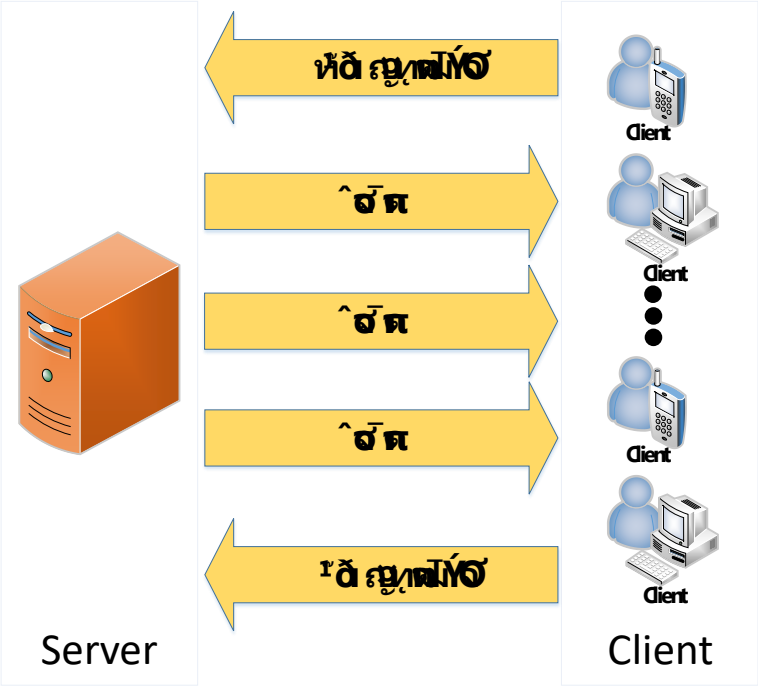


ภาพที่ 2.3 โครงสร้างข้อมูลของ AMQP

เว็บซ็อกเก็ต (Web Socket)

เว็บซ็อกเก็ต (Web Socket)[7] เป็นเทคโนโลยีเพื่อใช้ติดต่อสื่อสารระหว่างเว็บซ็อกเก็ตเซิฟเวอร์ (Web Socket Server) กับปลายทาง (Client) แบบตามเวลาจริง (Real-Time) ซึ่งปลายทาง (Client) จะสามารถรับข้อมูลจากฝั่งเว็บซ็อกเก็ตเซิฟเวอร์ (Web Socket Server) มาแสดงผลโดยที่ไม่ต้องโหลดหน้าเว็บ (Refresh) ซึ่งการทำงานจะคล้าย ๆ เทคโนโลยีเอแจ็กซ์ (Asynchronous JavaScript And XML: Ajax) คือหน้าเว็บสามารถแสดงข้อมูลที่มีการอัพเดทใหม่โดยที่ไม่ต้องโหลดหน้าเว็บ (Refresh) แต่เว็บซ็อกเก็ต (Web Socket) มีข้อดีกว่าเทคโนโลยีเอแจ็กซ์ (Asynchronous JavaScript And XML: Ajax) คือเว็บซ็อกเก็ต (Web Socket) ไม่ต้องร้องขอ (Request) เพื่อส่งคำร้องขอไปยังเซิร์ฟเวอร์และรอรับข้อมูลจากทางฝั่งเซิร์ฟเวอร์โดยการทำงานของเว็บซ็อกเก็ต (Web Socket) ทางฝั่งปลายทาง (Client) จะรอรับข้อความจากทางฝั่งเว็บซ็อกเก็ตเซิฟเวอร์ (Web Socket Server) อยู่ตลอดเวลาหลังจากที่ได้ส่งคำสั่งเชื่อมต่อเว็บซ็อกเก็ตเซิฟเวอร์ (Web Socket Server) ไปแล้วในครั้งแรกและปลายทาง (Client) จะรอรับข้อความจนกระทั่งปิดบราวเซอร์ (Browser) หรือเลิกการเชื่อมต่อไปซึ่งแสดงดังภาพที่ 2.4

โดยระบบรายงานสถานะไฟล์ควบคุมดาวเทียมไทยโชตได้เลือกใช้ซ็อกเก็ตไอโอ (Socket.io) ซึ่งเป็นโมดูล (Module) ที่รองรับการทำงานของเว็บซ็อกเก็ต (Web Socket) และส่งคำสั่งผ่านโนดเจเอส (NodeJS) ซึ่งทำให้การส่งข้อมูลบนเว็บแอปพลิเคชันมีการทำงานแบบเวลาจริง (Real-Time) และสามารถส่งข้อมูลให้กับปลายทาง (Client) ในระบบได้เช่นเดียวกับแรบบิทเอมคิล (RabbitMQ)



ภาพที่ 2.4 โครงสร้างการทำงานของเว็บซ็อกเก็ต (Web Socket)

จากที่ได้เลือกใช้แรบบิทเอมคิล (RabbitMQ) และซ็อกเก็ตไอโอ (Socket.io) กับระบบรายงานสถานะไฟล์ควบคุมดาวเทียมไทยโซต ซึ่งได้ข้อเปรียบเทียบความเหมาะสมในการใช้งานในตารางที่ 2

การเปรียบเทียบแรบบิทเอมคิล (RabbitMQ) และซ็อกเก็ตไอโอ (Socket.io)

ตารางที่ 2.2 ตารางเปรียนเทียมแรบบิทเอมคิล (RabbitMQ) และซ็อกเก็ตไอโอ (Socket.io)

RabbitMQ	Socket.io
- แอพพลิเคชั่นเซิฟเวอร์ (Application Server) เป็นตัวกลางในการกระจายข้อมูล - เป็นการส่งข้อความแบบเป็นลำดับ (Message Queue) - ใช้ทรัพยากรมาก - สามารถจัดการการใช้งานผ่านหน้าเว็บได้ - เหมาะสำหรับแอพพลิเคชั่น (Application)	- เป็นเฟรมเวิร์ค (Framework) สำหรับการสื่อสารแบบเวลาจริง (Real-Time) ทำงานได้ทุกอุปกรณ์ - ความเชื่อถือในการใช้งานเนื่องจากมีการใช้อย่างแพร่หลาย - ใช้ทรัพยากรน้อย - ต้องจัดการจากเครื่องเซิร์ฟเวอร์

ตารางที่ 2.2 ตารางเปรียนเทียมแรบบิทเอมคิล (RabbitMQ) และซ็อกเก็ตไอโอ (Socket.io) (ต่อ)

RabbitMQ	Socket.io
	- เหมาะสำหรับเว็บบราวเซอร์ (Web browser)

ในระบบที่พัฒนาขึ้นนี้ได้เลือกใช้ซ็อกเก็ตไอโอ (Socket.io) มาใช้กับระบบรายงานสถานะไฟล์ควบคุมดาวเทียมไทยโชตเนื่องจากเหมาะกับการทำระบบที่เป็นรูปแบบเว็บเซิร์ฟเวอร์มากกว่าแรบบิทเอมคิล (RabbitMQ) ดังตารางที่ 2.2

งานวิจัยที่เกี่ยวข้อง

สุธี ชูศรี ได้เสนอบทความวิจัยชื่อ ต้นแบบการนำทางระหว่างผู้ใช้อุปกรณ์สื่อสารเคลื่อนที่ด้วยระบบกำหนดตำแหน่งบนโลก A Prototype of Two-way Navigation for Mobile Users with Global Positioning System ซึ่งในงานวิจัยได้นำอัลกอลิทึมมาทดลองหาตำแหน่งในการนำทางแบบเรียลไทม์ เพื่อหาความแม่นนำในการระบุตำแหน่งของผู้ใช้งาน

ชาลี ประจักษ์วงศ์ ได้ทำการศึกษาเรื่อง “ระบบติดตามงาน Task Monitoring System” พัฒนาระบบสั่งงานและติดตามงาน โดยพัฒนาขึ้นในรูปแบบของ Web application ระบบอำนวยความสะดวกแก่ ผู้ใช้ให้สามารถใช้งานได้โดยง่าย รวดเร็ว และตรวจสอบความถูกต้องได้ สามารถใช้งานในสถานที่ต่างๆ ที่สามารถเชื่อมต่อเข้าสู่เครือข่ายอินเตอร์เน็ตได้

รุจาภา เจียมธโนปจัย ได้เสนอบทความวิจัยชื่อ การพัฒนาโปรแกรมระบบการติดตามดูแลผู้ป่วยวัณโรคที่ใช้ระบบยาระยะสั้นภายใต้การสังเกตตรงระดับอำเภอ THE DEVELOPMENT OF MONITORING SYSTEM FOR ATIENTS WITH TUBERCULOSIS UNDER DIRECTLY - OBSERVED TREATMENT SHORT COURSE PROGRAM FOR DISTRICT LEVEL พัฒนาโปรแกรมระบบการติดตามดูแลผู้ป่วยวัณโรคที่ใช้ระบบยาระยะสั้นภายใต้การสังเกตตรง โดยพัฒนาโปรแกรมทางการพยาบาลเรื่องระบบการติดตามดูแลผู้ป่วยวัณโรคทที่ใช้ระบบยาระยะสั้นภายใต้การสังเกตตรงระดับอำเภอโดยใช้กรอบแนวคิดของ Donabedian (1988) และแนวคิดการใช้หลักฐานเชิงประจักษ์ของไอโอวา (Iowa, 2002) และโปรแกรมที่พัฒนาขึ้นโดยการทำ Pilot Study ก่อนนำโปรแกรมไปใช้จริงนอกจากนี้ผู้ ใช้โปรแกรมควรได้รับการฝึกอบรมความรู้เกี่ยวกับการดูแลผู้ป่วยวัณโรคเพื่อให้มีความเข้าใจและเพิ่มทักษะก่อนการใช้โปรแกรม