การปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์

นายศุภชัย หอมพันธุ์

การศึกษาค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช
พ.ศ. 2557

Improvement of Wallboard Display System for Call Center using SignalR Technology

Mr. Supachai Homphun

An Independent Study Submitted in Partial Fulfillment of the Requirements for

the Degree of Master of Science in Information and Communication Technology

School of Science and Technology

Sukhothai Thammathirat Open University

2014

หัวข้อการศึกษาค้นคว้าอิสระ	การปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์
ชื่อและนามสกุล	นายศุภชัย หอมพันธุ์
แขนงวิชา	เทคโนโลยีสารสนเทศและการสื่อสาร
สาขาวิชา	วิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร. วิภา เจริญภัณฑารักษ์

การศึกษาค้นคว้าอิสระนี้ ได้รับความเห็นชอบให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรระดับปริญญาโท เมื่อวันที่ 28 สิงหาคม 2558

คณะกรรมการสอบการศึกษาค้นคว้าอิสระ

..... ประธานกรรมการ

(รองศาสตราจารย์ ดร.วิภา เจริญภัณฑารักษ์)

..... กรรมการ

(อาจารย์ ดร.ดวงดาว วิชาดากุล)

.....

(รองศาสตราจารย์ณัฏฐพร พิมพายน)

ประธานกรรมการประจำสาขาวิชาวิทยาศาสตร์และเทคโนโลยี

ชื่อการศึกษาค้นคว้าอิสระ การปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์

ผู้ศึกษา นายศุภชัย หอมพันธุ์ **รหัสนักศึกษา** 2559600420 **ปริญญา** วิทยาศาสตรมหาบัณฑิต
อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร. วิภา เจริญภัณฑารักษ์ **ปีการศึกษา** 2557

บทคัดย่อ

บทความของโครงการนี้นำเสนอการพัฒนาต้นแบบการปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์และการประเมินประสิทธิภาพของต้นแบบที่ได้พัฒนาขึ้น

การปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์ใช้ไลบารี่ของซิกแนลอาร์ทำงานร่วมกับเอเอสพีและเอเอสพีดอทเน็ตเอ็มวีซีประกอบไปด้วย (1) โมเดล (Model) ทำหน้าที่สร้างคลาสสำหรับสำหรับเก็บข้อมูลคอลเซ็นเตอร์จากฐานข้อมูล (2) วิว (View) ทำหน้าที่แสดงผลข้อมูลหน้าจอข้อมูลคอลเซ็นเตอร์ (3) คอนโทรลเลอร์ (Controller) ทำหน้าที่เชื่อมต่อการทำงานระหว่างฮับกับฟังก์ชั่นที่ถูกเรียกใช้งานโดยไคลเอนต์

ผลจากการดำเนินโครงการนี้เป็นการนำหลักการของสถาปัตยกรรมซิกแนลอาร์มาพัฒนาระบบแสดงข้อมูลคอลเซ็นเตอร์แบบเรียลไทม์ซึ่งสามารถประหยัดทรัพยากรของเครื่องเซิร์ฟเวอร์ได้มากกว่าเมื่อเทียบกับระบบเดิมที่ใช้อยู่ พบว่า ซิกแนลอาร์สามารถลดการทำงานของซีพียูในเครื่องเซิร์ฟเวอร์ 45% ลดการรับส่งข้อมูล 70% และได้รับข้อมูลในรูปแบบเวลาจริง แสดงให้เห็นว่าระบบที่พัฒนาขึ้นมีประสิทธิภาพโดยรวมอยู่ในระดับดี และจากประสิทธิภาพของระบบทำให้ศูนย์ข้อมูลคอลเซ็นเตอร์ได้รับข้อมูลแบบเรียลไทม์ ซึ่งสามารถนำข้อมูลที่ได้รับมาพัฒนางานในทีมเพื่อให้สามารถให้บริการลูกค้าได้อย่างรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น

คำสำคัญ ระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ เทคโนโลยีซิกแนลอาร์

Independent Study title: Improvement of Wallboard Display System for Call Center Using SignalR Technology
Author: Mr. Supachai Homphun; **ID:** 2559600420;
Degree: Master of Science (Information and Communication Technology)
Independent Study advisor: Dr. Vipa Jaroenpuntaruk, Associate Professor;
Academic year: 2014

Abstract

This research paper proposes Improvement of Wallboard Display System for Call Center using SignalR Technology new technique to enhance the presentation of data related to real time.

Improving the efficiency of the Wallboard display system information of call center with SignalR technology. Using a library of SignalR working with SSP and SSP.NETMVC consisting of (1) Model the function is generates a class for the data from database (2) View is the display screen of the call center and (3) Controller is acting connection between the hub and the function that are used by the client

According to the research methodology, the principles of architecture SignalR use real-time system can save server resources than compared to existing systems. SignalR can save CPU 45% and send-receive data 70% make a call Center Information can be obtained from development work in teams to serve customers quickly and more effectively.

Keywords : Improvement of Wallboard Display System for Call Center using SignalR Technology

กิตติกรรมประกาศ

การศึกษาค้นคว้าอิสระฉบับนี้สำเร็จได้ ด้วยความกรุณาของ รศ.ดร.วิภา เจริญภัณฑารักษ์ อาจารย์ที่ปรึกษาโครงการซึ่งได้ให้คำปรึกษา ข้อชี้แนะ และความช่วยเหลือในหลายสิ่งหลายอย่าง จนกระทั่งลุล่วงไปได้ด้วยดี ผู้จัดทำโครงการขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ขอกราบขอบพระคุณ คณาจารย์สาขาวิชาเทคโนโลยีสารสนเทศมหาวิทยาลัยสุโขทัยธรรมาธิราช ทุกท่านที่ได้ให้ความรู้ ให้คำแนะนำให้กำลังใจ

ขอบคุณและขอบใจ พี่ เพื่อน และน้องสาขาวิชาเทคโนโลยีสารสนเทศทุกคน ที่คอยถามไถ่ด้วยความห่วงใย รวมถึงผู้มีพระคุณทุกท่านที่มิได้เอ่ยนามไว้ ณ ที่นี้

ศุภชัย หอมพันธุ์
สิงหาคม 2558

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญตาราง ..... ฌ

สารบัญภาพ ..... ฎ

บทที่ 1 บทนำ 1

ความเป็นมาและความสำคัญของปัญหา 1

วัตถุประสงค์ 2

ขอบเขตของการดำเนินโครงการ 2

ขั้นตอนการดำเนินโครงการ 3

ประโยชน์ที่คาดว่าจะได้รับ 4

นิยามศัพท์เฉพาะ 4

บทที่ 2 วรรณกรรมที่เกี่ยวข้อง 7

ระบบ Call Center 7

ความหมายของ SignalR 10

การทำงานร่วมกันของ SignalR ร่วมกับ WebSocket 12

วิธีการทำงานของ SignalR ร่วมกับ ฮับ 14

ความต้องการพื้นฐานของง SignalR 15

การกำหนดค่า URL SignalR 17

ความปลอดภัยของ SignalR 18

ความหมายของระบบเรียลไทม์ (Real-time) 21

SQL server 2012 23

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีดำเนินการวิจัย 26
ขั้นตอนการดำเนินโครงการ 26
เครื่องมือที่ใช้ในการดำเนินโครงการ 27
บทที่ 4 การออกแบบและพัฒนาระบบ 28
ข้อมูลระบบแบบดั้งเดิม 28
การร้องของข้อมูลของระบบดั้งเดิม 29
ขั้นตอนการพัฒนาระบบใหม่ด้วยเทคโนโลยี SignalR 30
การวิเคราะห์และออกแบบระบบใหม่ 33
บทที่ 5 ผลการดำเนินโครงการ 53
การประเมินผล 53
ผลการประเมิน 53
บทที่ 6 อภิปรายผลและข้อเสนอแนะ 60
อภิปรายผล 60
ข้อเสนอแนะ 61
บรรณานุกรม 62
ประวัติผู้ศึกษา 67

สารบัญตาราง

หน้า

ตารางที่ 2.1 ตารางความต้องการ Transport Layer ของเว็บเบราว์เซอร์ 16
ตารางที่ 2.2 ตารางความต้องการชั้นการสื่อสารนำส่งข้อมูลของ
ระบบปฏิบัติการวินโดว์และ Silverlight 17
ตารางที่ 4.1 ตารางแสดงความหมายและรายละเอียดต่างๆ ของหน้าจอ Call center Dashboard
ส่วนการแสดงสายโทรเข้า 36
ตารางที่ 4.2 ตารางแสดงความหมายและรายละเอียดต่างๆ ของหน้าจอ Call center Dashboard
ส่วนการแสดงหมายเลขที่ให้บริการ 36
ตารางที่ 4.3 ตารางแสดงความหมายและรายละเอียดต่างๆ ของหน้าจอ Call center Dashboard
ส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงาน 37
ตารางที่ 4.4 ตารางแสดงความหมายและรายละเอียดต่างๆ ของหน้าจอ Call center Dashboard
ส่วนของการแสดงข้อมูลพนักงานที่กำลังใช้สายและพร้อมรับสาย 37
ตารางที่ 4.5 ตารางแสดงการออกแบบฐานข้อมูลตาราง Agent (พนักงาน) 38
ตารางที่ 4.6 ตารางแสดงการออกแบบฐานข้อมูลตาราง CallType (สถานะสายสนทนา) 38
ตารางที่ 4.7 ตารางแสดงการออกแบบฐานข้อมูลตาราง Chanel (ช่องทางติดต่อ) 39
ตารางที่ 4.8 ตารางแสดงการออกแบบฐานข้อมูลตาราง CallTransaction
(ประวัติสาย-สนทนา) 39
ตารางที่ 4.9 ตารางแสดงความหมายของคลาส ChangeSource 49
ตารางที่ 4.10 ตารางแสดงความหมายของคลาส ChangeInfo เป็นเหตุการณ์ที่เกี่ยวข้องกับ
การเชื่อมต่อข้อมูลระหว่างอุปกรณ์เครื่องเซอร์ฟเวอร์
ไปยังเซอร์ฟเวอร์ฐานข้อมูล 50
ตารางที่ 4.11 ตารางแสดงความหมายของคลาส ChangeType เป็นเหตุการณ์ที่เกี่ยวข้องกับประเภท
การเปลี่ยนแปลง 51
ตารางที่ 5.1 ตารางผลการประเมินระบบปัจจุบัน โดยการเปิดหน้าจอแสดงผลข้อมูล 1
หน้าจอเป็นเวลา 10 นาที 54
ตารางที่ 5.2 ตารางผลการประเมินระบบปัจจุบัน โดยการเปิดหน้าจอแสดงผลข้อมูล 3 หน้าจอ
พร้อมกันเวลาประเมิน 10 นาที 55

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 5.3 ตารางผลการประเมินระบบใหม่ที่ใช้เทคโนโลยี SignalR โดยการเปิดหน้าจอแสดงผลข้อมูล 1 เวลาประเมิน 10 นาที 56
ตารางที่ 5.4 ตารางผลการประเมินระบบใหม่ที่ใช้เทคโนโลยี SignalR โดยการเปิดหน้าจอแสดงผลข้อมูล 3 หน้าจอพร้อมกัน เวลาประเมิน 10 นาที 57

สารบัญภาพ

หน้า

ภาพที่ 2.1 รูปภาพแสดงการทำงานของ SignalR 11
ภาพที่ 2.2 ส่วนประกอบของ ASP.NET SignalR โดยที่ทางฝั่งเซิร์ฟเวอร์จะติดต่อผ่านสิ่งที่เรียกว่า Hub และ Hub Proxy 12
ภาพที่ 2.3 รูปภาพแสดงโทเด็นการเชื่อมต่อของ SignalR 19
ภาพที่ 2.4 รูปภาพแสดงสถานการณ์ร้องของการเชื่อมต่อของ Client ไปยัง Server เพื่อการตรวจสอบสิทธิ์การเข้าถึง 20
ภาพที่ 3.1 รูปภาพแสดงการสร้างโปรแกรมด้วย ASP.NET MVC Web Application 27
ภาพที่ 4.1 รูปภาพแสดงการทำงานของระบบดั้งเดิม 28
ภาพที่ 4.2 แสดงการทำงานของโปรแกรม Resource Monitor ที่ตรวจจับการร้องขอข้อมูลของระบบดั้งเดิม 30
ภาพที่ 4.3 รูปภาพแสดงการส่งผ่านข้อมูลที่มีการปรับปรุงในระบบฐานข้อมูล 32
ภาพที่ 4.4 รูปภาพแสดงการรับและส่งข้อมูลของ SignalR 33
ภาพที่ 4.5 รูปภาพรวมของหน้าจอ Call center Dashboard 34
ภาพที่ 4.6 รูปภาพหน้าจอส่วนการแสดงสายโทรเข้า 34
ภาพที่ 4.7 รูปภาพหน้าจอส่วนการแสดงหมายเลขที่ให้บริการ 35
ภาพที่ 4.8 รูปภาพหน้าจอส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงาน 35
ภาพที่ 4.9 รูปภาพหน้าจอส่วนของการแสดงข้อมูลพนักงานที่กำลังใช้สายและพร้อมรับสาย 36
ภาพที่ 4.10 รูปภาพแสดงรายการตารางฐานข้อมูล 40
ภาพที่ 4.11 รูปภาพแสดงรายการคลาสที่ใช้ภายในระบบ 40
ภาพที่ 4.12 รูปภาพแสดง Sequence diagram ของการเรียกใช้งานหน้าจอส่วนที่ 1 ส่วนการแสดงสายโทรเข้า 41
ภาพที่ 4.13 รูปภาพแสดง Sequence diagram ของการเรียกใช้งานหน้าจอส่วนที่ 2 ส่วนการแสดงหมายเลขที่ให้บริการและส่วนที่ 3 ส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงาน 42
ภาพที่ 4.14 รูปภาพแสดง Sequence diagram ของการเรียกใช้งานหน้าจอส่วนที่ 3 ส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงานและส่วนที่ 4 ส่วนของการแสดงข้อมูลพนักงานที่กำลังใช้สายและพร้อมรับสาย 43

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 5.1 รูปภาพแสดงการตรวจวัดค่า Network จากโปรแกรม Monitor Network Internet Explorer เวอร์ชัน 11 54
ภาพที่ 5.2 รูปภาพแสดงการตรวจวัดค่า CPU จากโปรแกรม Resource Monitor ของระบบปัจจุบัน 55
ภาพที่ 5.3 รูปภาพแสดงการตรวจวัดค่า Network จากโปรแกรม Monitor Network Internet Explorer เวอร์ชัน 11 จำนวน 3 หน้าจอ เวลาประเมิน 10 นาที 55
ภาพที่ 5.4 รูปภาพแสดงการตรวจวัดค่า CPU จากโปรแกรม Resource Monitor ของระบบใหม่ที่พัฒนาด้วย SignalR 56
ภาพที่ 5.5 รูปภาพแสดงการตรวจวัดค่า Network จากโปรแกรม Monitor Network Internet Explorer เวอร์ชัน 11 จำนวน1 หน้าจอ เวลาประเมิน 10 นาที 57
ภาพที่ 5.6 รูปภาพแสดงการตรวจวัดค่า CPU จากโปรแกรม Resource Monitor ของระบบใหม่ที่พัฒนาด้วย SignalR จำนวน 3 หน้าจอ 58
ภาพที่ 5.7 รูปภาพแสดงการตรวจวัดค่า Network จากโปรแกรม Monitor Network Internet Explorer เวอร์ชัน 11 จำนวน3 หน้าจอ เวลาประเมิน 10 นาที 58
ภาพที่ 5.8 รูปภาพแสดงรูปแบบ URL ของระบบปัจจุบันโดยการเรียกใช้ข้อมูลผ่าน Controller และ Action Class 59
ภาพที่ 5.9 รูปภาพแสดงรูปแบบ URL ของระบบใหม่ที่พัฒนาด้วย SignalR จะการใช้รูปแบบ URL แบบ Long polling 59
ภาพที่ 6.1 รูปภาพแสดงรูปแบบ URL ของระบบปัจจุบันโดยการเรียกใช้ข้อมูลผ่าน Controller และ Action Class 60
ภาพที่ 6.2 รูปภาพแสดงรูปแบบ URL ของระบบใหม่ที่พัฒนาด้วย SignalR จะการใช้รูปแบบ URL แบบ Long polling 60

บทที่ 1
บทนำ

1. ความเป็นมาและความสำคัญของปัญหา

Call Center เป็นช่องทางการติดต่อสื่อสารที่ประหยัด สะดวก และรวดเร็วที่สุดในการทำธุรกรรม เมื่อลูกค้าต้องการใช้บริการ หรือสอบถามข้อมูล ส่วนใหญ่ก็จะใช้โทรศัพท์ในการติดต่อสื่อสารกัน เมื่อลูกค้าได้รับข้อมูลที่ตรงกับความต้องการและประทับใจในการให้บริการก็จะนำไปสู่การใช้บริการในครั้งต่อๆ ไป ซึ่ง Call Center หมายถึงจุดบริการหลัก ที่สามารถดำเนินการตามความต้องการของลูกค้าได้อย่างมีประสิทธิภาพ ส่งเสริมทั้งภาพลักษณ์และการบริการที่น่าชื่นชมเป็นอย่างยิ่ง

ปัจจุบันการทำงานของระบบ Call Center จำเป็นต้องนำเสนอข้อมูลสถานการณ์ทำงานบนหน้าจอ Wallboard มีคุณสมบัติรองรับการแสดงข้อมูลสถานการณ์ทำงานบนหน้าจอขนาดใหญ่ (Wallboard System) สำหรับส่วนกลาง เพื่อแสดงข้อมูลสถานะของการทำงานของ Call Center บนหน้าจอคอมพิวเตอร์ และสามารถต่อเชื่อมเป็นชนิดจอภาพขนาดใหญ่ (Wall Display) ให้ฝ่ายบริหารจากส่วนกลาง สามารถตรวจสอบสถานะของการทำงานของเจ้าหน้าที่รับสาย (Agent) ได้ในรูปแบบ Real Time Monitoring โดยสามารถแสดงผลข้อมูลได้หลากหลาย ดังนี้

- Calls Waiting: แสดงจำนวนสายที่รออยู่ในคิว
- Calls Serviced Now: แสดงจำนวนสายที่กำลังสนทนาขณะนั้น
- Average Waiting Time: แสดงระยะเวลารอสายเฉลี่ยจากทุกสายที่อยู่ในคิว
- Total Calls: แสดงจำนวนสายที่เข้าทั้งหมด
- Answered: แสดงจำนวนสายที่ได้รับ
- Abandoned: แสดงจำนวนสายที่ไม่ได้รับ
- Total Talking Time: เวลาที่ใช้ในการสนทนาทั้งหมด

การนำเสนอข้อมูลข้อมูลสถานการณ์ทำงานบนหน้าจอ Wallboard ของระบบปัจจุบันมีการใช้เทคโนโลยีแบบดั้งเดิม คือการใช้ระบบที่มีการปรับปรุงข้อมูลจากการจับเวลาหน่วยเป็นวินาทีที่ให้ระบบต้องเข้าไปดาวน์โหลดข้อมูลทุกๆ 1 วินาที โดยการเรียกใช้งานผ่านเทคโนโลยีเอแจ็กซ์ (AJAX: Asynchronous JavaScript and XML) เป็นกลุ่มของเทคนิคในการพัฒนาเว็บแอปพลิเคชัน

เพื่อให้ความสามารถโต้ตอบกับผู้ใช้ได้ดีขึ้น โดยการรับส่งข้อมูลในฉากหลัง ทำให้ทั้งหน้าไม่ต้องโหลดใหม่ทุกครั้งที่มีการเปลี่ยนแปลง ซึ่งช่วยทำให้เพิ่มการตอบสนอง ความรวดเร็ว และการใช้งานโดยรวม แต่เทคโนโลยีดังกล่าวเป็นสาเหตุให้เครื่องเซิร์ฟเวอร์ทำงานหนักเนื่องจากในบางครั้งมีจำนวนเครื่อง Client เปิดเข้าใช้งานระบบพร้อมกันเป็นจำนวนมากก็จะยิ่งทวีคูณการทำงานของเครื่องแม่ข่าย และอาจเป็นสาเหตุที่ทำให้ระบบล่มได้

ดังนั้นผู้ดำเนินโครงการจึงคิดว่าน่าจะมีระบบที่สามารถแสดงผลข้อมูลได้แบบทันที (Real Time) และเพิ่มประสิทธิภาพให้กับ เครื่องลูกข่ายและเครื่องแม่ข่ายจึงได้นำเสนอเทคโนโลยีของ .Net ASP SignalR ที่ทำงานได้แบบ Real Time โดยการให้ซิกแนลอาร์ทำหน้าที่ตรวจสอบการเคลื่อนไหวข้อมูลในระบบฐานข้อมูลและทำการส่งข้อความแบบ broadcast message ไปยัง User ที่ใช้งานระบบอยู่ และยังสามารถประหยัดทรัพยากรบนเครื่องแม่ข่ายได้อีก เนื่องจากไม่จำเป็นต้องเชื่อมต่อกับเซร์ฟเวอร์ทุกๆ 1 วินาที แต่จะเป็นการตรวจจับการเพิ่มข้อมูล การปรับปรุงข้อมูลในระบบฐานข้อมูลแทน

2. วัตถุประสงค์

การดำเนินงานในครั้งนี้เป็นการศึกษาเกี่ยวกับ ASP.NET SignalR ซึ่งเป็นหนึ่งในไลบรารี่ใหม่ที่ใช้ในการพัฒนาระบบที่มีรูปแบบการติดต่อสื่อสารแบบเรียลไทม์ เพื่อเป็นข้อมูลสำหรับประกอบการพัฒนาระบบต้นแบบซอฟต์แวร์ที่สามารถแสดงผลข้อมูลได้แบบทันที เพื่อให้ได้มาซึ่งวัตถุประสงค์ของการดำเนินโครงการในครั้งนี้คือ

2.1 เพื่อพัฒนาต้นแบบระบบการแสดงข้อมูลการใช้งานโทรศัพท์ด้วยเทคโนโลยีซิกแนลอาร์

2.2 เพื่อประเมินต้นแบบระบบการแสดงผลแบบ Real-time ด้วยเทคโนโลยี SignalR

3. ขอบเขตของการดำเนินโครงการ

3.1 ต้นแบบที่พัฒนา จะประกอบไปด้วย

- ส่วนการแสดงสายโทรเข้า
- ส่วนการแสดงหมายเลขที่ให้บริการ
- ส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงาน
- ส่วนของการแสดงข้อมูลพนักงานที่กำลังใช้สายและพร้อมรับสาย

3.2 การประเมินประสิทธิภาพของระบบการแสดงผลข้อมูลแบบเรียลไทม์

การประเมินประสิทธิภาพโดยใช้ซอฟต์แวร์ประยุกต์การดูการทำงานของเครื่องเซิร์ฟเวอร์ จากระบบต้นแบบด้วยการเปรียบเทียบกับผลกับซอฟต์แวร์เดียวกันที่ใช้เทคโนโลยีแบบดั้งเดิมโดยงการทดสอบประสิทธิภาพดังนี้

3.2.1 การใช้งานของ CPU น้อยลง

3.2.2 การใช้งาน Send / Receive ข้อมูลน้อยลง

3.2.3 หน้าจอสามารถแสดงข้อมูลได้แบบ Real-time

3.3 ระบบต้นแบบที่พัฒนาขึ้นจะเน้นการแสดงผลข้อมูลแบบเรียลไทม์โดยใช้เทคโนโลยี SignalR เป็นตัวตรวจสอบข้อมูลที่มีการปรับปรุงในระบบฐานข้อมูลเท่านั้น

4. ขั้นตอนการดำเนินโครงการ

ขั้นตอนและวิธีการการดำเนินโครงการเรื่อง การปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์ มีรายละเอียดดังต่อไปนี้

4.1 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง ซึ่งได้แก่

4.1.1 ประเด็นปัญหาและวิธีการพัฒนาปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์ รวมถึงเทคโนโลยีที่เกี่ยวข้อง

4.1.2 กระบวนการแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์

4.1.3 รายละเอียด ลักษณะเฉพาะ และโครงสร้างข้อมูลของระบบคอลเซ็นเตอร์

4.1.4 รายละเอียดและวิธีการประยุกต์ใช้เทคโนโลยีซิกแนลอาร์

4.1.5 งานวิจัยและระบบที่เกี่ยวข้องกับการใช้เทคโนโลยีซิกแนลอาร์

4.2 วางแผนการดำเนินโครงการและพัฒนาปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์

4.3 ศึกษาและออกแบบวิธีการใช้งานเทคโนโลยีซิกแนลอาร์

4.4 เลือกเครื่องมือและพัฒนาระบบตามที่ได้ออกแบบไว้

4.4.1 ศึกษาแนวทางการปรับปรุงระบบแสดงผลข้อมูลของคอลเซ็นเตอร์จากระบบที่ใช้เทคโนโลยีแบบดั้งเดิมเป็นการใช้เทคโนโลยีใหม่

4.4.2 ปรับปรุงระบบแสดงผลข้อมูลของคอลเซ็นเตอร์จากระบบที่ใช้เทคโนโลยีแบบดั้งเดิมเป็นการใช้เทคโนโลยีใหม่

4.5 ทำการประเมินประสิทธิภาพระบบใหม่ด้วยซอฟต์แวร์ประยุกต์ การตรวจสอบการใช้งานทรัพยากรเครื่องเซิร์ฟเวอร์จากระบบต้นแบบเพื่อประเมินการใช้ทรัพยากรของเครื่องเซิร์ฟเวอร์จากระบบใหม่เปรียบเทียบกับระบบดั้งเดิม

4.6 จัดทำบทความวิชาการ

4.7 สรุปผลแนวทางการดำเนินโครงการและข้อเสนอแนะและจัดทำเล่มโครงงาน

5. ประโยชน์ที่คาดว่าจะได้รับ

5.1 ได้ระบบต้นแบบที่แสดงผลข้อมูลของคอลเซ็นเตอร์แบบเรียลไทม์ที่มีการใช้ทรัพยากรเครื่องเซิร์ฟเวอร์อย่างคุ้มค่า

5.2 ประหยัดทรัพยากรของเครื่อง Server เนื่องจากมีการใช้งาน CPU น้อยลง

5.3 ผู้ใช้งานสามาถนำข้อมูลที่ได้รับแบบ real-time ไปใช้งานเพื่อจัดสรรทรัพยากรบุคคลให้สามารถรับสายได้ทันเพื่อสร้างความประทับใจในการให้บริการกับลูกค้า

6. นิยามศัพท์เฉพาะ

6.1 SignalR : เทคโนโลยีประเภทโอเพนซอร์ทที่สนับสนุนการสื่อสารแบบ Real-time

6.2 Hub : ตัวกลางในการรับส่งข้อมูลระหว่างเครื่องเซิร์ฟเวอร์และเครื่อง Client ทำการบรรจุข้อมูลและพารามิเตอร์ในรูปแบบ Json เพื่อส่งข้อมูลไปกลับ เช่นพารามิเตอร์ A เก็บข้อมูลชื่อ Hub พารามิเตอร์ A เก็บข้อมูล Method การทำงาน พารามิเตอร์ B เก็บข้อความจากแอฟลิเคชั่น

6.3 Web Socket : เป็น API ที่ต่อจาก AJAX เทคนิคคือการ push ข้อมูลจากเซิร์ฟเวอร์มายังไคลเอนต์ เนื่องจากเทคนิคการส่งขอ้มูลแบบ HTTP แบบดั้งเดิมจะเปิดการเชื่อมต่อกับเซิร์ฟเวอร์เพื่อส่งขอ้มูล แล้วตัดการเชื่อมต่อเมื่อใช้เสร็จ ดังนั้นการขอข้อมูลจากเซิร์ฟเวอร์จึงทำได้ยาก เพราะต้องดึงข้อมูลจาก เซิร์ฟเวอร์ (polling) ซึ่งเปลืองการโหลดของเซิร์ฟเวอร์โดยเฉพาะกรณีที่ต้องเปิดการเชื่อมต่อ HTTP ค้างเอาไว้ (Long polling หรือ COMET) ดังนั้น WebSockets จึงเป็นเทคโนโลยีที่ช่วยแก้ปัญหานี้ โดยสร้างการ เชื่อมต่อแบบถาวรระหว่างเซิร์ฟเวอร์กับไคลเอนต์ เพื่อให้สองฝั่งส่งข้อมูลกันได้ตลอด โดยอาศัย Protocol TCP

6.4 Long Polling หรือ Asynchronous Polling คือการใช้ประโยชน์จาก HTTP Keep-Alive ซึ่งจะส่งให้เมื่อบราวเซอร์ส่งการร้องขอไปยังเซิร์ฟเวอร์แล้ว เซิร์ฟเวอร์ไม่จำเป็นต้องตอบ

กลับทันที การเชื่อมต่อแบบ HTTP ครั้งนั้นจะอยู่จนกว่าจะหมดอายุ (timeout) หรือจนกว่าเซิร์ฟเวอร์จะมีข้อมูลให้กับบราวเซอร์

6.5 Call Center : จุดบริการหลัก ที่สามารถดำเนินการตามความต้องการของลูกค้าได้อย่างมีประสิทธิภาพ ส่งเสริมทั้งภาพลักษณ์และการบริการที่น่าชื่นชมเป็นอย่างยิ่ง

6.1.1 Calls Waiting: แสดงจำนวนสายที่รออยู่ในคิว

6.1.2 Calls Serviced Now: แสดงจำนวนสายที่กำลังสนทนาขณะนั้น

6.1.3 Average Waiting Time: แสดงระยะเวลารอสายเฉลี่ยจากทุกสายที่อยู่ในคิว

6.1.4 Total Calls: แสดงจำนวนสายที่เข้าทั้งหมด

6.1.5 Answered: แสดงจำนวนสายที่ได้รับ

6.1.6 Abandoned: แสดงจำนวนสายที่ไม่ได้รับ

6.1.7 Total Talking Time: เวลาที่ใช้ในการสนทนาทั้งหมด

6.6 เอแจ็กซ์ (AJAX: Asynchronous JavaScript and XML) เป็นกลุ่มของเทคนิคในการพัฒนาเว็บแอปพลิเคชันเพื่อให้ความสามารถโต้ตอบกับผู้ใช้ได้ดีขึ้น โดยการรับส่งข้อมูลในฉากหลัง ทำให้ทั้งหน้าไม่ต้องโหลดใหม่ทุกครั้งที่มีการเปลี่ยนแปลง ซึ่งช่วยทำให้เพิ่มการตอบสนองความรวดเร็ว และการใช้งานโดยรวม

6.7 เซิร์ฟเวอร์ (server) หรือ เครื่องบริการ หรือ เครื่องแม่ข่าย คือ เครื่องหรือโปรแกรมคอมพิวเตอร์ซึ่งทำงานให้บริการ ในระบบเครือข่ายแก่ลูกข่าย (ซึ่งให้บริการผู้ใช้อีกทีหนึ่ง) เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์นี้ควรจะมีประสิทธิภาพสูง มีความเสถียร สามารถให้บริการแก่ผู้ใช้ได้เป็นจำนวนมาก ภายในเซิร์ฟเวอร์ให้บริการได้ด้วยโปรแกรมบริการ ซึ่งทำงานอยู่บนระบบปฏิบัติการอีกชั้นหนึ่ง

6.8 เครื่องลูกข่าย หรือ ไคลเอนต์ (client) เป็นระบบหรือแอปพลิเคชันที่สามารถเชื่อมต่อเข้ากับระบบคอมพิวเตอร์อื่นที่เรียกว่าเซิร์ฟเวอร์ได้ คำว่าไคลเอนต์เริ่มมีการใช้เรียกถึงคอมพิวเตอร์ที่ไม่สามารถเรียกใช้งานโปรแกรมคอมพิวเตอร์ในตัวเองได้ แต่สามารถใช้งานโปรแกรมนั้นผ่านทางระบบเครือข่าย

6.9 เอชทีเอ็มแอล (HTML: Hypertext Markup Language ภาษามาร์กอัปข้อความหลายมิติ) เป็นภาษามาร์กอัปหลักในปัจจุบันที่ใช้ในการสร้างเว็บเพจ หรือข้อมูลอื่นที่เรียกดูผ่านทางเว็บเบราว์เซอร์ ซึ่งตัวโค้ดจะแสดงโครงสร้างของข้อมูล ในการแสดง หัวข้อ ลิงก์ ย่อหน้า รายการ รวมถึงการสร้างแบบฟอร์ม เชื่อมโยงภาพหรือวิดีโอด้วย โครงสร้างของโค้ดเอชทีเอ็มแอลจะอยู่ในลักษณะภายในวงเล็บสามเหลี่ยม

6.10 เจซัน (JSON: JavaScript Object Notation เป็นฟอร์แมตสำหรับแลกเปลี่ยนข้อมูลคอมพิวเตอร์ ฟอร์แมต JSON นั้นอยู่ในรูปข้อความธรรมดา (plain text) ที่ทั้งมนุษย์และโปรแกรมคอมพิวเตอร์สามารถอ่านเข้าใจได้

6.11 ระบบเรียลไทม์ (Realtime) คือระบบที่สามารถให้การตอบสนองจากระบบอย่างทันทีทันใดเมื่อได้รับอินพุตเข้าไป ในทางอุดมคติระบบเรียลไทม์นี้จะเป็นระบบที่ไม่เสียเวลาในการประมวลผลหรืออาจจะกล่าวได้ว่าเวลาในการประมวลเป็นศูนย์ แต่ในทางปฏิบัติเครื่องคอมพิวเตอร์ที่ทำงานแบบเรียลไทม์นี้ไม่สามารถผลิตขึ้นมาได้ เราทำได้เพียงการลดเวลาการประมวลผลให้น้อยที่สุดจนไม่สามารถเห็นความแตกต่างของช่วงเวลาที่ป้อนอินพุตเข้าไปและได้รับเอ้าท์พุทออกมา เวลาของความแตกต่างนี้เรียกว่า “เวลาตอบสนอง” (response time) ซึ่งผู้ใช้งานทั่วไปต้องการเวลาตอบสนองให้น้อยที่สุดเพื่อประสิทธิภาพของระบบ

6.12 โพรโทคอล (protocol) คือข้อกำหนดซึ่งประกอบด้วยกฎต่าง ๆ สำหรับรูปแบบการสื่อสารเฉพาะรูปแบบใดรูปแบบหนึ่ง เพื่อให้การติดต่อสื่อสารในระบบเครือข่าย ทำงานได้ด้วยกันทั้งระบบ คล้ายกับมนุษย์สามารถใช้ภาษาอังกฤษเป็นภาษากลางในการสื่อสารถึงกันได้

6.13 IP Broadcasting คือการส่ง packets ของข้อมูลไปยัง Address พิเศษ ซึ่งถูกกำหนดให้เป็น Broadcast Address ซึ่ง Broadcast Address นี้จะไม่ได้เป็นหมายเลข IP ประจำเครื่องคอมพิวเตอร์ เครื่องใดเครื่องหนึ่ง ดังนั้นการจะส่งข้อมูลไปยัง Broadcast Address ได้นั้นจึงต้องอาศัย UDP เป็น Protocol ใน การสื่อสาร

6.14 พอดแคสต์ (Broadcast) คือการรับส่งข้อมูลแบบ Broadcasting IP Broadcasting นั้นก็คือการส่ง packets ของข้อมูลไปยัง Address พิเศษ ซึ่งถูกกำหนดให้เป็น Broadcast Address ซึ่ง Broadcast Address นี้จะไม่ได้เป็นหมายเลข IP ประจำเครื่องคอมพิวเตอร์ เครื่องใดเครื่องหนึ่ง ดังนั้นการจำส่งข้อมูลไปยัง Broadcast Address เป็นการรับส่งข้อมูลแบบ connectionless จึงใช้ UDP เป็น Protocol ในการสื่อสาร

บทที่ 2
วรรณกรรมที่เกี่ยวข้อง

1. ระบบ Call Center

ระบบ Call Center เป็นช่องทางการติดต่อสื่อสารที่ประหยัด สะดวก และรวดเร็วที่สุดในการทำธุรกรรม เมื่อลูกค้าต้องการใช้บริการ หรือสอบถามข้อมูล ส่วนใหญ่ก็จะใช้โทรศัพท์ในการติดต่อสื่อสารกัน เมื่อลูกค้าได้รับข้อมูลที่ตรงกับความต้องการและประทับใจในการให้บริการก็จะนำไปสู่การใช้บริการในครั้งต่อๆ ไป ซึ่ง Call Center หมายถึงจุดบริการหลัก ที่สามารถดำเนินการตามความต้องการของลูกค้าได้อย่างมีประสิทธิภาพ ส่งเสริมทั้งภาพลักษณ์และการบริการที่น่าชื่นชมเป็นอย่างยิ่ง

Call Center ส่วนใหญ่ถูกจัดตั้งขึ้นเพื่อเพิ่มประสิทธิภาพการให้บริการแก่ลูกค้าผ่านทางโทรศัพท์ โดยเน้นการพัฒนา ปรับปรุงวิธีการในการให้บริการ โดยใช้ความรู้ความสามารถของทรัพยากรบุคคลที่เลือกสรรภายในองค์กร ควบคู่ไปพร้อมกับการใช้ฐานความรู้ในระบบคอมพิวเตอร์ และพร้อมกันนั้นจะคำนึงถึงการให้บริการลูกค้าได้ตลอด 24 ชั่วโมง ตลอดทั้ง 7 วัน รวมไปถึงเพื่อมุ่งเน้นในการลดค่าใช้จ่าย เสริมสร้างประโยชน์ และความประทับใจ จากการให้บริการทั้งทางตรงและทางอ้อม

โดยทั่วไปเราจะเรียกพนักงานรับสายโทรศัพท์ใน Call Center ว่า Agent อาจจะมีตั้งแต่ระดับ 1 คนขึ้นไป จนกระทั่งถึงหลายร้อยคน คอยทำหน้าที่ในการตอบรับสายการโทรเรียกเข้าจากลูกค้า โดยมีวัตถุประสงค์ในการให้ข้อมูลด้านสินค้าและบริการ รับคำสั่งซื้อสินค้า สนับสนุนการให้บริการช่วยเหลือทางด้านเทคนิค รับดำเนินงานที่เกี่ยวข้องกับเรื่องร้องเรียนของลูกค้าหรือกิจกรรมต่างๆ ที่เกี่ยวกับธุรกรรม อย่างครบวงจรแก่ลูกค้า ซึ่งการติดตั้งระบบ Call Center อาจจะถูกติดตั้งในลักษณะแบบ Stand Alone หรือจะมีการเชื่อมโยงกับ Call Center อื่นๆ ที่มีอยู่แล้วก็ได้ โดยสามารถที่จะเชื่อมโยงข้อมูลต่างๆ ได้ทั้งจากระบบ Internet และ LAN

การนำอุปกรณ์เทคโนโลยีทางด้าน Voice Processing ซึ่งได้แก่ ระบบตอบรับโทรศัพท์อัตโนมัติ (Interactive Voice Response : IVR) หรือ Audio Text เข้ามาใช้ จะสามารถช่วยเพิ่มประสิทธิภาพในการให้บริการ ทำให้ลูกค้าได้รับความสะดวกสบายในการรับบริการจากระบบโทรศัพท์ได้

ตลอดเวลา เช่น บริการรับคำสั่งซื้อ รับจองสินค้า/บริการ รับฟังข้อมูลประชาสัมพันธ์หรือแนะนำสินค้า/บริการ เป็นต้น

ดังนั้นศูนย์บริการตอบรับทางโทรศัพท์ หรือ Call Center จึงเป็นการปฏิบัติงานที่รวมงานทางด้านการติดต่อสื่อสาร งานด้านฐานข้อมูล การให้การบริการ และการบริหารงาน ซึ่งในปัจจุบันองค์กรต่างๆ จึงได้นำเอาระบบ Call Center มาเป็นอีกหนึ่งกลยุทธ์ในการพัฒนาปรับปรุงคุณภาพการดำเนินงานอย่างเป็นรูปธรรมและสัมผัสได้

ระบบ Call Center โดยทั่วไปจะมีอุปกรณ์และ Application Software ดังต่อไปนี้

1) PABX ระบบตู้สาขาโทรศัพท์
2) Automatic Call Distribution (ACD)
3) Computer Telephony Integration (CTI)
4) Audio Text / Interactive Voice Response (IVR)
5) Speech Recognition
6) Automate Attendant
7) Voice Mailbox
8) Fax on Demand
9) Fax Broadcast
10) Database Storage, Database Server, PC Workstation
11) CRM Applications Software (Customer Relationship Management)
12) CMS Applications Software (Content management System)
13) DMS Applications Software (Document management System)
14) SMS (Short Message Service)
15) Voice Logger / Voice Recording Unit (VRU)
16) Screen Monitoring
17) Wallboard Display
18) Workforce Management
19) Internet & Web Self-services
20) Telephone Headset
21) Finger Scan
22) IP Camera เป็นต้น

ซึ่งอุปกรณ์ต่างๆ ที่กล่าวมาข้างต้น สามารถเลือกติดตั้งเพื่อให้เหมาะสมกับขนาดงบประมาณและความเหมาะสมกับการใช้งาน Call Center ของแต่ละองค์กรไป

Call Center ส่วนใหญ่จะมี PBX (ตู้สาขาโทรศัพท์) และระบบ Automatic Call Distributor เรียกย่อๆว่า ACD ซึ่งจะเป็นระบบการจัดการและบริหารงานเกี่ยวกับโทรศัพท์ โดยจะทำการโอนสายจากผู้ที่โทรเข้ามา ไปยัง Agent โดยอัตโนมัติ พนักงานจะรับสายในปริมาณเฉลี่ยที่เท่าๆ กัน หรือใกล้เคียงกัน อีกทั้งยังสามารถใช้ประโยชน์จากระบบ ACD ได้โดยการโอนสายของลูกค้าไปยัง Agent ที่มี Special Skill หรือมีความสามารถเฉพาะทางได้ เช่น พนักงานที่สามารถพูดภาษาต่างประเทศได้คล่อง, ผู้เชี่ยวชาญที่ต้องใช้ความรู้พิเศษเฉพาะด้าน หรือโอนสายไปยังกลุ่ม Agent ที่ต้องปฏิบัติงานกับเรื่องนั้นๆ โดยเฉพาะ

ในความพยายามที่จะทำให้การบริการลูกค้ามีประสิทธิภาพมากยิ่งขึ้นนั้น Call Center บางแห่งจึงพิจารณานำระบบที่ใช้การทำงานร่วมกัน ระหว่างระบบโทรศัพท์กับเครื่องคอมพิวเตอร์ (Computer Telephony Integration : CTI) มาใช้ ซึ่งเป็นเทคโนโลยีที่นำความสามารถของระบบ Voice Response เพื่อเชื่อมโยงการทำงานของคอมพิวเตอร์ Network มาผสานกับระบบโทรศัพท์ โดยจะทำการค้นหาและดึงข้อมูลจากฐานข้อมูลที่มีอยู่พร้อมกับการโอนสายไปยังพนักงานรับสาย โดยข้อมูลที่ได้รับจากการเชื่อมโยงระบบโทรศัพท์กับฐานข้อมูลในคอมพิวเตอร์ จะไปปรากฏบนหน้าจอคอมพิวเตอร์ของพนักงานรับสายให้โดยอัตโนมัติ (Screen Pop-up) รวมถึงในเรื่องสถิติ และหมายเลขโทรศัพท์ ของผู้ที่โทรเข้ามาCall Center ในแต่ละวัน (Caller ID)

การเชื่อมโยง CTI นี้ จะมีบทบาทที่สำคัญมากในการทำให้แต่ละขั้นตอนของการทำงานเป็นไปแบบมีประสิทธิภาพ ซึ่งจะสามารถแสดงเลขหมายที่โทรเข้ามายัง Call Center และทำการค้นหาข้อมูลลูกค้าที่ตรงกับหมายเลขโทรศัพท์ที่เรียกเข้ามา เพื่อทำให้ข้อมูลหรือประวัติส่วนตัวของผู้โทรนั้น ปรากฏบนหน้าจอคอมพิวเตอร์ของพนักงานได้โดยอัตโนมัติ

การนำระบบตอบรับโทรศัพท์อัตโนมัติ (Interactive Voice Response : IVR) เข้ามาใช้จะสามารถช่วยเพิ่มประสิทธิภาพในการให้บริการนอกช่วงเวลาทำการ หรือวันหยุดของบริษัทได้ โดยที่องค์กรบางองค์กรไม่สามารถจัดหาพนักงานมารับสายได้ตลอด 24 ชั่วโมง หรือไม่คุ้มค่ากับการว่าจ้างคนมาทำงานในช่วงเวลากลางคืน ระบบ IVR จะทำให้ลูกค้าได้รับความสะดวกสบายในการรับบริการจากระบบโทรศัพท์ได้ตลอดเวลา เช่น บริการรับสั่งซื้อ รับจองสินค้า/บริการ รับฟังข้อมูลประชาสัมพันธ์หรือแนะนำสินค้า/บริการ การลงทะเบียนเรียนในมหาวิทยาลัย การขอรับทราบผลการสอบผ่านระบบ IVR เป็นต้น และจะนำระบบ Fax on Demand มาใช้ด้วยในกรณีที่ลูกค้าต้องการรับเอกสารผ่านทางเครื่องรับเอกสาร (Fax)

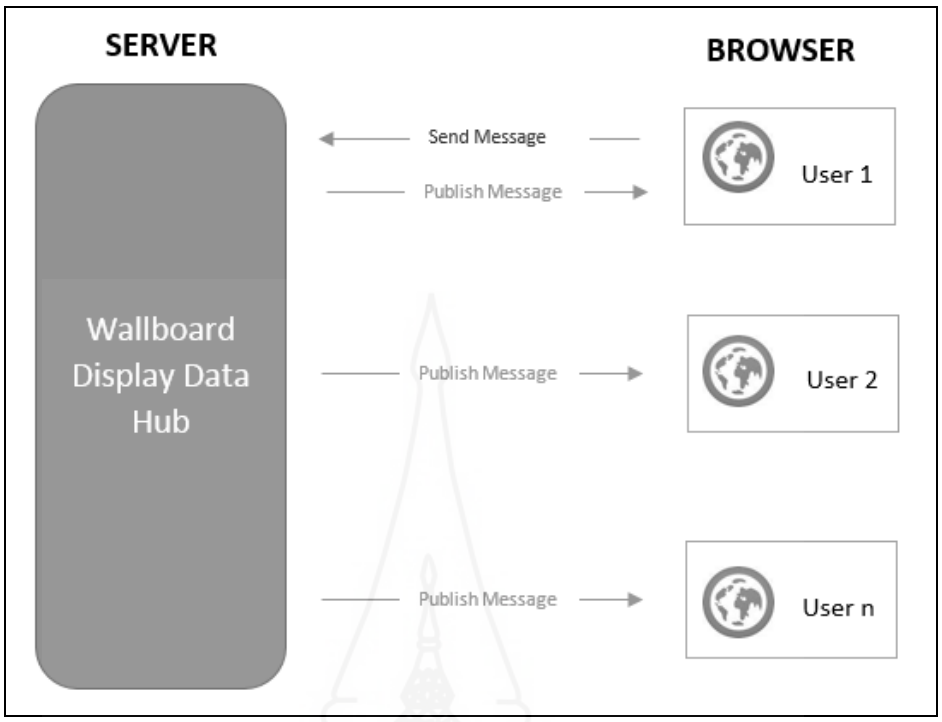
สำหรับสถาบันการเงินทั้งหลายโดยเฉพาะธนาคาร เป็นหนึ่งในกลุ่มผู้ใช้เทคโนโลยีด้าน Voice Processing ชัดเจนที่สุด โดยที่ลูกค้าสามารถสอบถามยอดบัญชี โอนเงิน อายัดบัตร ATM และเช็ค ชำระเงินค่าสินค้าและบริการอื่นๆ ชำระยอดบัตรเครดิต สอบถามข้อมูลโดยทั่วไป เช่น อัตราแลกเปลี่ยน อัตราดอกเบี้ย ฯลฯ ได้ทุกเวลา ผ่านระบบ IVR นั่นเอง

ประโยชน์ที่นอกเหนือจากการนำระบบ IVR มาใช้บริการแก่ลูกค้าแล้วนั้น การใช้งานภายในองค์กร ผู้บริหารองค์กรยังสามารถกด Password จากแป้นโทรศัพท์ เพื่อเข้าไปฟังข้อมูลที่อยู่ในฐานข้อมูลของระบบ Call Center ก็ได้ เช่น ยอดรวมการสั่งซื้อในแต่ละวัน ยอดขายที่ขายได้ หรือจะเช็คสต็อกสินค้าคงคลังที่ยังเหลืออยู่ เป็นต้น

2. ความหมายของ SignalR

ASP.NET SignalR เป็นหนึ่งในไลบรารี่ใหม่สำหรับการติดต่อสื่อสารแบบเรียลไทม์ ซึ่งเมื่อนำไปพัฒนาแล้ว ความสามารถของมันคือการที่ทำให้แอพลิเคชันเซิร์ฟเวอร์สามารถส่งข้อมูลกลับมายังไคลเอนท์ที่ติดต่อมันอยู่ได้เลย โดยที่ไคลเอนท์ไม่ต้องรีเฟรชหน้าบราวเซอร์เพื่อส่งคำสั่งมาร้องขออีกครั้งถึงจะได้ข้อมูลที่ต้องการและไม่ได้เป็นการทำงานแบบ Asynchronize

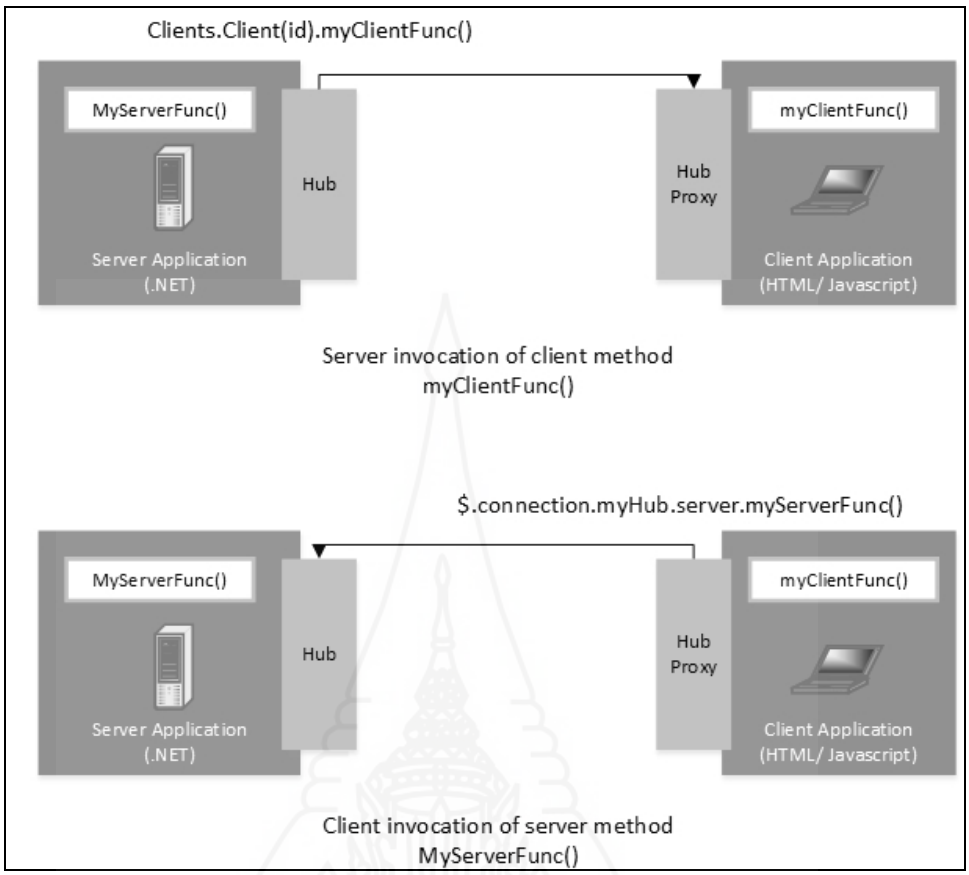
สำหรับ ASP.NET Application นั้นเราสามารถมาเพิ่มความสามารถของ SignalR เข้าไปในแอพลิเคชั่นเราได้ด้วยการเพิ่มความสามารถที่เรียกว่า “real-time” และขึ้นชื่อว่าเรียลไทม์หลายคนคงนึกถึงว่าส่วนใหญ่ก็คงเอาไประบบ Chat แต่ว่าหากผู้ใช้ ต้องการใช้งานระบบที่ไม่ต้องการให้ผู้ใช้งานรีเฟรชหน้าบราวเซอร์เองเพื่อรอดูข้อมูลใหม่ๆหรือมีงานที่ต้องการการโปรเซสผ่านเว็บเป็นระยะเวลานานๆนะมาใช้ SignalR ได้เลย ยกตัวอย่างเช่น พวกแอพลิเคชัน Dashboard, Monitoring พวกงานติดต่อสื่อสาร งานที่ต้องมีการทำงานพร้อมๆกันเป็นต้น



ภาพที่ 2.1 รูปภาพแสดงการทำงานของ SignalR

SignalR มีควาสามารถที่รองรับการอัพเดทข้อมูลที่มีปริมาณความถี่ที่สูงจากเซิร์ฟเวอร์ที่ส่งมายังไคลเอนท์ได้ เช่นแอพลิเคชั่นประเภทเกมส์ต่างๆ

SignalR นำเสนอไลบรารี่ง่ายๆ มาให้เราใช้ด้วยการสร้าง Remote Procedure Call (RPC) ระหว่าง Client และ Server โดยในฝั่งบราวเซอร์หรือไคลเอนท์นั้นก็จะใช้ Java Script เรียกไปยังแอพลิเคั่นเซิร์ฟเวอร์ผ่านทาง Server Side .NET Code ซึ่ง SignalR นั้นยังเตรียมเรื่องของการบริการเชื่อมต่อและการส่งข้อมูลไปแบบกลุ่มมาให้ได้ใช้กันด้วยและ SignalR ยังสามารถซัพพอร์ตเรื่องของการ Scale Out ไปยังหลายๆพันไคลเอนท์ผ่านทาง Service Bus, SQL Server และ Redis (NoSQL) ได้อีกด้วย



ภาพที่ 2.2 ส่วนประกอบของ ASP.NET SignalR โดยที่ทางฝั่งเซิร์ฟเวอร์จะติดต่อผ่านสิ่งที่เรียกว่า Hub และ Hub Proxy (ที่มา:www.asp.net/signalR)

3. การทำงานร่วมกันของ SignalR ร่วมกับ WebSocket

WebSockets เป็น API ที่ต่อจาก AJAX เทคนิคคือการ push ข้อมูลจากเซิร์ฟเวอร์มายังไคลเอนต์ เนื่องจากเทคนิคการส่งขอ้มูลแบบ HTTP แบบดั้งเดิมจะเปิดการเชื่อมต่อกับเซิร์ฟเวอร์เพื่อส่งขอ้มูล แล้วตัดการเชื่อมต่อเมื่อใช้เสร็จ ดังนั้นการขอข้อมูลจากเซิร์ฟเวอร์จึงทำได้ยาก เพราะต้องดึงข้อมูลจาก เซิร์ฟเวอร์ (polling)ซึ่งเปลืองการโหลดของเซิร์ฟเวอร์โดยเฉพาะกรณีที่ต้องเปิดการเชื่อมต่อ HTTP ค้างเอาไว้ (Long polling หรือ COMET) ดังนั้น WebSockets จึงเป็นเทคโนโลยีที่ช่วยแก้ปัญหานี้ โดยสร้างการ เชื่อมต่อแบบถาวรระหว่างเซิร์ฟเวอร์กับไคลเอนต์เพื่อให้สองฝั่งส่งข้อมูลกันได้ตลอด โดยอาศัย Protocol TCP ตัวอย่างการใช้งาน WebSockets ด้วย ws:// หรือถ้าต้องการการเชื่อมต่อแบบปลอดภยั wss:// ข้อดีคือสามารถส่งข้อมูลได้เร็วกว่า HTTP ข้อเสียคือ

เซิร์ฟเวอร์ต้องรองรับ WebSockets ด้วย ปัจจุบนั WebSockets เป็นมาตรฐานที่รับรองโดย IETF และกำลังผ่านกระบวนการเข้าเป็นมาตรฐาน ของ W3C

ในการทำงานของ SignalR จะทำการใช้คุณสมบัติของ WebSocket ใน HTML5 เป็นตัวเลือกแรกก่อนเลยในการรับและส่งข้อมูล แต่ถ้าไม่สามารถใช้ WebSocket ได้ก็จะเลี่ยงไปใช้วิธีการรับส่งข้อมูลที่เหมาะสมกับ WebSocket

การพัฒนาระบบด้วยการเขียน WebSocket ไปตรงๆ เลยก็ได้หรือไม่ก็อาจนำเอา SignalR มาผสมกันเข้าไปเพื่อใช้คุณสมบัติบางอย่างของ SignalR แต่เมื่อคุณเขียน WebSocket แบบมาตรฐานเลยคุณอย่าลืมเรื่องของความเข้ากันได้ของบราวเซอร์รุ่นต่างๆ ถ้าไม่ซัพพอร์ตจะทำอย่างไร แต่ถ้าคุณเลือก SignalR นั้นมันจะทำหน้าที่ป้องกันตรงส่วนนี้ให้คุณได้และก็เลือกสิ่งที่เหมาะสมในการรับส่งข้อมูลให้กับเรา

หากต้องการใช้ความสามารถของ SignalR อย่างเต็มที่เค้าบอกไว้ว่าให้ใช้แอพลิเคชั่นเซิร์ฟเวอร์ที่รันอยู่บนระบบปฏิบัติการ Windows Server 2012 หรือ Windows 8 พร้อมด้วยการติดตั้ง Microsoft.NET Framework เวอร์ชั่น 4.5 แต่ถ้ามีไม่ครบทาง Signal ก็จะเลี่ยงจากการใช้ WebSocket ไปใช้เทคโนโลยีเก่าๆ ในการรับและส่งข้อมูลให้แทน เอาละครับมาดูคุณสมบัติที่ทาง SignalR นำมาใช้ในการรับส่งข้อมูลครับแบ่งเป็น

1. รับส่งข้อมูลด้วยคุณสมบัติของ HTML 5

- Web Socket คือทั้งไคลเอนท์และเซิร์ฟเวอร์จะต้องใช้ WebSocket ได้ทั้งคู่ ด้วยการเชื่อมต่อแบบนี้ทางไคลเอนท์และเซิร์เวอร์จะติดต่อกันเหมือนจับมือกันตลอดเวลา แต่ก็มีข้อจำกัดเรื่องบราวเซอร์คือ Web Socket จะซัพพอร์ตเวอร์ชันล่าสุดของ Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox และใช้งานได้บางส่วนกับ Opera กับ Safari

- Server sent events การส่งข้อมูลทางเดียวจากเซิร์ฟเวอร์มายังไคลเอนต์ หลักการทำงานของ SSE คือเซิร์ฟเวอร์สามารถส่งข้อมูลไปยังไคลเอนต์ได้โดยตรง โดยที่ไคลเอนต์ไม่ต้องร้องขอ (GET Request) ก่อน

2. รับส่งข้อมูลด้วยคุณสมบัติ Comet

- Forever Frame คือวิธีการดั้งเดิมคือแบบสร้าง Iframe ไว้บนไคลเอนต์แล้วรับส่งข้อมูลผ่าน Iframe นี่แหละและใช้ได้แต่บน Microsoft Internet Explorer

- Ajax long polling การทำ Long Polling หรือรู้จักกันในอีกชื่อว่า Asynchronous Polling เป็นการใช้ประโยชน์จาก HTTP Keep-Alive ซึ่งจะส่งให้เมื่อบราวเซอร์ส่งการร้องขอไปยังเซิร์ฟเวอร์แล้ว เซิร์ฟเวอร์ไม่จำเป็นต้องตอบกลับทันที การเชื่อมต่อแบบ HTTP ครั้งนั้นจะอยู่จนกว่าจะหมดอายุ (timeout) หรือจนกว่าเซิร์ฟเวอร์จะมีข้อมูลให้กับบราวเซอร์

เมื่อเปรียบเทียบ Long Polling กับ Polling จะพบว่า จำนวนการร้องขอที่ต้องส่งนั้นมีจำนวนน้อยกว่า Polling แต่จะสังเกตุได้ว่า หากเซิร์ฟเวอร์มีข้อมูลบ่อยครั้ง Long Polling จะให้ผลเช่นเดียวกับ Polling ไม่ว่าจะเป็นการใช้งานสถาปัตยกรรมแบบ Polling หรือ Long Polling ก็ตามจะพบว่ามีการส่งคำร้องขอไปเป็นจำนวนมากระหว่างการเชื่อมต่อ โดยคำร้องขอแต่ละครั้งจะมี Overhead ของโปรโตคอล HTTP อยู่พอสมควร

ในกรณีของ WebSocket จะมีการส่งคำร้องขอไปยังเซิร์ฟเวอร์เพื่อขออัพเกรดการเชื่อมต่อแบบ HTTP ธรรมดาไปเป็น WebSocket โดยหลังการอัพเกรด บราวเซอร์และเซิร์ฟเวอร์จะสามารถส่งข้อมูลได้ทันทีเมื่อต้องการ โดยเป็นการสื่อสารสองทางเต็มรูปแบบ (full duplex)

4. วิธีการทำงานของ SignalR ร่วมกับ ฮับ

เมื่อรหัสฝั่งเซิร์ฟเวอร์เรียกวิธีบนไคลเอนต์แพ็คเกจถูกส่งผ่านการเชื่อมต่อ โดยการใช้งานที่มีชื่อและพารามิเตอร์ของวิธีการที่จะเรียกว่า (เมื่อวัตถุถูกส่งเป็นพารามิเตอร์วิธีการก็จะต่อเนื่องโดยใช้ JSON) ลูกค้าแล้วตรงกับชื่อวิธีการที่จะวิธีการที่กำหนดไว้ในรหัสฝั่งไคลเอ็นต์ หากมีการแข่งขันวิธีการที่ลูกค้าจะได้รับการดำเนินการโดยใช้ข้อมูลพารามิเตอร์ desterilized

```
10:38:03:1298 Session846.WebSocket'WebSocket #846' - Pushing 104 bytes from server WebSocket
81 66 7B 22 43 22 3A 22 42 2C 31 35 7C 43 2C 30    f{"C":"B,15|C,0
7C 44 2C 30 7C 45 2C 30 22 2C 22 4D 22 3A 5B 7B  |D,0|E,0","M":[{
22 48 22 3A 22 4D 6F 76 65 53 68 61 70 65 48 75  "H":"MoveShapeHu
62 22 2C 22 4D 22 3A 22 75 70 64 61 74 65 53 68  b","M":"updateSh
61 70 65 22 2C 22 41 22 3A 5B 7B 22 6C 65 66 74  ape","A":[{"left
22 3A 35 30 31 2E 30 2C 22 74 6F 70 22 3A 33 30  ":501.0,"top":30
32 2E 30 7D 5D 7D 5D 7D                          2.0}]}]}
```

ในตัวอย่างนี้ชื่อฮับจะถูกระบุด้วย พารามิเตอร์ H และ ชื่อวิธีการจะถูกระบุด้วยพารามิเตอร์ M และข้อมูลที่ถูกส่งไปจะถูกระบุด้วยพารามิเตอร์ A แอพลิเคชันที่สร้างข้อความนี้ถูกสร้างขึ้นในกระบวนการส่งข้อมูลแบบเรียลไทม์

SignalR ได้รับการสนับสนุนภายใต้ความหลากหลายของเซิร์ฟเวอร์และการกำหนดค่าของลูกค้า นอกจากนี้ในแต่ละตัวเลือกการขนส่งมีชุดของความต้องการของตัวเอง ถ้าความต้องการของระบบสำหรับการขนส่งไม่สามารถใช้ได้อย่างสง่างาม SignalR จะ failover เพื่อการขนส่งอื่นๆ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการขนส่งที่สนับสนุน SignalR ให้ดูที่ การขนส่งและ fallbacks

5. ความต้องการพื้นฐานของง SignalR

5.1 เซิร์ฟเวอร์สำหรับ SignalR

ส่วนประกอบเซิร์ฟเวอร์ SignalR มีความหลากหลายของการกำหนดค่าเซิร์ฟเวอร์ ส่วนนี้จะอธิบายเวอร์ชันที่สนับสนุนระบบปฏิบัติการที่สนับสนุนการทำงานของ Asp.net เซิร์ฟเวอร์ข้อมูลทางอินเทอร์เน็ตและส่วนประกอบอื่นๆ

5.1.1 ระบบปฏิบัติการเซิร์ฟเวอร์ที่สนับสนุนการทำงานของ SignalR

ส่วนประกอบเซิร์ฟเวอร์ SignalR สามารถใช้ได้ในเซิร์ฟเวอร์ต่อไปนี้หรือระบบปฏิบัติการที่รองรับเทคโนโลยี Signal และWeb Sockets ได้อาทิเช่น

- Windows Server 2012
- Windows Server 2008 R2
- Windows 8
- Windows 7
- Windows Azure
- สนับสนุนเซิร์ฟเวอร์ NET Framework 4.5 ขึ้นไป

5.1.2 IIS ที่สนับสนุนการทำงานของ SignalR

- IIS 8 หรือ IIS 8 Express
- IIS 7 และ 7.5 การสนับสนุนสำหรับ URL ที่มีสกุลเฉพาะ
- IIS จะต้องทำงานในโหมดแบบบูรณาการ โหมดคลาสสิกที่ไม่ได้รับการสนับสนุน ความล่าช้าข้อความได้ถึง 30 วินาทีอาจจะมีประสบการณ์ถ้า IIS กำลังทำงานในโหมดคลาสสิกโดยใช้เซิร์ฟเวอร์ส่งขนส่งเหตุการณ์
- การประยุกต์ใช้โฮสติ้งจะต้องมีการทำงานในโหมดความไว้วางใจอย่างเต็มรูปแบบ
- ความต้องการของระบบไคลเอ็นต์

5.1.3 เว็บเบราเซอร์ที่สนับสนุนการทำงานของ SignalR

โปรแกรมประยุกต์ที่ใช้ในเบราว์เซอร์ SignalR ต้องใช้รุ่น jQuery 1.6.4 หรือรุ่นที่ใหม่กว่า เช่น 1.7.2, 1.8.2 หรือตั้งแต่ 1.9.1 ขึ้นไป และสนับสนุนเว็บเบราเซอร์ดังต่อไปนี้

- Microsoft Internet Explorer 8, 9, 10, และ 11
- Mozilla Firefox: เวอร์ชั่นปัจจุบันและเวอร์ชั่นย้อนหลัง 1 เวอร์ชั่นสามารถใช้งานได้ทั้งทั้ง Windows และ Mac

- Google Chrome: เวอร์ชั่นปัจจุบันและเวอร์ชั่นย้อนหลัง 1 เวอร์ชั่นสามารถใช้งานได้ทั้ง Windows และ Mac

- Safari: : เวอร์ชั่นปัจจุบันและเวอร์ชั่นย้อนหลัง 1 เวอร์ชั่นสามารถใช้งานได้ทั้ง Mac และ iOS ทุกรุ่น

- โอเปร่า: เวอร์ชั่นปัจจุบันและเวอร์ชั่นย้อนหลัง 1 เวอร์ชั่นสามารถใช้งานได้บน Windows เท่านั้น

- เว็บเบราว์เซอร์ของระบบ Android

ตารางที่ 2.1 ตารางความต้องการ Transport Layer ของเว็บเบราว์เซอร์

ความต้องการ Transport Layer ของเว็บเบราว์เซอร์					
Transport	**Internet Explorer**	**Chrome (Windows or iOS)**	**Firefox**	**Safari (OSX or iOS)**	**Android**
WebSockets	10+	เวอร์ชันปัจจุบัน - 1	เวอร์ชันปัจจุบัน - 1	เวอร์ชันปัจจุบัน - 1	N/A
Server-Sent Events	N/A	เวอร์ชันปัจจุบัน - 1	เวอร์ชันปัจจุบัน - 1	เวอร์ชันปัจจุบัน - 1	N/A
Forever Frame	8+	N/A	N/A	N/A	4.1
Long Polling	8+	เวอร์ชันปัจจุบัน - 1	เวอร์ชันปัจจุบัน - 1	เวอร์ชันปัจจุบัน - 1	4.1

5.1.4 Windows Desktop และการประยุกต์ Silverlight

นอกเหนือจากการทำงานในเว็บเบราเซอร์ SignalR สามารถเป็นเจ้าภาพในไคลเอนต์ Windows แบบสแตนด์อโลนหรือโปรแกรม Silverlight Windows Desktop และ Silverlight SignalR การใช้งานมีความต้องการระบบต่อไปนี้

- การประยุกต์ใช้สุทธิ 4 ได้รับการสนับสนุนใน Windows XP SP3 หรือในภายหลัง

- การประยุกต์ใช้ .NET Framework 4.5 ได้รับการสนับสนุนบน Windows Vista จนถึงระบบปฏิบัติการปัจจุบันของระบบปฏิบัติการวินโดว์

ตารางที่ 2.2 ตารางความต้องการชั้นการสื่อสารนำส่งข้อมูลของระบบปฏิบัติการวินโดว์และ Silverlight

ความต้องการ Transport Layer ของระบบปฏิบัติการวินโดว์และ Silverlight		
Transport	**.NET application**	**Silverlight**
Web Sockets	Windows 8+ and .NET 4.5+	N/A
Forever Frame	N/A	N/A
Server-Sent Events	.NET 4+	5+
Long Polling	.NET 4+	5+

Silverlight คือ ดอทเน็ตปลั๊กอินที่ช่วยให้นักออกแบบ และนักพัฒนาสามารถพัฒนาแอพพลิเคชั่นประเภทมัลติมีเดียสมบูรณ์แบบสำหรับเบราว์เซอร์ได้ในหลายๆ เบราว์เซอร์ และสามารถรันได้บนหลายแพลตฟอร์ม กล่าวคือเราสามารถรันแอพพลิเคชั่นที่ทำด้วย Silverlight ได้ทั้งบน Firefox, Safari และที่แน่นอนก็คือ IE นอกจากนี้ Silverlight ยังสามารถรันได้ทั้งบนวินโดว์สและแมคอินทอชอีกด้วย

6. การกำหนดค่า URL SignalR

โดยค่าเริ่มต้นเส้นทาง URL ที่ลูกค้าจะใช้ในการเชื่อมต่อกับฮับของคุณคือ "/ signalr" (อย่าสับสน URL นี้กับ "/ signalr / ฮับ" URL ซึ่งเป็นที่สร้างขึ้นโดยอัตโนมัติไฟล์ JavaScript สำหรับข้อมูลเพิ่มเติมเกี่ยวกับพร็อกซี่ที่สร้างให้ดู. SignalR Hub API - ไคลเอนต์ใช้งาน JavaScript - พร็อกซี่สร้างขึ้นและสิ่งที่ไม่จำเป็น

อาจจะมีสถานการณ์พิเศษที่ทำให้ URL ไม่สามารถใช้งานได้กับ SignalR ตัวอย่างเช่น คุณมีโฟลเดอร์ในโครงการของคุณชื่อ signalr และคุณไม่ต้องการที่จะเปลี่ยนชื่อ ในกรณีที่คุณ

สามารถเปลี่ยน URL พื้นฐานตามที่แสดงในตัวอย่างต่อไปนี้ (แทนที่ "/ signalr" ในรหัสตัวอย่างที่มี URL ที่คุณต้องการ)

โค๊ด URL สำหรับการเชื่อมต่อเวอร์เวอร์

```
app.MapSignalR("/signalr", new HubConfiguration());
```

โค๊ด URL สำหรับการเชื่อมต่อจาวาสคริป แบบมีการสร้างพร็อกซี่อัตโนมัติ

```
$.connection.hub.url = "/signalr"
$.connection.hub.start().done(init);
```

โค๊ด URL สำหรับการเชื่อมต่อจาวาสคริป แบบไม่มีการสร้างพร็อกซี่อัตโนมัติ

```
var connection = $.hubConnection("/signalr", { useDefaultPath: false });
```

โค๊ด URL สำหรับการเชื่อมต่อโปรแกรมไคลแอนด์

```
var hubConnection = new HubConnection("http://contoso.com/signalr", useDefaultUrl: false);
```

7. ความปลอดภัยของ SignalR

7.1 การตรวจสอบและอนุมัติ

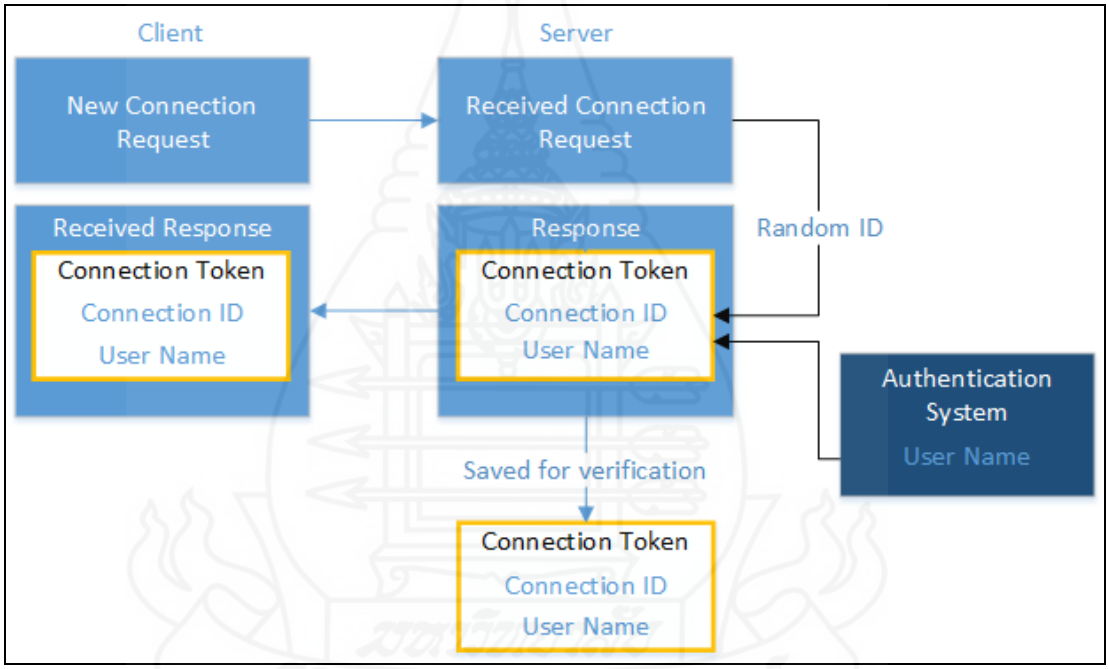
SignalR ไม่ได้มีคุณสมบัติใดๆ สำหรับการตรวจสอบผู้ใช้ แต่คุณรวมคุณสมบัติ SignalR ในโครงสร้างการตรวจสอบที่มีอยู่สำหรับการประยุกต์ใช้ คุณตรวจสอบผู้ใช้ตามที่คุณต้องการได้ตามปกติในใบสมัครของคุณและทำงานกับผลของการตรวจสอบในรหัส SignalR ของคุณ ตัวอย่างเช่นคุณอาจจะตรวจสอบผู้ใช้ของคุณด้วยรูปแบบการตรวจสอบ ASP.NET และจากนั้นในศูนย์กลางของการบังคับใช้ที่ผู้ใช้หรือบทบาทที่ได้รับอนุญาตให้เรียกวิธีการ ในศูนย์กลางของคุณคุณยังสามารถส่งข้อมูลการตรวจสอบเช่นชื่อผู้ใช้หรือว่าผู้ใช้เป็นบทบาทให้กับลูกค้า

SignalR อนุญาตให้แอตทริบิวต์ระบุว่าผู้ใช้สามารถเข้าถึงศูนย์กลางหรือวิธีการที่คุณใช้แอตทริบิวต์อนุญาตให้ทั้งฮับหรือโดยเฉพาะอย่างยิ่งวิธีการในการเป็นศูนย์กลาง โดยไม่ต้องแอตทริบิวต์อนุญาตวิธีส่วนกลางทั้งหมดบนฮับที่มีอยู่ให้กับลูกค้าที่เชื่อมต่อกับฮับ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับฮับให้ดูที่ การตรวจสอบและอนุมัติสำหรับ SignalR ฮับ ใช้คุณสมบัติของ Authorize แอตทริบิวต์ฮับ แต่ไม่ได้เชื่อมต่อแบบถาวร ในการบังคับใช้กฎการอนุมัติเมื่อใช้

Persistent Connection คุณต้องแทนที่ Authorize Request วิธี สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการเชื่อมต่อแบบถาวรให้ดูที่ การตรวจสอบและการอนุมัติสำหรับการเชื่อมต่อแบบถาวร SignalR

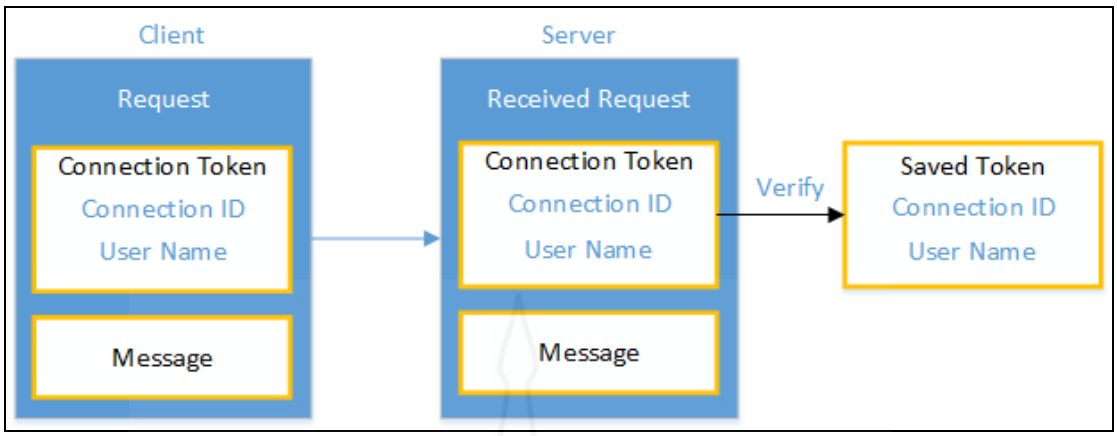
7.2 โทเค็นการเชื่อมต่อของ SignalR

SignalR ลดความเสี่ยงในการดำเนินการคำสั่งที่เป็นอันตรายโดยการตรวจสอบตัวตนของผู้ส่ง สำหรับการร้องขอแต่ละไคลเอ็นต์และเซิร์ฟเวอร์ผ่านโทเค็นการเชื่อมต่อที่มีการเชื่อมต่อรหัสและชื่อผู้ใช้สำหรับผู้ใช้รับรองความถูกต้อง รหัสการเชื่อมต่อที่ไม่ซ้ำกันในแต่ละระบุลูกค้าที่เชื่อมต่อ เซิร์ฟเวอร์สุ่มสร้างรหัสการเชื่อมต่อเมื่อการเชื่อมต่อใหม่จะถูกสร้างขึ้นและยังคงมีรหัสว่าในช่วงระยะเวลาของการเชื่อมต่อ กลไกการตรวจสอบสำหรับการประยุกต์ใช้เว็บให้ชื่อผู้ใช้ SignalR ใช้การเข้ารหัสและลายเซ็นดิจิตอลเพื่อปกป้องโทเค็นการเชื่อมต่อ



ภาพที่ 2.3 รูปภาพแสดงโทเค็นการเชื่อมต่อของ SignalR (ที่มา:www.asp.net/signalR)

สำหรับการร้องขอแต่ละเซิร์ฟเวอร์จะตรวจสอบเนื้อหาของโทเค็นเพื่อให้แน่ใจว่าการร้องขอมาจากผู้ใช้ที่ระบุ ชื่อผู้ใช้จะต้องสอดคล้องกับรหัสการเชื่อมต่อ โดยการตรวจสอบทั้งรหัสการเชื่อมต่อและชื่อผู้ใช้ SignalR ป้องกันไม่ให้ผู้ใช้ที่เป็นอันตรายจากการปลอมตัวได้อย่างง่ายดายผู้ใช้อื่น ถ้าเซิร์ฟเวอร์ที่ไม่สามารถตรวจสอบการเชื่อมต่อโทเค็นการร้องขอล้มเหลว



ภาพที่ 2.4 รูปภาพแสดงสถานการณ์ร้องของการเชื่อมต่อของ Client ไปยัง Server เพื่อการตรวจสอบสิทธิ์การเข้าถึง (ที่มา:www.asp.net/signalR)

7.3 เข้าร่วมกลุ่มเมื่อเชื่อมต่อ

โดยค่าเริ่มต้นการประยุกต์ใช้ SignalR โดยอัตโนมัติจะเปิดการกำหนดผู้ใช้กลุ่มที่เหมาะสมเมื่อเชื่อมต่อจากการหยุดชะงักชั่วคราวเช่นเมื่อมีการเชื่อมต่อจะลดลงและอีกครั้งที่จัดตั้งขึ้นก่อนที่จะหมดเวลาการเชื่อมต่อ เมื่อเชื่อมต่อลูกค้าผ่านโทเค็นกลุ่มที่มีรหัสในการเชื่อมต่อและกลุ่มที่ได้รับมอบหมาย โทเค็นกลุ่มที่มีลายเซ็นและการเข้ารหัส ลูกค้ายังคงมีรหัสเดียวกันหลังจากการเชื่อมต่อการเชื่อมต่อใหม่ ดังนั้นการเชื่อมต่อรหัสผ่านจากลูกค้าเชื่อมจะต้องตรงกับรหัสที่ใช้การเชื่อมต่อก่อนหน้านี้โดยลูกค้า การตรวจสอบนี้จะช่วยป้องกันผู้ใช้ที่เป็นอันตรายจากการผ่านการร้องขอไปเข้าร่วมกลุ่มไม่ได้รับอนุญาตเมื่อเชื่อมต่อ

แต่ก็เป็นสิ่งสำคัญที่จะทราบว่ากลุ่ม token ไม่หมดอายุ หากผู้ใช้เป็นกลุ่มในอดีตที่ผ่านมา แต่ไม่ได้รับอนุญาตจากกลุ่มที่ผู้ใช้ที่อาจจะไม่สามารถที่จะเลียนแบบโทเค็นกลุ่มที่มีกลุ่มต้องห้าม ถ้าคุณต้องการที่จะปลอดภัยในการจัดการผู้ใช้เป็นกลุ่มที่คุณจำเป็นต้องเก็บข้อมูลที่อยู่บนเซิร์ฟเวอร์เช่นในฐานข้อมูล จากนั้นให้เพิ่มตรรกะในการใช้งานของคุณที่ตรวจสอบบนเซิร์ฟเวอร์ว่าผู้ใช้อยู่ในกลุ่ม สำหรับตัวอย่างของการตรวจสอบการเป็นสมาชิกของกลุ่มให้ดูที่ การทำงานกับกลุ่มเข้าร่วมกลุ่มโดยอัตโนมัติใช้เฉพาะเมื่อมีการเชื่อมต่อจะเชื่อมหลังจากที่หยุดชะงักชั่วคราว หากผู้ใช้ยกเลิกการเชื่อมต่อโดยการนำออกจากโปรแกรมหรือรีสตาร์ทโปรแกรมที่ใช้งานของคุณต้องจัดการกับวิธีการเพิ่มผู้ใช้ที่ถูกต้องให้กับกลุ่ม สำหรับข้อมูลเพิ่มเติมโปรดดูที่ การทำงานกับกลุ่ม

7.4 ความปลอดภัยของ SignalR จากเว็บไซต์ปลอม

เว็บไซต์ปลอม (CSRF) คือการโจมตีที่เป็นอันตรายเว็บไซต์ส่งคำขอไปยังเว็บไซต์ที่มีช่องโหว่ที่ผู้ใช้อยู่ในขณะนี้เข้าสู่ระบบ SignalR ป้องกันไม่ให้ CSRF โดยการร้องขอใบรับรองสำหรับเว็บไซต์ที่เป็นอันตรายในการสร้างการร้องขอที่ถูกต้องสำหรับ SignalR ชั้น SSL (Secure Socket) โปรโตคอล

โปรโตคอล SSL ใช้การเข้ารหัสเพื่อรักษาความปลอดภัยการขนส่งของข้อมูลระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ หากใบสมัคร SignalR ของคุณส่งข้อมูลที่สำคัญระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ใช้ SSL สำหรับการขนส่ง สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการตั้งค่า SSL ให้ดู วิธีการตั้งค่า SSL บน IIS 7

8. ความหมายของระบบเรียลไทม์ (Real-time)

ดร.ธนารักษ์ ธีระมั่นคง (2549 : 15-16) ได้กล่าวถึงการประมวลผลแบบเรียลไทม์ไว้ว่า ระบบ เรียลไทม์เป็นระบบที่มีการประมวลผลเสร็จสิ้นภายในเวลาที่กำหนด โดยตอบสนองอะไรบางอย่าง เช่น การ แสดงผลบนหน้าจอตามคำสั่งของผู้ควบคุมเครื่องมือ ระบบเรียลไท์มแบ่งเป็น 2 ประเภท คือ (1) ประเภทที่ มีข้อจำกัดด้านเวลาแบบหลวมๆ เช่น การควบคุมเครื่องมืออุปกรณ์ของมนุษย์ และ (2) ประเภทที่มีข้อจำกัด เวลาเข้มงวดมาก เช่น การสื่อสารระหว่างอุปกรณ์กับอปุกรณ์ โดยแบบแรกจะเรียกว่า เรียลไทม์แบบอ่อน (soft real-time system) และแบบหลังจะเรียกว่าเรียลไทม์แบบแข็ง (hard real-time system)

ความหมายโดยรวมของระบบเรียลไทม์ (Real-time) คือระบบที่สามารถให้การตอบสนองจากระบบอย่าง ทันทีทันใดเมื่อได้รับอินพุตเข้าไป ในทางอุดมคติระบบเรียลไทม์นี้จะเป็นระบบที่ไม่เสียเวลาในการ ประมวลผลหรืออาจจะกล่าวได้ว่าเวลาในการประมวลเป็นศูนย์ แต่ในทางปฏิบัติเครื่องคอมพิวเตอร์ที่ทำงานแบบเรียลไทม์นี้ไม่สามารถผลิตขึ้นมาได้ เราทำได้เพียงการลดเวลาการประมวลผลให้น้อย ที่สุดจนไม่สามารถเห็นความแตกต่างของช่วงเวลาที่ป้อนอินพุตเข้าไปและได้รับเอ้าท์พุทออกมา เวลาของความแตกต่างนี้เรียกว่า “เวลาตอบสนอง” (response time) ซึ่งผู้ใช้งานทั่วไปต้องการเวลา ตอบสนองให้น้อยที่สุดเพื่อประสิทธิภาพของระบบ

ระบบเรียลไทม์นิยมนำไปใช้ในการควบคุม กระบวนการในทางอุตสาหกรรม ซึ่งปัจจุบันสามารถควบคุมด้วยระบบคอมพิวเตอร์ที่ทันสมัยให้ค่าเวลาตอบสนองที่ยอมรับได้สำหรับในวงการคอมพิวเตอร์ระบบเรียลไทม์เข้าใกล้อุดมคติมากขึ้น เนื่องจากความเร็วในการประมวลผลของซีพียู

8.1 ประเภทของ Real-time

จุดประสงค์อีกอย่างหนึ่งของระบบปฏิบัติการ คือ ระบบเวลาจริง (Real-time system) หมายถึงการ ตอบสนองทันที เช่นระบบ Sensor ที่ส่งข้อมูลให้คอมพิวเตอร์ เครื่องมือทดลองทางวิทยาศาสตร์ ระบบภาพ ทางการแพทย์ ระบบควบคุมในโรงงานอุตสาหกรรม ระบบหัวฉีดในรถยนต์ ระบบควบคุมการยิง ระบบ แขนกล และเครื่องใช้ในครัวเรือนทั้งหมด ระบบ (Real-Time Systems) จะทำงานได้กับงานที่มีข้อข้อกำหนดในเรื่อง Fixed-Time ระบบ Real-Time Systems มี 2 ระบบดังนี้

1) Hard real-Time Systems เป็นระบบที่ถูกรับรองว่าจะได้รับการตอบสนองตรงเวลา และหยุด รอไม่ได้ (ทำงานเสร็จตรงตามเวลา) ระบบนี้ไม่มีฮาร์ดิสก์หรือมีขนาดเล็ก การเก็บข้อมูลจะเก็บใน Short-Term memory หรือรอม (Rom) มีข้อเสียคือ มีปัญหากับระบบแบ่งส่วนเวลาและไม่มีการซัฟพอร์ต (Support)

2) Soft real-Time Systems เป็นระบบ less restrictive type ที่สามารถรอให้งานอื่นทำให้เสร็จ ก่อนได้ (ขาด Deadline) มีการจำกัด Utility และ เสียงในการใช้ในอุตสาหกรรมควบคุมหรือ หุ่นยนต์

ระบบเรียลไทม์มีประโยชน์ในการประยุกต์ใช้ใน Multimedia, Virtual reality และการสำรวจในน้ำและ ดาวเคราะห์ ระบบนี้ต้องการ Features ของระบบปฏิบัติการขั้นสูง

ระบบ Real-Time Systems จะมีเวลาในการทำงานจำกัดถ้าระบบให้ผลลัพธ์ได้ไม่ทันในช่วงเวลาที่กำหนดระบบนั้นๆ ก็จะล้มเหลว ซึ่งต่างจากระบบแบ่งเป็นส่วน (Time Sharing Systems) ซึ่งตอบสนอง ได้ทันที หรือระบบกลุ่ม (Batch Operating Systems) ซึ่งไม่มีการจำกัดเวลาในการตอบสนองเลย Real Time Programming and System เป็นระบบที่ว่าด้วยโปรแกรมที่ใช้สร้างแอพพลิเคชั่น (Application) เกี่ยวกับระบบเวลาจริง อย่างเช่น Real-time PCR ใช้ในการตรวจและเพิ่มจำนวน DNA และ RNA เป็นต้น

8.2 กิจกรรมการประมวลผล ในระบบ Real-time

การประมวลผลข้อมูลโดยทั่วไปประกอบไปด้วยขั้นตอนหลักๆ 3 ขั้นตอน ในแต่ละขั้นตอนมี รายละเอียดดังนี้

1) ส่วนนำข้อมูลเข้า (Input) การดำเนินการในขั้นตอนนี้ประกอบด้วยการรวบรวมข้อมูล (Collection) เป็นขั้นตอนแรกของการประมวลผลข้อมูล คือ การรวบรวม ข้อมูลที่จะนำมาประมวลผล เช่น ใบสั่งซื้อสินค้า ประวัติของพนักงาน ข้อมูลนักศึกษา และเตรียมข้อมูลเพื่อที่จะนำไปประมวลผลการบันทึกข้อมูล (Recording) เป็นขั้นตอนที่ต่อจากการรวบรวมข้อมูลและบันทึก

ข้อมูลลงในสื่อ (Media) บันทึกข้อมูล เช่น แผ่นดิสก์เก็ต เทปแม่เหล็ก จานแม่เหล็ก แผ่น CD การสอบทานข้อมูล (Verification) ป็นการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะนำไปประมวลผล

2) การดำเนินการ (Process) เป็นขั้นตอนที่นำข้อมูลซึ่งผ่านขั้นตอนแรกนำไปประมวลผล ในการ ประมวลผลอาจมีการประมวลผลย่อย ดังนี้

- การเรียงลำดับข้อมูล (Sorting) ข้อมูลที่แสดงออกมาทางรายงานอาจจะไม่ได้เรียงลำดับ การเรียงลำดับข้อมูลมี 2 อย่าง คือ เรียงลำดับข้อมูลจากมากไปน้อย หรือเรียงจากน้อยไปมาก การเรียงลำดับข้อมูลจะต้อง กำหนดข้อมูลที่จะนำมาจัดเรียงอาจเป็นข้อมูลตัวเลข หรือตัวอักษร เช่น เรียงข้อมูลตามชื่อพนักงานจากมาก ไปน้อย เรียงข้อมูลตามเงินเดือนจากมากไปน้อย เรียงเกรดนักศึกษาจากน้อยไปมาก ข้อมูลที่นำมาจัดเรียงเรา จะเรียกว่า Key Field
- การจัดหมวดหมู่ (Classification) เป็นการจัดกลุ่มข้อมูลที่เหมือนกันไว้ด้วยกันเช่น พนักงาน จำแนกตามแผนก พนักงานจำแนกตามตำแหน่งงานหนังสือจำแนกตามสำนักพิมพ์ หนังสือจำแนกตาม ประเภทหนังสือ
- การคัดเลือก (Selection) เป็นการคัดเลือกข้อมูลที่มีคุณสมบัติตรงตามความต้องการของผู้ใช้
- การคำนวณ (Calculation) เป็นการคำนวณ เช่น การบวก การลบ การคูณ การหาร ยกกำลัง
- การปรับปรุงข้อมูล (Update) คือการปรับปรุงเปลี่ยนแปลงข้อมูลให้ทันสมัยอยู่เสมอ การรวบรวม ข้อมูลเพื่อผลิตข้อมูลสารสนเทศ ต้องคำนึงถึงความถูกต้องของข้อมูล การปรับปรุงเลี่ยนแปลงข้อมูลประกอบด้วย การเพิ่ม การลบ การแก้ไขข้อมูลให้เป็นปัจจุบัน การปรับปรุงข้อมูลเป็นขั้นตอนที่สำคัญ

3) ส่วนข้อมูลออก หรือผลลัพธ์ (Output) การดำเนินการในขั้นนี้ประกอบด้วย การทำรายงาน (Report) การแสดงข้อมูลตามรูปแบบที่กำหนดอาจจะออกรายงานทางจอภาพ (Monitor) หรือทางเครื่องพิมพ์ (Printer) การจะออกรายงานแบบไหนขึ้นอยู่กับวัตถุประสงค์ของการใช้งาน การจัดเก็บ ควรจัดเก็บข้อมูลที่เป็นรายงานให้สะดวกต่อการค้นหาและง่ายต่อการนำมาใช้งาน

9. SQL server 2012

Microsoft SQL Server คือแพล็ตฟอร์มข้อมูลสารสนเทศและเซิร์ฟเวอร์ฐานข้อมูลแบบครบวงจรที่มอบเทคโนโลยีที่พร้อมสำหรับองค์กรขนาดใหญ่และเครื่องมือที่ช่วยให้บุคลากรได้

ประโยชน์จากข้อมูลที่มีอย่างเต็มที่โดยมีต้นทุนสำหรับเจ้าของ (Total Cost of Ownership) ที่ต่ำที่สุด ผู้ใช้จะได้รับประสิทธิภาพการทำงาน ความพร้อมในการให้บริการ และระบบความปลอดภัยที่สูงขึ้น ในขณะเดียวกันก็มีเครื่องมือในการจัดการและการพัฒนาซอฟต์แวร์ที่ช่วยให้เกิดการทำงานอย่างมีประสิทธิภาพ และมอบข้อมูลเชิงลึกที่ใช้งานได้เต็มที่ผ่าน BI ที่ผู้ใช้จัดการได้ด้วยตนเอง

Microsoft SQL Server เป็นแพล็ตฟอร์มที่สมบูรณ์และทำงานร่วมกับฐานข้อมูลอื่นได้ สร้างคุณค่าจากทักษะและทรัพยากรของไอทีเพื่อให้เกิดประโยชน์สูงสุด เพิ่มประสิทธิผลการทำงานและความคล่องตัวของแผนกไอที และสร้างแอพพลิเคชั่นใหม่ๆ ที่ทำงานอย่างยืดหยุ่นได้อย่างรวดเร็ว

พื้นฐานและโครงสร้าง SQL Server 2012 Database คำสั่งและการใช้งาน SQL Server 2012 จะมีสองตัวเลือก คือที่อยู่บนพื้นฐานของ computing power และที่อยู่บนพื้นฐานของผู้ใช้หรืออุปกรณ์ (users หรือ devices) (ดูตารางด้านล่างสำหรับการเปรียบเทียบทางเลือกของ new licensing ต่อ edition)

9.1 SQL Server Service Broker

เป็นเทคโนโลยีที่ช่วยให้งานที่ถูกกระทำบนระบบฐานข้อมูลมีความไว้วางใจได้ สามารถปรับขนาดของงานได้ และทำให้งานมีความปลอดภัยมากขึ้น Service Broker นั้นเป็นเทคโนโลยีที่อยู่ในส่วนของ Database Engine อีกที Service Broker จะใช้การสื่อสารด้วยข้อความเพื่องานต่างๆที่อยู่แยกกันให้สามารถทำงานร่วมกันได้ นอกจากนี้ Service Broker ยังมีเครื่องมือต่างๆที่จำเป็นสำหรับการสร้างฐานข้อมูลแบบกระจาย และเครื่องมือเหล่านี้จะช่วยลดระบะเวลาในการพัฒนาฐานข้อมูลลงได้อย่างมา นอกจากนี้ Service Broker ยังช่วยให้ผู้ใช้สามารถปรับขนาดของงาน ให้เหมาะสมกับปริมาณงานที่ได้รับ

```
ALTER DATABASE ‘databasename’ SET ENABLE_BROKER WITH ROLLBACK IMMEDIATE ;
```

ตัวอย่างคำสั่งในการเปิดใช้งาน Service Broker บน Microsoft SQL Server

9.2 SQL Dependency

SQL dependency เป็นการให้บริการโดย .Net Framework ถูกสร้างขึ้นบนโครงสร้างพื้นฐานของ Microsoft SQL Server ช่วยอำนวยความสะดวกให้กับ แอพลิเคชัน เกี่ยวกับการรับการแจ้งเตือนเมื่อมีการเปลี่ยนแปลงข้อมูลในระบบฐานข้อมูล Microsoft SQL Server จะช่วยให้ที่ส่ง

คำสั่ง (SQL Command) ระหว่าง โปรแกรมประยุกต์ กับ SQL Server สามารถแจ้งการดำเนินงานในคำสั่งที่ต้องการผลลัพธ์ที่แตกต่างกัน เช่นคำสั่ง SELECT และ EXECUTE

บทที่ 3
วิธีดำเนินการวิจัย

1. ขั้นตอนการดำเนินโครงการ

ขั้นตอนและวิธีการการดำเนินโครงการเรื่อง การปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์ มีรายละเอียดดังต่อไปนี้

1.1 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง ซึ่งได้แก่

1.1.1 ประเด็นปัญหาและวิธีการพัฒนาปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์ รวมถึงเทคโนโลยีที่เกี่ยวข้อง

1.1.2 กระบวนการแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์

1.1.3 รายละเอียด ลักษณะเฉพาะ และโครงสร้างข้อมูลของระบบคอลเซ็นเตอร์

1.1.4 รายละเอียดและวิธีการประยุกต์ใช้เทคโนโลยีซิกแนลอาร์

1.1.5 งานวิจัยและระบบที่เกี่ยวข้องกับการใช้เทคโนโลยีซิกแนลอาร์

1.2 วางแผนการดำเนินโครงการและพัฒนาปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์

1.3 ศึกษาและออกแบบวิธีการใช้งานเทคโนโลยีซิกแนลอาร์

1.4 เลือกเครื่องมือและพัฒนาระบบตามที่ได้ออกแบบไว้

1.4.1 ศึกษาแนวทางการปรับปรุงระบบแสดงผลข้อมูลของคอลเซ็นเตอร์จากระบบที่ใช้เทคโนลีแบบดั้งเดิมเป็นการใช้เทคโนโลยีใหม่

1.4.2 ปรับปรุงระบบแสดงผลข้อมูลของคอลเซ็นเตอร์จากระบบที่ใช้เทคโนลีแบบดั้งเดิมเป็นการใช้เทคโนโลยีใหม่

1.5 ทำการทดสอบระบบใหม่ด้วยซอฟต์แวร์ประยุกต์ การตรวจสอบการใช้งานทรัพยากรเครื่องเซิร์ฟเวอร์จากระบบต้นแบบเพื่อประเมินการใช้ทรัพยากรของเครื่องเซิร์ฟเวอร์จากระบบใหม่เปรียบเทียบกับระบบดั้งเดิม

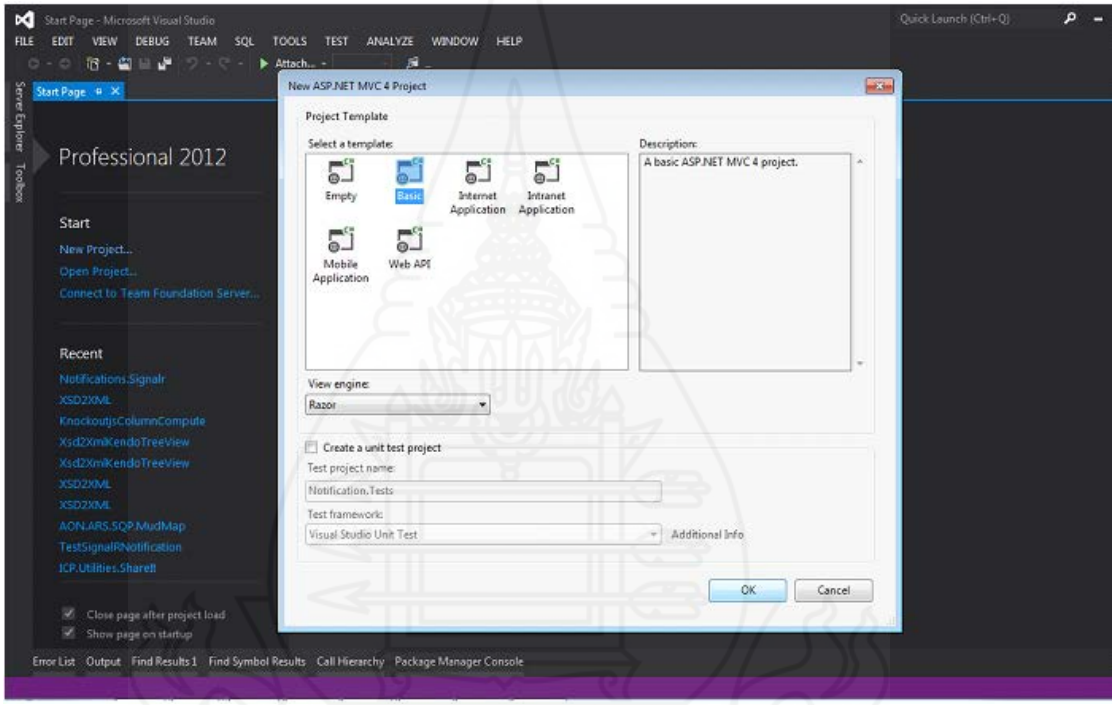
2. เครื่องมือที่ใช้ในการดำเนินโครงการ

2.1 Microsoft Visual Studio 2015 Express

2.2 ASP.NET MVC Web Application

2.3 Network tool ของ Internet Explore เวอร์ชัน 11

2.4 Resource Monitor tool

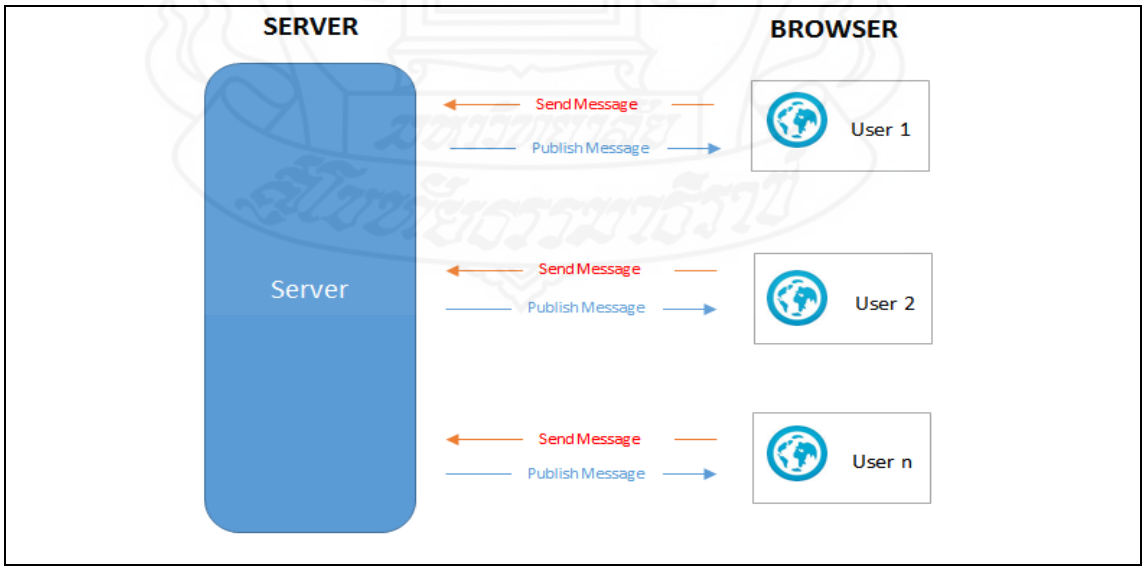


ภาพที่ 3.1 รูปภาพแสดงการสร้างโปรแกรมด้วย ASP.NET MVC Web Application

บทที่ 4
การออกแบบและพัฒนาระบบ

1. ข้อมูลระบบแบบดั้งเดิม

ระบบแบบดั้งเดิมมีการทำงานแบบ การปรับปรุงข้อมูลจากการจับเวลาหน่วยเป็นวินาที ที่ให้ระบบต้องเข้าไปดาวน์โหลดข้อมูลทุกๆ 1 วินาทีโดยการเรียกใช้งานผ่านเทคโนโลยีเอแจ็กซ์ (AJAX: Asynchronous JavaScript and XML) เป็นกลุ่มของเทคนิคในการพัฒนาเว็บแอบพลิเคชัน เพื่อให้ความสามารถโต้ตอบกับผู้ใช้ได้ดีขึ้น โดยการรับส่งข้อมูลในฉากหลัง ซึ่งทำให้เครื่องเซิร์ฟเวอร์ทำงานหนักเพราะมีกระบวนการต่างๆ ที่เครื่องเซิร์ฟเวอร์ต้องทำงานให้อาทิเช่น จำนวนสายที่เข้ามาทั้งหมด (Total Call) การคำนวณค่าผลรวมของเวลาการใช้สาย (Total Talk Time) การแสดงสถานะพนักงานรับสาย (Agent) ค่าเฉลี่ยการใช้งานสายต่อพนักงาน (Average Talk Time / Agent) การแบ่งสายตามหมายเลขโทรศัพท์ที่ให้บริการ (Latest Chanel) ซึ่งการทำงานเหล่านี้แยกการร้องขอข้อมูลจากเครื่องเซิร์ฟเวอร์ออกเป็นส่วนๆ ทุกๆ 1 วินาที ระบบต้องคำนวณค่าต่างๆ เหล่านี้เพื่อส่งข้อมูลมายังเครื่องไคลเอนต์ หากจำนวนเครื่องไคลเอนต์ มีการใช้งานมากกว่า 1 เครื่องขึ้นไป ก็จะทำให้การร้องขอข้อมูลจากเครื่องเซิร์ฟเวอร์มีจำนวนมากขึ้นตามลำดับ



ภาพที่ 4.1 รูปภาพแสดงการทำงานของระบบดั้งเดิม

2. การร้องของข้อมูลของระบบดั้งเดิม

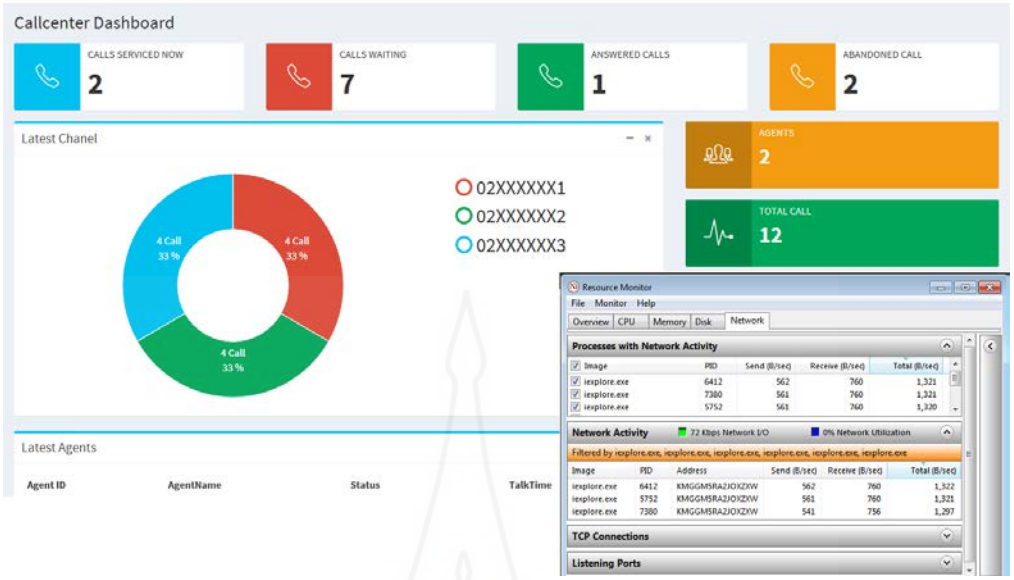
รูปแบบการร้องขอข้อมูลในระบบแบบดังเดิมจะใช้จาวาสคริปเพื่อร้องขอข้อมูลจากเครื่องเซิร์ฟเวอร์ทำการร้องขออยู่เบื้องหลังทุกๆ 1 วินาที

```
$(document).ready(function () {
    setTimeout("CountCall()", "1000");
});

function CountCall() {
    setTimeout("CountCall()", "1000");
    var url = '@Url.Action("GetCountCallR", "Control")';
    $.post(url, function (data) {
        var totalcall = 0;
        for (var i = 0; i < data.length; i++) {
            $("#" + data[i].CallTypeCode)[0].innerHTML = data[i].CallCount;
            totalcall = totalcall + data[i].CallCount;
        }
        $("#totalcall")[0].innerHTML = totalcall;
    });
}
```

โค๊ดการร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์โดยการใช้คุณสมบัติของฟังก์ชั่น “setTimeout” ของจาวาสคริป (JavaScript) เพื่อร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์โดยที่เว็บเซิร์ฟเวอร์จะทำการประมวลผลข้อมูลและส่งข้อมูลออกมาให้กับเว็บบราวเซอร์ในเครื่องไคลเอนต์

เมื่อเปิดหน้าต่างเว็บบราวเซอร์ขึ้นมามากกว่า 1 หน้าต่างหรือมี User ที่สามารถเปิดหน้าต่างบราวเซอร์นี้ขึ้นมาได้ พบว่าการทำงานจากโปรแกรม Resource Monitor มีอัตราการถ่ายโอนข้อมูลทุกๆ 1 วินาทีตามการร้องขอข้อมูลจาวาสคริปที่ได้กำหนดไว้ในหน้าโปรแกรม



ภาพที่ 4.2 แสดงการทำงานของโปรแกรม Resource Monitor ที่ตรวจจับการร้องขอข้อมูลของระบบดั้งเดิม

3. ขั้นตอนการพัฒนาระบบใหม่ด้วยเทคโนโลยี SignalR

จากที่ทราบปัญหาเบื้องต้นของระบบแบบดั้งเดิมที่ทำการร้องขอข้อมูลตลอดเวลาเป็นผลทำให้ Server ทำงานหนักและอาจเสี่ยงต่อการล่มของระบบได้ หากมีผู้ใช้งานเป็นจำนวนมากและมีข้อมูลที่มากขึ้นผู้ดำเนินโครงการจึงได้คิดที่จะนำเทคโนโลยี SignalR มาใช้เพื่อแก้ไขปัญหาดังกล่าวโดยมีขั้นตอนการพัฒนาระบบตามของวงจรชีวิตของการพัฒนาซอฟต์แวร์ หรือ Software Development Life Cycle (SDLC) เป็นโครงร่างหรือแนวทางวิธีการ เพื่อใช้ทำความเข้าใจและเพื่อใช้เป็นขั้นตอนการพัฒนา ระบบสารสนเทศ

3.1 ขั้นตอนการพัฒนาระบบใหม่ด้วยเทคโนโลยี SignalR

3.1.1 เปิดใช้งาน Service Broker บน Microsoft SQL Server

```
ALTER DATABASE Callcenter_New SET ENABLE_BROKER WITH ROLLBACK IMMEDIATE ;
```

3.1.2 เปิดใช้งาน SQL Dependency โดยการเริ่มต้นการใช้งานและการสิ้นสุดการใช้งานของไฟล์ Global.asax

```
public class MvcApplication : System.Web.HttpApplication
{
    String connString =
    ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;

     protected void Application_Start()
     {
        AreaRegistration.RegisterAllAreas();
        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);
        GlobalConfiguration.Configure(WebApiConfig.Register);
        SqlDependency.Start(connString);
     }

     protected void Application_End()
     {
         SqlDependency.Stop(connString);
     }
  }
```

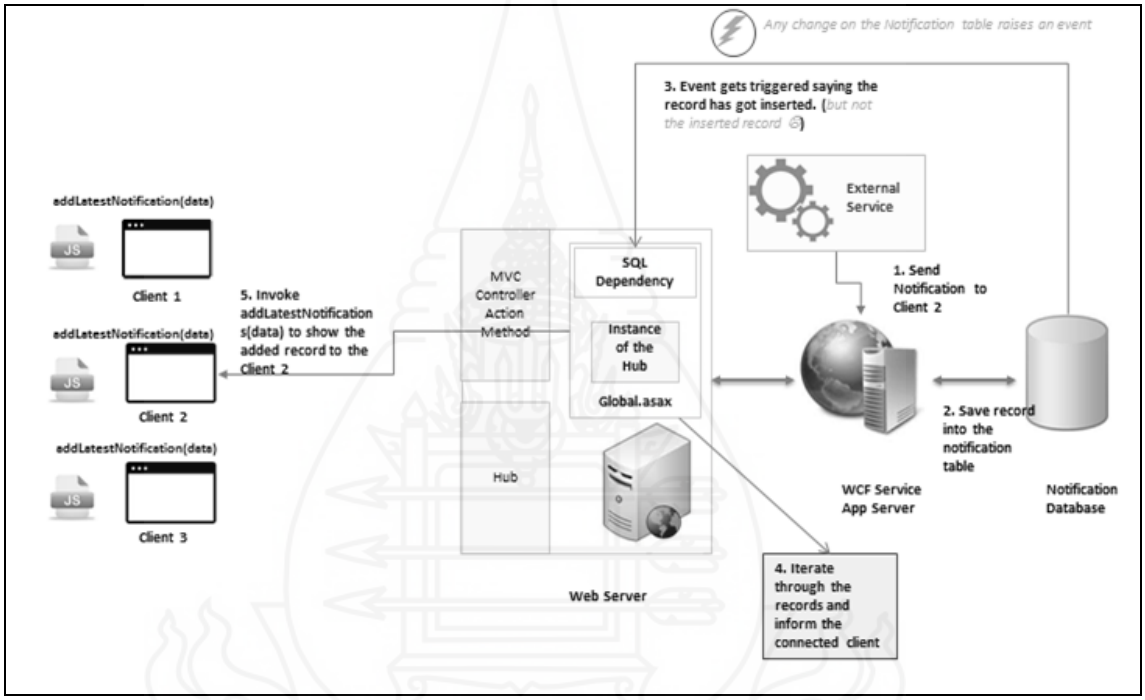
3.1.3 การลงทะเบียน SignalR เพื่อนำโค๊ดและสคริปที่พร้อมใช้งานจาก ASP.Net มาใช้งานบนโปรเจ็คที่สร้างขึ้นผ่านหน้าจอ Package Manager Console ที่ทำงานบน Visual studio 2015 ด้วยคำสั่ง

```
Install-Package Microsoft.AspNet.SignalR
```

3.1.4 ขั้นตอนการพัฒนาโปรแกรมที่จัดทำในส่วนของ SignalR Hub Class, Data Repository, SignalR Startup Class, การแสดงผลข้อมูลที่รับค่าจาก SignalR จะกล่าวแบบระเอียดในบทที่ 4 ในลำดับถัดไป

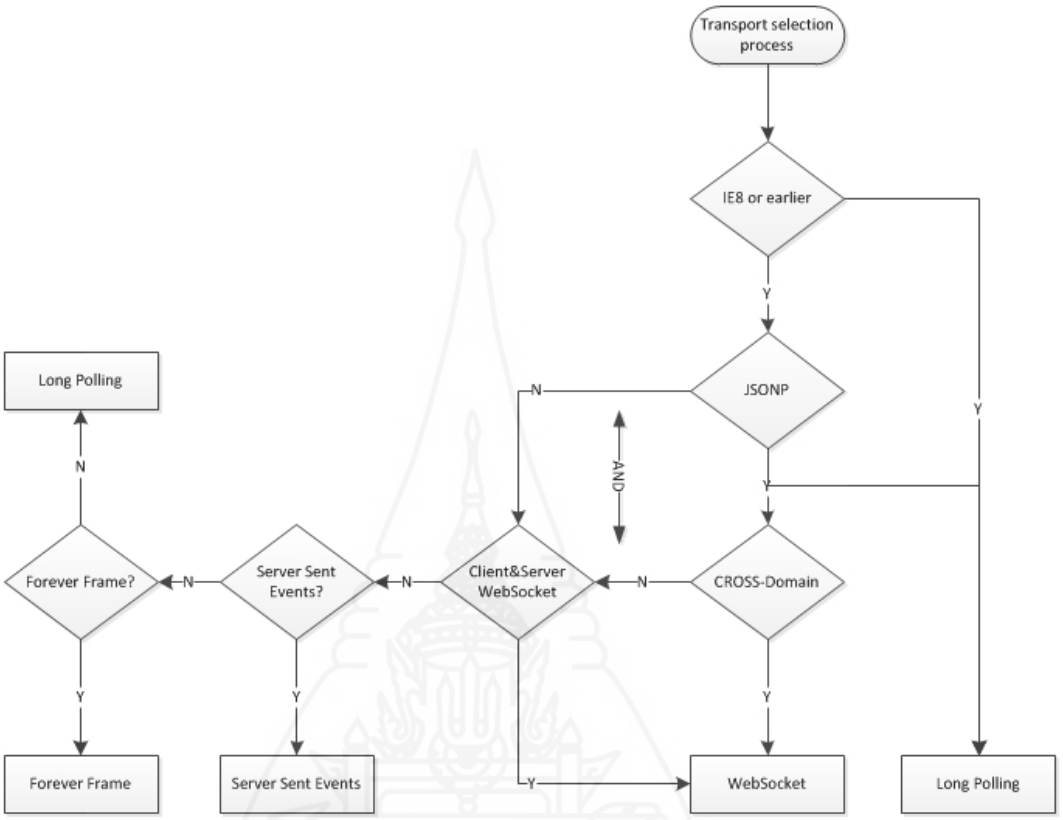
3.2 แผนภาพการทำงานของ SQL Server Service Broker

แผนภาพการทำงานร่วมกันระหว่าง SQL Server Service Broker และ SQL Dependency ร่วมกับการทำงานของ SignalR ที่สามารถตรวจสอบการการปรับปรุงข้อมูลในฐานข้อมูลและส่งผ่านข้อมูลออกมาให้กับเครื่องไคลแอนท์ได้



ภาพที่ 4.3 รูปภาพแสดงการส่งผ่านข้อมูลที่มีการปรับปรุงในระบบฐานข้อมูล (ที่มา:www.asp.net/signalR)

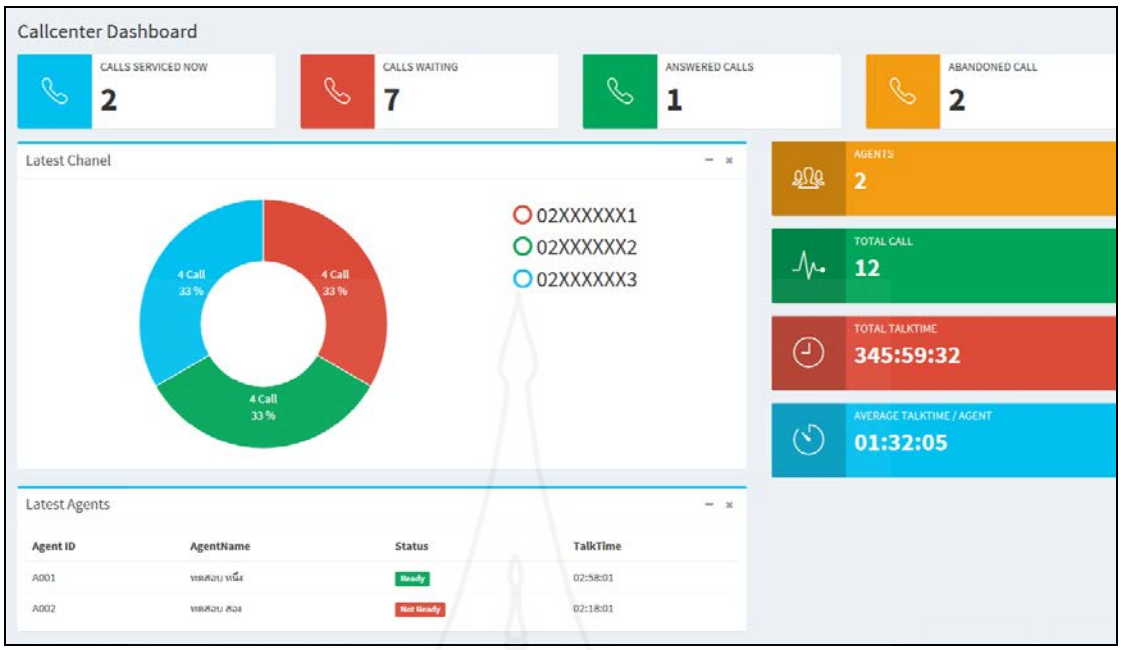
4. การวิเคราะห์และออกแบบระบบใหม่



ภาพที่ 4.4 รูปภาพแสดงการรับและส่งข้อมูลของ SignalR

4.1 การออกแบบหน้าจอแสดงผลข้อมูล

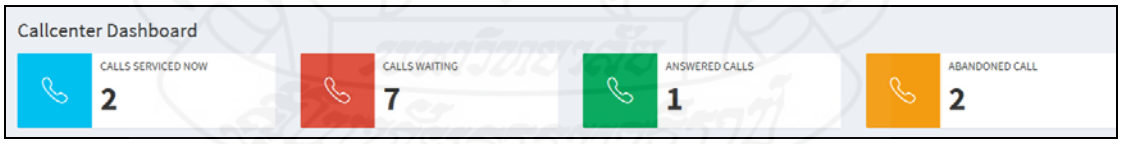
ใช้การแสดงผลหน้าจอที่มีอยู่ในระบบเดิมโดยนำโครงสร้างหน้าจอระบบเดิมมาปรับใช้ร่วมกับเทคโนโลยี SignalR



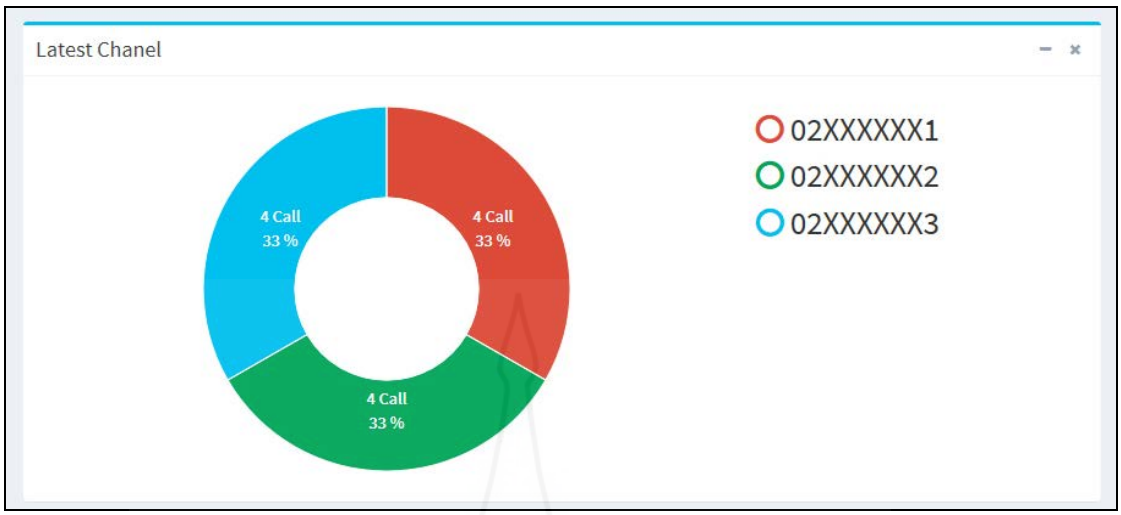
ภาพที่ 4.5 รูปภาพรวมของหน้าจอ Call center Dashboard

หน้าจอ Call center Dashboard ประกอบไปด้วย 4 ส่วนหลักได้แก่

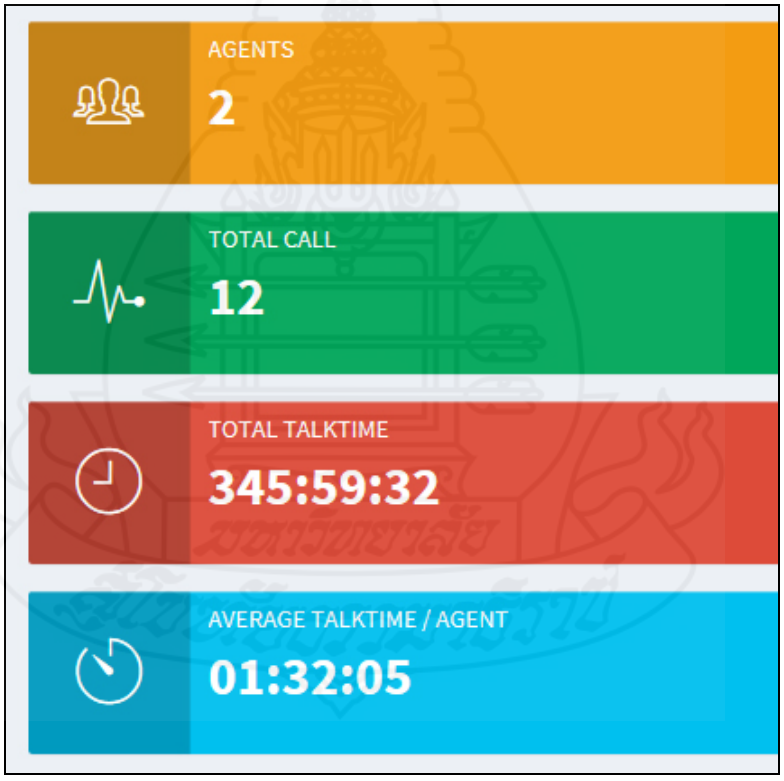
1) ส่วนการแสดงสายโทรเข้า
2) ส่วนการแสดงหมายเลขที่ให้บริการ
3) ส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงาน
4) ส่วนของการแสดงข้อมูลพนักงานที่กำลังใช้สายและพร้อมรับสาย



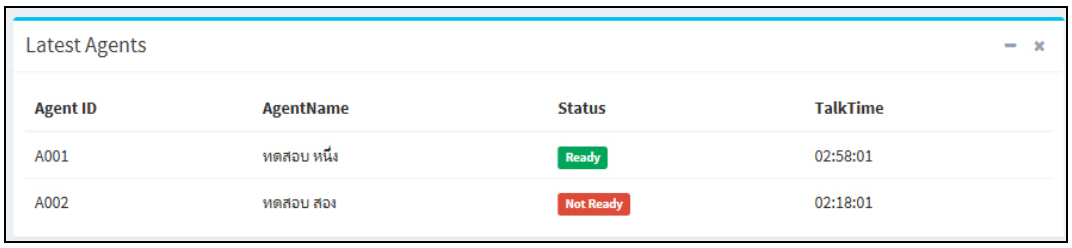
ภาพที่ 4.6 รูปภาพหน้าจอส่วนการแสดงสายโทรเข้า



ภาพที่ 4.7 รูปภาพหน้าจอส่วนการแสดงหมายเลขที่ให้บริการ



ภาพที่ 4.8 รูปภาพหน้าจอส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงาน



ภาพที่ 4.9 รูปภาพหน้าจอส่วนของการแสดงข้อมูลพนักงานที่กำลังใช้สายและพร้อมรับสาย

ตารางที่ 4.1 ตารางแสดงความหมายและรายละเอียดต่างๆ ของหน้าจอ Call center Dashboard ส่วนการแสดงสายโทรเข้า

หัวข้อ	ความหมาย
Calls Serviced Now	สายโทรศัพท์ที่กำลังอยู่ในคิวการสนทนาอยู่
Calls Waiting	สายโทรศัพท์ที่กำลังอยู่ระหว่างการรอสายสนทนา
Answered Calls	สายโทรศัพท์ที่ถูกรับสายเรียบร้อยแล้ว
Abandoned Call	สายที่ไม่ได้รับ
Latest Chanel	หมายเลขโทรศัพท์ที่ให้บริการ

ตารางที่ 4.2 ตารางแสดงความหมายและรายละเอียดต่างๆ ของหน้าจอ Call center Dashboard ส่วนการแสดงหมายเลขที่ให้บริการ

หัวข้อ	ความหมาย
Agents	จำนวนพนักงานที่ออนไลน์ในระบบ
Total Call	จำนวนผลรวมของสายโทรศัพท์ทั้งหมด
Total Talk Time	จำนวนผลรวมของเวลาในการสนทนา
Average TalkTime / Agent	จำนวนค่าเฉลี่ยของเวลาในการสนทนา ต่อพนักงาน

ตารางที่ 4.3 ตารางแสดงความหมายและรายละเอียดต่างๆ ของหน้าจอ Call center Dashboard ส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงาน

หัวข้อ	ความหมาย
Latest Chanel	จำนวนสายสนทนาของแต่ละ Chanel

ตารางที่ 4.4 ตารางแสดงความหมายและรายละเอียดต่างๆ ของหน้าจอ Call center Dashboard ส่วนของการแสดงข้อมูลพนักงานที่กำลังใช้สายและพร้อมรับสาย

หัวข้อ	ความหมาย
Latest Agents	แสดงรายชื่อพนักงานที่ออนไลน์ในระบบ
Agent ID	รหัสพนักงาน
AgentName	ชื่อพนักงาน
Status	สถานะ Ready คือพนักงานพร้อมรับงาน สถานะ Not Ready คือพนักงานอยู่ระหว่างการสนทนา
Talk Time	ผลรวมเวลาในการสนทาของพนักงาน

4.2 การออกแบบฐานข้อมูล

ฐานข้อมูลที่ใช้จะทำการจัดเก็บข้อมูลที่มาจาก Server PBX ที่สามารถจับเหตุการณ์ต่างๆที่ลูกค้าโทรเข้ามาใช้งานระบบคอลเซ็นต์เตอร์โดยมี Windows service ที่ใช้สำหรับกรองข้อมูลเหตุการณ์เหล่านั้นมาใส่ลงในฐานข้อมูลที่ถูกออกแบบไว้โดยมีตารางที่เกี่ยวข้องดังนี้

ตารางที่ 4.5 ตารางแสดงการออกแบบฐานข้อมูลตาราง Agent (พนักงาน)

ON	Attribute Name	Description	Data type (size)	Key type	Reference Table
1	AgentID	รหัสพนักงาน	int	PK	
2	AgentTitle	คำนำหน้าชื่อ	varchar(50)		
3	AgentFirstName	ชื่อ	varchar(50)		
4	AgentLastName	สกุล	varchar(50)		
5	IsActive	สถานะการใช้งาน	bit		
6	Status	สถานะการใช้สาย	varchar(50)		

ตารางที่ 4.6 ตารางแสดงการออกแบบฐานข้อมูลตาราง CallType (สถานะสายสนทนา)

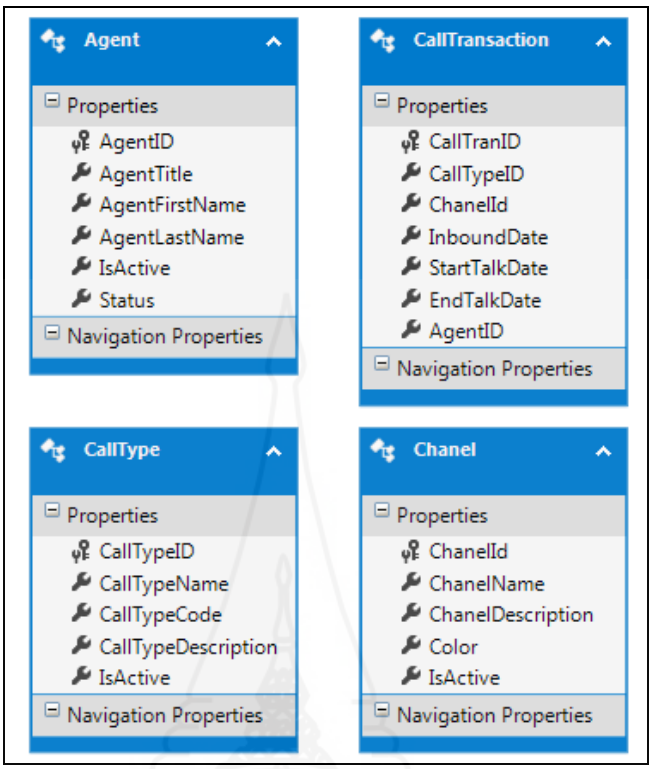
ON	Attribute Name	Description	Data type (size)	Key type	Reference Table
1	CallTypeID	รหัสสถานะ	int	PK	
2	CallTypeName	ชื่อสถานะ	varchar(50)		
3	CallTypeCode	รหัสย่อยสถานะ	varchar(50)		
4	CallTypeDescription	รายละเอียดสถานะ	varchar(50)		
5	IsActive	สถานะการใช้งาน	bit		

ตารางที่ 4.7 ตารางแสดงการออกแบบฐานข้อมูลตาราง Chanel (ช่องทางติดต่อ)

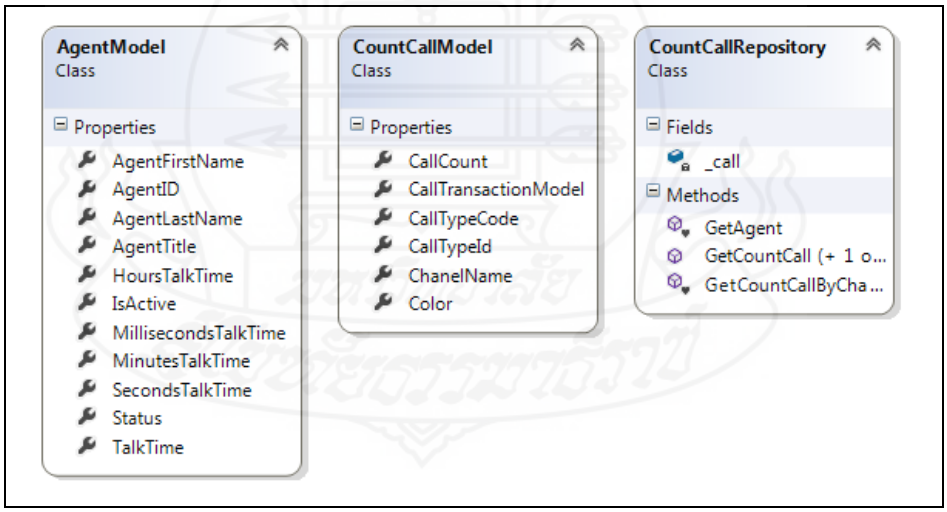
ON	Attribute Name	Description	Data type (size)	Key type	Reference Table
1	ChanelId	รหัส	int	PK	
2	ChanelName	เบอร์โทรที่ให้บริการ			
3	ChanelDescription	รายละเอียด			
4	Color	สีที่แสดงกราฟหน้าโปรแกรม			
5	IsActive	สถานะการใช้งาน			

ตารางที่ 4.8 ตารางแสดงการออกแบบฐานข้อมูลตาราง CallTransaction (ประวัติสาย-สนทนา)

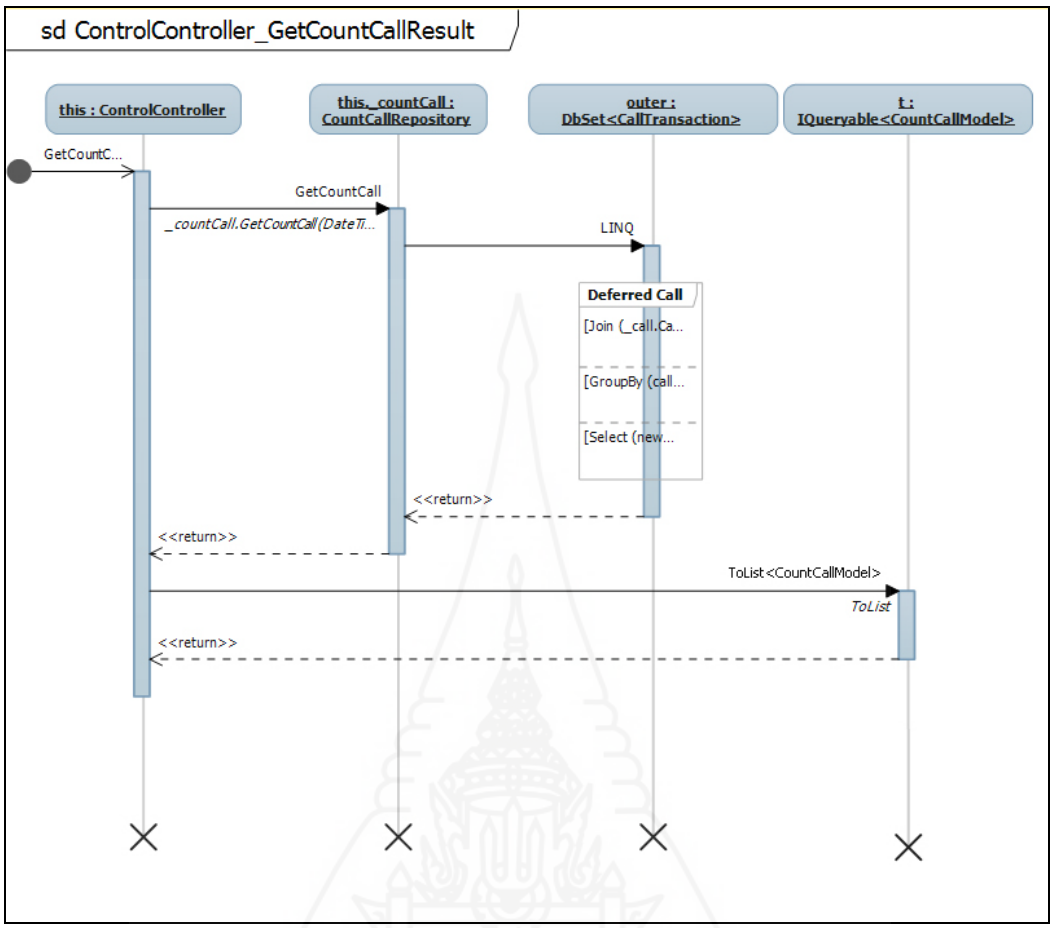
ON	Attribute Name	Description	Data type (size)	Key type	Reference Table
1	CallTranID	รหัส	int	PK	
2	CallTypeID	รหัสสถานะ	int		CallType
3	ChanelId	รหัสช่องทางติดต่อ	int		Chanel
4	InboundDate	วันที่รับสาย	DateTime		
5	StartTalkDate	วันที่-เวลาเริ่มสนทนา	DateTime		
6	EndTalkDate	วันที่-เวลาสิ้นสุดการสนทนา	DateTime		
7	AgentID	รหัสพนักงาน	int		Agent



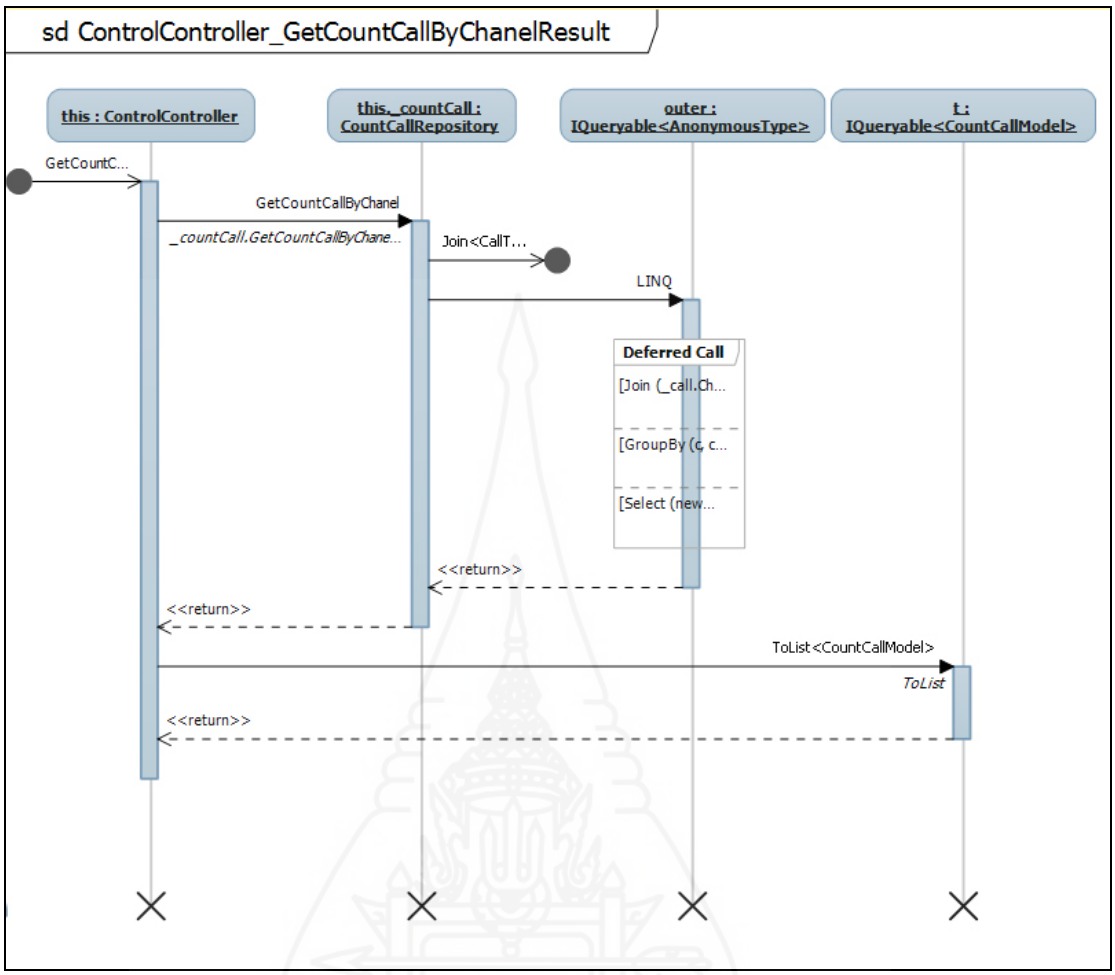
ภาพที่ 4.10 รูปภาพแสดงรายการตารางฐานข้อมูล



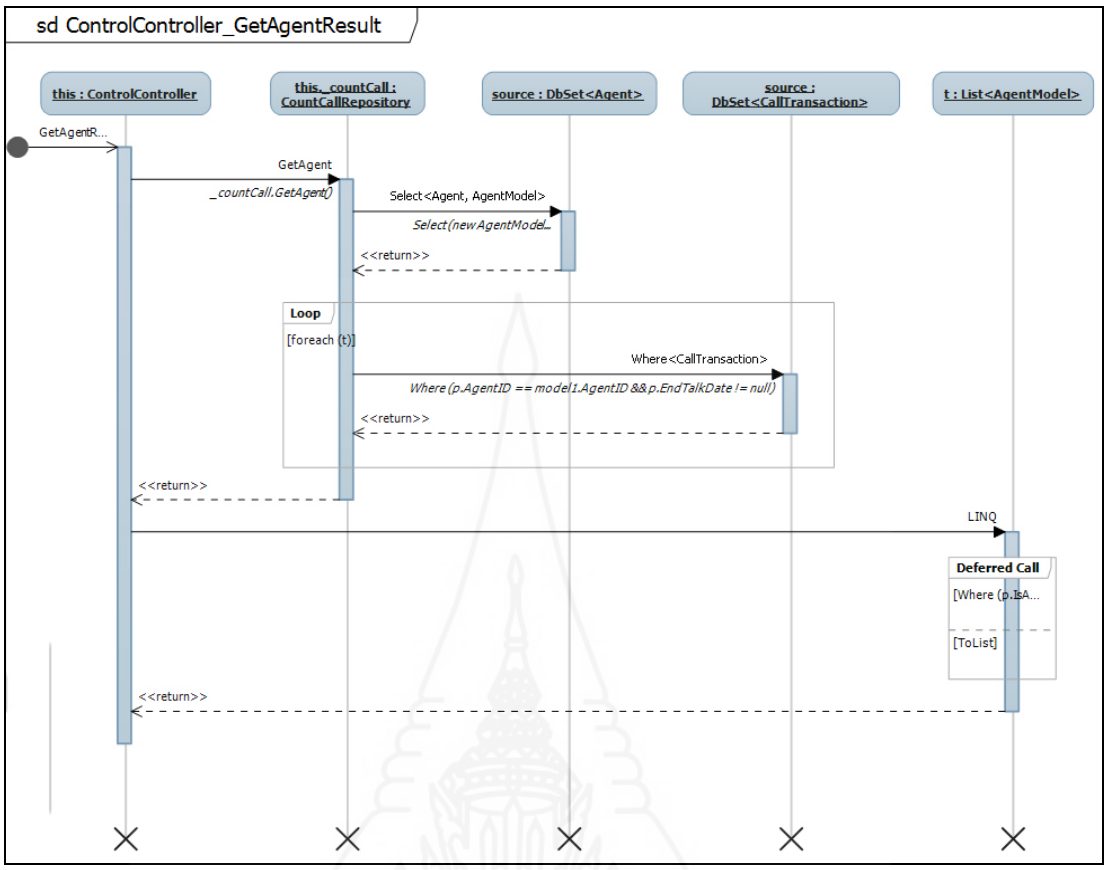
ภาพที่ 4.11 รูปภาพแสดงรายการคลาสที่ใช้ภายในระบบ



ภาพที่ 4.12 รูปภาพแสดง Sequence diagram ของการเรียกใช้งานหน้าจอส่วนที่ 1 ส่วนการแสดงสายโทรเข้า



ภาพที่ 4.13 รูปภาพแสดง Sequence diagram ของการเรียกใช้งานหน้าจอส่วนที่ 2 ส่วนการแสดงหมายเลขที่ให้บริการและส่วนที่ 3 ส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงาน



ภาพที่ 4.14 รูปภาพแสดง Sequence diagram ของการเรียกใช้งานหน้าจอส่วนที่ 3 ส่วนของสรุปยอดสายโทรเข้าและค่าเฉลี่ยต่อพนักงานและส่วนที่ 4 ส่วนของการแสดงข้อมูลพนักงานที่กำลังใช้สายและพร้อมรับสาย

4.3 ซอร์สโค้ดการทำงานของ SignalR

1) โค๊ดแสดงการทำงานของระบบการแจ้งเตือนการปรับปรุงฐานข้อมูลด้วยคุณสมบัติของ SQL Dependency

```
public sealed class ChangeNotifier : IDisposable
        {
                private SqlConnection _con;
                private SqlCommand _cmd;
                private SqlDependency _dependency;
                private String _connectionString;
```

```
        private String _dependencyCheckSql;

        ~ChangeNotifier()
        {
            Dispose();
        }

        public event EventHandler<ChangeEventArgs> Change;

        public Boolean Start(String connectionStringName, String
dependencyCheckSql)
        {
            new SqlClientPermission(PermissionState.Unrestricted).Demand();

            _connectionString =
ConfigurationManager.ConnectionStrings[connectionStringName].ConnectionString;
            _dependencyCheckSql = dependencyCheckSql;

            var result = SqlDependency.Start(_connectionString);

            _con = new SqlConnection(_connectionString);
            _con.Open();

            _cmd = _con.CreateCommand();
            _cmd.CommandText = _dependencyCheckSql;

            Setup(true);

            return (result);
        }
```

```
public Boolean Stop()
{
        var result = false;

        if (_cmd != null)
        {
                _cmd.Notification = null;
                _cmd.Dispose();
                _cmd = null;
        }

        if (_con != null)
        {
                _con.Close();
                _con = null;
        }

        if (_dependency != null)
        {
                result = SqlDependency.Stop(_connectionString);
                _dependency.OnChange -= OnChange;
                _dependency = null;
        }

        Change = null;

        return (result);
}
```

```
        private void Setup(Boolean initial)
        {
            if (initial == false)
            {
                _dependency.OnChange -= OnChange;
            }

            _cmd.Notification = null;
            _dependency = new SqlDependency(_cmd);
            _dependency.OnChange += OnChange;

            _cmd.ExecuteScalar();
        }

        private void OnChange(Object sender, SqlNotificationEventArgs e)
        {
            Setup(false);

            var handler = Change;

            if (handler != null)
            {
                handler(sender, new
ChangeEventArgs((ChangeInfo)(Int32)e.Info, (ChangeSource)(Int32)e.Source,
(ChangeType)(Int32)e.Type));
            }
        }

        public void Dispose()
        {
```

```
                Stop();
            }
        }
```

2) โค๊ดแสดงการทำงานของเหตุการณ์การตรวจการปรับปรุงข้อมูลในระบบฐานข้อมูล

```
public sealed class ChangeEventArgs : EventArgs
    {
        public ChangeEventArgs(ChangeInfo info, ChangeSource source, ChangeType type)
        {
            this.Info = info;
            this.Source = source;
            this.Type = type;
        }

        public ChangeInfo Info { get; private set; }

        public ChangeSource Source { get; private set; }

        public ChangeType Type { get; private set; }
    }
```

3) คลาสการเปรียบเทียบค่าข้อมูลการเปลี่ยนแปลงฐานข้อมูล

```
public enum ChangeInfo
    {
        AlreadyChanged = -2,
```

```
        Unknown = -1,
        Truncate = 0,
        Insert = 1,
        Update = 2,
        Delete = 3,
        Drop = 4,
        Alter = 5,
        Restart = 6,
        Error = 7,
        Query = 8,
        Invalid = 9,
        Options = 10,
        Isolation = 11,
        Expired = 12,
        Resource = 13,
        PreviousFire = 14,
        TemplateLimit = 15,
        Merge = 16,
}
public enum ChangeSource
{
        Client = -2,
        Unknown = -1,
        Data = 0,
        Timeout = 1,
        Object = 2,
        Database = 3,
        System = 4,
        Statement = 5,
        Environment = 6,
```

```
        Execution = 7,
        Owner = 8,
    }

    public enum ChangeType
    {
        Unknown = -1,
        Change = 0,
        Subscribe = 1,
    }
```

ตารางที่ 4.9 ตารางแสดงความหมายของคลาส ChangeSource

หัวข้อ	ความหมาย
AlreadyChanged = -2	การเปลี่ยนแปลงโครงสร้างฐานข้อมูล
Unknown = -1	ไม่สามารถระบุการเปลี่ยนแปลงของระบบฐานข้อมูล
Truncate = 0	การลบข้อมูลพร้อมทั้ง reset auto increment ให้เริ่มนับใหม่
Insert = 1	การเพิ่มข้อมูลใหม่
Update = 2	การปรับปรุงข้อมูล
Delete = 3	การลบข้อมูล
Drop = 4	การลบตาราง
Alter = 5	การปรับปรุงโครงสร้าง
Restart = 6	การเริ่มต้นใหม่
Error = 7	เกิดข้อผิดพลาด
Query = 8	การ Query ข้อมูล เช่นคำสั่ง Select * from
Invalid = 9	ข้อมูลผิด
Options = 10	กำหนดคุณสมบัติ
Isolation = 11	การ Lock ข้อมูลที่แตกต่างกัน
Expired = 12	หมดอายุ

ตารางที่ 4.9 (ต่อ)

หัวข้อ	ความหมาย
Resource = 13	การปรับปรุงคำอธิบาย
PreviousFire = 14	เวอร์ชั่นผิดพลาด
TemplateLimit = 15	ข้อผิดพลาดการออกแบบตาราง
Merge = 16	การผสานเซลล์

ตารางที่ 4.10 ตารางแสดงความหมายของคลาส ChangeInfo เป็นเหตุการณ์ที่เกี่ยวข้องกับการเชื่อมต่อข้อมูลระหว่างอุปกรณ์เครื่องเซอร์ฟเวอร์ไปยังเซอร์ฟเวอร์ฐานข้อมูล

หัวข้อ	ความหมาย
Client = -2,	การเปลี่ยนแปลงที่เกิดจากเครื่อง Client
Unknown = -1	ไม่สามารถระบุเหตุการณ์ได้
Data = 0	การเปลี่ยนแปลงของข้อมูล
Timeout = 1	เกินระยะเวลาที่กำหนด
Object = 2	การเปลี่ยนแปลงโครงสร้าง
Database = 3	การเปลี่ยนชื่อ Database
System = 4	การเปลี่ยนแปลงระบบ
Statement = 5	การเปลี่ยนแปลงรูปแบบคำสั่ง
Environment = 6	การเปลี่ยนแปลงค่า CLASSPATH
Execution = 7	การเรียกใช้ชุดคำสั่ง เช่นคำสั่ง exec stored
Owner = 8	การเปลี่ยนแปลงที่เกิดจากผู้มีสิทธิ์การทำรายการ

ตารางที่ 4.11 ตารางแสดงความหมายของคลาส ChangeType เป็นเหตุการณ์ที่เกี่ยวข้องกับประเภทการเปลี่ยนแปลง

หัวข้อ	ความหมาย
Unknown = -1	ไม่สามารถระบุเหตุการณ์ได้
Change = 0,	การเปลี่ยนแปลงข้อมูล
Subscribe = 1	การเปลี่ยนแปลงข้อกำหนดของข้อมูล

4.4 รูปแบบการติดต่อระหว่าง SignalR Hub และ HTML

```
<script src="~/Scripts/jquery.signalR-2.2.0.js"></script>
<script type="text/javascript" src="signalr/hubs"></script>
<script type="text/javascript">
    $(document).ready(function () {
        var notifier = $.connection.changeNotificationHub;
        notifier.client.countCallSignalR = CountCallSignalR;
        notifier.client.countCallByChanelSignalR = CountCallByChanelSignalR;
        notifier.client.getAgentSignalR = GetAgentSignalR;
        $.connection.hub.start();
    });

    function CountCallSignalR(data) {
        var totalcall = 0;
        for (var i = 0; i < data.length; i++) {
            $("#" + data[i].CallTypeCode)[0].innerHTML = data[i].CallCount;
            totalcall = totalcall + data[i].CallCount;
        }
        $("#totalcall")[0].innerHTML = totalcall;
    }
    function CountCallByChanelSignalR(data) {
```

```
        var donutData = [];
        for (var i = 0; i < data.length; i++) {
            donutData.push({ label: data[i].ChanelName, data: data[i].CallCount, color:
data[i].Color });
        }
        $.plot("#donut-chart", donutData, {
            series: {
                pie: {
                    show: true,
                    radius: 1,
                    innerRadius: 0.5,
                    label: {
                        show: true,
                        radius: 2 / 3,
                        formatter: labelFormatter,
                        threshold: 0.1
                    }
                }
            }
        });
    }

    function GetAgentSignalR(data) {
        var agentData = [];
        for (var i = 0; i < data.length; i++) {
            agentData.push({ AgentID: data[i].AgentID, AgentName: data[i].AgentFirstName + ' '
+ data[i].AgentLastName, TalkTime: data[i].TalkTime });
        }
        $("#countAgent")[0].innerHTML = data.length;
    }
```

บทที่ 5
ผลการดำเนินโครงการ

1. การประเมินผล

การดำเนินโครงการครั้งนี้เป็นการดำเนินโครงการและพัฒนาระบบเรื่อง การปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์ด้วยเทคโนโลยีซิกแนลอาร์ มีวัตถุประสงค์ของการดำเนินโครงการ (1) เพื่อปรับปรุงประสิทธิภาพระบบแสดงผลข้อมูลของคอลเซ็นเตอร์โดยการพัฒนาระบบการแสดงข้อมูลการใช้งานโทรศัพท์ด้วยเทคโนโลยีซิกแนลอาร์ (2) เพื่อพัฒนาต้นแบบระบบการแสดงผลแบบ Real-time ด้วยเทคโนโลยี SignalR ผู้ดำเนินการใช้หลักการประเมินประสิทธิภาพโดยการเปรียบเทียบการใช้งานทรัพยากรเครื่องเซอร์ฟเวอร์ดังนี้

การประเมินระบบปัจจุบันแบ่งการประเมินออกเป็น 2 รอบคือ

(1) ประเมินการเปิดการแสดงผลหน้าจอเดียวเป็นเวลา 10 นาที

(2) ประเมินการเปิดการแสดงผลหน้าจอ 3 หน้าจอเป็นเวลา 10 นาที

การประเมินระบบใหม่ที่ถูกพัฒนาโดยการนำเทคโนโลยี SignalR เข้ามาใช้ แบ่งการประเมินออกเป็น 2 รอบคือ

(1) ประเมินการเปิดการแสดงผลหน้าจอเดียวเป็นเวลา 10 นาที

(2) ประเมินการเปิดการแสดงผลหน้าจอ - หน้าจอเป็นเวลา 10 นาที

การประเมินระบบปัจจุบันและระบบใหม่ทำการประเมินบนฐานข้อมูลแบ่งข้อมูลออกเป็น 2 ประเภท ได้แก่ข้อมูลพนักงานรับสายจำนวน 50 ระเบียน (Record) และ ข้อมูลสายสนทนาที่เข้ามาในระบบจำนวน 1,000 ระเบียน (Record)

2. ผลการประเมิน

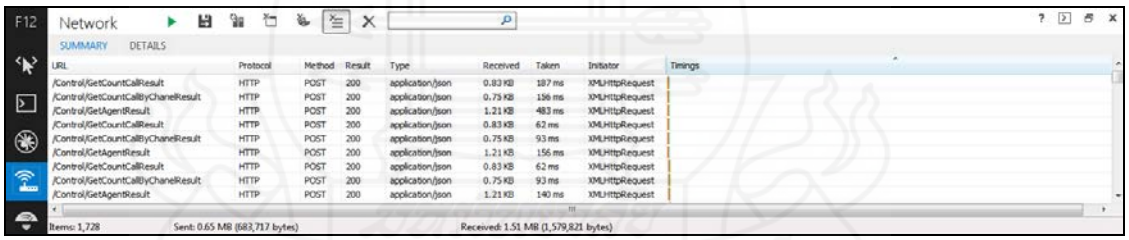
1. การพัฒนาระบบใหม่ที่นำเทคโนโลยี SignalR เข้ามาใช้ มีการนำหลักการของ SQL Dependency และ SQL service Broker เข้ามาใช้ร่วมด้วย โดยใช้ โปรแกรมสำเร็จรูปในการพัฒนา

ระบบซึ่งจะใช้ซอฟต์แวร์ของไมโครซอฟต์ชื่อว่า Microsoft visual studio 2013 เป็นตัวพัฒนาระบบ และใช้ Microsoft SQL server 2014 เป็นระบบฐานข้อมูล

2. ประเมินผลโดยการใช้โปรแกรมสำเร็จรูปในการวัดค่าการประเมินแบ่งออกเป็น 2 โปรแกรมได้แก่ (1) โปรแกรม Resource monitor เป็นตัววัดการใช้งานทรัพยากรเครื่องเซอร์ฟเวอร์ โดยการตรวจวัดค่า CPU ได้แก่ Thread CPU และAverage CPU (2) โปรแกรม Monitor Network Internet Explorer 11 เป็นตัววัดการรับส่งข้อมูลจากเครื่องไคลเอนต์ ไปยังเครื่องเซิร์ฟเวอร์และการส่งข้อมูลกลับจากเครื่องเซิร์ฟเวอร์มายังเครื่องไคลเอนต์ โดยทำการตรวจวัดค่า Network ได้แก่ Send, Receive และ Total

ตารางที่ 5.1 ตารางผลการประเมินระบบปัจจุบัน โดยการเปิดหน้าจอแสดงผลข้อมูล 1 หน้าจอเป็นเวลา 10 นาที

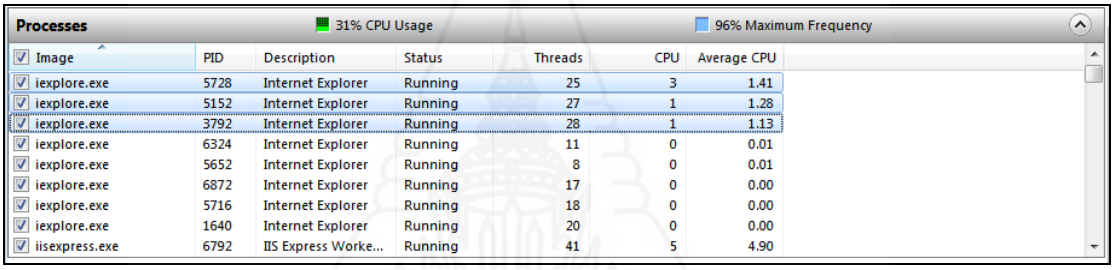
Web Browser	**CPU**		**Network**		
IE 11	**Thread**	**CPU**	**Average CPU**	**Send (B/sec)**	**Receive (B/sec)**	**Total (B/sec)**
Client 1	21	3	4.6	683,717	1,579,821	2,263,538



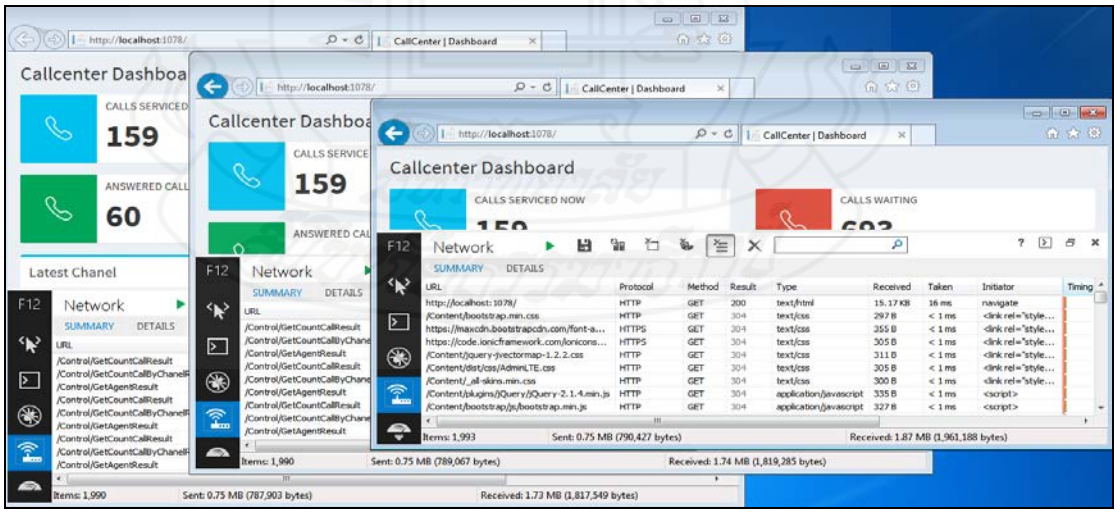
ภาพที่ 5.1 รูปภาพแสดงการตรวจวัดค่า Network จากโปรแกรม Monitor Network Internet Explorer เวอร์ชัน 11

ตารางที่ 5.2 ตารางผลการประเมินระบบปัจจุบัน โดยการเปิดหน้าจอแสดงผลข้อมูล 3 หน้าจอ พร้อมกันเวลาประเมิน 10 นาที

Web Browser	**CPU**		**Network**			
IE 11	**Thread**	**CPU**	**Average CPU**	**Send (B/sec)**	**Receive (B/sec)**	**Total (B/sec)**
Client 1	25	3	1.41	787,903	1,817,549	2,605,452
Client 2	27	1	1.28	789,067	1,819,285	2,608,352
Client 3	28	1	1.51	790,427	1,961,188	2,751,615



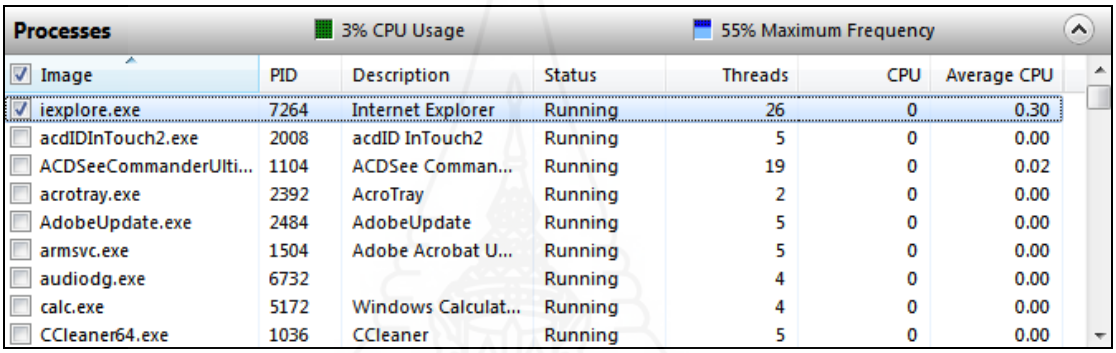
ภาพที่ 5.2 รูปภาพแสดงการตรวจวัดค่า CPU จากโปรแกรม Resource Monitor ของระบบปัจจุบัน



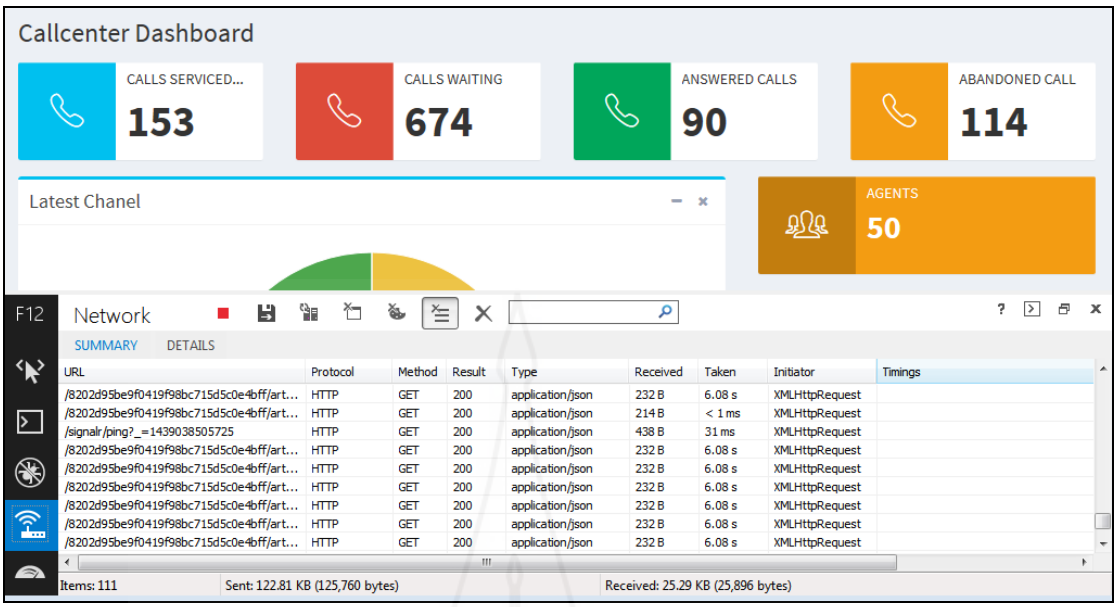
ภาพที่ 5.3 รูปภาพแสดงการตรวจวัดค่า Network จากโปรแกรม Monitor Network Internet Explorer เวอร์ชัน 11 จำนวน 3 หน้าจอ เวลาประเมิน 10 นาที

ตารางที่ 5.3 ตารางผลการประเมินระบบใหม่ที่ใช้เทคโนโลยี SignalR โดยการเปิดหน้าจอแสดงผลข้อมูล 1 เวลาประเมิน 10 นาที

Web Browser		**CPU**		**Network**		
IE 11	**Thread**	**CPU**	**Average CPU**	**Send (B/sec)**	**Receive (B/sec)**	**Total (B/sec)**
Client 1	26	0	0.30	125,760	25,896	151,656



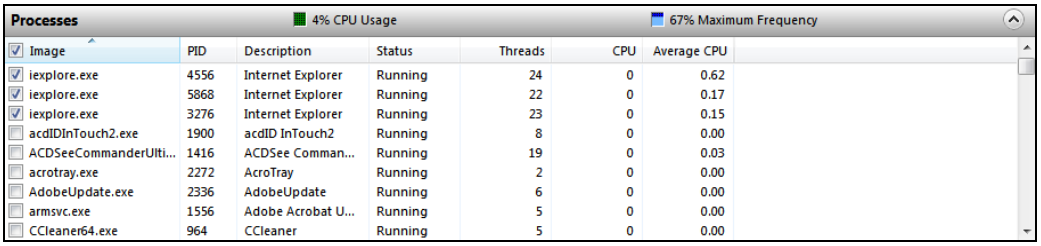
ภาพที่ 5.4 รูปภาพแสดงการตรวจวัดค่า CPU จากโปรแกรม Resource Monitor ของระบบใหม่ที่พัฒนาด้วย SignalR



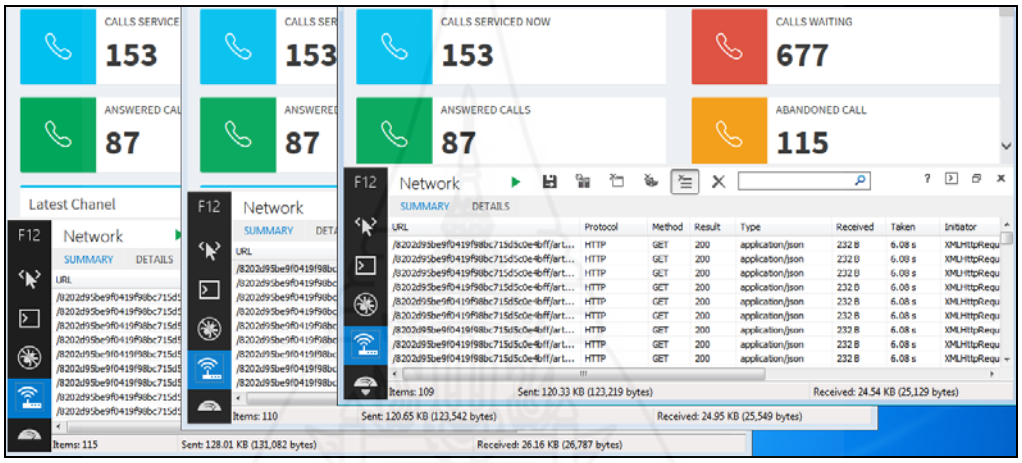
ภาพที่ 5.5 รูปภาพแสดงการตรวจวัดค่า Network จากโปรแกรม Monitor Network Internet Explorer เวอร์ชัน 11 จำนวน1 หน้าจอ เวลาประเมิน 10 นาที

ตารางที่ 5.4 ตารางผลการประเมินระบบใหม่ที่ใช้เทคโนโลยี SignalR โดยการเปิดหน้าจอแสดงผลข้อมูล 3 หน้าจอพร้อมกัน เวลาประเมิน 10 นาที

Web Browser	**CPU**			**Network**		
IE 11	**Thread**	**CPU**	**Average CPU**	**Send (B/sec)**	**Receive (B/sec)**	**Total (B/sec)**
Client 1	24	0	0.62	131,082	26,787	157,869
Client 2	22	0	0.17	123,542	25,549	149,091
Client 3	23	0	0.15	123,219	25,129	148,348



ภาพที่ 5.6 รูปภาพแสดงการตรวจวัดค่า CPU จากโปรแกรม Resource Monitor ของระบบใหม่ที่พัฒนาด้วย SignalR จำนวน 3 หน้าจอ



ภาพที่ 5.7 รูปภาพแสดงการตรวจวัดค่า Network จากโปรแกรม Monitor Network Internet Explorer เวอร์ชัน 11 จำนวน3 หน้าจอ เวลาประเมิน 10 นาที

จากผลการประเมินพบว่า ระบบปัจจุบันใช้งานทรัพยากรเครื่องผู้ให้บริการสูงกว่าระบบใหม่ที่ถูกพัฒนาด้วย SignalR เป็นอย่างมาก อีกทั้งหากมีการเปิด Browser จำนวนมากจำให้ให้การใช้งานเครื่องผู้ให้บริการ มีอัตราที่สูงขึ้นเมื่อเทียบกับการปล่อยข้อมูลโดยเครื่องผู้ให้บริการ แล้วให้เครื่องผู้ใช้บริการ เป็นผู้รับข้อมูลที่พัฒนาด้วย SignalR เครื่องผู้ใช้บริการ จะได้รับข้อมูลพร้อมกันและประหยัดทรัพยากรเครื่องผู้ให้บริการอย่างเห็นได้ชัด เนื่องจากเมื่อเปิด Browser เพิ่มขึ้นเครื่องแม่ข่ายจะไม่ถูกเพิ่มการใช้งานทรัพยากรเลยจะค่อนข้างคงที่ เมื่อเทียบกับระบบปัจจุบันที่มีอัตราการใช้งานที่สูงขึ้นเพราะเครื่องผู้ใช้บริการ ไม่ได้ประมวลผลข้อมูลเองภาระจะตกอยู่ที่เครื่องผู้ให้บริการทั้งหมด

ความแตกต่างระหว่าง URL ที่มีการติดต่อเพื่อร้องขอข้อมูลจาก Server ระหว่างระบบปัจจุบันและระบบใหม่ที่พัฒนาด้วย SignalR จะมีความแตกต่างกันในเรื่องของรูปแบบคำสั่งที่ถูก

ร้องขอไปยังเครื่อง Server โดยคำสั่งของระบบปัจจุบันจะใช้ URLประเภท application/ json ที่เป็นรูปแบบการเรียกใช้ข้อมูลผ่าน Controller และ Action Class การรับส่งข้อมูลแบบ json ดังภาพที่ 5.8 ส่วนระบบใหม่ที่พัฒนาด้วย SignalR จะใช้ URL ประเภท application/json เป็นรูปแบบURL ที่ใช้คุณสมบัติของ Long polling ดังภาพที่ 28

URL	Protocol	Method	Result	Type	Received	Taken	Initiator
/Control/GetCountCallResult	HTTP	POST	200	application/json	0.83 KB	312 ms	XMLHttpRequest
/Control/GetCountCallByChanelResult	HTTP	POST	200	application/json	0.75 KB	374 ms	XMLHttpRequest
/Control/GetAgentResult	(Pending...)	(Pen...	(Pendi...	(Pending...)	(Pendin...	(Pendin...	XMLHttpRequest
/Control/GetCountCallResult	HTTP	POST	200	application/json	0.83 KB	156 ms	XMLHttpRequest
/Control/GetCountCallByChanelResult	HTTP	POST	200	application/json	0.75 KB	281 ms	XMLHttpRequest
/Control/GetAgentResult	(Pending...)	(Pen...	(Pendi...	(Pending...)	(Pendin...	(Pendin...	XMLHttpRequest
/Control/GetCountCallResult	HTTP	POST	200	application/json	0.83 KB	171 ms	XMLHttpRequest
/Control/GetCountCallByChanelResult	HTTP	POST	200	application/json	0.75 KB	250 ms	XMLHttpRequest
/Control/GetAgentResult	(Pending...)	(Pen...	(Pendi...	(Pending...)	(Pendin...	(Pendin...	XMLHttpRequest

ภาพที่ 5.8 รูปภาพแสดงรูปแบบ URL ของระบบปัจจุบันโดยการเรียกใช้ข้อมูลผ่าน Controller และ Action Class

URL	Protocol	Method	Result	Type	Received	Taken	Initiator
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	195 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest

ภาพที่ 5.9 รูปภาพแสดงรูปแบบ URL ของระบบใหม่ที่พัฒนาด้วย SignalR จะการใช้รูปแบบ URL แบบ Long polling

บทที่ 6
อภิปรายผลและข้อเสนอแนะ

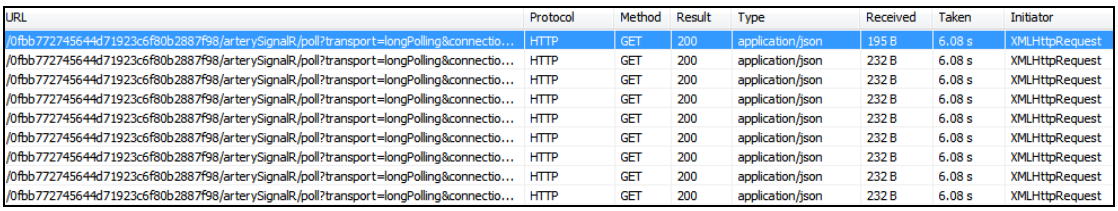
1. อภิปรายผล

จากผลการประเมินพบว่า ระบบปัจจุบันใช้งานทรัพยากรเครื่องผู้ให้บริการสูงกว่าระบบใหม่ที่ถูกพัฒนาด้วย SignalR เป็นอย่างมาก อีกทั้งหากมีการเปิด Browser จำนวนมากจำให้ให้การใช้งานเครื่องผู้ให้บริการ มีอัตราที่สูงขึ้นเมื่อเทียบกับการปล่อยข้อมูลโดยเครื่องผู้ให้บริการ แล้วให้เครื่องผู้ใช้บริการ เป็นผู้รับข้อมูลที่พัฒนาด้วย SignalR เครื่องผู้ใช้บริการ จะได้รับข้อมูลพร้อมกันและประหยัดทรัพยากรเครื่องผู้ให้บริการอย่างเห็นได้ชัด เนื่องจากเมื่อเปิด Browser เพิ่มขึ้นเครื่องแม่ข่ายจะไม่ถูกเพิ่มการใช้งานทรัพยากรเลยจะค่อนข้างคงที่ เมื่อเทียบกับระบบปัจจุบันที่มีอัตราการใช้งานที่สูงขึ้นเพราะเครื่องผู้ใช้บริการ ไม่ได้ประมวลผลข้อมูลเองภาระจะตกอยู่ที่เครื่องผู้ให้บริการทั้งหมด

ความแตกต่างระหว่าง URL ที่มีการติดต่อเพื่อร้องขอข้อมูลจาก Server ระหว่างระบบปัจจุบันและระบบใหม่ที่พัฒนาด้วย SignalR จะมีความแตกต่างกันในเรื่องของรูปแบบคำสั่งที่ถูกร้องขอไปยังเครื่อง Server โดยคำสั่งของระบบปัจจุบันจะใช้ URL ประเภท application/ json ที่เป็นรูปแบบการเรียกใช้ข้อมูลผ่าน Controller และ Action Class การรับส่งข้อมูลแบบ json ดังภาพที่ 5.8 ส่วนระบบใหม่ที่พัฒนาด้วย SignalR จะใช้ URL ประเภท application/json เป็นรูปแบบ URL ที่ใช้คุณสมบัติของ Long polling ดังภาพที่ 5.9

URL	Protocol	Method	Result	Type	Received	Taken	Initiator
/Control/GetCountCallResult	HTTP	POST	200	application/json	0.83 KB	312 ms	XMLHttpRequest
/Control/GetCountCallByChanelResult	HTTP	POST	200	application/json	0.75 KB	374 ms	XMLHttpRequest
/Control/GetAgentResult	(Pending...)	(Pen...	(Pendi...	(Pending...)	(Pendin...	(Pendin...	XMLHttpRequest
/Control/GetCountCallResult	HTTP	POST	200	application/json	0.83 KB	156 ms	XMLHttpRequest
/Control/GetCountCallByChanelResult	HTTP	POST	200	application/json	0.75 KB	281 ms	XMLHttpRequest
/Control/GetAgentResult	(Pending...)	(Pen...	(Pendi...	(Pending...)	(Pendin...	(Pendin...	XMLHttpRequest
/Control/GetCountCallResult	HTTP	POST	200	application/json	0.83 KB	171 ms	XMLHttpRequest
/Control/GetCountCallByChanelResult	HTTP	POST	200	application/json	0.75 KB	250 ms	XMLHttpRequest
/Control/GetAgentResult	(Pending...)	(Pen...	(Pendi...	(Pending...)	(Pendin...	(Pendin...	XMLHttpRequest

ภาพที่ 6.1 รูปภาพแสดงรูปแบบ URL ของระบบปัจจุบันโดยการเรียกใช้ข้อมูลผ่าน Controller และ Action Class



URL	Protocol	Method	Result	Type	Received	Taken	Initiator
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	195 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest
/0fbb772745644d71923c6f80b2887f98/arterySignalR/poll?transport=longPolling&connectio...	HTTP	GET	200	application/json	232 B	6.08 s	XMLHttpRequest

ภาพที่ 6.2 รูปภาพแสดงรูปแบบ URL ของระบบใหม่ที่พัฒนาด้วย SignalR จะการใช้รูปแบบ URL แบบ Long polling

เมื่อข้อมูลที่แสดงออกมาทางจอภาพที่ทุกคนมีสิทธิ์เห็นว่าสายโทรศัพท์ที่เข้ามามีจำนวนมากน้อยขนาดไหน ทางทีมคอลเซ็นเตอร์สามารถนำข้อมูลเหล่านั้นมาปรับใช้เพื่อพัฒนาความรวดเร็วในการให้บริการลูกค้าได้

2. ข้อเสนอแนะ

จากผลการประเมินที่ได้รับทำให้รู้ว่าเทคโนโลยี SignalR สามารถนำมาใช้ประโยชน์ในเรื่องของการติดต่อสื่อสารที่เป็นแบบเรียลไทม์ อาทิเช่นระบบ การสื่อสารผ่านอินเตอร์เน็ต ระบบการแสดงผลข้อมูลรายงานที่เป็นแบบเรียลไทม์ต่างๆ เกมส์ และ สื่อสังคมออนไลน์ เป็นต้น ซึ่งหวังว่าเทคโนโลยี SignalR คงจะเป็นประโยชน์ในการนำผลการดำเนินโครงการไปปรับใช้ได้จริงหรือนำไปพัฒนาแนวคิด หรือทฤษฎีในสาขาวิชาที่เกี่ยวข้อง อีกทั้งยังสามารถนำไปต่อยอดระบบการแสดงผลข้อมูลด้านธุรกิจที่คล้ายคลึงกันได้

สำหรับระบบคอลเซ็นเตอร์อื่นๆที่สามารถนำข้อมูลเหล่านี้ไปใช้สำหรับตั้งค่าจำนวนพนักงานที่ต้องรับสายแต่ละหมายเลขหรือใช้ในการบันทึกสายแบบอัตโนมัติเพื่อให้พนักงานติดต่อกลับภาพหลังก็สามารถทำได้ หรือแม่กระทั้งระบบรับสายเข้าที่สามารถรับสายแบบอัตโนมัติก็สามารถใช้ข้อมูลเหล่านี้ได้เช่นกัน

บรรณานุกรม

บรรณานุกรม

ธนารักษ์ ธีระมั่นคง และคณะ. (2549). เทคโนโลยีสมองกลฝังตัว. สำนักพิมพ์ ส.ส.ท.

B. Chacos, “Next-gen HTTP 2.0 protocol will require HTTPS encryption (most of the time) | PCWorld.” [Online]. Available: http://www.pcworld.com/article/2061189/next-gen-http-2-0-protocol-will-require-https-encryption-most-of-the-time-.html. ค้นเมื่อวันที่ 22 กรกฎาคม 2558

Microsoft (ปี 2558). คู่มือการพัฒนาระบบด้วย SignalR สืบค้นเมื่อวัน 01 เดือนกรกฎาคม, ปี 2558, จาก ชื่อเว็บไซต์: http://www.asp.net/signalr

Wikipedia (ปี 2558). Web Socket สืบค้นเมื่อ วัน01 เดือน กรกฎาคม, ปี 2558, จาก ชื่อเว็บไซต์: http://en.wikipedia.org/wiki/WebSocket

Kmitl (ปี 2558). Client Server สืบค้นเมื่อ วัน 10 เดือน กรกฎาคม, ปี 2558, จาก ชื่อเว็บไซต์: http://www.mindphp.com

Techbrij (ปี 2556). Database change notifications asp.net สืบค้นเมื่อ วัน 10 เดือน กรกฎาคม, ปี 2558, จาก ชื่อเว็บไซต์: http://techbrij.com/databasechangenotifications-asp-net-signalr-sqldependency

José M. Aguilar 2557 SignalRProgramming inMicrosoft ASP.NET สำนักพิมพ์ Krasis Consulting S.L.

P. Leggetter, “Real-Time Web Technologies Guide.” [Online]. Available: http://www.leggetter.co.uk/real-time-web-technologies-guide. ค้นเมื่อวันที่ 1 มิถุนายน 2558

N. Podbielski, “WebSocket libraries comparison - CodeProject.” [Online]. Available:http://www.codeproject.com/Articles/733297/WebSocket-libraries-comparison. ค้นเมื่อวันที่ 2 มิถุนายน 2558

“Real-Time Analytics with WebSockets: SignalR vs. Node.JS with Socket.IO – Round 1.” [Online]. Available:http://sim4all.com/blogging/?p=454. ค้นเมื่อวันที่ 7 มิถุนายน 2558

“SockJS, multiple channels, and why I dumped socket.io | Matt’s Hacking Blog on WordPress.com.” [Online]. Available: http://baudehlo.com/2013/05/07/sockjs-multiple-channels-and-why-i-dumped-socket-io/. ค้นเมื่อวันที่ 8 มิถุนายน 2558

"SockJS, multiple channels, and why I dumped socket.io | Matt's Hacking Blog on WordPress.com." [Online]. Available: http://baudehlo.com/2013/05/07/sockjs-multiple-channels-and-why-i-dumped-socket-io/.ค้นเมื่อวันที่ 8 มิถุนายน 2558

XSockets, "XSockets vs SignalR | xsockets.net." [Online]. Available: http://xsockets.net/xsockets-vs-signalr. ค้นเมื่อวันที่ 8 มิถุนายน 2558

Lightstreamer, "The Lightstreamer Blog: Benchmarking Socket.IO vs. Lightstreamer with Node.js." [Online]. Available: http://blog.lightstreamer.com/2013/05/benchmarking-socketio-vs-lightstreamer.html. ค้นเมื่อวันที่ 9 มิถุนายน 2558

E. Bozdag, A. Mesbah, and A. van Deursen, "A Comparison of Push and Pull Techniques for AJAX," 2007 9th IEEE International Workshop on Web Site Evolution, pp. 15–22, Oct. 2007. ค้นเมื่อวันที่ 25มิถุนายน 2558

M. Jõhvik, "Push-based versus pull-based data transfer in AJAX applications," University of Tartu, 2011. ค้นเมื่อวันที่ 25 มิถุนายน 2558

D. G. Puranik, D. C. Feiock, and J. H. Hill, "Real-Time Monitoring using AJAX and WebSockets," ECBS '13 Proceedings of the 20th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems, pp. 110–118, 2013. ค้นเมื่อวันที่ 1 กรกฎาคม 2558

X. Wang, "Library vs. Framework?" [Online]. Available: http://www.programcreek.com/2011/09/what-is-the-difference-between-a-java-library-and-a-framework/. ค้นเมื่อวันที่ 1 กรกฎาคม 2558

M. Simpson, "Unit? Integration? Functional? Wha? | Mark's Devblog." [Online]. Available:http://defragdev.com/blog/?p=611. ค้นเมื่อวันที่ 12 กรกฎาคม 2558

K. Johannessen, "Code base - GitHub." [Online]. Available: https://github.com/kjohann/MasterThesis. ค้นเมื่อวันที่ 12 กรกฎาคม 2558

I. Peter, "History of the world wide web," 2004. [Online]. Available: http://www.nethistory.info/Historyof the Internet/web.html.ค้นเมื่อวันที่ 20 กรกฎาคม 2558

R. Fielding, H. Frystyk, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.0," 1996. [Online]. Available:

http://www.w3.org/Protocols/HTTP/1.0/draft-ietf-http-spec.html. ค้นเมื่อวันที่ 20 กรกฎาคม 2558

H. Nielsen and J. Gettys, “Network performance effects of HTTP/1.1, CSS1, and PNG,” ACM SIGCOMM …, pp. 155–166, 1997. ค้นเมื่อวันที่ 20 กรกฎาคม 2558

M. Belshe and R. Peon, “SPDY Protocol - Draft 3.1 - The Chromium Projects.” [Online]. Available:http://www.chromium.org/spdy/spdy-protocol/spdy-protocol-draft3-1. ค้นเมื่อวันที่ 21 กรกฎาคม 2558

M. Belshe, A. Melnikov, M. Thomson, and R. Peon, “Hypertext Transfer Protocol version 2.0.” [Online]. Available:http://tools.ietf.org/html/draft-ietf-httpbis-http2-00. ค้นเมื่อวันที่ 22 กรกฎาคม 2558

“Hypertext Transfer Protocol Bis (httpbis) - Charter.” [Online]. Available: https://datatracker.ietf.org/wg/httpbis/charter/. ค้นเมื่อวันที่ 22 กรกฎาคม 2558

P. Lubbers, B. Albers, and F. Salim, Pro HTML5 Programming, 2nd ed. Springer, 2011, pp. 159–191. ค้นเมื่อวันที่ 22 กรกฎาคม 2558

D. Crane and P. McCarthy, Comet and Reverse Ajax:The Next-Generation Ajax 2.0. Heidelberg: Springer-Verlag, 2008, pp. 25, 40–41. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

D. Schiemann, “Comet Daily>Blog Archive>The forever-frame technique,” 2007. [Online]. Available:http://cometdaily.com/2007/11/05/the-forever-frame-technique/. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

E. Bidelman, “Stream Updates with Server-Sent Events - HTML5 Rocks,” 2011. [Online]. Available: http://www.html5rocks.com/en/tutorials/eventsource/basics/. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

I. Hickson, “Server-Sent Events,” 2012. [Online]. Available: http://dev.w3.org/html5/eventsource/. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

I. Fette and A. Melnikov, “The WebSocket Protocol.” [Online]. Available: http://tools.ietf.org/html/rfc6455.ค้นเมื่อวันที่ 25 กรกฎาคม 2558

Walker-Morgan, “WebSockets becomes proposed standard,” 2011. [Online]. Available: http://www.h-online.com/open/news/item/WebSockets-becomes-proposed-standard-1394315.html. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

P. Lubbers, B. Albers, and F. Salim, “WebSocket.org | About WebSocket.” [Online]. Available: http://www.websocket.org/aboutwebsocket.html. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

I. Hickson, “The WebSocket API,” 2012. [Online]. Available: http://dev.w3.org/html5/websockets/. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

P. Lubbers and F. Greco, “WebSocket.org | The Benefits of WebSocket.” [Online]. Available: http://www.websocket.org/quantum.html. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

“Internet Explorer Browser.” [Online]. Available: http://www.w3schools.com/browsers/browsers_explorer.asp. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

LearnBoost, “Socket.IO: the cross-browser WebSocket for realtime apps.” [Online]. Available: http://socket.io/. ค้นเมื่อวันที่ 25 กรกฎาคม 2558

ประวัติผู้ศึกษา

ชื่อ	นายศุภชัย หอมพันธุ์
วัน เดือน ปีเกิด	10 กันยายน 2530
สถานที่เกิด	จังหวัดแพร่
ประวัติการศึกษา	มัธยมศึกษา โรงเรียนสาธิตเทศบาลบ้านเชตวันจังหวัดแพร่
	วิทยาศาสตรบัณฑิต มหาวิทยาลัยสยาม
	ศิลปศาสตรบัณฑิต มหาวิทยาลัยสุโขทัยธรรมาธิราช
สถานที่ทำงาน	บริษัท เทเลอินเทล จำกัด 318 ราชเทวี กรุงเทพฯ 10400
ตำแหน่ง	Programmer