

บทที่ 4

โครงสร้างการทำงานในส่วนของเครื่องใช้บริการ

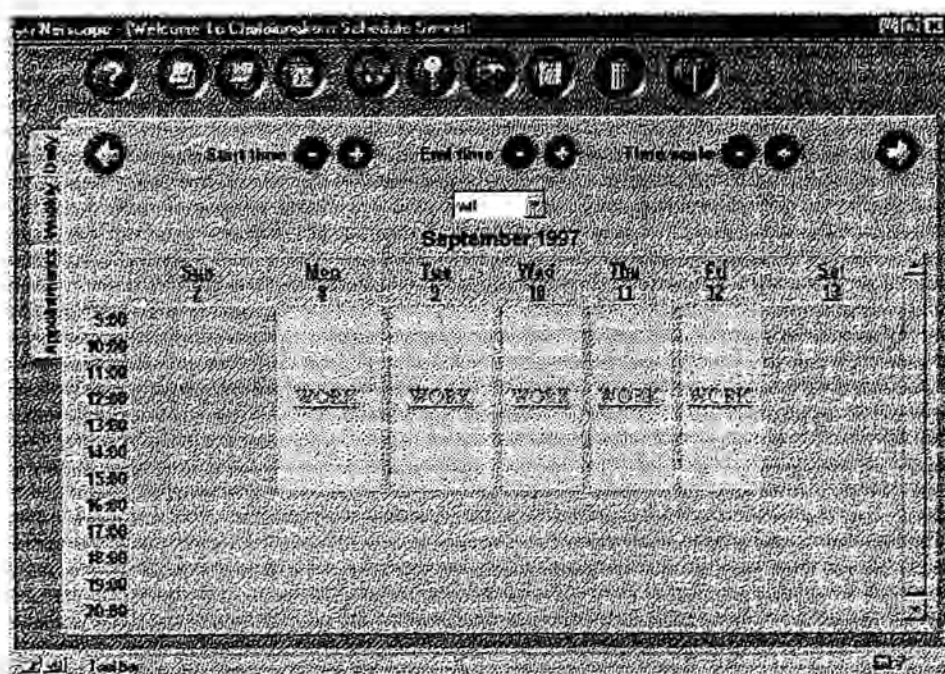
การทำงานของฝั่งเครื่องใช้บริการนั้นจะถูกพัฒนาโดยใช้ HTML เป็นส่วนสำหรับการอินเตอร์เฟลกับผู้ใช้และภาษา javascript เวอร์ชัน 1.1 เป็นส่วนของฟังก์ชันการทำงานต่างๆ ถ้าเปรียบเทียบอัตราส่วนของการทำงานในส่วนของเครื่องใช้บริการกับเครื่องบริการแล้ว งานมากกว่า 70% จะเป็นส่วนของเครื่องใช้บริการหรือถูกพัฒนาให้อยู่ในรูป javascript สาเหตุที่ผู้วิจัยออกแบบเช่นนั้นเนื่องจาก CSS ทำงานรองรับผู้ใช้หลายคนพร้อมกัน ดังนั้นหากมีผู้ใช้ใช้งานพร้อมกันยิ่งมากเท่าไร เครื่องบริการก็จะยิ่งทำงานหนักมากขึ้นเท่านั้น และถ้าหากออกแบบให้งานของเครื่องบริการมีมาก ก็ยิ่งทำให้ภาระของเครื่องบริการมากขึ้น เป็นผลทำให้ประสิทธิภาพการทำงานโดยรวมตกต่ำลงและทำให้การตอบรับระหว่างเครื่องบริการกับเครื่องใช้บริการช้าลง ซึ่งผู้วิจัยได้ตระหนักถึงข้อเสียดังกล่าวจึงได้ออกแบบให้งานในส่วนของเครื่องบริการมีน้อยที่สุดเท่าที่เป็นไปได้ จึงส่งผลให้ประสิทธิภาพการทำงานโดยรวมดีขึ้นมาก ยิ่งถ้าหากเครื่องใช้บริการมีประสิทธิภาพสูงเท่าไรก็จะส่งผลให้การทำงานของโปรแกรม javascript เร็วขึ้นเท่านั้น

จุดเริ่มต้นของการทำงานในฝั่งเครื่องใช้บริการคือ HTML หน้าแรกสุดที่ใช้สำหรับการล็อกอินดังรูปที่ 4.1



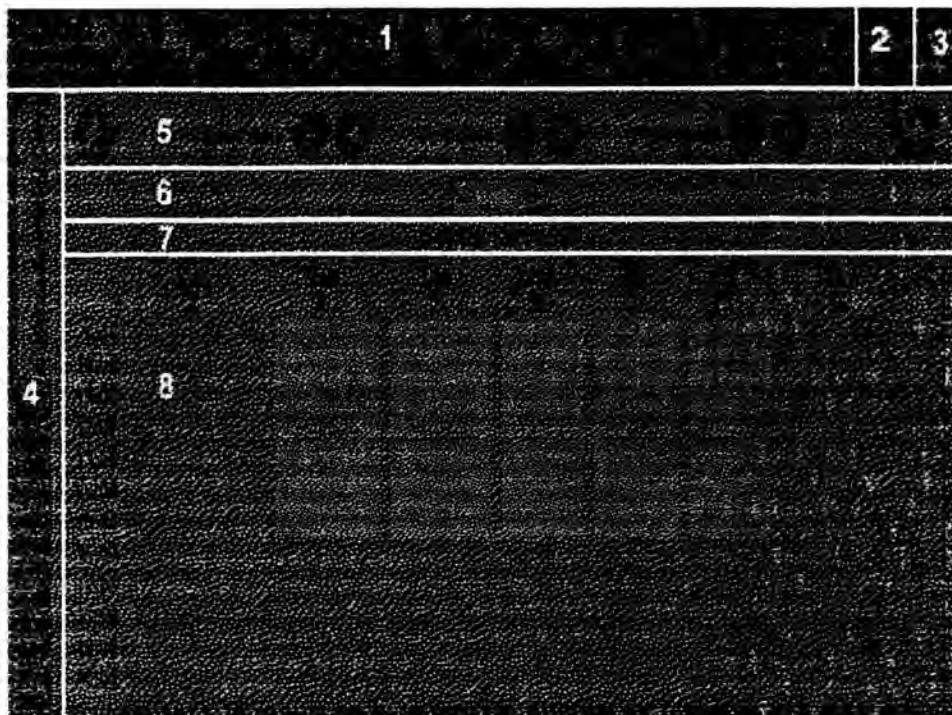
รูปที่ 4.1 HTML หน้าแรกสำหรับการล็อกอิน

ซึ่ง ณ จุดนี้เป็นหน้าที่ของผู้ใช้ที่จะป้อน user name และ password ของตนเองลงไป โดยจะมี javascript ที่คอยตรวจสอบก่อนการล็อกอินว่าผู้ใช้ได้ใช้ WWW Browser ตามรุ่นที่กำหนดหรือไม่ ถ้าใช้จริงจะยอมให้มีการล็อกอิน สาเหตุที่จะต้องมีการตรวจสอบเช่นนี้เพราะหากอนุญาตให้ browser อื่นที่นอกเหนือจากที่กำหนดล็อกอินเข้ามาได้ จะเป็นผลให้ WWW Browser เหล่านั้นอาจจะทำงานโปรแกรม javascript ผิดพลาดได้ ซึ่งขั้นตอนของการล็อกอินนี้จะเรียกใช้ฟังก์ชัน main ของ kernel ที่ได้ถูกกล่าวถึงในบทที่ 3 หากการล็อกอินประสบความสำเร็จ WWW Browser ก็จะทำให้การสร้างวินโดวใหม่ดังรูปที่ 4.2



รูปที่ 4.2 main page

หากผู้ใช้ได้เห็นวินโดวนี้ก็หมายถึงการล็อกอินประสบความสำเร็จ ซึ่งวินโดวที่เกิดขึ้นมานี้ก็คือ main page ที่ได้กล่าวถึงไปแล้วในส่วนของฟังก์ชัน main ของ kernel ในบทที่ 4 ซึ่งใน main page นี้ถูกแบ่งออกเป็นเฟรมย่อยๆ ดังรูปที่ 4.3



รูปที่ 4.3 การแบ่งเฟรมของ main page

- **เฟรมที่ 1** เป็นเฟรมที่มีชื่อว่า "Client_Function" โดยแฟ้ม HTML ที่เป็นเจ้าของเฟรมนี้คือ client_function.html ซึ่งเป็นแฟ้มที่มีขนาดใหญ่ที่สุดและเป็นแฟ้มที่เก็บฟังก์ชันของ javascript กว่า 80% ของเครื่องใช้บริการทั้งหมด ทั้งนี้เพื่อให้เฟรมอื่นสามารถเรียกใช้ฟังก์ชันที่อยู่ในเฟรมนี้ได้เพื่อลดความซ้ำซ้อนของฟังก์ชัน นอกจากนี้ยังทำหน้าที่ในการแสดงปุ่มฟังก์ชันทูลบาร์และไอคอน (Toolbar and Icon) ซึ่งแต่ละปุ่มจะมีการเรียกใช้ฟังก์ชันที่แตกต่างกันไป ตัวอย่างเช่น ปุ่มเรียกปฏิทิน ปุ่มสร้างการนัดหมายใหม่ เป็นต้น
- **เฟรมที่ 2** เป็นเฟรมที่มีชื่อว่า "Appointment_Loader" เป็นเฟรมที่มีไว้สำหรับรองรับการเรียกใช้งานของฟังก์ชัน Get_Daily_Appointment, Get_Weekly_Appointment และ Get_List_Appointment ซึ่งเฟรมนี้จะมีขนาดเล็กมากและถูกทำให้กลมกลืนไปกับเฟรม "Client_Function" ทั้งนี้เพราะเป็นเฟรมที่ไม่ต้องแสดงผลใดๆ ทั้งสิ้น ดังนั้นผู้วิจัยจึงได้ทำการซ่อนเฟรมนี้ไว้

- **เฟรมที่ 3** เป็นเฟรมที่มีชื่อว่า "Reminder Loader" เป็นเฟรมที่ถูกซ่อนเช่นเดียวกับ "Appointment Loader" แต่มีหน้าที่ที่แตกต่างกันคือจะรองรับการเรียกใช้งานของฟังก์ชัน Get_Reminder_Appointment
- **เฟรมที่ 4** เป็นเฟรมที่มีชื่อว่า "Menu" โดยเพิ่ม HTML ที่เป็นเจ้าของเฟรมนี้คือ menu.html เฟรมนี้มีหน้าที่ในการแสดงเมนูให้ผู้ใช้เลือก ซึ่งมีอยู่ด้วยกัน 3 เมื่อดังต่อไปนี้
 - **Daily** เป็นเมนูในการแสดงตารางเวลารายวัน หากผู้ใช้กดเลือกเมนูนี้จะเป็นการเรียกใช้ฟังก์ชัน Get_Daily_Appointment ไปที่เฟรม Appointment Loader จากนั้นจึงเรียกใช้ฟังก์ชันการแสดงตารางเวลารายวันชื่อ Show_Daily_Schedule ซึ่งอยู่ในเฟรม Client_Function
 - **Weekly** เป็นเมนูในการแสดงตารางเวลารายสัปดาห์ หากผู้ใช้กดเลือกเมนูนี้จะเป็นการเรียกใช้ฟังก์ชัน Get_Weekly_Appointment ไปที่เฟรม Appointment Loader จากนั้นจึงเรียกใช้ฟังก์ชันการแสดงตารางเวลารายวันชื่อ Show_Weekly_Schedule ซึ่งอยู่ในเฟรม Client_Function
 - **Appointments** เป็นเมนูในการแสดงลิสต์ของเรคคอร์ดการนัดหมาย โดยจะแสดงทีละ 20 เรคคอร์ด ซึ่งจะเรียงลำดับจากประเภทของการนัดหมายและวันเดือนปี หากเกิดการเลือกเมนูนี้ก็จะเรียกใช้ฟังก์ชัน Get_List_Appointment ในเฟรม Appointment Loader จากนั้นจึงมีการเรียกใช้ฟังก์ชัน Show_List_Appointment เพื่อการแสดงผล
- **เฟรมที่ 5** เป็นเฟรมที่มีชื่อว่า "Change" โดยเพิ่ม HTML ที่เป็นเจ้าของเฟรมนี้คือ change.html ซึ่งเฟรมนี้มีหน้าที่ในการแสดงปุ่มฟังก์ชันที่เกี่ยวข้องกับแต่ละเมนูของเฟรม Menu โดยตรง โดยปุ่มเหล่านี้ก็ได้แก่ ปุ่มลูกศรซ้ายขวาเพื่อเลื่อนวันที่ ปุ่มปรับเปลี่ยนความละเอียดในการแสดงตารางเวลา เป็นต้น ซึ่งฟังก์ชัน javascript ของการทำงานของปุ่มเหล่านี้จะอยู่ที่เฟรม Client_Function
- **เฟรมที่ 6** เป็นเฟรมที่มีชื่อว่า "Allow_User_Selection" เป็นเฟรมที่ถูกใช้งานในกรณีที่ผู้ใช้ได้รับสิทธิ์ดูตารางเวลาของผู้ใช้คนอื่น โดยจะแสดงรายชื่อของผู้ใช้เหล่านี้ออกมาในลักษณะของคอมโบบ็อกซ์ (Combo box) หากผู้ใช้เลื่อนแถบสว่างไปที่ชื่อผู้ใช้อื่นแล้วก็จะสามารถมองเห็นตารางเวลาของผู้ใช้คนนั้นๆ ได้
- **เฟรมที่ 7,8** เฟรมที่ 7 มีชื่อว่า "Header" ส่วนเฟรมที่ 8 มีชื่อว่า "Schedule" ทั้ง 2 เฟรมเป็นเฟรมที่ใช้แสดงผลของการเลือกเมนู ซึ่งแต่ละเมนูจะให้ผลที่แตกต่างกันดังต่อไปนี้

- Daily เฟรมที่ 8 จะแสดงตารางเวลารายวัน ส่วนเฟรมที่ 7 จะแสดงวันเดือนปีของวันนั้นๆ
- Weekly เฟรมที่ 8 จะแสดงตารางเวลารายสัปดาห์ ส่วนเฟรมที่ 7 จะแสดงเดือนและปีของสัปดาห์นั้นๆ
- Appointments เฟรมที่ 8 จะแสดงลิสต์ของเรคคอร์ดข้อมูลการนัดหมายจำนวนไม่เกิน 20 เรคคอร์ด ส่วนเฟรมที่ 7 จะแสดงว่าผู้ใช้ได้สร้างเรคคอร์ดการนัดหมายเอาไว้เป็นจำนวนทั้งหมดเท่าไร รวมทั้งจำนวนหน้าของลิสต์ เช่นหากมี 100 เรคคอร์ดก็จะมี 5 หน้าเป็นต้น

4.1 สถานภาพการทำงานของโปรแกรม

การทำงานส่วนของเครื่องใช้บริการมีอยู่ด้วยกัน 2 สถานภาพ (State) ดังต่อไปนี้

- 4.1.1 **สถานภาพของการว่างงาน (Idle State)** เป็นช่วงเวลาที่ไม่มีการทำงานใดๆ เกิดขึ้นบนเครื่องใช้บริการยกเว้นระบบเตือนความจำ (ซึ่งจะถูกกล่าวถึงในส่วนต่อไป) เท่านั้น เป็นสถานภาพที่เครื่องใช้บริการรอรับการสั่งงานจากผู้ใช้ หากผู้ใช้มีการสั่งการ สถานภาพของเครื่องใช้บริการก็จะถูกแปรเปลี่ยนเป็นสถานภาพของการประมวลผล
- 4.1.2 **สถานภาพของการประมวลผล (Processing State)** เป็นช่วงเวลาที่เครื่องใช้บริการกำลังประมวลผลการทำงานที่ผู้ใช้ได้สั่งการครั้งล่าสุด ซึ่งขณะที่เครื่องใช้บริการกำลังประมวลผลอยู่นี้ จะไม่มีการโต้ตอบหรือรับการสั่งการใดๆ จากผู้ใช้ทั้งสิ้น ในสถานภาพนี้จะเกิดวินโดว์พิเศษขึ้นมาดังรูปที่ 4.4



รูปที่ 4.4 วินโดว์แสดงสถานภาพของการประมวลผล

ซึ่งเป็นวินโดว์ที่บอกข้อความแก่ผู้ใช้ว่าให้หยุดรอจนกว่าเครื่องใช้บริการจะประมวลผลเสร็จ แต่วินโดว์นี้ไม่เพียงแคบอกข้อความเท่านั้น แต่มันจะทำหน้าที่เป็นเกราะกำบังให้กับเครื่องใช้บริการหรือ main page ด้วย โดยระหว่างที่แสดงวินโดว์นี้ผู้ใช้ไม่สามารถแตะต้องหรือสั่งการ main page ได้เลย และหากผู้ใช้ฝ่าฝืนโดยการปิดวินโดว์นี้โดยพลการ จะเป็นผลทำให้การทำงานของเครื่องใช้บริการทั้งหมดถูกหยุดลงและ

main page ก็จะทำให้การปิดตัวเองด้วยด้วย สาเหตุที่ต้องทำเช่นนี้เนื่องจากสถานะภาพของการประมวลผลนั้นจะถูกขัดจังหวะไม่ได้เด็ดขาด มีเช่นนั้นการทำงานของเครื่องใช้บริการที่ตามมาภายหลังนั้นอาจจะคลาดเคลื่อนหรือผิดพลาดได้

แต่ในบางเหตุการณ์ ผู้ใช้อาจจำเป็นต้องปิดวินโดวนี้ ในกรณีที่สถานะภาพของการประมวลผลยาวนานจนเกินไป ซึ่งอาจจะมีสาเหตุมาจากปัญหาทางด้านเครือข่ายหรือที่เครื่องบริการ ดังนั้นหากผู้ใช้ปิดวินโดวนี้ ก็สามารถล็อกอินเข้ามาใหม่และใช้งานได้ตามปกติอีกครั้ง

การที่เครื่องใช้บริการเข้าสู่สถานะภาพของการประมวลผลมีอยู่ด้วยกัน 2 กรณี

- **กรณีของการล็อกอิน** หลังจากที่ผู้ใช้ได้ทำการล็อกอินเข้าไปแล้ว เครื่องใช้บริการจะเข้าสู่สถานะภาพของการประมวลผลโดยอัตโนมัติ เนื่องจากเป็นช่วงเวลาที่เครื่องใช้บริการจะต้องนำ main page รวมทั้งเพิ่มข้อมูล HTML ที่ประจำอยู่ในแต่ละเฟรมจากเครื่องบริการมาให้เสร็จสมบูรณ์เสียก่อน ดังนั้นเครื่องใช้บริการจะไม่ยอมให้ผู้ใช้สั่งการใดๆ จนกว่าเพิ่ม HTML ที่บรรจุ javascript ที่เป็นฟังก์ชันการทำงานทั้งหมดจะถูกนำมาจนเสร็จสมบูรณ์เสียก่อน
- **กรณีที่ผู้ใช้สั่งการเครื่องใช้บริการ** หลังจากที่ผู้ใช้สั่งการทำงานกับเครื่องใช้บริการ เครื่องก็จะเข้าสู่สถานะภาพของการประมวลผล เมื่อประมวลผลเสร็จ วินโดว "Please wait..." ก็จะถูกปิดลงแต่เครื่องใช้บริการจะยังคงอยู่ที่สถานะภาพของการประมวลผลต่อไปอีก 1 วินาทีจึงจะกลับสู่สถานะภาพของการว่างงาน ทั้งนี้เพื่อป้องกันในกรณีที่ผู้ใช้กดปุ่มสั่งการเครื่องใช้บริการรัวติดต่อกัน ทั้งนี้การสั่งงานฟังก์ชันบางอย่างจะมีการติดต่อไปยังเครื่องบริการเพื่อดึงข้อมูล ตัวอย่างเช่น การเลื่อนตารางเวลารายสัปดาห์ไปยังอาทิตย์ถัดไป โดยกดปุ่มลูกศรทางขวาของเฟรม Change ซึ่งจะต้องนำเอาข้อมูลของการนัดหมายของสัปดาห์ถัดไปมาจากเครื่องบริการ ถ้าหากไม่มีการป้องกันดังกล่าวและผู้ใช้กดปุ่มลูกศรรัวติดต่อกันหลายครั้ง จะเป็นผลให้เครื่องบริการต้องทำงานตามจำนวนครั้งที่ผู้ใช้ได้กดไป ทำให้งานของเครื่องบริการหนักจนเกินไป และทำให้ประสิทธิภาพการทำงานของเครื่องบริการโดยรวมตกลง

ในสถานะภาพของการประมวลผลนี้ ระบบเตือนความจำจะถูกหยุดการทำงานและจะเริ่มต้นทำงานอีกครั้งเมื่อกลับเข้าสู่สถานะภาพของการว่างงาน

4.2 ระบบเตือนความจำ (Reminder System)

ระบบเตือนความจำนั้นเป็นฟังก์ชันที่คอยตรวจสอบเรคคอร์ดการนัดหมายของวันเดือนปีปัจจุบันว่ามีการนัดหมายใดบ้างที่ผู้ใช้ต้องการให้มีการเตือน ซึ่งเรคคอร์ดการนัดหมายเหล่านี้จะต้องมีค่าของฟิลด์ชื่อ Remind เป็น 1 และวันเวลาที่ฟังก์ชันนี้ใช้ในการเทียบเพื่อการเตือนคือวันเวลาที่ถูกตั้งไว้บนเครื่องเครื่องใช้บริการ ซึ่งถือได้ว่าเป็นหัวใจการทำงานของระบบเตือนความจำ ดังนั้นผู้ใช้ทุกคนต้องแน่ใจว่าวันเวลาของเครื่องคอมพิวเตอร์ที่กำลังใช้อยู่ตรงตรงกับเวลาปัจจุบันหรือไม่

ฟังก์ชันนี้จะคอยตรวจสอบข้อมูลการนัดหมายทุก 1 นาที โดยจะตรวจสอบเวลาเริ่มต้นของการนัดหมายใดๆ เทียบกับเวลาปัจจุบัน ผนวกกับจำนวนนาทีที่ผู้ใช้ต้องการให้มีการเตือนล่วงหน้า (Remind_Before_Time) ตัวอย่างเช่น การนัดหมายเกิดขึ้น ณ เวลา 9.00 และค่าของ Remind_Before_Time เป็น 15 นาที ในกรณีนี้ระบบเตือนความจำจะเตือนผู้ใช้เมื่อเวลา 8.45 เป็นต้น