



## รายงานสหกิจศึกษาฉบับสมบูรณ์

การพัฒนาระบบการจัดเก็บชิ้นงานวิจัยและพัฒนาหลังจากเสร็จสิ้น

กระบวนการผลิตบนระบบเว็บแอปพลิเคชัน

ภาคการจัดการระบบตอบสนองทางข้อมูล

Research and Development materials inventory system

on Web Application

(Data Response Service)

นายชัยทัศน์ มีมา

สาขาวิศวกรรมสารสนเทศ

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา (ภาษาไทย)

การพัฒนากระบวนการจัดเก็บงานวิจัยและพัฒนาหลังจาก  
เสร็จสิ้นกระบวนการผลิตบนระบบเว็บแอปพลิเคชัน  
ภาคการจัดการระบบตอบสนองทางข้อมูล

ชื่อ-สกุล นักศึกษา นายชัยทัศน์ มีมา

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา วิศวกรรมสารสนเทศ

ชื่อ-สกุล อาจารย์นิเทศ ผศ.บุญชนะ ภูระหงษ์

ชื่อ-สกุล ผู้นิเทศงาน นายหัสตินทร์ ตั้งสกุล

สถานประกอบการ บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด

### บทคัดย่อ

การพัฒนากระบวนการจัดเก็บงานวิจัยและพัฒนาหลังจาก เสร็จสิ้นกระบวนการผลิตบนระบบเว็บแอปพลิเคชันเป็นการวิจัยเชิงประติมากรรม มีวัตถุประสงค์เพื่อศึกษาและพัฒนาวิธีการเพิ่มศักยภาพระบบการทำงานในกระบวนการผลิตที่มีการเก็บข้อมูลของผลิตภัณฑ์ที่ใช้งานเพื่อลดระยะชั้นตอน และเวลาในการทำงานของผู้ใช้งานให้น้อยที่สุด โดยงานที่ได้ต้องมีคุณภาพที่ดีที่สุดเช่นกัน

มีวิธีการดำเนินงานคือการสร้างเว็บแอปพลิเคชันผ่านเครือข่ายภายในองค์กร ซึ่งอ้างอิงจากข้อมูลที่มีอยู่ โดยมีการเขียนโปรแกรมเป็นภาษา HTML, CSS, Java และ SQL เป็นหลักในการพัฒนาทั้งด้าน Front-End ในส่วนการออกแบบวิธีการใช้งาน และ Back-end ในส่วนกับการเชื่อมต่อกับข้อมูลในฐานข้อมูล

จากผลการวิจัยพบว่า สามารถเพิ่มประสิทธิภาพในการทำงานของแผนกโดยการสำรวจจากผู้ใช้งานจริงถือว่าเป็นพึงพอใจอย่างมาก เพราะว่าทางพนักงานที่ใช้งานสามารถนำข้อมูลไปประยุกต์ใช้ได้ทันที ทำให้ลดขั้นตอนการทำงานและส่งผลถึงประสิทธิภาพการทำงานให้สามารถทำงานได้ง่ายและสะดวกขึ้นได้อย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Cooperative Title:** Research and Development materials inventory system on  
Web Application (Data Response Service)

**Student intern name:** Mr. Chaiyatat Meema

**Faculty:** Engineering                      **Department:** Information Engineering

**Program:** Information Engineering

**Advisor name:** Asst.Prof. Boonchana Purahong

**Mentor name:** Mr. Hussadin Thungsakul

**Company:** Seagate Technology (Thailand) Ltd.

## ABSTRACT

The purpose of this research is study and increase working systems potential in the production process in factory that have data collection of active products to shorten the procedure and reduce the processing time in common to a least. The achieve work project must done in the best quality as well.

Developing a web application is the suitable way to do this project by making the application online on company's server using HTML, CSS, Java and SQL Language for the main technique by develop both Front-end for the design and Back-end for connect with database.

The results of this study were increased of working systems potential in company's department and decrease the processing time. Survey results from real users considered very satisfied to use our web application.

## กิตติกรรมประกาศ

จากที่ได้เข้าร่วมโครงการสหกิจศึกษาและเริ่มปฏิบัติงานที่ บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด ตั้งแต่วันที่ 5 สิงหาคม ถึง วันที่ 29 พฤศจิกายน 2562 ทำให้ได้รับความรู้ ความสามารถและประสบการณ์ ทางด้านต่าง ๆ ที่สามารถนำมาพัฒนาตัวเองมากขึ้น และรายงานสหกิจศึกษา ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดี โดยมีนางสาวกัญชพร พุ่มนตรี ที่ช่วยเหลือ และจัดการภาคติดต่อประสานงานกับผู้ใช้งาน ที่ทำให้งานวิจัยมีความสมบูรณ์ อีกทั้งด้วยความอนุเคราะห์จาก ผู้ช่วยศาสตราจารย์บุญยชนะ ภูระหงษ์ อาจารย์ที่ปรึกษา งานสหกิจศึกษา ที่ได้สละเวลาให้ความรู้คำปรึกษาตรวจสอบ และแก้ไขข้อผิดพลาดต่าง ๆ ตลอดจนการกำกับดูแล และคอยติดตามความก้าวหน้า ทำให้การวิจัยนี้ประสบความสำเร็จไปได้ด้วยดี ทางผู้วิจัย ขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณบิดา มารดา และญาติพี่น้องที่ให้การสนับสนุนและเป็นกำลังใจ ที่ดีให้เสมอมา และสนับสนุนด้านทุนทรัพย์ในการศึกษา รวมไปถึงผู้สนับสนุนทุกท่านที่มีส่วนร่วมในการจัดทำรายงานสหกิจศึกษาครั้งนี้ ซึ่งมิได้กล่าวชื่อนามในที่นี้

ท้ายที่สุด ผู้วิจัยขอขอบพระคุณเพื่อน ๆ ทุกคนที่คอยติดตามและให้กำลังใจ รวมถึงท่านอื่น ๆ ที่มีได้กล่าวชื่อนามไว้ ณ ที่นี้ ที่มีส่วนทำให้รายงานสหกิจศึกษาสำเร็จลุล่วง ไปได้ด้วยดี ผู้วิจัยคาดหวังเป็นอย่างยิ่งว่ารายงานสหกิจศึกษาฉบับนี้เป็นประโยชน์ ไม่มากก็น้อย สำหรับผู้ที่สนใจศึกษารายละเอียดต่อไป

นายชัยทัศน์ มีมา

# สารบัญ

หน้า

บทคัดย่อ .....	I
ABSTRACT .....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญตาราง .....	VIII
สารบัญภาพ .....	IX
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	3
1.4 วิธีดำเนินการวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	4
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	6
2.1 Hypertext Transfer Protocol (HTTP) .....	6
2.10.1 HTTP - Request.....	6
2.10.2 HTTP - Response .....	8
2.2 ภาษา Java.....	10
2.2.1 ประวัติของภาษา Java .....	10
2.2.2 โครงสร้างของภาษา Java.....	11
2.2.3 ข้อดีของ ภาษา Java .....	12
2.2.4 ข้อเสียของ ภาษา Java.....	12
2.2.5 จุดมุ่งหมายในการพัฒนาภาษา Java.....	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
2.3 ภาษา HTML .....	13
2.3.1 ส่วนประกอบของ HTML .....	13
2.3.2 โครงสร้างของ HTML.....	14
2.3.3 การสร้างเว็บเพจ HTML .....	16
2.4 ภาษา JSP.....	16
2.4.1 ประเภทของ JSP.....	16
2.4.2 ข้อดีของ JSP.....	17
2.4.3 ข้อเสียของ JSP .....	18
2.5 ภาษา Javascript.....	18
2.5.1 โครงสร้างคำสั่งของ Javascript.....	18
2.5.2 ลักษณะการทำงานของ JavaScript .....	19
2.5.3 ข้อดีและข้อเสียของ JavaScript.....	20
2.6 รูปแบบข้อมูล JSON.....	20
2.6.1 ประเภทของ JSON.....	21
2.6.2 โครงสร้างของ JSON.....	21
2.6.3 จุดเด่นของ JSON.....	22
2.7 ภาษา SQL.....	23
2.7.1 ประวัติภาษา SQL.....	23
2.7.2 การใช้งานภาษา SQL .....	24
2.7.3 SQL Syntax.....	24
2.8 ภาษา CSS .....	25
2.8.1 โครงสร้างของ CSS .....	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

หน้า

2.8.2 การใช้งาน CSS .....	26
2.9 Apache POI Library .....	28
2.10 JQUERY Library .....	29
2.10.1 โครงสร้างของ JQuery.....	29
2.11 กลุ่มเทคโนโลยี AJAX.....	30
2.11.1 ส่วนประกอบของ AJAX .....	30
2.11.2 ประวัติของ AJAX .....	31
2.12 ระบบฐานข้อมูล (Database System).....	31
2.12.1 วิวัฒนาการของ Database .....	32
2.12.2 การเข้าถึง Database .....	32
2.12.3 ข้อดีของการใช้ Database .....	33
2.12.4 ข้อเสียของการใช้ Database.....	33
2.13 Oracle Database.....	34
2.13.1 ข้อดีของ Oracle.....	34
2.14 Web Application.....	35
2.14.1 ส่วนประกอบฝั่งผู้ใช้งาน (Client-side Technology) .....	35
2.14.2 ส่วนประกอบฝั่ง Server (Server-side Technology).....	37
2.15 WildFly Application Server .....	39
2.16 โปรแกรม NetBeans IDE .....	40
<b>บทที่ 3 วิธีดำเนินการวิจัย .....</b>	<b>41</b>
3.1 ภาพรวมแนวคิดของงานวิจัย .....	41
3.2 ขั้นตอนการดำเนินงาน.....	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

หน้า

3.2.1	ขั้นตอนการรวบรวมข้อมูลและออกแบบโครงสร้างระบบที่ใช้.....	41
3.2.2	ขั้นตอนการออกแบบระบบการทำงานและสร้างเว็บแอปพลิเคชัน .....	43
3.2.3	ขั้นตอนการสร้างระบบเสริมการทำงานเพื่ออำนวยความสะดวก .....	48
3.3	ขั้นตอนการทดสอบการใช้งานของโปรแกรมตัวทดลอง.....	49
3.4	ขั้นตอนการพัฒนาโปรแกรมตัวเต็ม .....	50
<b>บทที่ 4 ผลการดำเนินงาน .....</b>		<b>51</b>
4.1	สรุปผลการวิจัย.....	51
4.1.1	รูปแบบหน้าหลักของเว็บไซต์ .....	51
4.1.2	รูปแบบหน้าฟังก์ชัน Check in Check out.....	52
4.1.3	รูปแบบหน้าเมนูฟังก์ชันของการค้นหาชิ้นงานประเภทต่าง ๆ.....	53
4.1.4	รูปแบบหน้าเมนูฟังก์ชันสำหรับการแสดงกราฟ .....	55
4.1.5	รูปแบบหน้าแสดงรายชื่อของผู้ใช้งานทั้งหมดในทุกฝ่ายทุกแผนกที่เกี่ยวข้อง .....	57
4.1.6	รูปแบบหน้าเมนูฟังก์ชันการเบิกงานสำหรับฝ่าย Planner .....	57
4.1.7	รูปแบบหน้าแสดงสถานะของการเบิกงานต่าง ๆ (ฝ่าย Inventory Manager) .....	60
4.1.8	รูปแบบหน้าแสดงสถานะของการเบิกงานต่าง ๆ (ฝ่าย Production) .....	61
4.1.9	รูปแบบของการได้รับ Email ของทุกฝ่ายงานที่เกี่ยวข้อง .....	62
<b>บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ .....</b>		<b>63</b>
5.1	สรุปผลการวิจัย.....	63
5.2	ปัญหาและอุปสรรคที่พบ .....	64
5.3	ข้อเสนอแนะและแนวทางในอนาคต.....	64
<b>เอกสารอ้างอิง .....</b>		<b>65</b>
<b>ประวัติผู้จัดทำ.....</b>		<b>66</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

	หน้า
ตารางที่ 3.1 รายการเครื่องมือที่ใช้งานในระบบ.....	41
ตารางที่ 3.2 ตำแหน่ง(Role) ในการจำกัดสิทธิ์ต่าง ๆ .....	46



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ

หน้า

ภาพที่ 1.1	Workflow การทำงานของโปรเจกต์ .....	4
ภาพที่ 1.2	Timeline การทำงานทั้งหมดของโปรเจกต์.....	5
ภาพที่ 2.1	การสื่อสารผ่าน Internet .....	6
ภาพที่ 2.2	การ Request จาก Client ไป Server .....	6
ภาพที่ 2.3	การแบ่งส่วนของ Request message .....	7
ภาพที่ 2.4	Request-Line .....	7
ภาพที่ 2.5	Message-body .....	8
ภาพที่ 2.6	การ Response หลังจากการ Request.....	8
ภาพที่ 2.7	Status-Line .....	9
ภาพที่ 2.8	โครงสร้างของภาษา Java.....	11
ภาพที่ 2.9	รูปแบบคำสั่งในส่วน <head></head>.....	14
ภาพที่ 2.10	ตัวอย่างการเขียนในรูปแบบ JSON.....	21
ภาพที่ 2.11	ตัวอย่างการเขียนในรูปแบบ JSON ในลักษณะของ Master – Detail.....	22
ภาพที่ 2.12	โครงสร้างตัวอย่างคำสั่ง SQL .....	24
ภาพที่ 2.13	โครงสร้างของ CSS.....	25
ภาพที่ 2.14	การเขียน CSS แบบ Embedded .....	26
ภาพที่ 2.15	การเขียน CSS code แบบ External.....	27
ภาพที่ 2.16	ตัวอย่างการเขียน CSS code แบบ External.....	27
ภาพที่ 2.17	Apache POI.....	28
ภาพที่ 2.18	ตัวอย่างเอกสาร Microsoft Office ที่สร้างผ่าน Apache POI .....	28
ภาพที่ 2.19	โครงสร้างระบบ Web application .....	35
ภาพที่ 2.20	WildFly Application Server .....	39
ภาพที่ 2.21	โปรแกรม NetBeans IDE 7.1.2.....	40
ภาพที่ 3.1	WildFly.....	41
ภาพที่ 3.2	Oracle Database .....	41
ภาพที่ 3.3	กระบวนการการทำงานของ web application .....	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 3.4 Workflow การทำงานของระบบ Check In / Check Out .....	44
ภาพที่ 3.5 Workflow การทำงานของระบบ Search by Tray ID.....	44
ภาพที่ 3.6 คำสั่ง SQL สำหรับกรองแถว .....	45
ภาพที่ 3.7 คำสั่ง SQL สำหรับดูจำนวนโดยรวม.....	45
ภาพที่ 3.8 Workflow การทำงานของระบบ Authentication.....	46
ภาพที่ 3.9 คำสั่ง Java สำหรับออกจากระบบอัตโนมัติเมื่อเข้าสู่ระบบค้างไว้ .....	46
ภาพที่ 3.10 ตัวอย่างข้อมูลใน JSON.....	47
ภาพที่ 3.11 Workflow การทำงานของระบบเบิกงาน .....	47
ภาพที่ 3.12 ตัวอย่างคำสั่ง Java ในการสร้างไฟล์ผ่าน Apache POI.....	48
ภาพที่ 3.13 เว็บไซต์หลักที่ Deploy ให้ผู้ใช้งาน .....	49
ภาพที่ 3.14 การ Deploy Web Application ผ่าน WildFly Server.....	49
ภาพที่ 4.1 หน้าการเข้าสู่ระบบ.....	51
ภาพที่ 4.2 เมนูหลัก.....	52
ภาพที่ 4.3 เมนู Check in / Check out.....	52
ภาพที่ 4.4 หน้าหลักของการค้นหา.....	53
ภาพที่ 4.5 หน้าการค้นหาโดยตัวแปร AAA.....	53
ภาพที่ 4.6 หน้าการค้นหาโดยตัวแปร BBB .....	54
ภาพที่ 4.7 ข้อมูลทั้งหมดของผลิตภัณฑ์ ID AAAAAAAA .....	54
ภาพที่ 4.8 หน้าหลักของการแสดงกราฟ .....	55
ภาพที่ 4.9 หน้าการแสดงผลกราฟของผลิตภัณฑ์ AAA.....	55
ภาพที่ 4.10 รายละเอียดของกราฟนั้น ๆ ของผลิตภัณฑ์ AAA โดยแบ่งตามอายุของผลิตภัณฑ์ .....	56
ภาพที่ 4.11 รายละเอียดข้อมูลที่จำเป็นทั้งหมดของของผลิตภัณฑ์ AAA .....	56
ภาพที่ 4.12 ข้อมูลของผู้ใช้งาน.....	57
ภาพที่ 4.13 เมนูหลักของการเบิกงาน .....	57
ภาพที่ 4.14 หน้าหลักของการเบิกงานของฝ่าย Planner ตาม Request ที่ได้รับ .....	58
ภาพที่ 4.15 หน้าแสดงเลข Order ของการเบิกงาน .....	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 4.16	รูปแบบของ Auto Email.....	59
ภาพที่ 4.17	หน้าสถานะของแต่ละ Order การเบิกงาน สำหรับฝ่าย Inventory Manager.....	60
ภาพที่ 4.18	ข้อมูลที่จำเป็นทั้งหมดสำหรับให้ฝ่าย Inventory Manager เข้ามากรอก .....	60
ภาพที่ 4.19	หน้าสถานะของแต่ละ Order การเบิกงาน สำหรับฝ่าย Production .....	61
ภาพที่ 4.20	ข้อมูลที่จำเป็นทั้งหมดสำหรับให้ฝ่าย Production เข้ามากรอก.....	61
ภาพที่ 4.21	Auto Email ทั้งหมดของการเบิกงาน .....	62



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากบริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด เป็นบริษัทที่เป็นผู้นำด้านการผลิตฮาร์ดดิสก์ไดรฟ์ แม็คเคนติกดิสก์ และหัวเขียน-อ่าน รายใหญ่ที่สุดในโลก มีสำนักงานใหญ่ตั้งอยู่ที่เมืองสก็อตต์สวอเล่ย์ รัฐแคลิฟอร์เนีย ประเทศสหรัฐอเมริกา ต่อมาได้ขยายกิจการไปยังภูมิภาคต่าง ๆ ทั้งทวีปยุโรปและทวีปเอเชีย ปัจจุบันได้กระจายแหล่งที่ตั้งไปทั่วทุกภูมิภาค ซึ่งแบ่งเป็นฝ่ายดีไซน์ตั้งอยู่ที่สหรัฐอเมริกาทำหน้าที่ออกแบบผลิตภัณฑ์ใหม่ตามความต้องการของลูกค้า ส่วนฝ่ายฐานการผลิต ตั้งอยู่ที่รัฐมินเนโซต้า ประเทศสหรัฐอเมริกา และในต่างประเทศ ได้แก่ ไอร์แลนด์เหนือ สิงคโปร์ จีน มาเลเซีย และไทย โดยบริษัท ซีเกท เทคโนโลยี ในประเทศไทย ได้รับการส่งเสริมการลงทุนจากสำนักงานคณะกรรมการส่งเสริมการลงทุน (BOI) เพื่อทำการผลิตและส่งออกส่วนประกอบชิ้นต้นและชิ้นสุดท้ายของฮาร์ดดิสก์ไดรฟ์ สำหรับโรงงานผลิตในประเทศไทย มี 2 แห่ง คือที่โรงงานสาขาเทพารักษ์ จังหวัดสมุทรปราการ เป็นโรงงานผลิตชิ้นส่วนบางขั้นตอนที่เกี่ยวกับหัวบันทึกและอ่านข้อมูลเป็นหลัก ที่ผลิตเฉพาะ Head Gimbal Assembly (HGA) กับโรงงานสาขาโคราช จังหวัดนครราชสีมา เป็นโรงงานประกอบฮาร์ดดิสก์ และเป็นโรงงานเดียวในอุตสาหกรรมนี้ ที่มีกระบวนการผลิตตั้งแต่ชิ้นต้นจนถึงชิ้นตอนสุดท้าย แบ่งเป็น Slider, HSA และ Drive Operations ซึ่งในปัจจุบันมีการปรับเปลี่ยนกระบวนการผลิตเป็นแบบ Automation คือ การใช้เครื่องจักรกลทั้งหมดในสายการผลิต โดยใช้คอมพิวเตอร์ควบคุมการทำงานแทนการใช้แรงงานคน

การเข้าร่วมในโครงการสหกิจศึกษาครั้งนี้ผู้วิจัยได้เข้าร่วมทำงานในแผนก Head Research & Development Operations (HDO) ของบริษัทฯ ซึ่งมีหน้าที่ในการสนับสนุนงานวิจัยและพัฒนาผลิตภัณฑ์ให้มีเทคโนโลยีที่สูงขึ้น โดยในการผลิต Hard Disk Drive แผนกทำการรับ Slider จากสาขาโคราช เพื่อประกอบเป็น HGA, HSA และ Drive ที่แผนก HDO โดยเฉพาะ ซึ่งสามารถประกอบงานจนจบขั้นตอนการผลิตฮาร์ดดิสก์ได้ทันที ซึ่งไม่จำเป็นต้องส่งงานไปผลิตต่อที่สาขาโคราช หรือที่สาขาอื่น ๆ ซึ่งต่างจากแผนกอื่น เพราะที่สาขาเทพารักษ์ ผลิตแค่ HGA อย่างเดียว ซึ่งในการผลิตขั้นตอนต่อไปมีความจำเป็นต้องไปทำต่อที่สาขาอื่น

แผนก HDO ทำการประกอบและทดสอบทางไฟฟ้า ของงานวิจัยและพัฒนาที่ได้รับ Wafer, Suspension และ Air bearing designs มาจากประเทศสหรัฐอเมริกา หรือมาจากทางยุโรป เพื่อนำข้อมูลที่ได้ทั้ง ทางไฟฟ้าและทางกล ไปทำการปรับปรุงพัฒนาผลิตภัณฑ์ใหม่ ๆ ให้กับทางบริษัทฯ ในระยะเวลา ที่สั้นที่สุด ในกระบวนการผลิต ได้มีการเก็บข้อมูลของผลิตภัณฑ์ ที่ใช้งาน ซึ่งในปัจจุบันได้มี กระบวนการผลิตแบบใหม่ที่เรียกว่า Slider Test Socket คือ Process แรกที่ใช้คัดเลือก Slider ที่คุณภาพไม่ต้อออก และเลือกชิ้นงานที่มีคุณภาพสูง ๆ เพื่อมาประกอบขึ้นเป็น HGA เพื่อเป็น การลดต้นทุนในสายการผลิต

ในกระบวนการทำงานข้างต้น จำเป็นต้องมีการเก็บข้อมูลในด้านต่าง ๆ เกี่ยวกับผลิตภัณฑ์ ให้เป็นระบบอยู่บนเว็บแอปพลิเคชัน เพื่อความสะดวกสบายของผู้ใช้งาน ซึ่งระบบดั้งเดิมของแผนกมีการจัดการเก็บข้อมูลยังมีรูปแบบและโครงสร้างไม่ตีเท่าที่ควร การบันทึกข้อมูลแต่ละครั้ง ในส่วนของ Locator ต้องใช้ลักษณะของการบันทึกด้วยมือ (Manual) ซึ่งต้องนำไปกรอกใส่สูตร ที่บันทึกไว้ใน Excel file ก่อนการนำ Locator ที่ได้จากสูตร มากรอกลงในฐานข้อมูล Oracle ซึ่งใช้เวลานานพอสมควรในการบันทึกแต่ละครั้ง จึงอาจทำให้เกิดความผิดพลาดได้ ถ้าหากบันทึกข้อมูลผิด (Human Error) ทำให้ ต้องใช้เวลานานในการทำงานแต่ละส่วนของ แต่ละฝ่ายงาน จากปัญหาข้างต้น ทางผู้วิจัยจึง เล็งเห็นถึงแนวแก้ไขปัญหา โดยการพัฒนาระบบการจัดเก็บงานวิจัยและพัฒนาหลังจากเสร็จสิ้นกระบวนการผลิตบนระบบเว็บแอปพลิเคชัน ให้มีโครงสร้าง และรูปแบบ ที่ดีขึ้นกว่าเดิม ลดระยะขั้นตอนและเวลาในการทำงานของผู้ใช้งาน ให้น้อยที่สุด โดยงานที่ได้จะต้องมีคุณภาพที่ดีที่สุดเช่นกัน

## 1.2 วัตถุประสงค์ของการวิจัย

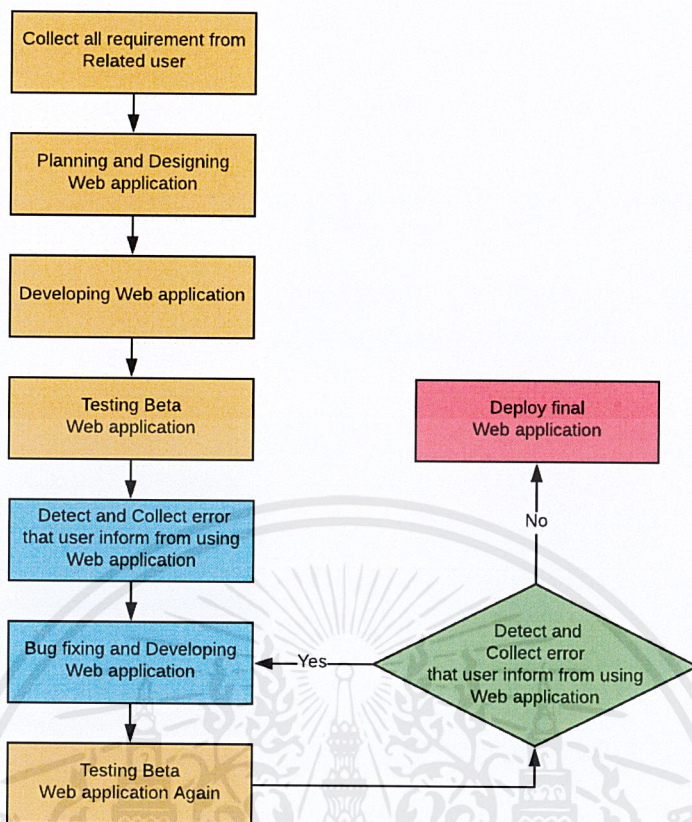
1. เพื่อเพิ่มประสิทธิภาพและลดระยะเวลาของการทำงานในระบบดั้งเดิม
2. เพื่อเพิ่มความสะดวกสบายและความพึงพอใจของผู้ใช้งาน
3. เพื่อลดภาระงานที่ไม่จำเป็นของฝ่ายผลิตในการบันทึกข้อมูลลงระบบ
4. เพื่อให้ฝ่ายวิศวกรฝ่ายวางแผนการผลิต และฝ่ายควบคุมการผลิต สามารถค้นหา และตรวจสอบข้อมูลได้ในทันที และมีความรวดเร็วขึ้นกว่าระบบเดิม
5. เพื่อให้ผู้ใช้งานได้เห็นหน้าสรุปข้อมูลที่ชัดเจนขึ้น
6. เพื่อให้ผู้ใช้งานเก็บและตรวจสอบข้อมูลลงฐานข้อมูลได้สะดวกและง่ายขึ้น เพื่อนำไปใช้ประโยชน์ในขั้นตอนงานต่อ ๆ ไป

### 1.3 ขอบเขตของการวิจัย

1. เว็บไซต์สามารถดูข้อมูลของผลิตภัณฑ์ได้
2. เว็บไซต์สามารถให้ผู้ที่เกี่ยวข้องติดตามขั้นตอนของการผลิตได้
3. เว็บไซต์สามารถให้แอดมินจำกัดสิทธิ์การเข้าถึงของผู้ใช้ในบางหน้าบนเว็บไซต์ได้
4. เว็บไซต์สามารถให้ฝ่ายการผลิตเพิ่มหรือแก้ไขข้อมูลของผลการผลิตได้
5. เว็บไซต์สามารถเก็บข้อมูลลงฐานข้อมูลได้
6. เว็บไซต์สามารถ Export ข้อมูลออกมาเป็น File Excel จากฐานข้อมูลได้
7. เว็บไซต์สามารถให้ผู้ใช้งานรับเข้า-เบิกจ่ายผลิตภัณฑ์ภายในระบบได้
8. เว็บไซต์สามารถค้นหาผลิตภัณฑ์และชิ้นงาน เพื่อนำไปเบิกงานกับฝ่ายวางแผนการผลิตได้
9. เว็บไซต์สามารถส่ง Email ข้อมูลออกไปให้แก่ผู้ใช้งานได้

### 1.4 วิธีดำเนินการวิจัย

1. รวบรวมข้อมูล ข้อกำหนดและความต้องการต่าง ๆ จากผู้ใช้งานที่เกี่ยวข้อง
2. วางแผนและออกแบบการพัฒนาโปรแกรม
3. พัฒนาโปรแกรม
4. ทดสอบการใช้งานของโปรแกรมตัวทดลอง
5. นำโปรแกรมตัวทดลองไปใช้งานจริง
6. เก็บข้อผิดพลาดหรือปัญหาที่ผู้ใช้งานพบ นำมาวางแผนแก้ไข
7. พัฒนาโปรแกรมจากข้อผิดพลาดหรือปัญหาที่เกิดขึ้น
8. ทดสอบการใช้งานโปรแกรมตัวจริง
9. นำโปรแกรมไปใช้งานจริง



ภาพที่ 1.1 Workflow การทำงานของโปรเจกต์

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการเข้าร่วมโครงการสหกิจศึกษา กับ บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด จำแนกออกได้เป็น 2 ส่วน คือ

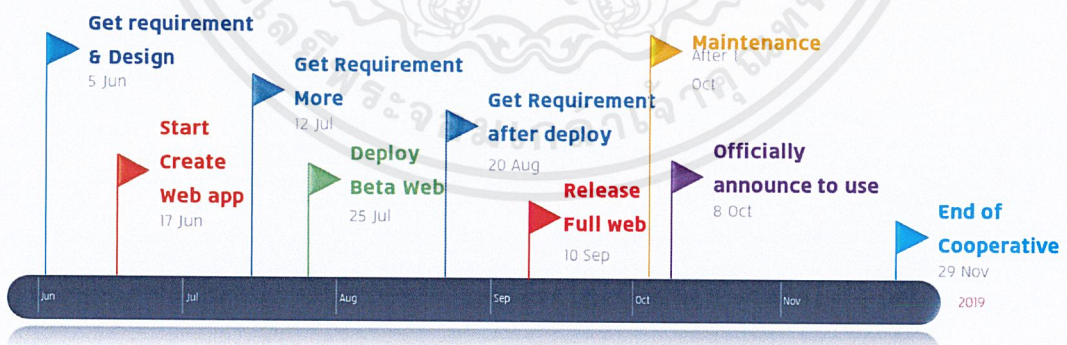
### 1.5.1 ประโยชน์ต่อบริษัท

- ลดการใช้ทรัพยากรบุคคล
- ลดความผิดพลาดที่เกิดจากการดำเนินงานของมนุษย์
- เพิ่มประสิทธิภาพในการทำงานของบุคลากรฝ่ายวิจัยและพัฒนา ฝ่ายวางแผนการผลิต และมีระบบการทำงานที่ดีขึ้น
- ช่วยอำนวยความสะดวกและลดระยะเวลาในการทำงานของบุคลากร ฝ่ายวิจัยและพัฒนา ฝ่ายวางแผนการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.5.2 ประโยชน์ต่อผู้วิจัย

- ได้เรียนรู้วัฒนธรรมขององค์กร และสภาพแวดล้อมในการทำงานในชีวิตจริง
- ได้เรียนรู้เกี่ยวกับระบบการทำงานภายในบริษัทและการปรับตัวเพื่อให้สามารถทำงานร่วมกับผู้อื่นในบริษัทได้
- ได้พัฒนาทักษะการสื่อสารอย่างมีประสิทธิภาพและมีมนุษยสัมพันธ์ที่ดีต่อผู้คนที่อยู่ในองค์กร
- ได้พัฒนาทักษะการนำเสนองานในที่สาธารณะอย่างมีประสิทธิภาพมากยิ่งขึ้น
- ได้เรียนรู้ในการรับมือกับปัญหา และการแก้ไขปัญหาที่เกิดขึ้นจากการทำงาน
- สามารถเรียงลำดับความสำคัญในการทำงาน และการจัดการกับปัญหาได้
- ได้นำความรู้ที่ได้เรียนมาจากสาขาวิชาวิศวกรรมสารสนเทศมาประยุกต์ใช้กับงานที่ได้รับมอบหมายในบริษัทได้อย่างมีประสิทธิภาพ
- สามารถนำความรู้และทักษะการทำงานมาปรับใช้ให้สอดคล้องกับการทำงานในอนาคต
- ได้พัฒนาทักษะในด้านการเขียนโปรแกรมด้วยภาษาในการเขียนโปรแกรมคอมพิวเตอร์ที่หลากหลาย
- ได้เรียนรู้และพัฒนาทักษะการใช้งานโปรแกรม NetBeans IDE กับโปรแกรม Oracle SQL Developer



ภาพที่ 1.2 Timeline การทำงานทั้งหมดของโปรเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

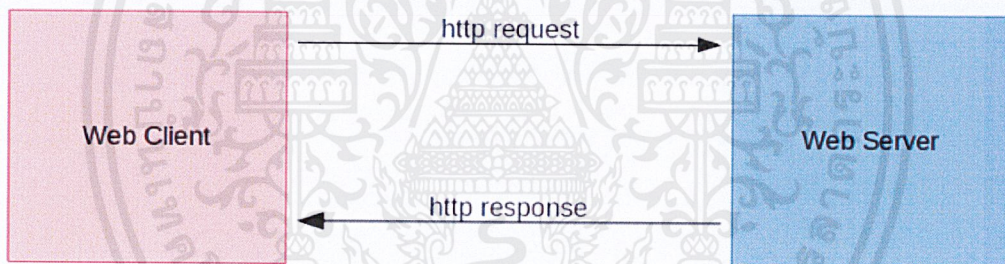
## บทที่ 2

### แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื่องจากการพัฒนาระบบการจัดเก็บงานวิจัยและพัฒนาหลังจากเสร็จสิ้นกระบวนการผลิตบนระบบเว็บแอปพลิเคชัน มีการปรับเปลี่ยนแพลตฟอร์มโครงสร้างและรูปแบบของเว็บใหม่ ดังนั้นผู้วิจัยจึงต้องศึกษาทฤษฎีที่เกี่ยวข้องและเป็นประโยชน์ต่าง ๆ เพื่อที่จะนำความรู้เหล่านี้มาประกอบรวมกันเพื่อทำให้งานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดี โดยแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องกับงานวิจัยชิ้นนี้ มีดังต่อไปนี้

#### 2.1 Hypertext Transfer Protocol (HTTP)

คือโพรโทคอลสื่อสารผ่าน internet ใช้ในการรับและส่งข้อมูล ในการแลกเปลี่ยนข้อมูลระหว่าง Client (ไคลเอนต์) และ Server (เซิร์ฟเวอร์) โดยการส่งจาก Client ไป Server เรียกว่า http request ส่วนข้อมูลที่ Server ตอบกลับมาที่ Client เรียกว่า http Response



ภาพที่ 2.1 การสื่อสารผ่าน Internet

##### 2.10.1 HTTP - Request

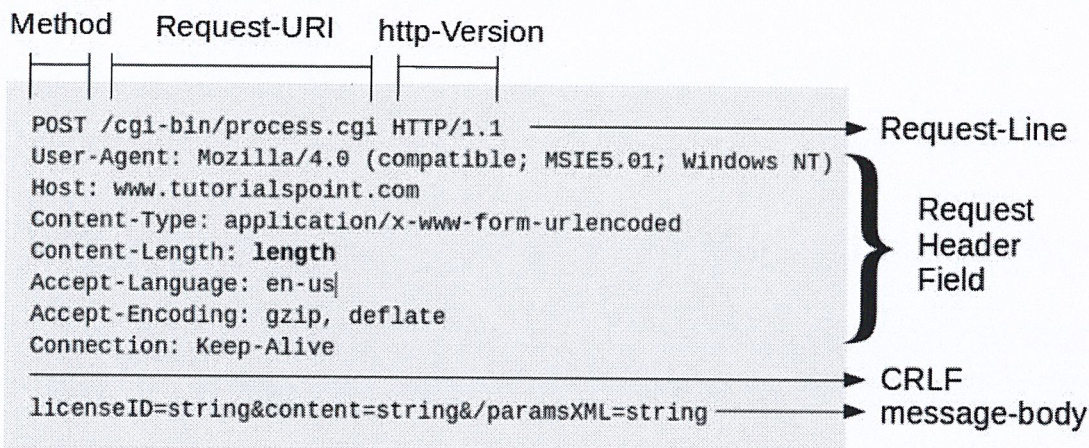
รูปแบบของ request จาก Client ไป Server เป็นดังต่อไปนี้

```
Request = Request-Line
          *( (general-header
            | request-header
            | entity-header) CRLF)
          CRLF
          [ message-body ]
```

ภาพที่ 2.2 การ Request จาก Client ไป Server

เมื่อเอาโครงสร้างของ Request message ไปเทียบกับข้อมูลตอนที่ใช้งานจริง โดยแบ่งแต่ละส่วนได้แบบภาพข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.3 การแบ่งส่วนของ Request message

### Request-Line

Request-Line เริ่มด้วย token method ตามด้วย Request-URI และ version ของ protocol จบด้วย CRLF ทุกส่วนถูกคั่นด้วยช่องว่าง (อักขระ SP)

```
Request-Line = Method SP Request-URI SP http-Version CRLF
```

ภาพที่ 2.4 Request-Line

### Request Method

http request ทำหน้าที่กำหนด action ที่จะกระทำต่อข้อมูล method โดยมีลักษณะเป็นตัวพิมพ์ใหญ่ทั้งหมดเสมอ

โดย method ที่สำคัญมี 4 ตัวคือ GET, POST, PUT, DELETE

- GET

เป็นการเรียกเก็บข้อมูลจาก URI ที่กำหนด method GET ควรใช้ในการดึงข้อมูลเท่านั้นและต้องไม่มีผลกระทบใด ๆ กับข้อมูล

- POST

ใช้สำหรับการสร้างข้อมูลใหม่โดยส่งข้อมูลผ่าน body

- PUT - ใช้สำหรับแทนที่ข้อมูลที่มีทั้งหมดด้วยข้อมูลใหม่ที่ส่งขึ้นไป

- DELETE - ใช้สำหรับลบข้อมูลที่มีอยู่ ของเป้าหมายที่กำหนดโดย URI

### Request Header Fields

header จะให้ผู้ใช้สามารถส่งผ่านข้อมูลเพิ่มเติมเกี่ยวกับ Request ของผู้ใช้ เช่น เป็นภาษาอะไร, ชนิดข้อมูลเป็นแบบไหน และนี่คือรายชื่อของ header ที่สำคัญบางอย่างที่สามารถเลือกใช้ได้

- Accept-Charset
- Accept-Encoding
- Accept-Language
- Connection
- Content-Length
- Content-Type
- Cookie
- Host
- User-Agent

### Message-body

เป็นส่วนสุดท้ายของ Request message เป็นส่วนที่ใช้สำหรับส่งข้อมูลให้กับ server อาจมีหรือไม่มีการส่งก็ได้ ขึ้นอยู่กับบริการที่เรียกด้วย

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://clearforest.com/">string</string>
```

ภาพที่ 2.5 Message-body

### 2.10.2 HTTP - Response

การ Response จะเกิดขึ้นหลังจากส่ง Request ไปที่ Server แล้ว Server ก็จะมีการตอบรับกลับมา ซึ่งเรียกข้อความที่ตอบกลับมานว่า http-Response message โดย Response message ประกอบไปด้วย 3 ส่วนหลักคือ Status-Line, Response Headers และ Message-body

```
Response = Status-Line
          *(( general-header
            | response-header
            | entity-header ) CRLF)
          CRLF
          [ message-body ]
```

ภาพที่ 2.6 การ Response หลังจากการ Request

## Status-Line

Status-Line เริ่มด้วย version ของ protocol ตามด้วย status-code และ Reason-Phase จบด้วย CRLF ทุกส่วนถูกคั่นด้วยช่องว่าง (อักขระ SP)

```
Status-line = HTTP-version SP Status-code SP Reason-Phase CRLF
```

## ภาพที่ 2.7 Status-Line

### Status-code

http status-code แบ่งออกเป็นหมวดหมู่ตามเลขที่อยู่ตัวหน้าสุด และที่ยกมาไว้เป็น Response ที่มักจะเจอ โดย Response อื่น ๆ

2xx (สำเร็จ) หมายความว่า การ request นั้นได้รับแล้วและกระทำตาม method สำเร็จโดย Server

- 200 Ok เป็นมาตรฐานของ HTTP Response นั้น Success สำหรับ GET, PUT หรือ POST
- 201 Create เป็น Response สำหรับข้อมูลใหม่ได้ถูกสร้างขึ้น ใช้สำหรับ POST
- 204 No Content เป็น Response สำหรับ request ที่ดำเนินการ Success แต่ไม่ได้ return ข้อมูลกลับ

### 3xx (Redirection)

- 304 Not Modified เป็น status code ที่บอกว่า client ได้รับการ Response แล้วอยู่ใน cache ไม่จำเป็นต้องส่งผ่านข้อมูลเดิมอีกครั้ง

4xx (Client error) โดย status ในกลุ่มนี้จะบอก client ว่า request ที่เข้ามา error

- 400 Bad Request บอกว่า request ที่ส่งมาโดย client นั้นไม่ถูกต้องดำเนินการ และ Server ไม่เข้าใจว่า request เกี่ยวกับอะไร
- 401 Unauthorized บอกว่า client ไม่ได้รับอนุญาตในการเข้าถึงข้อมูลและควรส่ง credential มาพร้อม request
- 403 Forbidden บ่งบอกว่า request นั้นถูกต้องและ client ได้รับการอนุญาต แต่ Client ไม่ได้รับการอนุญาตให้เข้าถึงข้อมูลด้วยเหตุผลบางประการ
- 404 Not Found บอกว่า resource ที่ request ไม่ว่างใช้งานตอนนี้
- 405 Gone บ่งบอกว่าข้อมูลที่ต้องการนั้นไม่มีอยู่แล้ว หรืออาจทำการย้ายข้อมูลไปที่อื่นแล้ว

### 5xx (Server error)

- 500 Internal Server Error บอกว่าการ request นั้นถูกต้อง แต่ server มีความสับสนและอาจบริการด้วยเงื่อนไขที่คาดการณ์ไม่ได้
- 503 Service Unavailable บอกว่า server ใช้การไม่ได้ หรือไม่วางที่จะรับและดำเนินการ request โดยส่วนใหญ่แล้ว server อยู่ในช่วงบำรุงรักษา

## 2.2 ภาษา Java

เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented) ที่ได้ถูกพัฒนาขึ้น เพื่อจุดประสงค์ในการใช้แทนภาษาซีพลัสพลัส (C++) โดยมีรูปแบบ ที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แม้ว่าชื่อ Java มีความคล้ายคลึงกับ JavaScript แต่ทั้งสองภาษานี้ไม่ได้มีความเกี่ยวข้องกันแต่อย่างใด ปัจจุบันมาตรฐานของภาษา Java ดูแลโดย Java Community Process ซึ่งเป็นกระบวนการอย่างเป็นทางการที่อนุญาตให้ผู้ที่สนใจเข้าร่วมกำหนดความสามารถใน Java Platform ได้ โปรแกรมที่เขียนขึ้นถูกสร้างให้อยู่ภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

### 2.2.1 ประวัติของภาษา Java

ภาษา Java เป็นภาษาโปรแกรมเชิงวัตถุที่พัฒนาขึ้นโดย “เจมส์ กอสลิง” และทีมวิศวกรของเขา จากบริษัทซันไมโครซิสเต็ม (Sun Microsystems) ในปี ค.ศ. 1991 ซึ่งเป็นบริษัทผู้ขายระบบ Unix ที่มีชื่อว่า Solaris ชื่อเดิมนั้นคือภาษา Oak ภายหลังจากที่มีการพัฒนาต่อมาเรื่อย ๆ ก็ได้มีการเปลี่ยนชื่อเป็น ภาษา Java ตามชื่อกาแฟ ที่ทีมพัฒนาใช้ดื่ม ภาษา Java พัฒนามาจากโครงการกรีน (The Green Project) คือโครงการที่ต้องการพัฒนาระบบซอฟต์แวร์เพื่อควบคุมเครื่องใช้ไฟฟ้าขนาดเล็กภายในบ้าน ที่มีค่าใช้จ่ายต่ำ ใช้งานง่าย มีข้อผิดพลาดน้อยที่สุด และสามารถปรับใช้กับอุปกรณ์ต่าง ๆ โดยไม่ต้องคอมไพล์โปรแกรมใหม่ ส่งผลให้ไม่จำกัดอยู่กับเครื่องหรือระบบปฏิบัติการ เพียงแค่ระบบปฏิบัติการเดียว

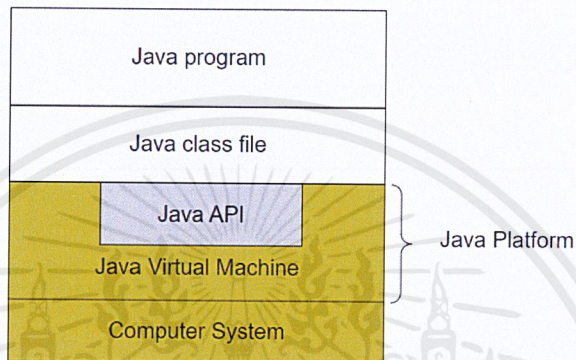
ภาษา Java เริ่มได้รับความนิยมอย่างแพร่หลายตั้งแต่ปี ค.ศ. 1995 ภาษา Java ยังสามารถนำไปใช้เป็นภาษาสำหรับอุปกรณ์แบบฝังต่าง ๆ เช่น โทรศัพท์ และอุปกรณ์ขนาดพกพาในแบบต่าง ๆ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกทั้งยังได้รับความนิยมนำไปใช้กับอุปกรณ์ที่ใช้สำหรับเข้าสู่อินเทอร์เน็ต โดยไม่ต้องใช้คอมพิวเตอร์ นอกจากนี้แล้วภาษา Java ยังถูกใช้งานในคอมพิวเตอร์แบบเอ็นซี (NC) ที่เน้นการทำงาน เป็นเครือข่ายจากเครื่อง Server ทำให้การติดต่อสื่อสารผ่านเครือข่ายใช้ช่องทางการสื่อสารน้อยกว่าการดึงมาทั้งโปรแกรมเป็นอย่างมาก

## 2.2.2 โครงสร้างของภาษา Java

โครงสร้างของภาษา Java แบ่งออกได้เป็น 4 ส่วน ดังนี้



ภาพที่ 2.8 โครงสร้างของภาษา Java

- Java programming Language คือ โปรแกรมที่เขียนขึ้นมาด้วยภาษา Java (.java) ในรูปของ text ที่สามารถอ่านได้ เรียกว่า Source code (ซอร์สโค้ด) ซึ่ง Source code จะถูกแปลง (compile) เป็น Java class file
- Java class file คือ Source code ที่ถูกแปลงเป็น .class หรือ byte code (ไบนารีโค้ด) ที่อยู่ในรูปของคำสั่งที่ Java Virtual Machine (Java VM) เข้าใจ
- Java API คือ กลุ่มของ ready-made software components หรือ กลุ่มส่วนประกอบสามารถนำมาใช้ได้โดยไม่ต้องเขียนเอง ประกอบไปด้วยไลบรารีของคลาสและอินเทอร์เฟซ
- Java Virtual Machine (JVM) คือ ส่วนที่ไปติดต่อสั่งงานโดยตรงต่อคอมพิวเตอร์ภายในประกอบด้วย Class loader ทำหน้าที่โหลด Class file จากโปรแกรมและจาก Java API และส่วน Execution engine ทำหน้าที่แปล byte code ซึ่งมีหลายรูปแบบ เช่น Just in time, Adaptive optimizer

### 2.2.3 ข้อดีของ ภาษา Java

1. ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุช่วยให้สามารถใช้คำหรือชื่อ ต่าง ๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่ายขึ้น

2. โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java มีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลง แก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็ยังสามารถถูก compile และ run บนเครื่องพีซีธรรมดาได้

3. ภาษา Java มีความซับซ้อนน้อยกว่าภาษา C++ เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java มีจำนวน code น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ทำให้ใช้งานได้ง่ายกว่า และลดความผิดพลาดได้มากขึ้น

### 2.2.4 ข้อเสียของ ภาษา Java

ภาษา Java ทำงานได้ช้ากว่า native code (โปรแกรมที่ compile ให้อยู่ในรูปของภาษาเครื่อง) หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น อย่างเช่น C หรือ C++ ทั้งนี้ก็เพราะว่าโปรแกรมที่เขียนขึ้นด้วยภาษา Java จะถูกแปลงเป็นภาษากลางก่อน แล้วเมื่อโปรแกรมทำงานคำสั่งของภาษากลางนี้จะถูกเปลี่ยนเป็นภาษาเครื่องอีกทีหนึ่งทีละคำสั่ง (หรือกลุ่มของคำสั่ง) ณ runtime ทำให้ทำงานช้ากว่า native code ซึ่งอยู่ในรูปของภาษาเครื่องแล้วตั้งแต่ compile โปรแกรมที่ต้องการความเร็วในการทำงานจึงไม่นิยมเขียนด้วย Java

### 2.2.5 จุดมุ่งหมายในการพัฒนาภาษา Java

1. เขียนโปรแกรมในเชิงวัตถุ
2. โปรแกรมที่เขียนไม่ขึ้นกับแพลตฟอร์ม (Platform) ใด ๆ
3. เหมาะสมกับการนำไปใช้ในระบบเครือข่าย เพราะมีไลบรารีที่เหมาะสม
4. โปรแกรมสามารถเรียกใช้งานจากระยะไกลได้อย่างปลอดภัย

## 2.3 ภาษา HTML

ย่อมาจาก Hyper Text Markup Language คือภาษาคอมพิวเตอร์ที่ใช้ในการแสดงผลของเอกสารบน website หรือที่เรียกกันว่าเว็บเพจ ภาษา HTML ถูกพัฒนาขึ้นมาจากภาษา SGML (Standard Generalized Markup Language) โดย Tim Berners-Lee และถูกกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C) และจากการพัฒนาทางด้าน Software ของ Microsoft ทำให้ภาษา HTML เป็นอีกภาษาหนึ่งที่ใช้เขียนโปรแกรมได้ หรือที่เรียกว่า HTML Application มีโครงสร้างการเขียนที่อาศัยตัวกำกับ เรียกว่า Tag ควบคุมการแสดงผลของข้อความ รูปภาพ หรือวัตถุอื่น ๆ เรียกใช้เอกสารเหล่านี้โดยการใช้โปรแกรมเว็บเบราว์เซอร์ (Web Browser) เช่น Mozilla Firefox, Opera, Netscape navigator, Internet Explorer ฯลฯ เป็นต้น ในปัจจุบัน HTML เป็นมาตรฐานหนึ่งของ ISO ซึ่งจัดการโดย World Wide Web Consortium (W3C) ในปัจจุบันทาง W3C ผลักดันรูปแบบของ HTML แบบใหม่ที่เรียกว่า XHTML ซึ่งเป็นลักษณะของโครงสร้าง XML แบบหนึ่งที่มีหลักเกณฑ์ในการกำหนดโครงสร้างของโปรแกรมที่มีรูปแบบที่มาตรฐานกว่า มาทดแทนใช้ HTML รุ่น 4.01 ที่ใช้กันอยู่ในปัจจุบันขณะที่ HTML รุ่น 5 ยังคงอยู่ในระหว่างการพิจารณาในการใช้งาน

### 2.3.1 ส่วนประกอบของ HTML

HTML มีโครงสร้างการเขียนโดยอาศัย Tag ซึ่ง Tag ในภาษา HTML ส่วนมากมี Tag เปิด และ Tag ปิด ในการควบคุมการแสดงผลของข้อความ รูปภาพ หรือวัตถุอื่น ๆ แต่ละ Tag อาจมีส่วนขยาย เรียกว่า Attribute สำหรับจัดรูปแบบเพิ่มเติม โดยมี tag หลัก ๆ คือ `<html>...</html>` ในการใช้งาน HTML ต้องเริ่มด้วย `<html>` และปิดด้วย `</html>` เสมอ ส่วนภายใน Element `<html>` ประกอบด้วยส่วนของ `<head>...</head>` ที่ใช้กำหนดรายละเอียดต่าง ๆ เกี่ยวกับเว็บเพจ กำหนดสไตล์ CSS และ Script ต่าง ๆ และส่วนของ `<body>...</body>` เป็นส่วนที่แสดงเนื้อหาที่แสดงทางหน้าจอทั้งหมดมีส่วนประกอบ ได้แก่ ข้อความ รูป ภาพ ลิงค์ ตาราง ลิสต์ เป็นต้น

โดย Attributes เป็นส่วนขยายความสามารถของ Tag ใช้สำหรับจัดรูปแบบ เช่น ขนาด สี ระยะห่าง เป็นต้น ค่าของ attribute อยู่ในเครื่องหมาย Double quote ("...") เช่น `<p align="center">ข้อความในพารากราฟนี้จัดวางอยู่กึ่งกลางหน้าจอ </p>`, `<hr width="100" color="red" noshade>` ใช้สร้างเส้นคั่นยาว 100 pixel สีแดงทึบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3.2 โครงสร้างของ HTML

### 1. ส่วนประกาศ DOCTYPE

<!DOCTYPE> ควรใส่ในไฟล์เอกสารทุก ๆ หน้า โดยวางไว้บรรทัดแรกเสมอ เพื่อบอกให้เว็บเบราว์เซอร์ทราบว่า ใช้คำสั่ง HTML รุ่นใด และบอกชนิดของเอกสาร (Document Type Definition: DTD) ที่ใช้ ซึ่งช่วยให้เว็บเบราว์เซอร์แปลเอกสารได้อย่างถูกต้อง ดังนี้

บอกรุ่นของ HTML ที่ใช้

- สำหรับ HTML รุ่นดั้งเดิม

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML Level 1//EN">
```

- สำหรับ HTML 2.0

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

- สำหรับ HTML 3.2

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
```

- สำหรับ HTML 4.0

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

### 2. คำสั่งในหัวข้อของ head (Head Section)

<HEAD>...</HEAD> Head Section เป็นส่วนที่ใช้อธิบายเกี่ยวกับข้อมูลเฉพาะของหน้านั้น ๆ เช่น ชื่อเรื่องของ หน้าเว็บ (Title) ชื่อผู้จัดทำเว็บ (Author) คีย์เวิร์ดสำหรับการค้นหา (Keyword) โดยมี Tag สำคัญ คือ

```
<HEAD>
  <TITLE>ข้อความอธิบายชื่อเรื่องของเว็บ</TITLE>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
  <META NAME="Author" CONTENT="ชื่อผู้พัฒนาเว็บ">
  <META NAME="Keywords" CONTENT="ข้อความ 1, ข้อความ 2 ">
</HEAD>
```

ภาพที่ 2.9 รูปแบบคำสั่งในส่วน <head></head>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 2.9 แสดงให้เห็นส่วนประกอบที่อยู่ภายใน tag <HEAD>...</HEAD> มีดังนี้

#### TITLE

- ข้อความที่ใช้เป็น TITLE ไม่ควรพิมพ์เกิน 64 ตัวอักษร ไม่ต้องใส่ลักษณะพิเศษ เช่น ตัวหนา เอียง หรือสี โดยข้อความในส่วนนี้จะแสดงผลใน title bar ของเว็บเบราว์เซอร์

#### META

- Tag META จะไม่ปรากฏผลบนเบราว์เซอร์ แต่เป็นส่วนสำคัญในการจัดอันดับบัญชีเว็บ สำหรับผู้ให้บริการสืบค้นเว็บ (Search Engine เช่น google) ส่วน charset=utf-8 มีไว้บอกว่าใช้ชุดตัวอักษรแบบใดในการแสดงผล ในที่นี้หมายถึง UTF-8 ที่รองรับภาษาไทย

#### KEYWORD

- ดังภาพที่ 2.9 แสดงให้เห็นว่าสามารถใช้ keywords มากกว่า 1 คำได้โดยใช้เครื่องหมาย (,) ในการคั่นระหว่างคำ

การพิมพ์ชุดคำสั่ง HTML สามารถพิมพ์ได้ทั้งตัวพิมพ์เล็ก ตัวพิมพ์ใหญ่ หรือผสม การย่อหน้า เว้นบรรทัด หรือช่องว่าง สามารถกระทำได้อิสระ โปรแกรมเบราว์เซอร์ไม่สนใจเกี่ยวกับระยะเว้นบรรทัดหรือย่อหน้า หรือช่องว่าง

### 3. คำสั่งในส่วนของ (Body Section)

Body Section เป็นส่วนเนื้อหาหลักของหน้าเว็บ ซึ่งการแสดงผลจำเป็นต้องใช้ Tag จำนวนมาก ขึ้นอยู่กับลักษณะของข้อมูล เช่น ข้อความ รูปภาพ เสียง วิดีโอ หรือไฟล์ต่าง ๆ ส่วนเนื้อหาเอกสารเว็บ เป็นส่วนการทำงานหลักของหน้าเว็บ ประกอบด้วย Tag มากมาย ตามลักษณะของข้อมูลที่ต้องการนำเสนอ การป้อนคำสั่งในส่วนนี้ ไม่มีข้อจำกัดสามารถป้อนติดกัน หรือ 1 บรรทัดต่อ 1 คำสั่งก็ได้ แต่มักยึดรูปแบบที่อ่านง่าย คือ การทำย่อหน้าในชุดคำสั่งที่เกี่ยวข้องกัน ทั้งนี้ให้ป้อนคำสั่งทั้งหมดภายใต้ Tag <BODY> ... </BODY> และแบ่งกลุ่มคำสั่งได้ตามลักษณะการแสดงผลต่าง ๆ เช่น กลุ่มคำสั่งเกี่ยวกับการจัดรูปแบบเอกสาร กลุ่มคำสั่งจัดแต่งรูปแบบตัวอักษร กลุ่มคำสั่งการทำเอกสารแบบรายการ (List) กลุ่มคำสั่งเกี่ยวกับการทำลิงค์ กลุ่มคำสั่งจัดการรูปภาพ กลุ่มคำสั่งจัดการตาราง กลุ่มคำสั่งควบคุมเฟรม

### 2.3.3 การสร้างเว็บเพจ HTML

การสร้างเว็บเพจ โดยใช้ภาษา HTML สามารถทำได้โดยใช้ Text Editor ต่าง ๆ เช่น Notepad Editplus หรือออสซี่โปรแกรมที่เป็นเครื่องมือช่วยสร้างเว็บเพจ เช่น Microsoft Front Page Dream Weaver ซึ่งอำนวยความสะดวกในการสร้างหน้า HTML ในลักษณะ What You See Is What You Get และยังสามารถนำสคริปต์มาแทรก ตัดต่อ สร้างลูกเล่นสีสันให้กับเว็บเพจได้ การเรียกใช้งานหรือทดสอบการทำงานของเอกสาร HTML ใช้เว็บเบราว์เซอร์ต่าง ๆ เช่น Internet Explorer (IE), Mozilla Firefox, Safari, Opera หรือ Google Chrome เป็นต้น

## 2.4 ภาษา JSP

Java server Pages เป็นภาษาที่พัฒนาโดย บริษัท Sun Microsystems (ซึ่ง ไมโครซิสเต็ม) ดังนั้น JSP ก็คือภาษา script (สคริปต์) ที่ทำงานอยู่ในส่วนของฝั่ง Server ซึ่งเรียกว่า server-side scripting (เซิร์ฟเวอร์-ไซด์-สคริปต์) หมายถึง เป็นการประมวลผลการทำงานต่าง ๆ ของภาษาสคริปต์ JSP จะถูกรัน หรือคอมไพล์ที่ฝั่ง Server ให้เสร็จเรียบร้อยก่อน ก่อนนำผลลัพธ์ที่ได้ส่งกลับคืนไปยัง Client เครื่องคอมพิวเตอร์ที่ไปร้องขอบริการและรับบริการอย่างใดอย่างหนึ่งจาก Server

JSP เป็นเทคโนโลยีไว้สำหรับพัฒนาแอปพลิเคชันบนเว็บไซต์ในรูปแบบของ Server และ Client แสดงผลและโต้ตอบกับ User Interface ผ่านเว็บเบราว์เซอร์โดยอาศัยการทำงานร่วมกับพวก Client Tag เช่น ภาษา Html, JavaScript, CSS และ JQuery ให้ได้ผลลัพธ์ตามที่ต้องการ ซึ่ง JSP ก็เหมือนกับโปรแกรม ASP, PHP และ .Net รูปแบบการทำงานนั้นไม่แตกต่างกัน แต่แตกต่างกันตรงที่ JSP เป็น Subset (ซับเซต) ของภาษา Java โดยรูปแบบการเขียนนั้น ใช้รูปแบบคำสั่งและชุด SDK (เอส ดี เค) ของ Java และใน JSP มีนามสกุลของไฟล์เป็น .jsp

### 2.4.1 ประเภทของ JSP

การพัฒนาเว็บไซต์ หรือเว็บแอปพลิเคชัน ด้วยภาษา JSP สามารถแยกย่อยได้อีกประมาณ 2-3 รูปแบบ และแต่ละรูปแบบการวางโครงสร้าง และรูปแบบการเขียนก็ต่างกัน แต่พื้นฐานแล้วยังใช้ Syntax ภาษา Java เหมือนกันโดยสรุป ๆ แล้ว แยกออกเป็น 3 รูปแบบคือ

1. Scriptlets คือการเขียน JSP ร่วมกับ HTML หรือ JavaScript ในหน้าเดียวกัน ให้มองว่าเหมือนพวก ASP และ PHP แต่ใช้นามสกุลไฟล์ .jsp ส่วน Code นั้นก็สามารถเขียนร่วมกับ HTML หรือสลับกันไปมาได้

2. Servlet เป็นการเขียน JSP ขั้นสูงซึ่งแยกในส่วนของ Interface (HTML) กับส่วนของ Code ที่เป็นภาษา Java ออกจากกัน ให้มองถึงรูปแบบการเขียน ASP.Net ที่แยก .ASPX และ .VB ไว้คนละไฟล์

3. Framework การเขียน Web Application ด้วย JSP มี Framework มารองรับการเขียนหลาย ๆ ตัวเช่น Spring, Struts และ Hibernate โดยรูปแบบการเขียนจะเป็น Pattern ในรูปแบบของ MVC

ซึ่งการเลือกใช้ JSP ในรูปแบบไหนนั้นก็ขึ้นอยู่กับความต้องการของผู้ใช้ เช่น หากเขียนเว็บไซต์ทั่ว ๆ ไป ขนาดไม่ใหญ่มากก็ควรใช้แค่ Scriptlets แต่ถ้าเป็นเว็บไซต์ที่ขนาดใหญ่ขึ้น ต้องการความเป็นระเบียบเรียบร้อยก็ควรเลือกใช้ Servlet และในงานที่มีขนาดใหญ่เขียนกันหลาย ๆ คนก็ควรเลือกใช้ Framework ซึ่งเหมาะสมอย่างยิ่งสำหรับการเขียน Web Application ที่ขนาดใหญ่ และแยกย่อยออกเป็นหลาย ๆ ระบบ

#### 2.4.2 ข้อดีของ JSP

1. JSP สามารถใช้งานได้แบบไม่มีค่าใช้จ่ายและสามารถรองรับได้ทุก Platform ไม่ว่าจะเป็น Windows, Linux และ iOS รวมทั้ง Software อื่น ๆ ที่จำเป็นต่อการใช้งาน และการพัฒนายังไม่มีค่าใช้จ่ายเช่นเดียวกัน

2. JSP ที่พัฒนาด้วยภาษา Java มีประสิทธิภาพการทำงานสูงมากและ JSP ยังมีการจัดการจัดสรรค่าตัวแปรในเว็บเซิร์ฟเวอร์ดีมาก เช่น เมื่อสร้างตัวแปรแล้วสามารถนำไปใช้งานได้ในระดับ Page Session และแอปพลิเคชันได้อย่างง่ายดาย ซึ่งข้อดีในส่วนนี้ช่วยลดการทำงานที่ซ้ำซ้อนของ Server ได้อย่างดี ซึ่งเหมาะสมอย่างยิ่งกับแอปพลิเคชันที่รองรับการทำงานขนาดใหญ่ได้

### 2.4.3 ข้อเสียของ JSP

ภาษา JSP ไม่สามารถทำงานร่วมกับภาษา PHP ได้ และค่อนข้างยากในการจัดเก็บ และในการย้ายไฟล์จาก Server หนึ่งไปยังอีก Server หนึ่ง เพราะโดยทั่วไปแต่ละ webserver มีลักษณะการจัดเก็บไฟล์ไม่เหมือนกัน แต่เมื่อ Java Servlet Specification เวอร์ชัน 2.2 ออกมา Server ที่ใช้รัน Servlet v2.2 ถูกบังคับให้มีความต้องการสนับสนุนการจัดเก็บไฟล์แบบหนึ่ง ที่เรียกว่า Web Application

## 2.5 ภาษา Javascript

JavaScript คือ ภาษาคอมพิวเตอร์ยุคใหม่สำหรับการเขียนโปรแกรมบนระบบ อินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง เป็นภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ดูมีการเคลื่อนไหว โดยสามารถเขียนโปรแกรม JavaScript เพิ่มเข้าไปในเว็บเพจ เพื่อใช้ประโยชน์สำหรับงานด้านต่าง ๆ ทั้งการคำนวณ การแสดงผล การรับ-ส่งข้อมูล และที่สำคัญคือ สามารถโต้ตอบกับผู้ใช้ได้อย่างทันที ซึ่งมีวิธีการทำงานในลักษณะแปลความและดำเนินงานไปที่ละคำสั่ง (interpret) หรือเรียกว่า Object Oriented Programming ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่ง Client และ ทางฝั่ง Server

### 2.5.1 โครงสร้างคำสั่งของ Javascript

โครงสร้างคำสั่งภาษา Java Script มีรูปแบบที่ขึ้นต้นด้วย คำสั่ง หรือนิพจน์ต่าง ๆ จากนั้นต้องตามด้วยเครื่องหมาย ; เพื่อบอกให้ทราบว่าจบคำสั่งแล้ว เช่น

```
<script language="javascript">
    document.write("KMITL");
    alert("ITE");
</script>
```

1. document หมายถึง เนื้อหาในหน้าต่างทั้งหมด

2. . (จุด) ใส่เพื่อแสดงว่ามีคำสั่งมาทำงานกับ document ซึ่งในที่นี้ถือว่า document นั้นเป็นตัวถูกระทำ เรียกว่า object

3. Write เป็นวิธีการกระทำ (method) ที่เขียนสิ่งต่าง ๆ ลงใน document เนื่องจากภาษา Java Script นั้นใช้เครื่องหมาย ; เพื่อแยกแต่ละคำสั่งออกจากกัน ดังนั้นสามารถเขียนคำสั่งต่าง ๆ ให้ต่อกันโดยไม่จำเป็นต้องขึ้นบรรทัดใหม่ ได้ เช่น

```
document.write("Hello Word"); alert("Hello Word");
```

4. เครื่องหมาย ( ) ใช้กับ Properties, Method หรือ นิพจน์และคำสั่ง ต่าง ๆ เช่น

```
document.write("KMITL");
alert("KMITL");
Test = (Answer = 10) ? "correct" : "Wrong"
```

5. เครื่องหมาย { } ใช้ในการเริ่มต้น และสิ้นสุดคำสั่ง และ ฟังก์ชันต่าง ๆ เช่น

```
if (Test = "correct") {alert(Correct Answer);}
function Test1() {
    document.write("KMITL"); }
```

## 2.5.2 ลักษณะการทำงานของ JavaScript

JavaScript เป็นภาษาสคริปต์เชิงวัตถุ หรือเรียกว่า อ็อบเจกต์โอเรียนเต้ด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ตสำหรับผู้เขียนเอกสารด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ทำงานร่วมกับภาษา HTML และภาษาจาวาได้ทั้งทางฝั่ง Client (ไคลเอนต์) และ ทางฝั่ง Server (เซิร์ฟเวอร์) โดยมีลักษณะการทำงานดังนี้

- Navigator JavaScript เป็น Client-Side JavaScript ซึ่งหมายถึง JavaScript ที่ถูกแปลทางฝั่ง Client ( หมายถึงฝั่งเครื่องคอมพิวเตอร์ของผู้ใช้ ไม่ว่าจะเป็นเครื่องพีซี เครื่องแมคอินทอช หรือ อื่น ๆ ) จึงมีความเหมาะสมต่อการใช้งานของผู้ใช้ทั่วไปเป็นส่วนใหญ่
- LiveWire JavaScript เป็น Server-Side JavaScript ซึ่งหมายถึง JavaScript ที่ถูกแปลทางฝั่ง Server ( หมายถึงฝั่งเครื่องคอมพิวเตอร์ของผู้ให้บริการเว็บ โดยอาจเป็นเครื่องของซันซิลิคอมกราฟิกส์ หรืออื่น ๆ ) สามารถใช้ได้เฉพาะกับ LiveWire ของเน็ตสเคป โดยตรง

### 2.5.3 ข้อดีและข้อเสียของ JavaScript

การทำงานของ JavaScript เกิดขึ้นบนเบราว์เซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าใช้เครื่อง Server อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่อง Server (เรียกว่า server-side script) ดังนั้นจึงต้องใช้บน Server ที่สนับสนุนภาษาเหล่านั้นเท่านั้น อย่างไรก็ตาม จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือไม่สามารถรับและส่งข้อมูลต่าง ๆ กับ Server โดยตรง เช่น การอ่านไฟล์จาก Server เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบน Server เป็นต้น ดังนั้นงานลักษณะนี้จึงยังคงต้องอาศัยภาษา server-side script อยู่

## 2.6 รูปแบบข้อมูล JSON

JSON หรือ JavaScript Object Notation คือ Standard format รูปแบบของข้อมูลตัวอักษร (String) ที่ใช้สำหรับแลกเปลี่ยนข้อมูลที่มีขนาดเล็ก ใช้ในการสร้าง object ขึ้นมาเพื่อส่งข้อมูลระหว่าง application หรือ Applications Program Interface (API) โดย format มีรูปแบบเป็น คู่ Key-Value หรือเป็นแบบ Array และสามารถนำมาใช้แทน XML format ได้ JSON เป็น format ที่ได้รับการใช้งานจาก JavaScript มาก่อน แต่ปัจจุบันมีภาษา programming หลายชนิดที่ เริ่มใช้งาน JSON โดนสามารถสร้างและแปลง format ไปมาได้ JSON สามารถรับส่งชุดค่าตัวแปรได้ทั้งฝั่ง Client (ไคลเอนต์) และฝั่ง Server (เซิร์ฟเวอร์)

โดยทั้ง 2 ฝั่งสามารถใช้การเข้ารหัส JSON Encode (รหัสเจสัน) และ JSON Decode (ถอดรหัสเจสัน) เพื่ออ่านค่าตัวแปรเหล่านั้น ซึ่งอยู่ในรูปแบบของตัวแปรชุด และสำหรับตัวแปร JSON นั้นไม่จำกัดแค่รับส่งข้อมูลผ่านเว็บเบราว์เซอร์เท่านั้น แต่ยังสามารถนำ JSON ไปประยุกต์กับการรับส่งข้อมูลในรูปแบบอื่น ๆ ได้ เช่น การจัดเก็บข้อมูลในรูปแบบของชุดของตัวอักษรในข้อความหรือการรับส่งผ่านเว็บเซอร์วิส (Web Service)

### 2.6.1 ประเภทของ JSON

1. Number - เป็นตัวเลขเท่านั้น
2. String - Unicode ข้อความ ที่ใช้เครื่องหมาย double-quote (") เป็นตัวบ่งบอก และสามารถใช้ backslash syntax ได้
3. Boolean - ค่า True or False
4. Array ชุดข้อมูล ซึ่งเป็นชนิดใดก็ได้ของ JSON ที่ใช้สัญลักษณ์วงเล็บเหลี่ยม (square bracket) เช่น [var1,var2] เป็นตัวแสดง และคั่นด้วย comma แต่ละค่าใน array
5. Object ชุดข้อมูลที่เป็นคู่ Key-Value แบบ strings ใช้สัญลักษณ์วงเล็บปีกกา (curly bracket) เช่น {key1:value1, key2:value2} ใช้ comma เป็นตัวแบ่งแต่ละคู่ และใช้ colon เป็นตัวแบ่งระหว่าง key และ value
6. Null คำว่าง

### 2.6.2 โครงสร้างของ JSON

JSON นั้นใช้ลักษณะของภาษา JavaScript จะไม่ถูกมองว่าเป็นภาษาโปรแกรม แต่กลับถูกมองว่าเป็นภาษาในการแลกเปลี่ยนข้อมูลมากกว่าซึ่งในปัจจุบันมีไลบรารี (Library) ของภาษาโปรแกรมอื่น ๆ ที่ใช้ประมวลผลข้อมูลในรูปแบบ JSON ตัวอย่างการเขียนในรูปแบบ JSON มีดังนี้

```
[  
  {"firstname":"name", "lastname":"name"},  
  {"firstname":"name1", "lastname":"name2"}  
]
```

ภาพที่ 2.10 ตัวอย่างการเขียนในรูปแบบ JSON

นอกจาก JSON นั้นยังสามารถจัดเก็บข้อมูลที่เป็นลักษณะของ Master - Detail ที่เพิ่มความซับซ้อนของข้อมูลได้อีกด้วย ยกตัวอย่างการจัดเก็บข้อมูลดัง ภาพที่ 2.11

```
[
  {
    "firstname" : "name",
    "lastname" : "name",
    "address" : [
      {
        "address1" : "address",
        "province" : "Bangkok",
        "country" : "Thailand"
      }
    ]
  }
]
```

ภาพที่ 2.11 ตัวอย่างการเขียนในรูปแบบ JSON ในลักษณะของ Master – Detail

### 2.6.3 จุดเด่นของ JSON

1. JSON เข้าใจง่าย
2. JSON ไม่จำเป็นต้องใช้ tag เปิดปิด
3. JSON มีการเก็บข้อมูลแบบลำดับชั้น (Hierarchical)
4. JSON สามารถส่งค่าผ่านทาง JavaScript ได้
5. JSON มีคำสั่งที่สั้น กระชับ และสามารถถูกอ่านหรือเขียนได้เร็ว
6. JSON ไม่มีจำกัดคำเฉพาะ เช่น public private เป็นต้น
7. JSON สามารถเก็บค่าแบบตัวแปรชุด (Array) ได้

## 2.7 ภาษา SQL

SQL หรือ Structured Query Language เป็นภาษามาตรฐานที่ใช้ในการเข้าถึงฐานข้อมูล อยู่ใน Relational Database Management System (RDBMS) โดยสามารถใช้งาน ภาษา SQL ได้จากโปรแกรมต่าง ๆ ที่ต้องกระทำการกับระบบฐานข้อมูล เช่น ค้นหาข้อมูล เปลี่ยนแปลง เพิ่ม ลบ ข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลในรูปแบบตารางที่มีลักษณะเป็น คอลัมน์ (column) และ แถว (row) เรียกข้อมูลเหล่านี้ว่าถูกเก็บอยู่ใน ตาราง (table) ซึ่ง SQL ยังสามารถสร้างตารางขึ้นมาใหม่ (create) รวมถึงลบ (drop) และเปลี่ยนแปลงค่า (alter) ของตารางได้สำหรับคำสั่ง SQL ประกอบไปด้วย Data Definition Language (DDL) ใช้ในการสร้างฐานข้อมูล, Data Manipulation Language (DML) ใช้ เพิ่ม ลบ และเปลี่ยนแปลงข้อมูลในตาราง และ Data Control Language (DCL) ใช้ในการกำหนดสิทธิการอนุญาต ซึ่ง SQL นั้นถือว่าเป็นภาษามาตรฐานกลางที่ใช้ในระบบฐานข้อมูลต่าง ๆ โดยเป็นมาตรฐานของ ANSI (American National Standard Institute)

### 2.7.1 ประวัติภาษา SQL

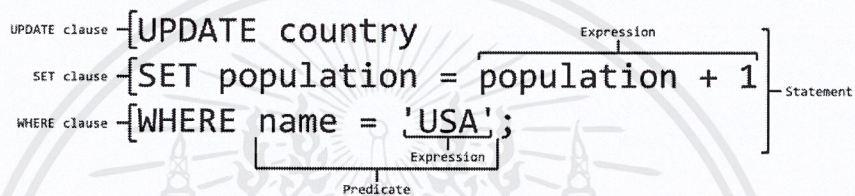
จากเอกสารของ ดร.เอ็ดการ์ เอฟ. คอตต์ แดลงต่อสาธารณชนในเดือนมิถุนายน ค.ศ.1970 ในนิตยสารของเอซีเอ็ม (Association for Computing Machinery : ACM) แบบจำลองของคอตต์ที่วางเอาไว้ได้กลายเป็นผลงานที่ยอมรับทั่วโลก สำหรับระบบการจัดการฐานข้อมูลในปัจจุบัน

ในช่วงปี ค.ศ. 1970 ณ ศูนย์พัฒนาคอมพิวเตอร์ซานโฮเซของไอบีเอ็มได้วิจัยและพัฒนาฐานข้อมูลที่เรียกว่า System R ขึ้นตามแบบจำลองของคอตต์ โดยใช้ระบบภายในภายใต้ชื่อว่า ภาษาสอบถามเชิงโครงสร้างภาษาอังกฤษ หรือ ซีเควล (Structured English Query Language : SEQUEL) ซึ่งภายหลังถูกเปลี่ยนชื่อมาเป็น เอสคิวแอล (SQL) เนื่องจากชื่อซีเควล (SEQUEL) ช้ำกับชื่อเครื่องหมายการค้าของบริษัทผลิตเครื่องบิน Hawker-Siddeley ของสหราชอาณาจักร ถึงแม้ว่าเอสคิวแอลพัฒนามาจากแนวความคิดของ ดร.คอตต์ แต่การพัฒนาทั้งหมดไม่ได้มาจาก ดร.คอตต์ แต่พัฒนาจาก โดนัลด์ ดี. แชมเบอร์ลิน (Donald D. Chamberlin) และ เรย์มอนด์ เอฟ. บอยซี (Raymond F. Boyce) จากไอบีเอ็ม ซึ่งเป็นผู้ที่ทำให้ SQL เป็นที่นิยมมากขึ้น

## 2.7.2 การใช้งานภาษา SQL

1. ใช้กับเว็บไซต์ เพื่อแสดงผลข้อมูลจากฐานข้อมูล DBMS เช่น Microsoft Access, SQL Server, MySQL, Oracle
2. ใช้ร่วมกับระบบฐานข้อมูล RDBMS เช่น MS SQL Server, IBM DB2, Oracle, MySQL และ Microsoft Access
3. ใช้ในการกำหนดในระบบวิเคราะห์ข้อมูล (Analysis Tools) ที่เปิดช่องให้สามารถ ทำการใส่ หรือ ปรับปรุง SQL ได้ด้วยตัวเอง

## 2.7.3 SQL Syntax



ภาพที่ 2.12 โครงสร้างตัวอย่างคำสั่ง SQL

จากภาพที่ 2.12 ภาษา SQL นั้นถูกแบ่งออกมาเป็นส่วนประกอบต่าง ๆ เรียกส่วนเหล่านี้ตามรูปแบบ ดังนี้

- Clauses คือองค์ประกอบหนึ่งของ statement และ query ส่วนนี้เป็น Optional ซึ่งมีหรือไม่มีก็ได้
- Expressions คือการสร้างผลลัพธ์ออกมาในรูปของ table ที่ประกอบด้วย column และ row ต่าง ๆ จากข้อมูลดิบ
- Predicates คือรูปแบบเงื่อนไขที่มีผลลัพธ์หรือที่เรียกว่า Boolean มีค่าเป็น true, false, unknown
- Queries คือการดึงข้อมูลตามเงื่อนไข (clause) เป็นส่วนสำคัญใน SQL
- Statements คือส่วนที่มีผลต่อโครงสร้างข้อมูล ทำหน้าที่จัดการข้อมูล transactions, program flow, session และวิเคราะห์ปัญหาโดยจำเป็นต้องจบด้วย semicolon (;) ซึ่งเป็นสิ่งจำเป็นที่ต้องใส่สำหรับการเขียน ภาษา SQL
- Insignificant whitespace หรือช่องว่าง สำหรับใน SQL statement และ query นั้นไม่สนใจ ทำให้ SQL สามารถเขียนในรูปแบบที่หลากหลาย

## 2.8 ภาษา CSS

ภาษา CSS หรือ Cascading Style Sheet มักเรียกโดยย่อว่า "สไตลชีต" เป็นภาษาที่ใช้สำหรับส่วนของการจัดรูปแบบการแสดงผล ตกแต่งเอกสาร HTML และ XHTML โดยที่ CSS กำหนดเกณฑ์ในการระบุรูปแบบ (หรือ "Style") ของเนื้อหาในเอกสารให้มีหน้าตา สี สัน ระยะห่าง พื้นหลัง เส้นขอบและอื่น ๆ ตามที่ต้องการ มีลักษณะเป็นภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะ ซึ่งการระบุรูปแบบ (Style) นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสารไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลลัพท์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลง เนื้อหาเอกสารบ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วกันทุกหน้าเอกสารภายในเว็บไซต์เดียวกัน โดยเกณฑ์ในการกำหนดรูปแบบ (Style) เอกสาร HTML ถูกเพิ่มเข้ามาครั้งแรกใน HTML 4.0 เมื่อปีพ.ศ. 2539 ในรูปแบบของ CSS level 1 Recommendations ที่กำหนดโดยองค์กร World Wide Web Consortium หรือ W3C ถือว่าเป็นภาษาหนึ่งในการตกแต่งเว็บไซต์ที่ได้รับความนิยมอย่างแพร่หลาย

### 2.8.1 โครงสร้างของ CSS



ภาพที่ 2.13 โครงสร้างของ CSS

จากภาพที่ 2.13 สามารถแบ่งโครงสร้าง CSS ได้ดังนี้

1. selector สามารถเป็น HTML Tag ต่าง ๆ เช่น <body> <p> หรือเป็น Class name หรือ ID ที่ตั้งชื่อไว้ได้
2. property คือ คุณสมบัติในการจัดรูปแบบการแสดงผลต่าง ๆ เช่น color สำหรับกำหนดสี font-size สำหรับกำหนดขนาดตัวอักษร
3. value เป็นค่าที่กำหนดให้กับ property ต่าง ๆ เช่น color: white

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.8.2 การใช้งาน CSS

การเขียน CSS สามารถเขียนได้ในที่ต่าง ๆ ใน source code ดังนี้

### 1. การเขียน CSS แบบ Inline

เป็นการเขียน CSS ลงไปในแท็ก HTML ที่ต้องการให้เกิดการแสดงผลเลย และมีผลต่อแท็กนั้น ๆ เพียงตัวเดียวเท่านั้น โดยมี Syntax ดังนี้

```
<tag style="property:value; property:value; . . .">
```

แอททริบิวต์ style ใช้ในการกำหนดค่าสไตลหรือฟอเพอร์ตีต่าง ๆ ซึ่งอยู่ในรูปของ "property:value" หากมีมากกว่าหนึ่งชุดให้ใช้เครื่องหมาย Semicolon ( ; ) คั่น ตัวอย่างเช่น

```
<h1 style="color:#0000FF"> THIS IS CSS </h1>
```

แต่การเขียน CSS แบบ Inline ไม่เป็นที่นิยมมากนัก เนื่องจากต้องเขียนแต่ละ CSS ลงไปในแท็ก HTML ดังนั้นจึงไม่สามารถนำสไตลไปประยุกต์ใช้กับแท็กอื่น ๆ ได้

### 2. การเขียน CSS แบบ Embedded

เป็นการเขียน CSS code ลงในไฟล์ HTML เช่นกัน แต่ทำการรวม CSS code รูปแบบทั้งหมดที่เขียนไว้ภายใต้แท็ก <head> โดยมี Syntax ดังนี้

```
<head>  
  <title></title>  
  <style type="text/css">  
    selector {  
      color: #0000FF;  
      font-style: italic;  
      . . .  
    }  
  </style>  
</head>
```

ภาพที่ 2.14 การเขียน CSS แบบ Embedded

การกำหนดสไตลถูกกำหนดไว้ภายในแท็ก <style> ซึ่งอยู่ภายในแท็ก <head> การกำหนดสไตลเริ่มต้นด้วยการระบุแท็กที่ต้องการ (Selector) จากนั้นจึงเป็นสไตล "property:value" ต่าง ๆ ภายในเครื่องหมายปีกกาแต่ละชุด ถูกคั่นด้วยเครื่องหมาย Semicolon ";"

การเขียนแบบ Embedded ยังคงไม่เป็นที่นิยม เพราะต้องเขียนรวมอยู่ในไฟล์ HTML เช่นเดียวกับแบบ Inline หากจะนำไปใช้กับไฟล์ HTML อื่น ๆ จะทำการได้ยาก ดังนั้นการเปลี่ยนแปลง แก้ไขจึงค่อนข้างยุ่งยาก เพราะต้องตามแก้ CSS ในทุก ๆ ไฟล์

### 3. การเขียน CSS code แบบ External

เป็นการเขียนแบบแยกไฟล์ของ CSS code ที่เขียนไว้ในไฟล์แยกต่างหาก โดยมี Syntax ดังนี้

```
HTML :
<head>
<title></title>
<link rel="stylesheet" type="text/css" ref="fileName.css" />
</head>

CSS:
selector {
property:value;
property:value;
. . .
}
```

ภาพที่ 2.15 การเขียน CSS code แบบ External

รูปแบบจะนี้แยก CSS code ไปไว้ในไฟล์แยก โดยไฟล์นั้นมีนามสกุลเป็น ".css" ส่วนโครงสร้างการเขียน Code เหมือนกับ แบบ Embedded คือเริ่มต้นด้วยการระบุแท็ก (Selector) ตามด้วยสไตล์หรือเพอร์ตี "property:value" ต่าง ๆ ภายในเครื่องหมายปีกกา และไม่จำเป็นต้องมีแท็ก <style> จากนั้นเมื่อต้องการเรียกใช้ CSS ในไฟล์ HTML ให้ใส่แท็กเพื่อลิงก์มายังไฟล์ CSS ด้วยแท็ก <link> ตัวอย่างเช่น

```
HTML :
<head>
<title></title>
<link rel="stylesheet" type="text/css" href="External.css" />
</head>

CSS (External.css):
text {
color: #0000FF;
font-style: italic;
}
tr:hover {
background: gainsboro;
}
```

ภาพที่ 2.16 ตัวอย่างการเขียน CSS code แบบ External

จากภาพที่ 2.16 ข้างต้น ไฟล์ HTML และ CSS ต้องอยู่ใน Directory เดียวกัน และสามารถลิงค์ไปยังไฟล์ CSS ได้มากกว่าหนึ่งไฟล์ด้วย ทำให้สามารถ แยกเป็นไฟล์ CSS ตามแต่ละจุดประสงค์ได้ เช่น Menu.css สำหรับจัดการกับเมนู Layout.css จัดการกับเลย์เอาต์ ซึ่งการเขียนแบบ External เป็นรูปแบบ ที่ได้รับความนิยมที่สุดเพราะสามารถนำกลับมาใช้ใหม่ได้ ทำให้การออกแบบจัดเรียง รูปลักษณะของหน้าเว็บเป็นไปได้ง่ายตายและสะดวกต่อผู้พัฒนา

## 2.9 Apache POI Library

เป็น Java API Library ที่ช่วยในการเขียนและอ่านเอกสาร ของ Microsoft Office โดย สามารถแบ่ง Component ใน POI ได้เป็นส่วน ๆ ดังนี้

- Excel (SS=HSSF+XSSF): ใช้สำหรับอ่านเขียนเอกสาร .xls และ .xlsx
- Word (HWPf+XWPF): ใช้สำหรับอ่านเขียนเอกสาร .doc และ .docx
- PowerPoint (HSLF+XSLF): ใช้สำหรับอ่านเขียนเอกสาร .ppt และ .pptx
- OpenXML4J (OOXML)
- OLE2 Filesystem (POIFS)
- OLE2 Document Props Outlook (HSMF)
- Visio (HDGF)
- TNEF (HMEF)
- Publisher (HPBF)



ภาพที่ 2.17 Apache POI

A screenshot of a Microsoft Office spreadsheet showing an employee list. The spreadsheet has columns for Employee ID, First Name, Last Name, Birth Date, Address, City, and Salary. The data is as follows:

Employee ID	First Name	Last Name	Birth Date	Address	City	Salary
2303	Buffet	Jimmy	04/04/1975	The Alamo	San Antonio	75,324.76
2304	Cartman	Eric	08/12/1973	Empire State Building	New York	59,000.03
2307	Jefferson	George	10/01/1981	Mockingbird Lane	Fargo	159,000.34

ภาพที่ 2.18 ตัวอย่างเอกสาร Microsoft Office ที่สร้างผ่าน Apache POI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.10 JQUERY Library

jQuery เป็น JavaScript Library ที่มีการรวบรวม function ของ JavaScript ต่าง ๆ ให้อยู่ในรูปแบบ Patterns Framework ที่สะดวกและง่ายต่อการใช้งาน มีความยืดหยุ่นรองรับต่อการใช้งาน Cross Browser คือไม่ว่าใช้งานบนเว็บเบราว์เซอร์ใด ใน Library ของ jQuery จะมีการเลือกใช้ function ที่เหมาะสมต่อการทำงานและแสดงผลในเว็บเบราว์เซอร์ที่กำลังรันอยู่ ซึ่งช่วยลดปัญหาการทำงานที่ผิดพลาดในฝั่งของ Client ได้ จากปัญหาก่อนนี้ นักโปรแกรมเมอร์ทั้งหลายในสมัยก่อน ๆ มักทดสอบโปรแกรม และพัฒนาบน IE แต่เนื่องจากในปัจจุบันได้มีหลายเว็บเบราว์เซอร์ได้เกิดขึ้นมากมาย เช่น Chrome, Firefox หรือ Safari และบางคำสั่งของ JavaScript ไม่ทำงานหรือไม่ support ในเว็บเบราว์เซอร์บางตัว ด้วยเหตุผลนี้ การใช้ jQuery มาเป็นทางเลือกก็สามารถช่วยแก้ปัญหาเหล่านี้เป็นได้อย่างดี ทั้งยังสะดวกต่อการใช้งาน เพราะเป็น syntax ที่เข้าใจง่าย และเขียนได้ในรูปแบบที่สั้น ๆ รองรับการทำงานทั้งใน HTML รูปแบบเดิม หรือ CSS, element, DOM element, effect การจัดการ Event ต่าง ๆ หรือแม้กระทั่งการพัฒนา Ajax ด้วย jQuery ก็สามารถทำได้ง่ายโดย Syntax เหล่านี้ยังคงทำงานอยู่ภายใต้คำสั่งของภาษา JavaScript แต่การเรียกใช้งาน Framework หรือ function ต่าง ๆ ถูกกำหนดรูปแบบโดย Patterns ที่ได้ถูกออกแบบไว้ใน Library ของ jQuery

### 2.10.1 โครงสร้างของ JQuery

jQuery การทำงานส่วนมากใน Web Form ให้ความสำคัญในการอ้างถึง (Selectors) element ในรูปแบบต่าง ๆ โดยใช้ Syntax สั้น ๆ เพียง

```
$("#txtname").val("myText");
```

นอกจากนี้ jQuery ยังมีความสามารถอ้างถึง (Selectors) element ได้หลายหลายรูปแบบมาก เช่น

1. `$(this).action()`

ใช้ Selectors อ้างถึง element ใน ปัจจุบัน

2. `$("#test").action()`

ใช้ Selectors อ้างถึง element ที่อยู่ภายใต้ `id="test"`

3. `$("p").action()`

ใช้ Selectors อ้างถึง element ที่อยู่ภายใต้ แท็กทั้งหมด `<p>` หรือ `<div>` หรือ `<span>`

4. `(".test").action()`

ใช้ Selectors อ้างถึง element ที่อยู่ภายใต้ แท็กทั้งหมด ที่เรียกใช้ css style sheet เช่น `class="test"`

## 2.11 กลุ่มเทคโนโลยี AJAX

AJAX หรือ Asynchronous JavaScript and XML เป็นการนำเทคโนโลยีต่าง ๆ มาทำงานร่วมกัน เพื่อให้การใช้งานเว็บเพจประหยัดและสะดวกขึ้น โดยปกติเมื่อผู้ใช้เข้าเว็บหรือร้องขอข้อมูลจากทาง Sever ต้องรอ Sever ประมวลผลเสร็จก่อน จึงจะส่งกลับมาแสดงผลบนเว็บเพจของผู้ใช้ จึงเกิดการ Refresh ใหม่ทุกครั้งที่ยังร้องขอข้อมูล จึงมีการนำ AJAX มาช่วยในการทำงานให้รวดเร็วขึ้นโดย AJAX สามารถลดการ Refresh ทำให้การแสดงผลนุ่มนวลขึ้นโดยหัวใจสำคัญในการทำงานของ AJAX คือ XMLHttpRequest Object ซึ่งทำหน้าที่ขอข้อมูลทางฝั่ง Server โดยข้อมูลที่ขอนั้นเป็นข้อมูลเล็ก ๆ และใช้ JavaScript นำมาแสดงผลบนหน้าจอเดิมโดยไม่ต้อง Refresh หน้าจอใหม่

หลักการทำงานของ AJAX คือ JavaScript ชุดหนึ่งที่ทำหน้าที่ร้องขอ (Request) ข้อมูลจากทาง Server ผ่านทาง XMLHttpRequest และทำการนำค่าที่ได้นั้นออกมาแสดงที่หน้าเบราว์เซอร์ โดยข้อมูลที่รับเข้ามานั้นเป็น XML หรือข้อความ (text)

### 2.11.1 ส่วนประกอบของ AJAX

1. JavaScript
2. Cascading Style Sheet (CSS)
3. Document Object Model (DOM)
4. XMLHttpRequest Object (XHR Object)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.11.2 ประวัติของ AJAX

AJAX ถูกบัญญัติศัพท์ขึ้นโดย Jese James Garrett ขึ้นเมื่อปี พ.ศ. 2548 โดย AJAX ไม่ได้เป็นเทคโนโลยีหรือภาษาโปรแกรมชนิดใหม่ แต่เป็นการรวมกลุ่มของเทคโนโลยีที่มีใช้อยู่แล้วอย่าง HTML CSS ECMAScript XMLHttpRequest และ XML โดยวิวัฒนาการของ AJAX เริ่มต้นเมื่อปี ค.ศ. 2002 ไมโครซอฟท์ได้ทำการคิดค้น XMLHttpRequest ขึ้นมาเพื่อเป็นทางเลือกในการเขียนโปรแกรมบนเว็บเพจเพื่อใช้ติดต่อกับเว็บเซิร์ฟเวอร์ ในการแลกเปลี่ยนข้อมูล ซึ่งในขณะนั้นมีแต่เพียง Internet Explorer เท่านั้นที่มีความสามารถนี้ ต่อมาเว็บเบราว์เซอร์อื่น ๆ เช่นจาก Mozilla Firefox ได้นำแนวคิดของ XMLHttpRequest ไปใส่ในเบราว์เซอร์ด้วย จึงเริ่มทำให้มีการใช้อย่างกว้างขวางขึ้น จนปัจจุบันได้กลายเป็นมาตรฐานที่ทุกเว็บเบราว์เซอร์ต้องมี

ในช่วงแรก ไมโครซอฟท์เป็นผู้ที่นำ XMLHttpRequest โดยใช้ใน Outlook Web Access ที่มาพร้อมกับ Microsoft Exchange Server 2000 ต่อมาเว็บไซต์อย่างกูเกิล (Google) ได้เปิดบริการใหม่ชื่อ จีเมล (Gmail) ซึ่งใช้ XMLHttpRequest เป็นหัวใจหลัก ในการดึงข้อมูลจากเว็บเซิร์ฟเวอร์ จึงทำให้แนวคิดและเทคนิคการพัฒนาเว็บแอปพลิเคชันด้วย AJAX เริ่มเป็นที่รู้จักกันกว้างขวางมากขึ้น

## 2.12 ระบบฐานข้อมูล (Database System)

คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบมีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้ต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ แฟ้มข้อมูลในระบบฐานข้อมูลประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูลเกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (data base management system) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่าย สะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

โดยส่วนประกอบมีแฟ้มข้อมูล (File) ระเบียบ (Record) และเขตข้อมูล (Field) และถูกจัดการด้วยระบบเดียวกัน โปรแกรมคอมพิวเตอร์เข้าไปดึงข้อมูลที่ต้องการได้อย่างรวดเร็ว ซึ่งอาจเปรียบฐานข้อมูลเสมือนเป็น electronic filing system

### 2.12.1 วิวัฒนาการของ Database

Database ถูกพัฒนาขึ้นมาตั้งแต่ปี 1960 เริ่มต้นจาก hierarchical และ network databases จนมาถึงปี 1980 มีการนำเอา object-oriented-databases (OODBMS) มาใช้งาน ซึ่งเป็นพื้นฐานของระบบ relation database ที่ใช้กันอยู่ในปัจจุบันนี้

ในอีกมุมหนึ่ง ซึ่งสามารถจัดแบ่งประเภทของ database ตามรูปแบบของ ชนิดข้อมูลได้ เช่น ตัวเลข,ตัวอักษร หรือ รูปภาพ บางครั้งก็อาจแบ่งตามความนิยมของ relational database เช่น distributed database, cloud database หรือ NoSQL database

### 2.12.2 การเข้าถึง Database

ปัจจุบันมีด้วยกัน 2 แบบคือ

#### 1. Database management system (DBMS)

เป็นระบบการจัดการฐานข้อมูล หรือซอฟต์แวร์ที่ดูแลจัดการเกี่ยวกับฐานข้อมูล โดยอำนวยความสะดวกให้แก่ผู้ใช้ทั้งในด้านการสร้าง การปรับปรุงแก้ไข การเข้าถึงข้อมูล และการจัดการเกี่ยวกับระบบแฟ้มข้อมูลทางกายภาพภายในฐานข้อมูลซึ่งต่างไปจากระบบแฟ้มข้อมูลคือหน้าที่เหล่านี้เป็นของโปรแกรมเมอร์ในการติดต่อฐานข้อมูล ไม่ว่าจะด้วยการใช้คำสั่งในกลุ่ม DML หรือ DDL หรือด้วยโปรแกรมต่าง ๆ ทุกคำสั่งที่ใช้กระทำกับฐานข้อมูล ถูกโปรแกรม DBMS นำมาแปล (Compile) เป็นการกระทำต่าง ๆ ภายใต้อำนาจคำสั่งนั้น ๆ เพื่อนำไปกระทำกับตัวข้อมูลใน ฐานข้อมูลต่อไป

#### 2. Relational database management system (RDBMS)

ระบบฐานข้อมูลเชิงสัมพันธ์ เป็นการเก็บข้อมูลในรูปของตาราง (table) ในแต่ละตารางแบ่งออกเป็นแถว ๆ และในแต่ละแถวแบ่งเป็นคอลัมน์ (Column) ซึ่งในการเชื่อมโยงกันระหว่างข้อมูลในตารางต่าง ๆ เชื่อมโยงโดยใช้การอ้างอิงจากข้อมูลในคอลัมน์ที่กำหนดไว้ ถูกพัฒนาขึ้นในปี 1970 เพื่อเข้าถึง ฐานข้อมูลแบบ relational และยังคงได้รับความนิยม จนถึงปัจจุบัน

### 2.12.3 ข้อดีของการใช้ Database

1. หลีกเลี่ยงความขัดแย้งของข้อมูล การจัดเก็บข้อมูลแบบแฟ้มข้อมูล โดยข้อมูลเรื่องเดียวกันอาจมีอยู่หลายแฟ้มข้อมูล ซึ่งก่อให้เกิดความขัดแย้งของข้อมูลได้ (Inconsistency)
2. สามารถใช้ข้อมูลร่วมกันได้ ฐานข้อมูลเป็นการจัดเก็บข้อมูลรวมไว้ด้วยกัน เมื่อผู้ใช้ต้องการข้อมูลจากฐานข้อมูล ซึ่งเป็นข้อมูลที่มาจากแฟ้มข้อมูลที่แตกต่างกันให้ทำได้ง่าย
3. สามารถลดความซ้ำซ้อนของข้อมูล การจัดเก็บข้อมูลในลักษณะแฟ้มข้อมูล อาจทำให้ข้อมูลประเภทเดียวกัน ถูกเก็บไว้หลาย ๆ แห่ง ทำให้เกิดความซ้ำซ้อน (Redundancy) การนำข้อมูลมารวมเก็บไว้ในฐานข้อมูลช่วยลดปัญหาความซ้ำซ้อนได้
4. รักษาความถูกต้อง ฐานข้อมูลบางครั้งอาจมีข้อผิดพลาดขึ้น เช่น การป้อนข้อมูลผิด ซึ่งระบบการจัดการฐานข้อมูลสามารถระบุกฎเกณฑ์เพื่อควบคุมความผิดพลาดที่อาจเกิดขึ้นได้
5. สามารถกำหนดความเป็นมาตรฐานเดียวกันได้ เพราะในระบบฐานข้อมูล จะมีบุคคลที่บริหารฐานข้อมูลกำหนดมาตรฐานต่าง ๆ ในการจัดเก็บข้อมูลลักษณะเดียวกัน
6. สามารถกำหนดระบบความปลอดภัยของข้อมูลได้ ผู้บริหารระบบฐานข้อมูล สามารถกำหนดการเรียกข้อมูลผู้ใช้แต่ละคนให้แตกต่างกันตามหน้าที่ความรับผิดชอบได้ง่าย
7. ความเป็นอิสระของข้อมูลและโปรแกรม โปรแกรมที่ใช้ในแต่ละแฟ้มข้อมูล จะมีความสัมพันธ์กับแฟ้มข้อมูลโดยตรงถ้าหากมีการแก้ไขเปลี่ยนแปลงโครงสร้างข้อมูล ก็ทำการแก้ไขโปรแกรมนั้น ๆ

### 2.12.4 ข้อเสียของการใช้ Database

1. มีต้นทุนสูง ระบบฐานข้อมูลก่อให้เกิดต้นทุนสูง เช่น ซอฟต์แวร์ที่ใช้ในการจัดการระบบฐานข้อมูล บุคลากร ต้นทุนในการปฏิบัติงาน และ ฮาร์ดแวร์ เป็นต้น
2. การเริ่มใช้ระบบฐานข้อมูลอาจก่อให้เกิดความซับซ้อนได้ เช่น การจัดเก็บข้อมูล การออกแบบฐานข้อมูล การเขียนโปรแกรม เป็นต้น
3. การเสี่ยงต่อการหยุดชะงักของระบบ เนื่องจากข้อมูลถูกจัดเก็บไว้ในลักษณะเป็นศูนย์รวม (Centralized Database System) ความล้มเหลวของการทำงานบางส่วน ในระบบอาจทำให้ระบบฐานข้อมูลทั้งระบบหยุดชะงักได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.13 Oracle Database

Oracle คือ โปรแกรมจัดการฐานข้อมูล ผลิตโดยบริษัทออราเคิล ซึ่งเป็นโปรแกรมจัดการฐานข้อมูลเชิงสัมพันธ์ หรือ DBMS (Relational Database Management System) ตัวโปรแกรมนี้ทำหน้าที่เป็นตัวกลางคอยติดต่อ ประสาน ระหว่างผู้ใช้และฐานข้อมูล ทำให้ผู้ใช้งานสามารถใช้งานฐานข้อมูลได้สะดวกขึ้น เช่นการค้นหาข้อมูลต่าง ๆ ภายในฐานข้อมูลที่ง่ายและสะดวก โดยผู้ใช้ไม่จำเป็นต้องทราบถึงโครงสร้างภายในของฐานข้อมูลก็สามารถเข้าใช้ฐานข้อมูลนั้นได้

### 2.13.1 ข้อดีของ Oracle

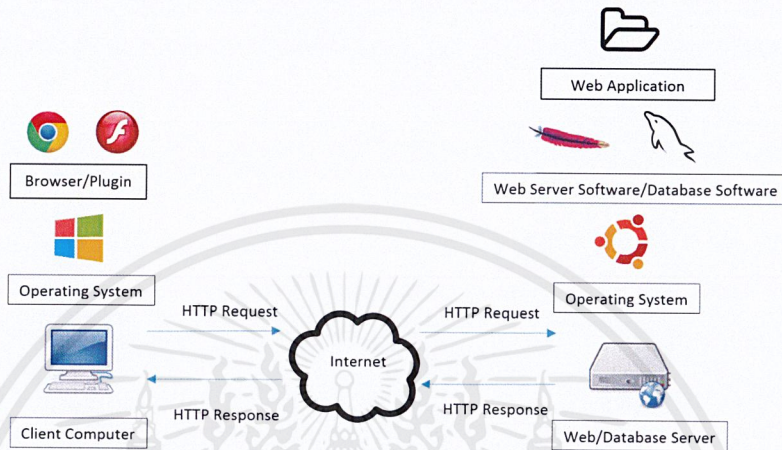
1) เทคโนโลยี Rollback Segment ถูกนำมาใช้ในโปรแกรม Oracle ประโยชน์ Rollback Segment คือ สามารถจัดการกับข้อมูลในกรณีที่เกิดการล้มเหลวของระบบ หรือ ภาวะระบบไม่สามารถให้บริการได้ ด้วยเทคโนโลยี Rollback Segment จะจัดการ Instance Recovery ข้อมูลไม่ให้เกิดความเสียหายอันเนื่องมาจากการล้มเหลวของระบบได้อย่างดีมาก

2) Oracle ยังมีส่วนที่เรียกว่า Timestamp ทำงานเกี่ยวข้องกับ Concurrency Control เป็นส่วนที่จัดการการทำงานกับหลาย ๆ Transaction ในเวลาเดียวกัน โดยทุก ๆ Transaction มีการระบุ Timestamp เป็นตัวกำหนดเวลาเริ่มต้นของการประมวลผล (Process) ซึ่งช่วยในการขจัดปัญหาหลักของ Concurrency Problems

3) Oracle ใช้ได้กับฐานข้อมูลกว่า 80 แพลตฟอร์ม ซึ่งครอบคลุมเกือบทุกแพลตฟอร์มที่มีอยู่ในปัจจุบัน เริ่มตั้งแต่เครื่องคอมพิวเตอร์บนเมนเฟรม มินิคอมพิวเตอร์ และพีซี บนระบบปฏิบัติการตั้งแต่ Window 9x, NT, Window CE, UNIX, SOLARIS, LINUX โดยที่ในทุกระบบมีโครงสร้างการเหมือนกัน ๆ หมด คำสั่งที่ใช้ก็เป็นแบบเดียวกัน สามารถทำงานร่วมกันได้ สามารถนำข้อมูลจากพอร์ตหนึ่งไปพอร์ตอื่นได้อย่างไม่มีปัญหา

## 2.14 Web Application

เว็บแอปพลิเคชัน เป็นแอปพลิเคชันที่ถูกเขียนขึ้นมาเพื่อเป็นเบราว์เซอร์ (Browser) สำหรับการใช้งานเว็บเพจ (Webpage) ต่าง ๆ ซึ่งถูกปรับแต่งให้แสดงผลแต่ส่วนที่จำเป็น เพื่อเป็นการลดทรัพยากรในการประมวลผล ของตัวเครื่อง Client ทำให้โหลดหน้าเว็บไซต์ได้เร็วขึ้น อีกทั้งยังสามารถใช้งานผ่านอินเทอร์เน็ต (Internet) และอินทราเน็ต (Intranet) ในความเร็วต่ำได้



ภาพที่ 2.19 โครงสร้างระบบ Web application

จากภาพที่ 2.19 สามารถแบ่งส่วนประกอบออกได้เป็น 2 ส่วน ดังนี้

### 2.14.1 ส่วนประกอบฝั่งผู้ใช้งาน (Client-side Technology)

#### 1. เว็บเบราว์เซอร์ (Web Browser)

เว็บเบราว์เซอร์ เป็นซอฟต์แวร์ที่ผู้ใช้งานใช้ในการเข้าถึงเว็บแอปพลิเคชัน โดยที่เมื่อเริ่มต้น ผู้ใช้งานทำการใส่ URL หรือว่าชื่อของเว็บไซต์ที่ต้องการเข้าใช้งาน เช่น <https://www.google.com> เมื่อเบราว์เซอร์ได้รับชื่อของเว็บไซต์ ก็จะทำการแปลงจากชื่อของเว็บไซต์เป็น IP address ผ่านทาง DNS หลังจากนั้น เว็บเบราว์เซอร์ทำการสร้าง HTTP request เพื่อส่งคำร้องไปยังเว็บเซิร์ฟเวอร์ผ่านทางเครือข่ายอินเทอร์เน็ต เมื่อได้รับ HTTP response จากเว็บเซิร์ฟเวอร์ เว็บเบราว์เซอร์ ทำหน้าที่ในการอ่าน และแปลง HTTP response ให้เป็นข้อมูลที่ใช้ในการแสดงผลให้กับผู้ใช้งาน ดังนั้นหน้าที่ของเว็บเบราว์เซอร์ ประกอบไปด้วย การรับข้อมูลและคำสั่งจากผู้ใช้งาน, แปลงคำสั่งของผู้ใช้งานให้เป็น HTTP request เพื่อส่งไปให้กับเว็บเซิร์ฟเวอร์, ประมวลผล HTTP response และเรียกใช้ Plugin, แปลงภาษา HTML, CSS, JavaScript ให้ข้อมูลสำหรับแสดงผลให้กับผู้ใช้งาน

## 2. ส่วนต่อความสามารถเว็บและเบราว์เซอร์ (Web Plugin และ Browser Add-on/Extension)

- Web Plugin (ส่วนต่อความสามารถเว็บ) คือโปรแกรมที่ถูกเขียนให้ทำงานร่วมกับเว็บเบราว์เซอร์ Web Plugin ที่เป็นที่ยูจกันดีเช่น Adobe Flash, PDF reader, Silverlight, Java Applet, และอื่น ๆ ซึ่ง Web Plugin เหล่านี้ จะถูกเบราว์เซอร์เรียกใช้ก็ต่อเมื่อเว็บไซต์ที่เข้าใช้งานมีเนื้อหาที่ต้องแสดงผลโดย Plugin เช่น Adobe Flash Plugin จะถูกเรียกใช้ โดยเบราว์เซอร์ก็ต่อเมื่อเจอเนื้อหาที่ต้องใช้ Flash Player ในการแสดงผล

- Browser Add-one/Extension (ส่วนเพิ่มความสามารถเบราว์เซอร์) เป็นโปรแกรมที่ใช้ในการเพิ่มความสามารถให้กับเบราว์เซอร์ เช่น ส่วนเพิ่มความสามารถที่ช่วยในการจัดการไฟล์ดาวน์โหลด ส่วนเพิ่มความสามารถที่ช่วยในการดาวน์โหลดไฟล์วิดีโอ เป็นต้น ซึ่งส่วนเพิ่มความสามารถเบราว์เซอร์เหล่านี้ เน้นเพิ่มความสามารถให้กับเบราว์เซอร์ มากกว่าการประมวลผลเนื้อหาเว็บไซต์

- Web Plugin เป็นส่วนที่ถูกเรียกใช้โดยเว็บเบราว์เซอร์เพื่อส่งต่อเนื้อหาไปให้กับโปรแกรมภายนอกเช่น Adobe Flash/ PDF reader ในการประมวลผล

- Browser Extension เน้นการเพิ่มความสร้างมาให้กับเบราว์เซอร์ แต่ไม่ได้ประมวลผลหรือแสดงเนื้อหาในเว็บไซต์

## 3. ระบบปฏิบัติการ (Operating System)

ระบบปฏิบัติการทำหน้าที่ในการจัดการทรัพยากรของเครื่องคอมพิวเตอร์ ทำหน้าที่ในการรับ HTTP request จากเบราว์เซอร์และส่งต่อไปให้ กับอินเทอร์เน็ต DNS ในระบบปฏิบัติการทำหน้าที่ในการแปลง URL ให้เป็น IP Address เพื่อค้นหาเครื่องเซิร์ฟเวอร์ สร้างการเชื่อมต่อ (TCP connection) ระหว่างเครื่องผู้ใช้งานและเครื่องเซิร์ฟเวอร์ ดังนั้นการทำงานของระบบปฏิบัติการเป็นสิ่งที่ผู้ใช้งานมองไม่เห็นแต่ก็มีความสำคัญมาก

## 2.14.2 ส่วนประกอบฝั่ง Server (Server-side Technology)

เว็บเซิร์ฟเวอร์ที่ทำหน้าที่เป็นผู้ให้บริการแก่ผู้ใช้งานเว็บไซต์ประกอบด้วยเทคโนโลยีและซอฟต์แวร์หลายส่วนทำงานร่วมกัน โดยซอฟต์แวร์หลักที่ใช้ในการให้บริการของเว็บเซิร์ฟเวอร์ประกอบไปด้วย 4 ส่วนประกอบหลัก

### 1. เว็บแอปพลิเคชัน (Web Application)

เว็บแอปพลิเคชัน ถือเป็นหัวใจหลักของเว็บไซต์ เนื่องจากทำหน้าที่ติดต่อกับผู้ใช้งาน รับและแสดงข้อมูล ประมวลผลข้อมูล จัดการข้อมูลในฐานข้อมูล และอื่น ๆ เรียกได้ว่าเว็บแอปพลิเคชันเป็น ซอฟต์แวร์ที่ให้บริการผู้ใช้งานทั่วโลกผ่านอินเทอร์เน็ต

- Front-End Web Technology เป็น ส่วนของเทคโนโลยีที่ใช้ ในการสร้างส่วนติดต่อกับผู้ใช้งาน ในการสร้างเว็บแอปพลิเคชัน Front-End Technology ที่เป็นที่แพร่หลายได้แก่ HTML, CSS, และ JavaScript ซึ่งภาษาคอมพิวเตอร์เหล่านี้ ถูกใช้อย่างแพร่หลายในการสร้างส่วนติดต่อกับผู้ใช้งานของเว็บแอปพลิเคชัน ความหลากหลายของ Front-End Web Technology ถูกจำกัดด้วยมาตรฐานกลางที่ออกโดยองค์กร ที่ไม่แสดงหากำไรอย่าง World Wide Web Consortium (W3C) ซึ่งเป็นผู้กำหนดมาตรฐาน HTML, CSS และ JavaScript เพื่อให้ผู้พัฒนาเบราว์เซอร์ให้แสดงผลข้อมูลในรูปแบบเดียวกัน เพื่อความสะดวกแก่ผู้ใช้งานและนักพัฒนา ซึ่งเบราว์เซอร์ในปัจจุบันต่างรองรับการประมวลผลของ HTML, CSS และ JavaScript โดยสมบูรณ์ แม้ว่าจะมีความแตกต่างในการแสดงผลไปบ้าง ในบางเบราว์เซอร์
- Back-End Web Technology หมายถึง ส่วนของเทคโนโลยีที่ เป็นส่วนประมวลผลตรรกะและการทำงานของเว็บแอปพลิเคชัน ไม่ว่าจะเป็น การตรวจสอบสิทธิ์การเข้าใช้ การเรียกดูและจัดเก็บข้อมูล การทำงานของเว็บแอปพลิเคชันในส่วนของ Back-End เริ่มหลังจากเว็บแอปพลิเคชันได้รับ HTTP request มาจากผู้ใช้งาน ทำการประมวลผล และส่งข้อมูลกลับไปให้ กับผู้ใช้งาน เทคโนโลยีที่ใช้ในการพัฒนา Back-End ของเว็บแอปพลิเคชันมีความหลากหลายกว่า Front-End เนื่องจากไม่มีข้อจำกัด ด้านมาตรฐานกลาง

## - 2. เว็บเซิร์ฟเวอร์ซอฟต์แวร์ (Web Server Software)

Web server software เป็นโปรแกรมที่ทำงานอยู่บน web server ซึ่งหน้าที่หลักของ web server software คือการประมวลผล HTTP request ที่ได้รับมาและตอบกลับด้วย HTTP response ให้กับผู้ใช้งาน ปัจจุบันมี web server software หลายตัวที่ถูกใช้งานอย่างแพร่หลายเช่น Apache HTTP server, Internet Information Service (IIS) และ Nginx ยังมี web server software ตัวอื่นอีกมากในท้องตลาดที่ไม่ได้กล่าวถึงในที่นี้ อย่างไรก็ตาม web server software ที่ได้รับความนิยมอย่างแพร่หลายมากที่สุดในปัจจุบันคือ Apache HTTP server และผู้ใช้งานมักใช้คู่กับ PHP (ตัวแปลภาษา PHP) และ MySQL (ฐานข้อมูล)

## 3. ระบบปฏิบัติการ (Operating System)

ระบบปฏิบัติการบนฝั่งของ Server มีหน้าที่ในการจัดการกับทรัพยากรของเครื่อง Server เช่น CPU memory และ bandwidth เป็นต้น เนื่องจาก web application เป็นบริการที่เปิดให้ผู้ใช้งานเข้าถึงได้ตลอดเวลา ดังนั้นระบบปฏิบัติการบน Server จึงต้องมีความเสถียรและสามารถจัดการกับทรัพยากรของเครื่องได้อย่างมีประสิทธิภาพ

## 4. เว็บเซิร์ฟเวอร์ (Web Server)

คือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเครื่องบริการเว็บไซต์ (Website) แก่ผู้ร้องขอ (Request) ด้วยโปรแกรมประเภทเว็บเบราว์เซอร์ (Web Browser) ที่ร้องขอข้อมูลผ่านโพรโทคอลเฮกซ์ทีทีพี (HTTP = Hyper Text Transfer Protocol) เครื่องบริการส่งข้อมูลให้ผู้ร้องขอในรูปของข้อความ ภาพ เสียง หรือสื่อผสม เครื่องบริการเว็บเพจมักเปิดบริการพอร์ต 80 (HTTP Port) ให้ผู้ร้องขอได้เชื่อมต่อและนำข้อมูลไปใช้ เช่น Internet Explorer หรือ Firefox การเชื่อมต่อเริ่มด้วยการระบุ Web Address หรือ (URL) Uniform Resource Locator เช่น <http://www.google.com> หรือ <http://www.thaiall.com> เป็นต้น โปรแกรมที่นิยมใช้เป็นเครื่องบริการเว็บ คือ Apache Web Server หรือ Microsoft IIS ส่วนบริการที่นิยมติดตั้งเพิ่ม เพื่อเสริมความสามารถของเครื่องบริการ เช่น ตัวแปลภาษาสคริปต์ ระบบฐานข้อมูล ระบบจัดการผู้ใช้ และระบบจัดการเนื้อหา เป็นต้น

## 2.15 WildFly Application Server

Application Server คือ Server ที่รันโปรแกรมประยุกต์ได้ด้วย โดยการทำงานสอดคล้องกับ Client (ไคลเอนต์) เช่น Mail Server (รัน MS Exchange Server) Proxy Server (รัน Proxy Server) หรือเว็บเซิร์ฟเวอร์ (รันโปรแกรมเว็บเซิร์ฟเวอร์ เช่น Xitami, Apache) ใช้ในการประมวลผลคือมาประมวลผลที่ Client ทำให้ลดภาระการประมวลผลที่ Server เหมาะกับงานที่ไม่ต้องใช้ประสิทธิภาพ Server สูงมากนัก

WildFly มีคุณสมบัติที่สามารถตอบสนองความต้องการ ไม่ว่าจะเป็นทางด้าน เว็บเซอร์วิส (Web Service) การออกแบบที่เป็น Enterprise มีกลไกการทำงานที่ไม่สลับซับซ้อน แต่ได้ประสิทธิภาพสูง มีเซอร์วิสต่าง ๆ รองรับการใช้งาน

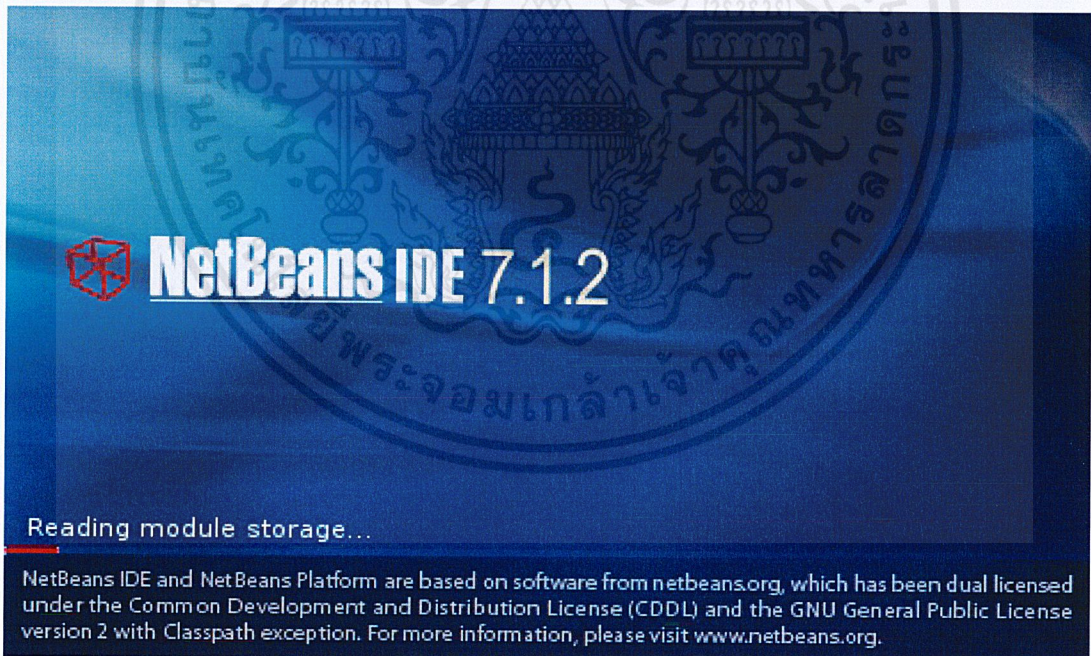


ภาพที่ 2.20 WildFly Application Server

## 2.16 โปรแกรม NetBeans IDE

เน็ทบีนส์ (NetBeans) เป็นเครื่องมือที่ได้รับความนิยมมากที่สุดใน การเขียนเว็บไซต์ และเว็บแอปพลิเคชันด้วยภาษา Java เนื่องจากเน็ทบีนส์มี Editor ที่รองรับการเขียน HTML และ JavaScript ได้อย่างดีเยี่ยม อีกทั้งยังมีอินเทอร์เฟซอื่น ๆ ที่รองรับการเขียน JSP ได้อย่างครบครัน และมีรูปแบบที่รองรับการเขียนด้วย Framework ได้ทันที จึงทำให้มีประโยชน์อย่างมาก เมื่อใช้ในการเขียน JSP

ปัจจุบัน NetBeans IDE ได้รับความนิยมมากยิ่งขึ้น และได้รับการพัฒนาให้มีความสามารถสูงยิ่งขึ้นเรื่อย ๆ นอกจากใช้ในการพัฒนาแอปพลิเคชันด้วยภาษา Java แล้ว ยังสามารถพัฒนาแอปพลิเคชันอื่น ๆ ได้อีกหลากหลายโดยติดตั้งโปรแกรมเสริมได้จากเว็บไซต์ หรือผ่าน Update Center ของเน็ทบีนส์ เช่น ภาษาซี/ซีพลัสพลัส (C/C++), Ruby, UML, SOA, เว็บแอปพลิเคชัน, Java EE, Mobility (Java ME), Java FX, JavaScript, PHP เป็นต้น ในเวอร์ชัน 6.0 เป็นต้นไป มีการรวมโปรแกรมเสริมต่าง ๆ ที่สำคัญเข้าในตัวติดตั้งของ NetBeans โดยสามารถเลือกติดตั้งได้ภายหลัง



ภาพที่ 2.21 โปรแกรม NetBeans IDE 7.1.2

## บทที่ 3

### วิธีดำเนินการวิจัย

#### 3.1 ภาพรวมแนวคิดของงานวิจัย

งานวิจัยนี้เป็นการวิจัยเชิงประติมากรรม เพื่อนำเสนอการพัฒนากระบวนการจัดเก็บงานวิจัยและพัฒนาหลังจากเสร็จสิ้นกระบวนการผลิตบนระบบเว็บแอปพลิเคชัน เพื่ออำนวยความสะดวกสบายแก่ผู้ใช้ มีโครงสร้างและรูปแบบที่ดีขึ้นกว่าระบบเดิม และลดระยะขั้นตอนและเวลาในการทำงานของผู้ใช้งานให้น้อยที่สุด

#### 3.2 ขั้นตอนการดำเนินงาน

##### 3.2.1 ขั้นตอนการรวบรวมข้อมูลและออกแบบโครงสร้างระบบที่ใช้

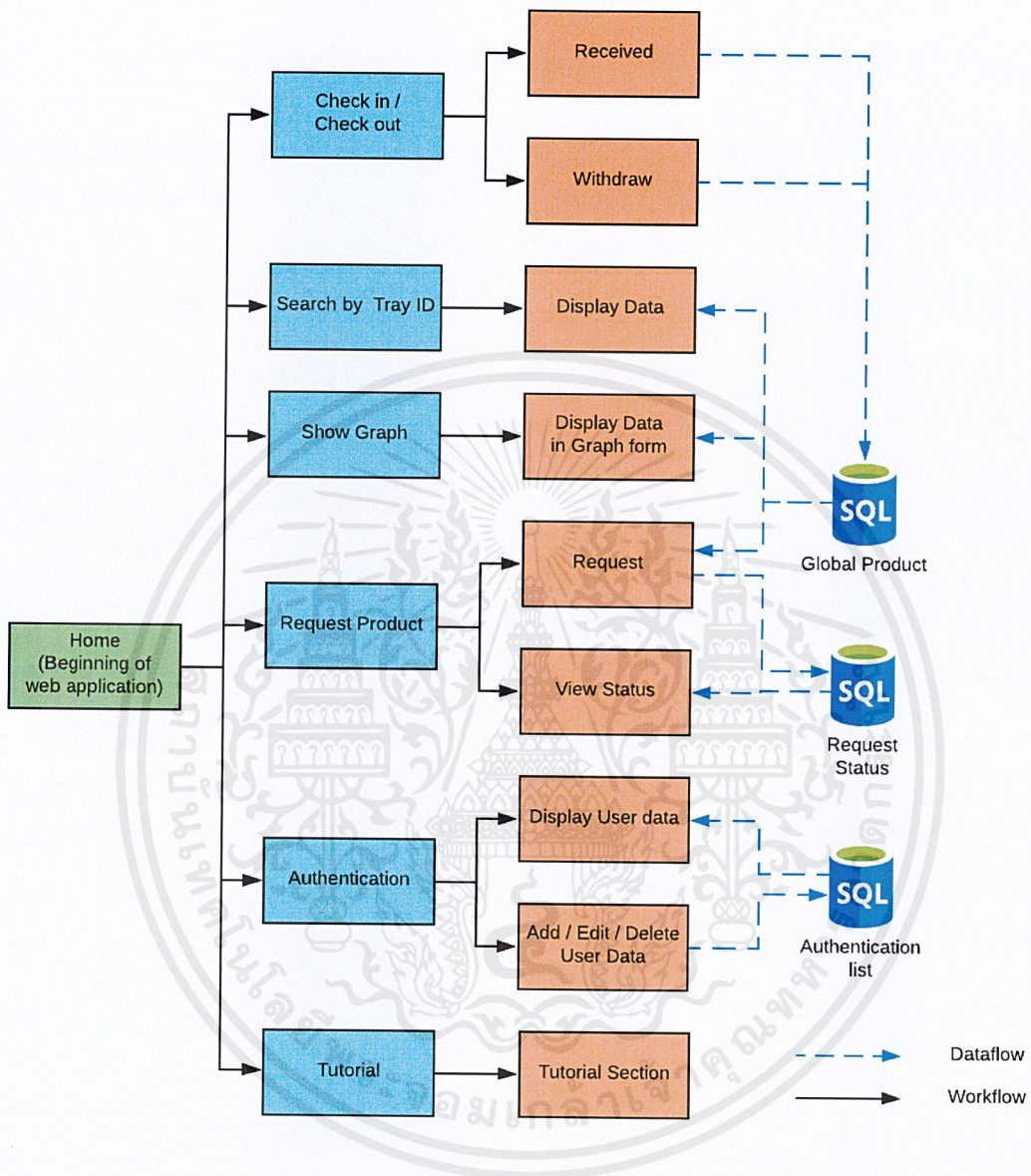
1. รวบรวมข้อมูล ข้อกำหนด และความต้องการต่าง ๆ จากผู้ใช้งานที่เกี่ยวข้องและพิจารณาจัดสรรเครื่องมือที่วางแผนเพื่อใช้ในระบบ

เครื่องมือ	สาเหตุที่เลือกใช้
 ภาพที่ 3.1 WildFly	มีความยืดหยุ่นในการเปลี่ยนแปลงระบบ เนื่องจากแอปพลิเคชัน Java EE โดยมีภาษา Java เป็นหลักทำให้สามารถส่งต่อแก้ไขและปรับใช้ได้ง่าย เพราะภาษาโปรแกรมที่ใช้อย่างกว้างขวาง
 ภาพที่ 3.2 Oracle Database	สามารถสร้าง Procedure (PL/SQL) ที่ช่วยในการดึงข้อมูลที่มีความซับซ้อนได้สะดวกขึ้น อีกทั้ง Oracle Database สามารถรองรับ Platform ที่หลากหลายไม่ว่าจะเป็น Windows, Linux, Solaris, HP-UX, OS X, z/OS และ AIX

ตารางที่ 3.1 รายการเครื่องมือที่ใช้งานในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หลังจากทำการรวบรวม Requirement ต่าง ๆ ที่ได้รับจากผู้ใช้งาน จึงนำข้อมูลทั้งหมดมาเขียนกระบวนการทำงานของเว็บแอปพลิเคชันโดยสังเขป



ภาพที่ 3.3 กระบวนการการทำงานของ web application

### 3.2.2 ขั้นตอนการออกแบบระบบการทำงานและสร้างเว็บแอปพลิเคชัน

#### 1. ทำการวางแผนว่าฟังก์ชันแต่ละฟังก์ชันมีความสามารถในการทำงานอะไรได้บ้าง

- ฟังก์ชัน Check in และ Check out

สำหรับการนำสินค้า งานวิจัยเข้าและออกจากระบบ

- ฟังก์ชัน Search by Tray ID

สำหรับการค้นหา ชิ้นงานวิจัยต่าง ๆ ในฐานข้อมูล เพื่อนำไปใช้ในการทำงานในการวิจัยต่อไป

- ฟังก์ชัน Request Product

สำหรับการขอเบิกสินค้า งานวิจัยออกจากระบบ

- ฟังก์ชัน Add user

สำหรับเพิ่ม แก้ไข ลบ รายชื่อและกำหนดสิทธิ์การเข้าถึงของผู้ใช้งานแต่ละคน

- ฟังก์ชัน Tutorial

แสดงแนวทางและแนะนำวิธีการใช้งาน ฟังก์ชันต่าง ๆ บนเว็บแอปพลิเคชันสำหรับผู้ใช้งาน เนื่องจากเป็น โปรแกรมใหม่

- ฟังก์ชัน Add user

สำหรับเพิ่ม แก้ไข ลบ รายชื่อและกำหนดสิทธิ์การเข้าถึงของผู้ใช้งานแต่ละคน

2. ทำการสร้าง Service ของระบบ Check In / Check Out โดยการรับการรับข้อมูล จาก ภาคติดต่อประสานงานกับผู้ใช้ที่ทำการใส่หมายเลขสินค้าส่งเข้ามา ระบบจะดึงข้อมูลจากฐานข้อมูล ทำการจำแนกประเภทข้อมูลเข้าก่อนว่าทำการรับสินค้าเข้ามาหรือรับสินค้าออก โดยแบ่งเป็นกรณีดังนี้

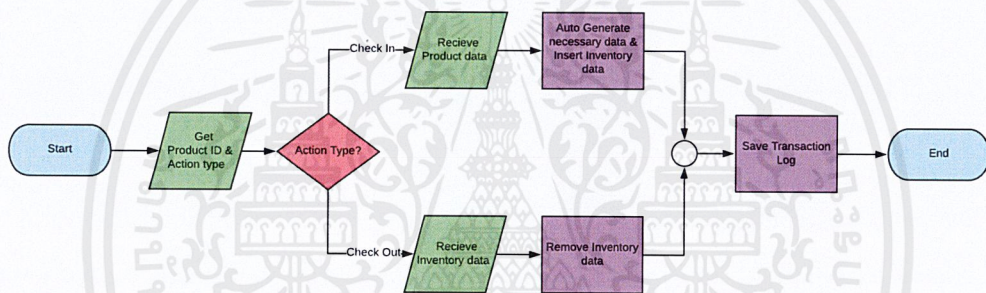
- กรณีรับสินค้าเข้า (Check In)

ทำการตรวจสอบก่อนว่ามีข้อมูลสินค้าในระบบหรือไม่ ถ้ามี จะทำการยกเลิกกระบวนการและแจ้งเตือน แต่ถ้าไม่มีระบบจะดำเนินการกรอกข้อมูลเข้าฐานข้อมูลคลังอัตโนมัติ

- กรณีรับสินค้าออก (Check Out)

ทำการตรวจสอบก่อนว่ามีข้อมูลสินค้าในระบบหรือไม่ ถ้าไม่มีจะทำการยกเลิกกระบวนการและแจ้งเตือน แต่ถ้ามีสินค้าอยู่ในระบบ จะดำเนินการดึงข้อมูลออกจากฐานข้อมูลคลังสินค้า

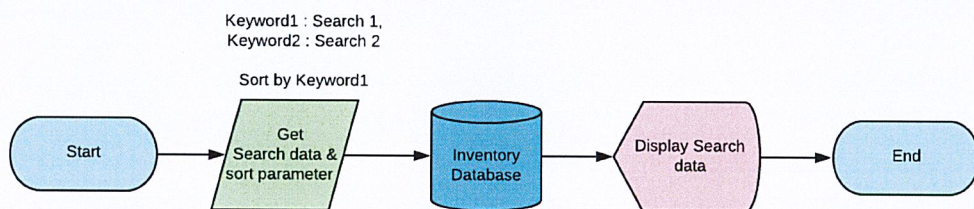
ระบบทำการบันทึกประวัติหลังกระบวนการทำงานทุกอย่างเพื่อตรวจสอบเกี่ยวกับเปลี่ยนแปลงของสินค้าในภายหลัง



ภาพที่ 3.4 Workflow การทำงานของระบบ Check In / Check Out

3. หลังจากนั้นจึงทำการสร้าง Service ในการดูชิ้นงาน Search by Tray ID โดยฟังก์ชันนี้มีหน้าที่เป็นหน้าที่แสดงรายละเอียด ของชิ้นงานวิจัยทั้งหมด เพื่อให้ผู้ที่ต้องการใช้ข้อมูลสามารถ เข้ามาค้นหารายละเอียดต่าง ๆ ที่ต้องการของชิ้นงานวิจัยนั้น ๆ ได้โดยง่าย

โดยหลักการคือเริ่มทำการรับค่าข้อมูลเป็นลักษณะเป็น “หัวข้อ : คำที่ค้นหา” ทำให้รับข้อมูลได้หลากหลายและค้นหาหลายตัวแปรพร้อมกันได้



ภาพที่ 3.5 Workflow การทำงานของระบบ Search by Tray ID

ซึ่งการแสดงผลข้อมูลจำเป็นต้องคำนึงถึงความรวดเร็วในการตอบสนอง ดังนั้น การค้นหาข้อมูลจากกลุ่มข้อมูลจำนวนมาก จึงจำเป็นต้องใช้ระบบ Pagination หรือ การจัดแบ่งหน้า เพื่อลดการดึงข้อมูลจากฐานข้อมูลเป็นระยะเวลาที่นาน

```
SELECT * FROM(
  SELECT
    table.*,
    row_number() over (order by [การจัดเรียงข้อมูล]) rown
  FROM(
    [Code SQL หลัก] ORDER BY [การจัดเรียงข้อมูล]) table
)
WHERE rown < (([เลขหน้า] * [ขนาดแถวต่อ 1 หน้า]) + 1)
and rown >= ((([เลขหน้า] - 1) * [ขนาดแถวต่อ 1 หน้า]) + 1)
```

ภาพที่ 3.6 คำสั่ง SQL สำหรับกรองแถว

จาก SQL Code ข้างต้น สามารถกรองแถวที่ออกมาได้โดยไม่ต้องดึงข้อมูลมาทั้งหมด ทำให้ลดขนาด Packet ที่ตอบสนองกลับไปยังส่วนผู้ใช้ทำให้ดึงข้อมูลได้ไวขึ้นภายในระยะเวลาไม่ถึง 1 วินาที แต่ต้องทำการส่งค่าเพิ่มเช่นเลขหน้า แต่ถ้าไม่มีค่าดังกล่าวจะทำการกำหนดเป็นเลขหน้า 1 (หน้าแรก) โดยอัตโนมัติ

4. ทำการสร้างระบบการดูชิ้นงาน Show Graph โดยฟังก์ชันนี้มีหน้าที่แสดงจำนวนทั้งหมด ของชิ้นงานวิจัยแต่ละชนิด ในรูปแบบของกราฟแท่ง โดยใช้งาน java class เป็นตัวกลางสำหรับการเชื่อมต่อ ระหว่างเว็บแอปพลิเคชันกับระบบฐานข้อมูล โดยแบ่งเป็น 2 ส่วนคือ

- ส่วนดูจำนวนโดยรวม แบ่งข้อมูลจำนวนทั้งหมดของแต่ละชนิดและแสดงเป็นกราฟ/ตารางอย่างเหมาะสมโดยใช้วิธีการ Sum หรือรวมจำนวนแต่ละชนิดโดยใช้คำสั่ง SQL ดังนี้

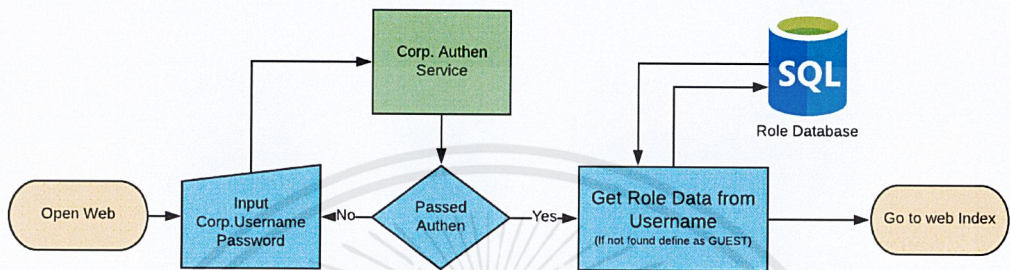
```
coalesce(sum(case when TYPE like [ชื่อชนิด] then QUANTITY end),0)
```

ภาพที่ 3.7 คำสั่ง SQL สำหรับดูจำนวนโดยรวม

โดยส่งข้อมูลจำนวนชิ้นงานกลับไปเพื่อนำไปแสดงผลผ่านภาคติดต่อประสานงานกับผู้ใช้

- ดูรายละเอียดชิ้นงานในชนิดนั้น ๆ แสดงรายชื่องานทั้งหมดลงในตาราง โดยใช้หลักการคล้ายกับข้อ 3. ส่วนการดูชิ้นงาน Search by Tray ID โดยมีการกรองจาก Parameter ที่กำหนดไว้

5. ทำการเริ่มสร้างระบบ Authentication หรือ ระบบการยืนยันตัวตนบนระบบเว็บแอปพลิเคชัน โดยใช้โปรแกรม NetBeans IDE ในติดต่อข้อมูลการยืนยันตัวตนด้วย Java Servlet (ทั้ง .JSP และ .Java) ด้วย POST request method ภายใน Service ใช้การตรวจสอบ Username และ Password ผ่านระบบ Login service ของทางบริษัทมาใช้ในระบบที่มีอยู่แล้ว ทำการดึงข้อมูลจากฐานข้อมูลเพื่อเลือกบทบาทหน้าที่ที่จัดการได้ในเว็บแอปพลิเคชันนี้ เพื่อป้องกันข้อมูลสำคัญทางธุรกิจ



ภาพที่ 3.8 Workflow การทำงานของระบบ Authentication

โดยแบ่งตำแหน่ง(Role)ในการทำงาน 3 ตำแหน่งหลัก ๆ เพื่อในการจำกัดสิทธิ์ว่าบุคคลใดสามารถเข้าถึงตรงส่วนไหนได้บ้าง ดังตารางที่ 3.2

Role	Check in / Check out	Search by Tray ID	Show Graph	Tutorial	Add user
Admin	●	●	●	●	●
Staff	●	●	●	●	
Guest		●	●	●	

ตารางที่ 3.2 ตำแหน่ง(Role) ในการจำกัดสิทธิ์ต่าง ๆ

ซึ่งต้องทำระบบเพิ่ม ลบ และแก้ไขรายชื่อผู้ให้สิทธิ์ใช้งานร่วมด้วย เพื่อให้ตำแหน่ง Admin สามารถจัดการสิทธิ์เพื่อปกป้องข้อมูลสำคัญ

และทำการกำหนดเวลาเป็นเวล.ใช้งานเป็นเวลา 40 นาทีเพื่อให้ผู้ใช้งานที่ลืมทำการออกจากระบบ จะไม่มีการเข้าสู่ระบบค้างไว้ผ่านคำสั่งดังนี้

```
session.setMaxInactiveInterval(40 * 60)
```

ภาพที่ 3.9 คำสั่ง Java สำหรับออกจากระบบอัตโนมัติเมื่อเข้าสู่ระบบค้างไว้

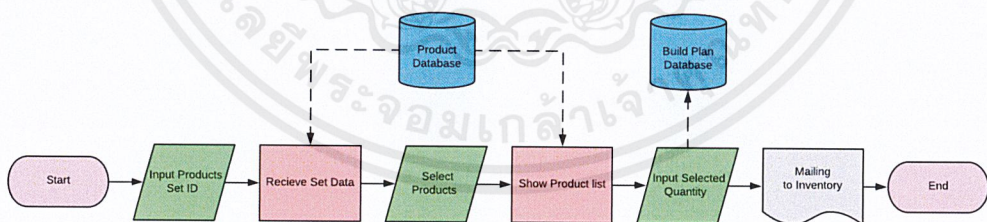
6. สร้างระบบการเบิกงาน ของผู้ใช้งานฝ่ายวางแผน (ฝ่าย Planner) โดยการส่งข้อมูลไปเข้าสู่ระบบเบิกงานต้องใช้ JSON เข้ามาช่วยส่งข้อมูลทุกส่วน เนื่องข้อมูล Data ที่สร้างขึ้นมาไม่มี Unique Key (คีย์เฉพาะ) ชัดเจน

โครงสร้าง JSON มีข้อมูลดังนี้

```
{
  "Build": [
    {
      "Adata1": "A1",
      "Adata2": "A2"
    },
    {
      "Bdata1": "B1",
      "Bdata2": "B2"
    }
  ],
  "CRIB": [
    {
      "Xdata1": "X1",
      "Xdata2": "X2"
    }
  ]
}
```

ภาพที่ 3.10 ตัวอย่างข้อมูลใน JSON

ซึ่ง Java Service ใช้ Library “javax.json.\*” ในการรับข้อมูลที่ส่งมาจากภาคติดต่อประสานงานกับผู้ใช้



ภาพที่ 3.11 Workflow การทำงานของระบบเบิกงาน

### 3.2.3 ขั้นตอนการสร้างระบบเสริมการทำงานเพื่ออำนวยความสะดวก

#### 1. สร้างระบบการสร้างไฟล์ Excel ผ่าน Java Class ด้วย Library Apache POI

โดยทำการรับค่า Parameter ที่ทำการ Request ผ่าน HTTP และส่งเป็นรูปแบบเอกสาร กลับไปโดยลักษณะคำสั่งการสร้างไฟล์ Excel เป็นดังนี้

```
response.setContentType("application/vnd.ms-excel");//xls
response.setHeader("Cache-Control", "no-cache");
response.setHeader("Pragma", "no-cache");
response.setHeader("Content-Disposition", "attachment; filename=test.xls");

HSSFWorkbook workbook = new HSSFWorkbook();
HSSFSheet table = workbook.createSheet("Test Sheet");
HSSFRow row = table.createRow(0);
HSSFCell cell;
cell = row.createCell(0);
cell.setCellValue("Data Test");

String[] head = {"Head1","Head2","Head3"};
String[] body = {"aaa","bbb","ccc"};

CellStyle style = workbook.createCellStyle();
style.setFillForegroundColor((short) dataType[a][1]);
style.setFillPattern(FillPatternType.SOLID_FOREGROUND);

row = table.createRow(1);
for(int i = 0;i<head.length;i++){
    cell = row.createCell(i);
    cell.setCellValue(head[i]);
    cell.setCellStyle(style);
}

row = table.createRow(2);
for(int j = 0;j<body.length;j++){
    cell = row.createCell(j);
    cell.setCellValue(body[j]);
}

table.createFreezePane(0, 2);

try (OutputStream outputStream = response.getOutputStream()) {
    workbook.write(outputStream);
    outputStream.flush();
}
System.out.println("xls written successfully");
```

ภาพที่ 3.12 ตัวอย่างคำสั่ง Java ในการสร้างไฟล์ผ่าน Apache POI

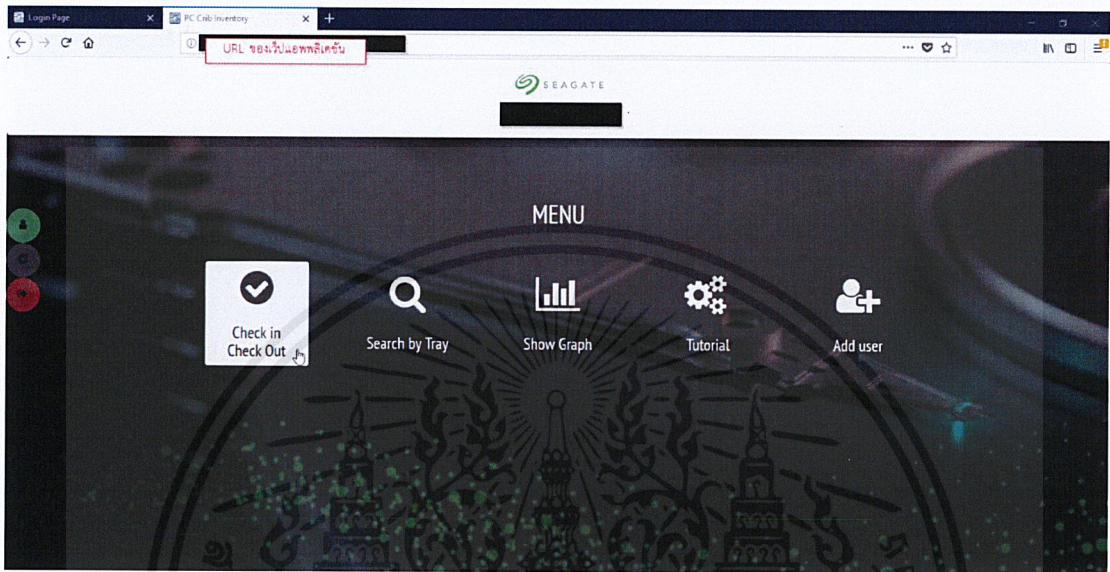
#### 2. สร้างระบบการส่ง Auto Email ผ่าน Java Class ด้วยการรับค่า Body HTML

เพื่อทำการเข้าใน Email และทำงานส่งผ่าน Mailing host (SMTP) ของบริษัท และส่งข้อมูลตอบกลับเป็นสถานะการส่งว่าสำเร็จหรือไม่

### 3.3 ขั้นตอนการทดสอบการใช้งานของโปรแกรมตัวทดลอง

#### 3.3.1 ขั้นตอนการ Launch โปรแกรมตัวทดลอง

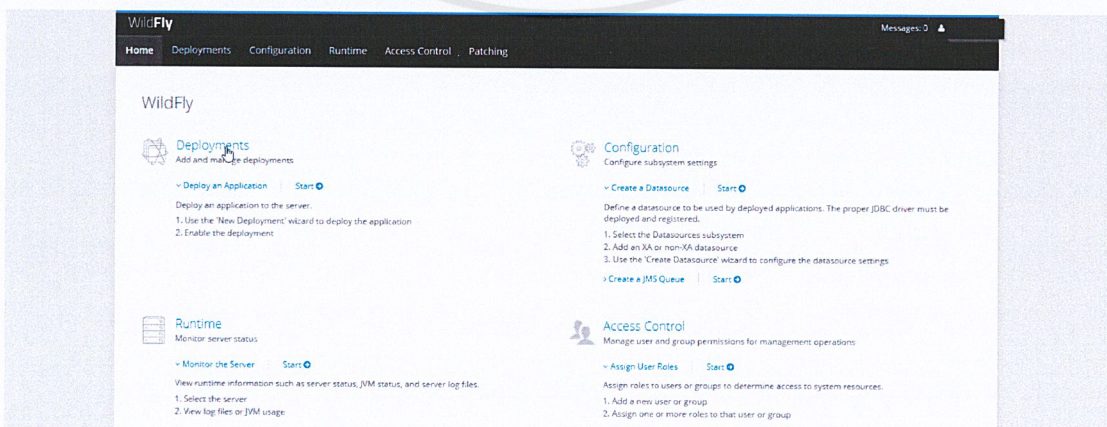
หลังจากที่ทำการออกแบบเว็บแอปพลิเคชันโดยสังเขปเสร็จเรียบร้อยแล้ว ต่อมาก็ทำการ Deploy (ปล่อยให้ใช้) ตัวโปรแกรมตัวทดลอง เพื่อให้ผู้ใช้ได้ทดลองใช้งานกับงานจริงและแจ้งปัญหากลับมา ดังภาพที่ 3.13



ภาพที่ 3.13 เว็บไซต์หลักที่ Deploy ให้ผู้ใช้งาน

#### 3.3.2 ขั้นตอนการรวบรวมข้อผิดพลาดและพัฒนาโปรแกรม

หลังจากที่ Deploy เว็บแอปพลิเคชันตัวทดลองให้ผู้ใช้ได้ลองใช้กับงานจริงเป็นระยะเวลาหนึ่ง โดยประมาณ 2 สัปดาห์ ผู้พัฒนาทำการเก็บข้อมูล ของข้อผิดพลาด และปัญหาที่เกิดขึ้นกับเว็บแอปพลิเคชัน จากที่ผู้ใช้งานได้แจ้งข้อมูลไว้ และทำการวิเคราะห์ปัญหาทั้งหมดที่เกิดขึ้น นำมาแก้ไขและพัฒนาตัวเว็บแอปพลิเคชันให้ดียิ่งขึ้น ก่อนจะทำการ Deploy ให้ผู้ใช้งานได้ใช้งานอีกครั้งหนึ่ง



ภาพที่ 3.14 การ Deploy Web Application ผ่าน WildFly Server

### 3.4 ขั้นตอนการพัฒนาโปรแกรมตัวเต็ม

#### 3.4.1 ขั้นตอนการรวบรวมความต้องการเพิ่มเติม

ทำการรวบรวมข้อมูล ข้อกำหนดและความต้องการต่าง ๆ เพิ่มเติมจากผู้ใช้งาน เพื่อนำมาทำเป็นฟังก์ชันใหม่ ๆ ที่เป็นประโยชน์มากขึ้นแก่ผู้ที่ใช้งานเว็บแอปพลิเคชัน

#### 3.4.2 ขั้นตอนการพัฒนาเพื่อเป็นโปรแกรมตัวเต็ม

ขั้นตอนสุดท้ายคือการตรวจสอบความถูกต้องของฟังก์ชันทั้งหมดบนเว็บแอปพลิเคชันอีกครั้งก่อนที่จะทำการ Deploy เว็บเวอร์ชันสมบูรณ์ ขึ้นออนไลน์อีกครั้งให้ผู้ใช้ได้ใช้งานจริง



## บทที่ 4

### ผลการดำเนินงาน

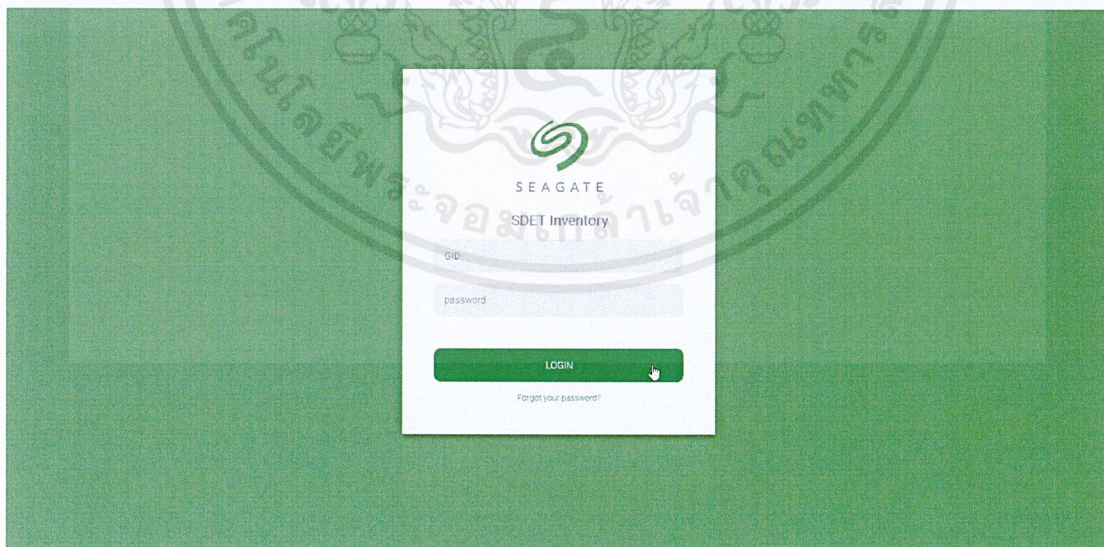
ในบทนี้จะกล่าวถึงสรุปผลการวิจัย ข้อจำกัดการวิจัย งานวิจัยในอนาคตและผลงาน โดยแต่ละส่วนที่กล่าวมานั้นมีรายละเอียดดังนี้

#### 4.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอการพัฒนากระบวนการจัดเก็บงานวิจัยและพัฒนาหลังจากเสร็จสิ้นกระบวนการผลิตบนระบบเว็บแอปพลิเคชัน เพื่ออำนวยความสะดวกสบายแก่ผู้ใช้ มีโครงสร้างและรูปแบบที่ดีขึ้นกว่าระบบเดิม และลดระยะขั้นตอนและเวลาในการทำงาน ของผู้ใช้งานให้น้อยที่สุด ซึ่งมีผลการวิจัย เป็นหน้าตารูปแบบหน้าเว็บแอปพลิเคชัน โดยข้อมูลบางส่วนเช่น ข้อมูลเกี่ยวกับผลิตภัณฑ์ของบริษัทไม่สามารถเปิดเผยได้เพราะถือเป็นความลับทางธุรกิจของบริษัท บางภาพจึงต้องทำการสร้างข้อมูลใหม่ขึ้นมา (Mock Data) หรือ ปิดข้อมูลนั้น ๆ ไว้ ดังนี้

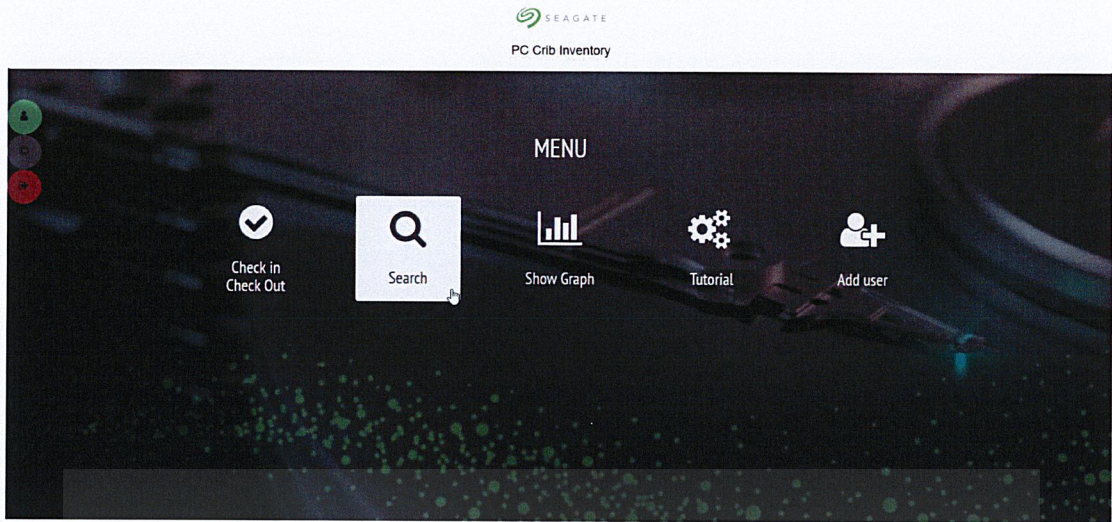
##### 4.1.1 รูปแบบหน้าหลักของเว็บไซต์

รูปแบบหน้าหลักของเว็บไซต์ มีการขอความต้องการให้เข้าสู่ระบบด้วยรหัสพนักงานของผู้ใช้งานก่อน เข้าเลือกใช้งานฟังก์ชันต่าง ๆ ที่ได้สร้างไว้



ภาพที่ 4.1 หน้าการเข้าสู่ระบบ

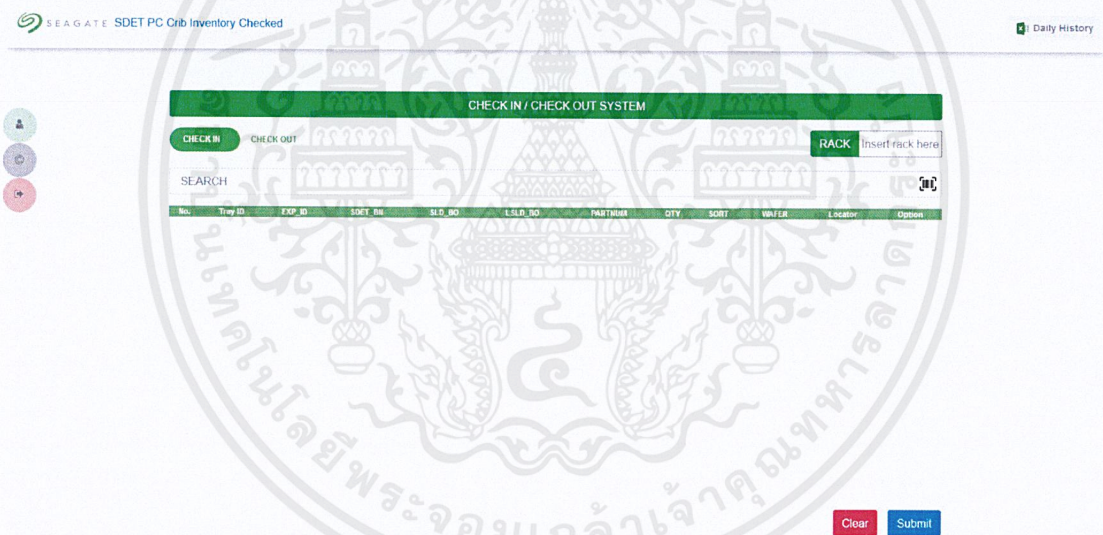
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.2 เมนูหลัก

#### 4.1.2 รูปแบบหน้าฟังก์ชัน Check in Check out

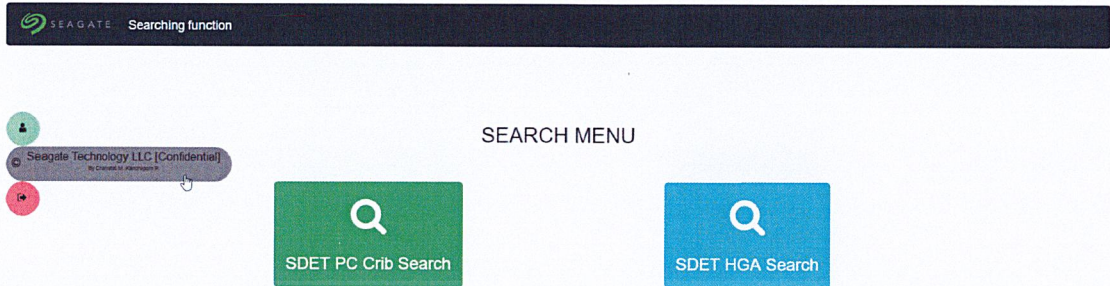
สำหรับการรับสินค้าเข้า เบิกสินค้าออก จากระบบ Inventory System



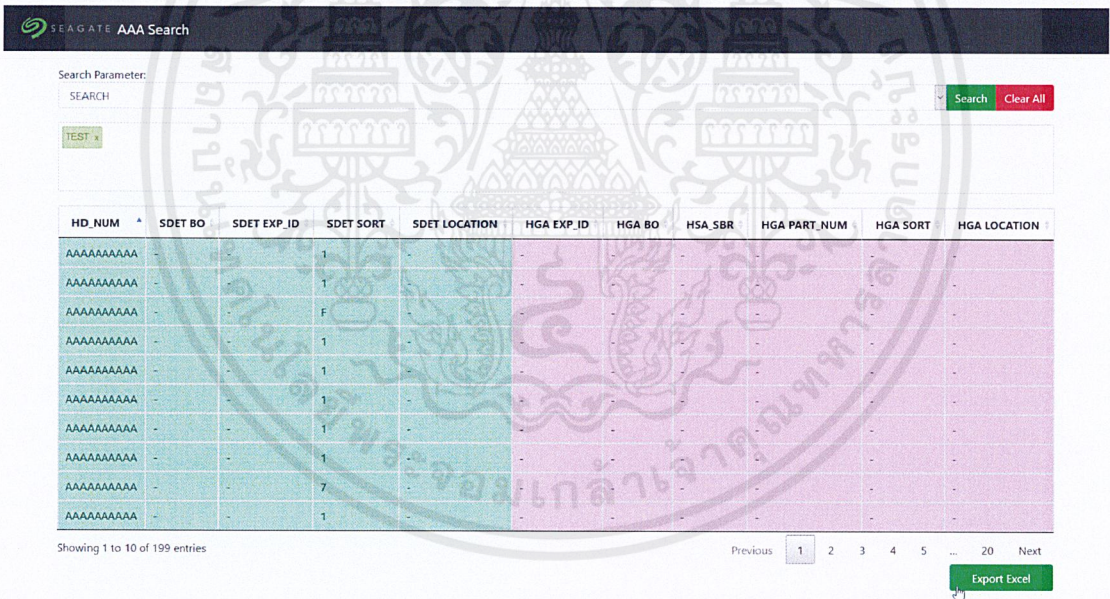
ภาพที่ 4.3 เมนู Check in / Check out

#### 4.1.3 รูปแบบหน้าเมนูฟังก์ชันของการค้นหาชิ้นงานประเภทต่าง ๆ

แสดงชิ้นงานทั้งหมดในระบบฐานข้อมูล Oracle ที่เก็บชิ้นงานจริงอยู่ใน Inventory room ค้นหาตามชนิดต่าง ๆ ของชิ้นงาน

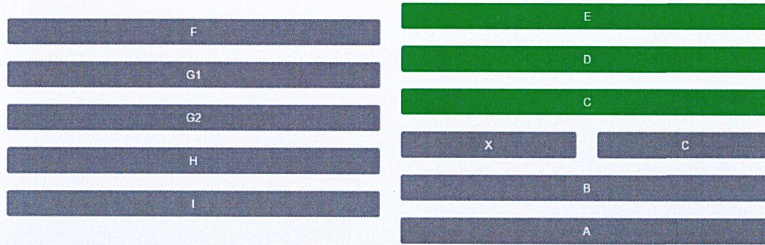


ภาพที่ 4.4 หน้าหลักของการค้นหา



ภาพที่ 4.5 หน้าการค้นหาโดยตัวแปร AAA

### Rack MAP



ภาพที่ 4.6 หน้าการค้นหาโดยตัวแปร BBB

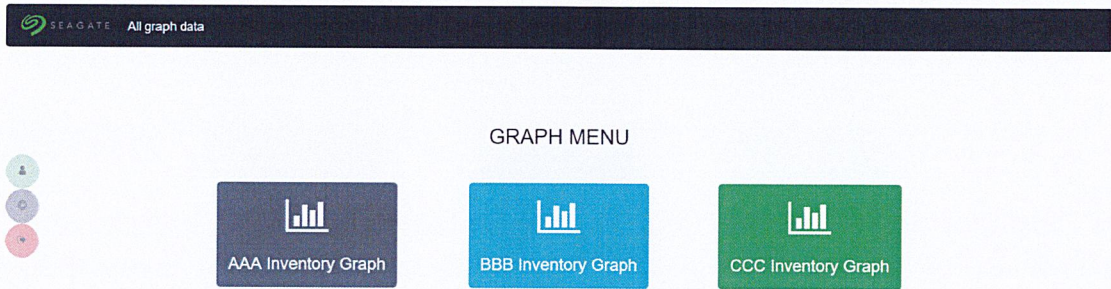
- รูปแบบหน้าฟังก์ชันแสดงข้อมูลของชิ้นงานแบบลงรายละเอียด ตามตัวแปรพารามิเตอร์ต่าง ๆ เช่น ชั้นที่เก็บชิ้นงาน



ภาพที่ 4.7 ข้อมูลทั้งหมดของผลิตภัณฑ์ ID AAAAAAAAA

#### 4.1.4 รูปแบบหน้าเมนูฟังก์ชันสำหรับการแสดงกราฟ

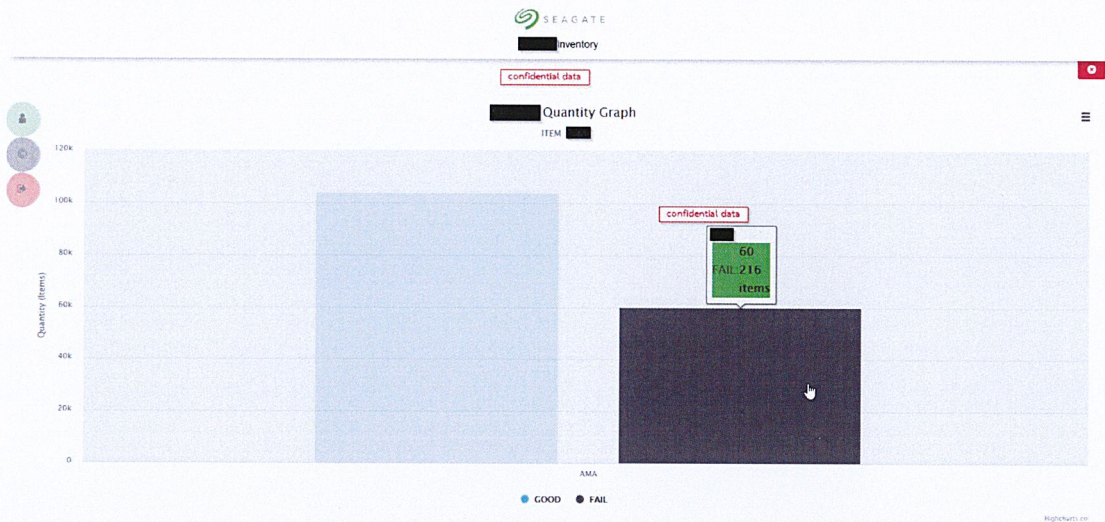
แสดงจำนวนทั้งหมด ของชิ้นงานแต่ละประเภท ในรูปแบบของกราฟแท่ง



ภาพที่ 4.8 หน้าหลักของการแสดงกราฟ

PRODUCT NAME	GOOD	FAIL	TOTAL
A	104108	60216	164324
B	3932	0	3932
C	250912	46048	296960
D	5221	784	6005
E	166	120	286
F	47916	10081	57997
G	30745	2392	33137
H	92772	45865	138637
I	131764	322	132086
TOTAL	667536	165828	833364

ภาพที่ 4.9 หน้าการแสดงผลกราฟของผลิตภัณฑ์ AAA



ภาพที่ 4.10 รายละเอียดของกราฟนั้น ๆ ของผลิตภัณฑ์ AAA โดยแบ่งตามอายุของผลิตภัณฑ์

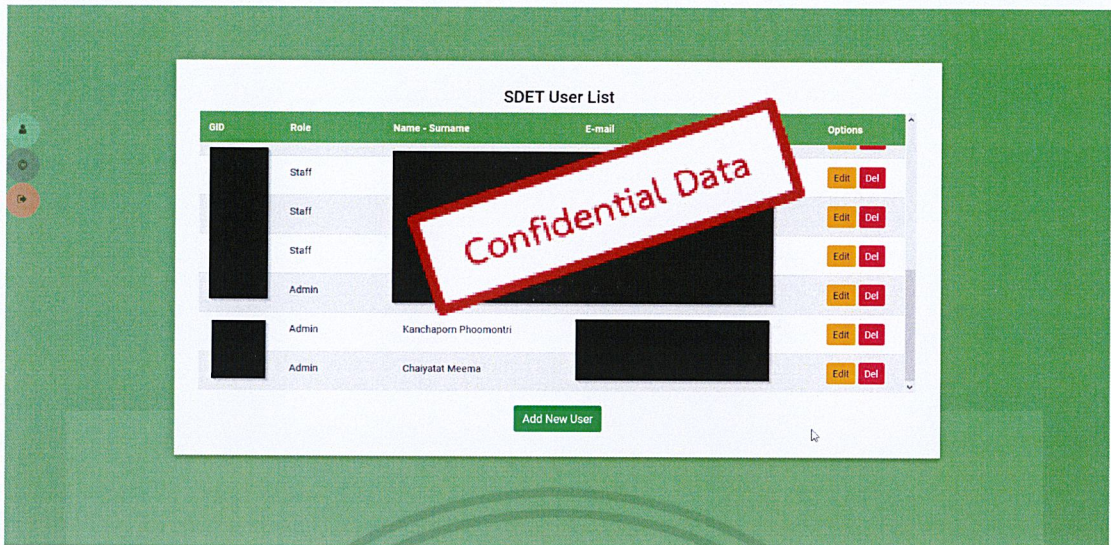
Good Item : High Aging  
Items which have aging more than 150 days

No.	AGING	PRODUCT	EXP_ID	SDET BIN	SORT	QTY	PARTNUM	TAB	RACK	LOCATOR	DATE Received	AUTHOR	AIRBEARINGDESIGN	BLD INTENT	DESCR
1															
2															
3															

Confidential Data

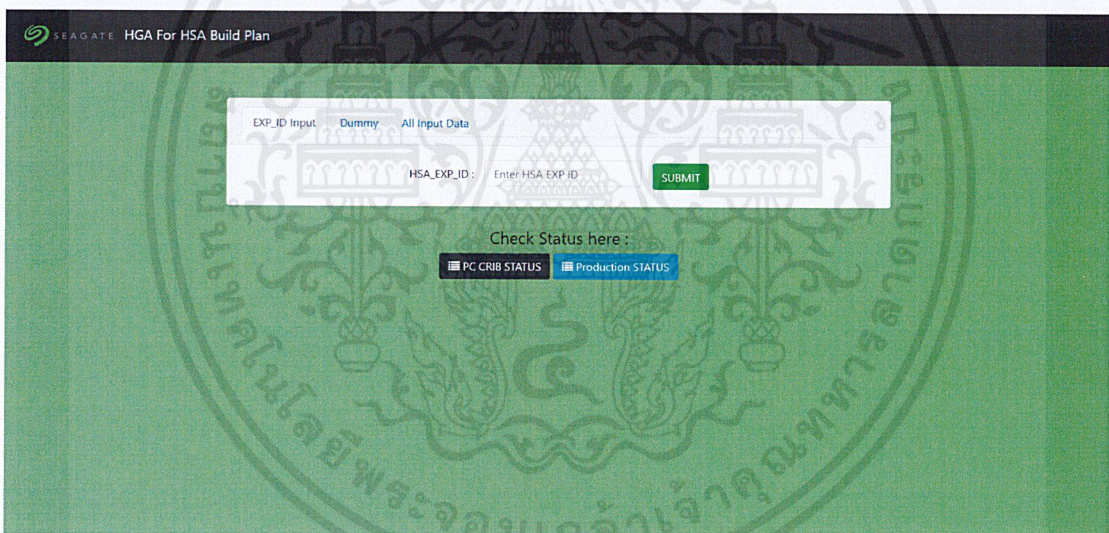
ภาพที่ 4.11 รายละเอียดข้อมูลที่จำเป็นทั้งหมดของของผลิตภัณฑ์ AAA

#### 4.1.5 รูปแบบหน้าแสดงรายชื่อของผู้ใช้งานทั้งหมดในทุกฝ่ายทุกแผนกที่เกี่ยวข้อง



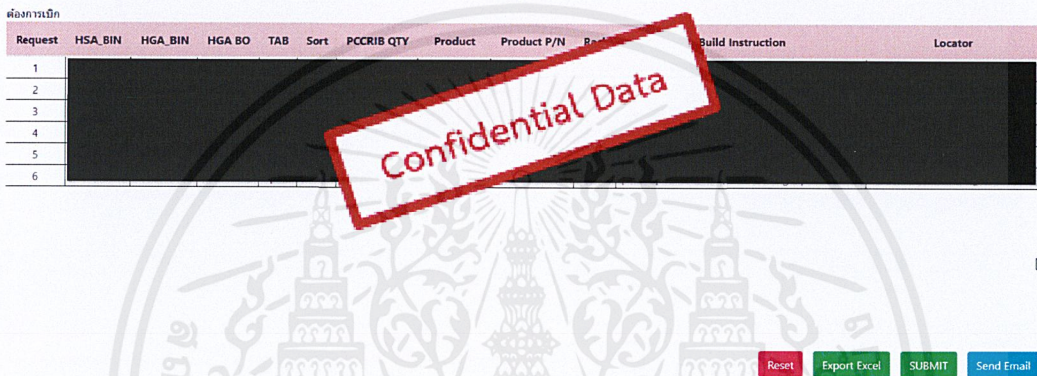
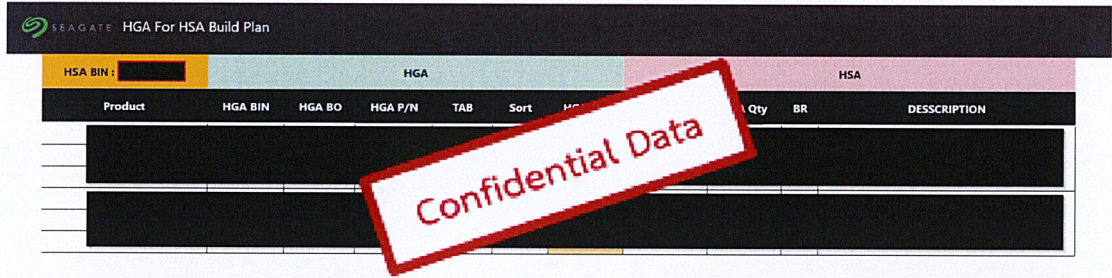
ภาพที่ 4.12 ข้อมูลของผู้ใช้งาน

#### 4.1.6 รูปแบบหน้าเมนูฟังก์ชันการเบิกงานสำหรับฝ่าย Planner



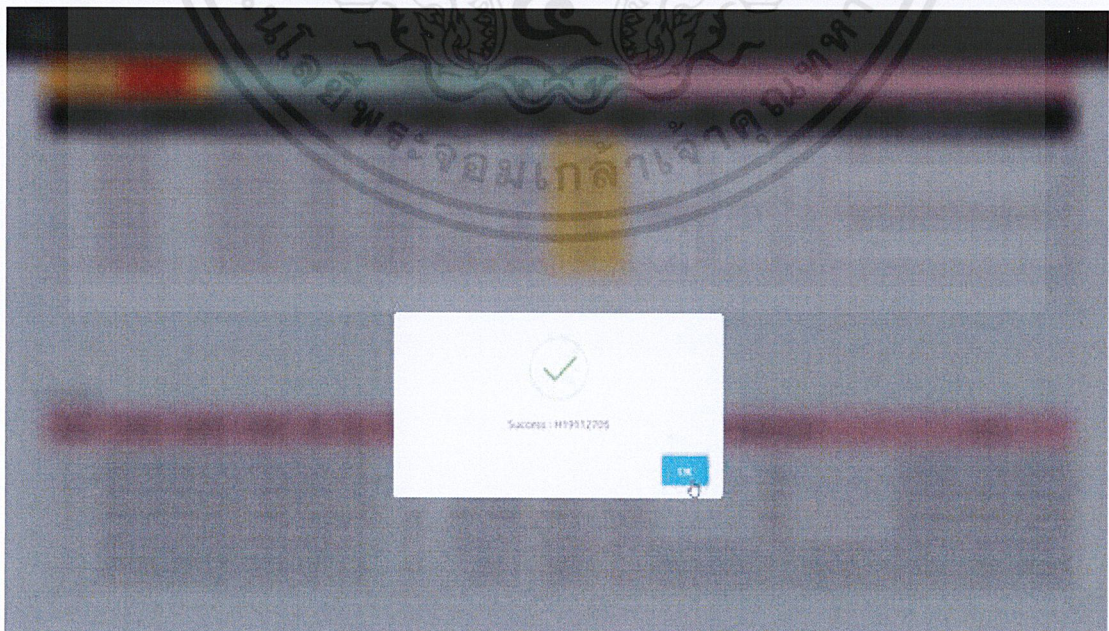
ภาพที่ 4.13 เมนูหลักของการเบิกงาน

- รูปแบบหน้าแสดงผลจากการกรอกข้อมูลค้นหาตามที่ได้กรอกในหน้าเมนูหลักของการเบิกงาน แสดงข้อมูลที่จำเป็นทั้งหมดของผลิตภัณฑ์นั้น ๆ โดยมีช่อง Request ให้ผู้ใช้งานกรอกลงไปเพื่อบันทึกลงในฐานข้อมูล



ภาพที่ 4.14 หน้าหลักของการเบิกงานของฝ่าย Planner ตาม Request ที่ได้รับ

- รูปแบบหน้าการแสดงผลว่า สำเร็จในการบันทึกข้อมูลที่ ต้องการเบิก โดยแสดงออกมาเป็นเลข Order ของการเบิกงาน



ภาพที่ 4.15 หน้าแสดงเลข Order ของการเบิกงาน

- รูปแบบหน้าของ Auto Email ของการเบิกงานที่ส่งไปที่ฝ่ายที่เกี่ยวข้องทั้งหมด

intern-test@seagate.com  
to me  
2:53 PM (1 minute ago) ☆ ↶ ⋮  
Hi All! Pls follow HAMR/PMR Flow

HSA BIN	HGA						HSA			
Product	HGA BIN	HGA BO	HGA P/N	TAB	Sort	HGA Qty	HSA SBR	HSA Qty	BR	DESCRIPTION
Confidential Data										

ผลการเบิก

Request	HSA_BIN	HGA_BIN	HGA BO	TAB	Sort	PCCRIB QTY	Product	Product P/N	Rack	Build Instruction	Locator
11	Confidential Data										
35											
13											
12											
3											
42											

ภาพที่ 4.16 รูปแบบของ Auto Email

#### 4.1.7 รูปแบบหน้าแสดงสถานะของการเบิกงานต่าง ๆ (ฝ่าย Inventory Manager)

สำหรับฝ่ายงานที่มีหน้าที่ในการจัดเก็บผลิตภัณฑ์ ซึ่งแสดงเลข Order ของการเบิกงานนั้น ๆ พร้อมกับข้อมูลที่เกี่ยวข้อง เช่น วันที่ทำการเบิกงาน ชื่อผู้ทำการเบิกงาน



SEAGATE HGA For HSA Build Plan

No.	ORDER ID	HSA BIN	Date	Author	Status
1	H19112601	AAA	2019-11-26 08:54:09.0	Intern Test	FINISHED
2	H19112501	AAA	2019-11-25 13:08:56.0	Intern Test	FINISHED
3	H19112212	AAA	2019-11-22 10:31:48.0	Intern Test	FINISHED
4	H19112211	AAA	2019-11-22 10:13:38.0	Intern Test	FINISHED
5	H19112210	AAA	2019-11-22 10:02:28.0	Intern Test	RETURN
6	H19112208	AAA	2019-11-22 09:27:27.0	Intern Test	PENDING
7	H19112203	AAA	2019-11-22 09:09:47.0	Intern Test	PENDING
8	H19112201	AAA	2019-11-22 08:53:31.0	Test ament	PENDING
9	H19112101	AAA	2019-11-21 14:30:01.0	Test ament	RETURN
10	H19111901	AAA	2019-11-19 11:30:17.0	Test ament	FINISHED

ภาพที่ 4.17 หน้าสถานะของแต่ละ Order การเบิกงาน สำหรับฝ่าย Inventory Manager

- รูปแบบหน้าแสดงรายละเอียดข้อมูลที่จำเป็นทั้งหมดของ Order การเบิกงานนั้น ๆ สำหรับฝ่าย Inventory Manager ให้กรอกข้อมูลเพิ่มเติม เพื่อส่งต่อไปที่ฝ่ายงานถัดไป

ORDER : **H19112704**  
Status: FINISHED | Requested By: Intern Test | 2019-11-27 10:35:18.0

ORDER ID	Product	HGA BIN	HGA BO	HGA P/N	TAB	Sort	HSA_REQUIRE	HSA SBA	HSA Qty	BR	DESCRIPTION
H19112704											
H19112704											
H19112704											
H19112704											
H19112704											
H19112704											

Actual QTY	Request QTY	ORDER ID	HSA BIN	HGA BIN	HGA BO	TAB	Sort	PCCRIB QTY	Product	Product P/N	Build Instruction	Locator
12	30	H19112704										
12	10	H19112704										
12	18	H19112704										

RETURN SUBMIT SEND EMAIL

ภาพที่ 4.18 ข้อมูลที่จำเป็นทั้งหมดสำหรับให้ฝ่าย Inventory Manager เข้ามากรอก

#### 4.1.8 รูปแบบหน้าแสดงสถานะของการเบิกงานต่าง ๆ (ฝ่าย Production)

สำหรับฝ่ายงาน Production ที่มีหน้าที่จัดการเกี่ยวกับผลิตภัณฑ์ในไลน์การผลิต ซึ่งแสดงเลข Order ของการเบิกงานนั้น ๆ พร้อมกับข้อมูลที่เกี่ยวข้อง ยกตัวอย่างเช่น วันที่ทำการเบิกงาน ชื่อผู้ทำการเบิกงาน

No.	ORDER ID	HSA BIN	Date	Author	PC Crib Status	Production Status
1	H19112704	ABCDE	2019-11-27 10:35:18.0	Intern Test	FINISHED	FINISHED
2	H19112703	ABCDE	2019-11-27 09:13:25.0	Intern Test	PENDING	FINISHED
3	H19112702	ABCDE	2019-11-27 09:09:03.0	Intern Test	PENDING	FINISHED
4	H19112701	ABCDE	2019-11-27 09:06:47.0	Intern Test	FINISHED	FINISHED

ภาพที่ 4.19 หน้าสถานะของแต่ละ Order การเบิกงาน สำหรับฝ่าย Production

- รูปแบบหน้าแสดงรายละเอียดข้อมูลที่จำเป็นทั้งหมดของ Order การเบิกงานนั้น ๆ สำหรับฝ่าย Production ให้กรอกข้อมูลเพิ่มเติม เพื่อเป็นการเสร็จสิ้นกระบวนการเบิกงาน

ORDER : **H19112704**  
PC Crib Status: FINISHED | Requested By: Intern Test | 2019-11-27 10:35:18.0

ORDER ID	Product	HGA BIN	HGA BO	HGA P/N	TAB	Sort	Received qty	HSA REQUIRE	HSA SBR	HSA Qty	BR	DESCRIPTION
H19112704							1					
H19112704							2					
H19112704							3					
H19112704							4					
H19112704							5					
H19112704							6					

Actual QTY	Request QTY	HSA BIN	HGA BIN	HGA BO	TAB	Sort	PCCRIB QTY	Product	Product P/N	Rack	Build Instruction	Locator
12	H19112704 30											
12	H19112704 10											
12	H19112704 18											

SUBMIT SEND EMAIL

ภาพที่ 4.20 ข้อมูลที่จำเป็นทั้งหมดสำหรับให้ฝ่าย Production เข้ามากรอก

#### 4.1.9 รูปแบบของการได้รับ Email ของทุกฝ่ายงานที่เกี่ยวข้อง

ซึ่งเนื้อความใน Email แจ้งเกี่ยวกับกระบวนการดำเนินงานดำเนินทั้งหมดในแต่ละขั้นตอนของการเบิกงาน โดยมี 4 Emails จากฟังก์ชัน Auto Email ที่ได้ทำใส่ไว้ในแต่ละหน้า Web application มีดังนี้

<input type="checkbox"/> ☆ intern-test	Order : H19112806 - PC Crib - Return BIN : AMA-06477 - Hi All, Pls Return HAMR/PMR Flow. HGA HSA ORDER ID Product HGA BIN HGA ...	3:04 PM
<input type="checkbox"/> ☆ intern-test	Order : H19112806 - Production - Received BIN : AMA-06477 - Hi All, Pls follow HAMR/PMR Flow. HGA HSA ORDER ID Product HGA BIN ...	3:04 PM
<input type="checkbox"/> ☆ intern-test	Order : H19112806 - PC Crib - เก็บ Actual Qty BIN : AMA-06477 - Hi All, Pls follow HAMR/PMR Flow. HGA HSA ORDER ID Product HGA B...	3:03 PM
<input type="checkbox"/> ☆ intern-test	Order : H19112806 - Planner - เก็บ HGA for build HSA BIN : AMA-06477 - Hi All, Pls follow HAMR/PMR Flow. HSA BIN : AMA-06477 HGA...	3:03 PM

ภาพที่ 4.21 Auto Email ทั้งหมดของการเบิกงาน



## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

ผู้วิจัยได้มีโอกาสเข้าร่วมโครงการสหกิจศึกษาที่บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด ในแผนก Head Development Operations ซึ่งเป็นแผนกสำหรับวิจัยและพัฒนาสินค้า ตั้งแต่วันที่ 5 สิงหาคม พ.ศ. 2562 ถึงวันที่ 29 พฤศจิกายน พ.ศ. 2562 ในขณะที่เข้าร่วมโครงการสหกิจศึกษา ผู้วิจัยได้รับมอบหมายงาน ให้สร้างระบบการพัฒนากระบวนการจัดเก็บงานวิจัยและพัฒนา หลังจากเสร็จสิ้นกระบวนการผลิตบนระบบเว็บแอปพลิเคชัน ภาคการจัดการระบบตอบสนองทางข้อมูล ผู้วิจัยได้เรียนรู้วิธีการทำงานจริง ของบริษัทฯ ที่เริ่มตั้งแต่การเรียนรู้ เกี่ยวกับศัพท์เทคนิคเฉพาะที่ใช้กันภายในบริษัทฯ การเข้าไปดูกระบวนการผลิต เพื่อให้เห็นกระบวนการทำงาน และปัญหาที่เกิดขึ้นจริงจากกลุ่มผู้ใช้งาน การเข้าร่วมประชุม เพื่อรวบรวมความต้องการต่าง ๆ จากผู้ใช้งาน การออกแบบและวิเคราะห์ พัฒนาระบบ การนำระบบไปทดสอบการใช้งาน ไปจนถึงการนำระบบไปใช้งานจริง

จากการดำเนินการสร้างระบบการพัฒนาการจัดเก็บงานวิจัยและพัฒนาหลังจากเสร็จสิ้นกระบวนการผลิตบนระบบเว็บแอปพลิเคชัน ภาคการจัดการระบบตอบสนองทางข้อมูล ทางผู้วิจัย ได้มีการปรับปรุงระบบให้มีความสอดคล้องกับ ขั้นตอนการดำเนินงานของระบบที่ใช้งานจริง ตรงตามความต้องการของผู้ใช้งาน ทำให้ผู้ใช้งานเห็นภาพรวมของระบบได้ชัดเจนยิ่งขึ้น อีกทั้งยังช่วยแก้ปัญหาให้ผู้ใช้งานเพื่อลดความผิดพลาด ที่เกิดจากการจัดบันทึกข้อมูลของผลการผลิต พบว่าสามารถเพิ่มประสิทธิภาพในการทำงาน ของแผนก ช่วยลดขั้นตอนที่ยุ่งยากและซับซ้อนในการทำงานของผู้ใช้งาน ทำให้ประหยัดเวลาในการทำงาน ใช้เวลาน้อยที่สุด รวมถึงช่วยลดค่าใช้จ่ายในการดำเนินงานบางส่วน โดยงานที่ได้ต้องมีคุณภาพที่ดีที่สุดเช่นกัน

## 5.2 ปัญหาและอุปสรรคที่พบ

5.2.1 เอกสารหรือข้อมูลบางอย่างเป็นความลับของทางบริษัท ไม่สามารถเปิดเผยได้ ทำให้ยากต่อการนำข้อมูลมาใช้ในการเขียนงานวิจัยครั้งนี้

5.2.2 ความรู้และประสบการณ์การทำงานในการเขียนภาษาคอมพิวเตอร์ (Programming language) ที่มีนั้น อาจไม่เพียงพอต่อการทำงาน ในช่วงเริ่มต้น ของการพัฒนาเว็บแอปพลิเคชัน ทำให้การทำงานในช่วงแรก เกิดความล่าช้าเล็กน้อย

5.2.3 ในช่วงที่มีการประชุมเก็บข้อมูลจากผู้ใช้งาน ทางผู้วิจัยไม่เข้าใจ ศัพท์เฉพาะทางเทคนิคของบริษัทฯ บางคำที่ผู้ใช้งานพูด ต้องให้ผู้นิเทศงานช่วยอธิบายงานให้อีกรอบว่าต้องทำอะไรบ้าง ทำให้เกิดปัญหาในการสื่อสารการทำงานในช่วงแรกนั้นเป็นไปค่อนข้างช้า

5.2.4 การสื่อสาร และการสรุปผลต่าง ๆ ของคนภายในองค์กรยังไม่เพียงพอ เนื่องจากต้องพัฒนาเว็บไซต์นี้ให้ผู้ใช้งานหลายฝ่ายงานได้ใช้ ซึ่งแต่ละฝ่ายมีความต้องการที่แตกต่างกัน ทำให้การดำเนินงานบางส่วนไม่ตรงตามความต้องการของทุก ๆ ฝ่าย หรืออาจจะไม่เกิดประสิทธิภาพเต็มที่ เช่น ความต้องการของผู้ใช้ที่ยังระบุได้ไม่ชัดเจนและตรงตาม ความต้องการเท่าที่ควร ซึ่งทำงานค่อนข้างล่าช้าในบางส่วน

## 5.3 ข้อเสนอแนะและแนวทางในอนาคต

5.3.1 ควรมีการเก็บสถิติจำนวนคนที่เข้ามาใช้งานเว็บไซต์ในแต่ละหน้า

5.3.2 ควรมีการเก็บประวัติการเข้าใช้งานของผู้ใช้ว่าเข้ามายังระบบไหนเวลาใดและได้เข้ามาทำอะไรกับระบบบ้าง

5.3.3 ควรปรับปรุงในส่วนของฐานข้อมูลหลักให้เก็บข้อมูลต่าง ๆ ให้มีความเป็นอิสระต่อกันมากกว่านี้ จะได้ไม่เกิดความซ้ำซ้อนกัน

## เอกสารอ้างอิง

Choi Lertkunakorn. (2557). *JAVA PROGRAMMING PART I*. เข้าถึงได้จาก SlidePlayer:

<http://slideplayer.in.th/slide/2179596/>

Computer2know. (13 มีนาคม 2560). *Apache POI คืออะไร*. เข้าถึงได้จาก Computer2know:

<http://computer2know.blogspot.com/2017/03/apache-poi.html>

Mindphp. (14 มีนาคม 2560). *Oracle คืออะไร ออราเคิล คือโปรแกรมจัดการฐานข้อมูล*. เข้าถึงได้

จาก Mindphp: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2283-oracle-คืออะไร-20.html>

Ohm Pathorn. (11 มกราคม 2559). *ส่วนประกอบของ Web Application (Website)*. เข้าถึงได้

จาก TAMA Blog: <https://blog.tamacorp.co/ส่วนประกอบของ-web-application-website/>

Suphakit Annopporchai. (2560). *รวมคำสั่ง SQL Command พื้นฐานเบื้องต้น (Basic SQL*

*Tutorial)*. เข้าถึงได้จาก Saixiii: <https://saixiii.com/sql-command/>

Tanabodin Kamol. (4 พฤศจิกายน 2562). *http คืออะไร?* เข้าถึงได้จาก Medium:

<https://medium.com/icreativesystems/basic-http-3a2b05e5aa19>

Unknown(นามปากกา). (9 ตุลาคม 2555). *CSS >> Syntax(โครงสร้างคำสั่ง)*. เข้าถึงได้จาก

CSS@Aumrin: <http://cssaumrin.blogspot.com/2012/10/css-syntax.html>

นันทวัฒน์ ไชยรัตน์. (13 กันยายน 2560). *โครงสร้างของภาษา HTML*. เข้าถึงได้จาก HelloMyWeb:

[https://www.hellomyweb.com/course/html/structure\\_of\\_html/](https://www.hellomyweb.com/course/html/structure_of_html/)

## ประวัติผู้จัดทำ



หัวข้อโครงการ การพัฒนาระบบการจัดเก็บชิ้นงานวิจัยและพัฒนา  
หลังจากเสร็จสิ้นกระบวนการผลิตบนระบบเว็บแอปพลิเคชัน  
ภาคการจัดการระบบตอบสนองทางข้อมูล

ชื่อ - สกุล นายชัยทัศน์ มีมา  
คณะ วิศวกรรมศาสตร์  
ภาควิชา วิศวกรรมคอมพิวเตอร์  
สาขาวิชา วิศวกรรมสารสนเทศ

### ประวัติส่วนตัว

วันเดือนปีเกิด 25 มกราคม พ.ศ. 2541  
ที่อยู่ 144 หมู่ 2 ตำบลหนองแม่ไก่ อำเภอโพธิ์ทอง จังหวัดอ่างทอง  
14120

### ประวัติการศึกษา

2559-ปัจจุบัน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
(ระดับปริญญาตรี)  
2553-2558 โรงเรียนโพธิ์ทอง“จินตามณี” (ระดับมัธยมศึกษา)  
2548-2552 โรงเรียนชุมชนวัดศีลขันธาราม(วิทยาคม) (ระดับประถมศึกษา)  
2547-2548 โรงเรียนวัดพระญาติการาม(ศุภสุขบำรุง) (ระดับประถมศึกษา)