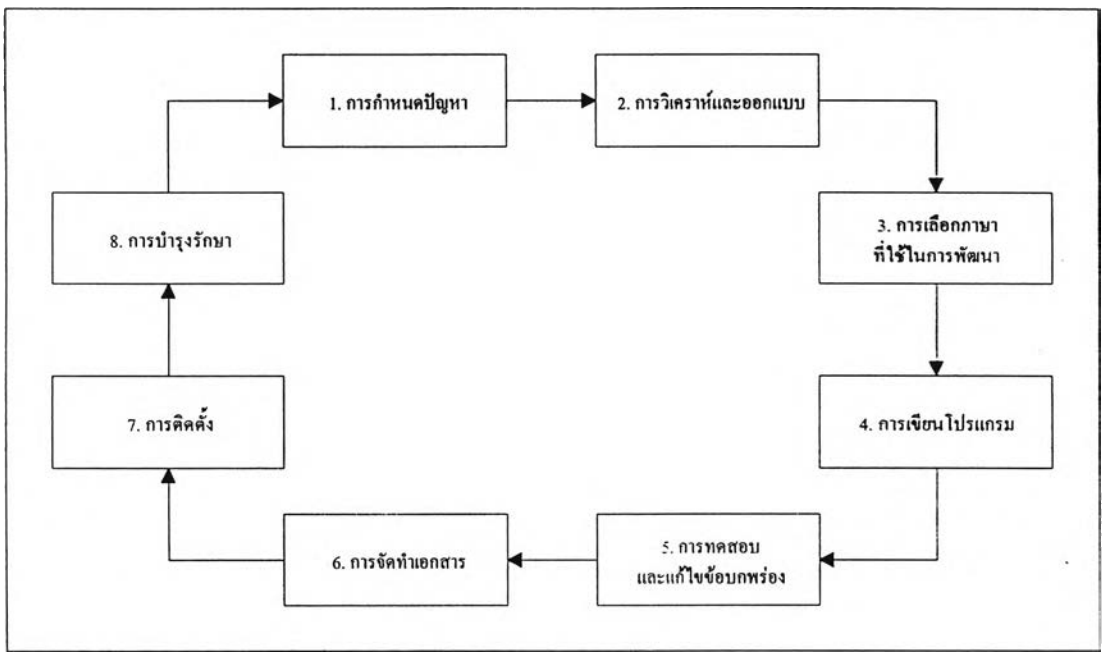
บทที่ 2

แนวคิดและทฤษฎี

2.1 การออกแบบและพัฒนาโปรแกรมประยุกต์

การออกแบบและพัฒนาโปรแกรมประยุกต์ [1] สามารถแบ่งออกได้เป็น 8 ขั้นตอน ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 วงจรการพัฒนาโปรแกรม

1. การกำหนดปัญหา

โปรแกรมประยุกต์ถูกพัฒนาขึ้นเพื่อช่วยในการแก้ไขปัญหา หรือตอบสนองความต้องการของผู้ใช้ ดังนั้นการเก็บรวบรวมสารสนเทศต่าง ๆ จึงมีความสำคัญอย่างมาก การเก็บรวบรวมข้อมูลสามารถทำได้หลายวิธี เช่น การสัมภาษณ์ความต้องการจากผู้ใช้ การทำแบบสอบถาม และการศึกษาจากระบบงานเดิม

ภายหลังจากการกำหนดปัญหา จะต้องกำหนดสิ่งที่ต้องการจากโปรแกรมที่พัฒนาขึ้น เช่น ลักษณะรูปแบบ ปริมาณของรายงาน หรือผลลัพธ์จากการใช้โปรแกรม และการกำหนดชนิดและจำนวนข้อมูลที่ต้องใช้ในการทำงานของโปรแกรม

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2. การวิเคราะห์และออกแบบ

เป็นการวิเคราะห์และหาแนวทางในการแก้ไขปัญหาที่กำหนดขึ้นจากขั้นตอนการกำหนดปัญหา แล้วทำการออกแบบกระบวนการที่ใช้ในการแก้ปัญหา การวิเคราะห์และออกแบบระบบมีเทคนิคต่าง ๆ กันหลายวิธี

3. การเลือกภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนา (Language Selection)

การพัฒนาโปรแกรมประยุกต์ ควรเลือกภาษาคอมพิวเตอร์ที่เหมาะสมกับระบบงานที่จะพัฒนา

4. การเขียนโปรแกรม

ผู้พัฒนาจะนำสิ่งที่ได้จากขั้นตอนการวิเคราะห์และออกแบบมาเขียนเป็นชุดคำสั่งตามโครงสร้างของภาษาที่เลือกใช้ ลักษณะของชุดคำสั่งที่มีประสิทธิภาพ คือ ต้องไม่สลับซับซ้อน มีคำอธิบายชุดคำสั่งเพื่อให้ง่ายและสะดวกในการอ่าน หรือแก้ไขในภายหลัง

5. การทดสอบและแก้ไขข้อบกพร่อง

เพื่อให้ผู้พัฒนามั่นใจได้ว่าโปรแกรมสามารถทำงานได้อย่างถูกต้องและแก้ไขข้อผิดพลาดก่อนที่จะนำไปใช้งานจริง จึงต้องมีการทดสอบการทำงานของโปรแกรม โดยที่การทดสอบสามารถแบ่งได้เป็น 2 ลักษณะ คือ

5.1 การทดสอบด้วยชุดข้อมูล ทำการทดสอบได้ 3 ลักษณะ คือ

5.1.1 ทดสอบด้วยชุดข้อมูลที่ถูกต้อง

5.1.2 ทดสอบด้วยชุดข้อมูลที่มีข้อผิดพลาด

5.1.3 ทดสอบด้วยชุดข้อมูลที่อยู่นอกขอบเขตที่กำหนด

โปรแกรมที่ดีจะต้องสามารถจัดการกับข้อผิดพลาดที่เกิดขึ้นจากชุดข้อมูลที่มีข้อผิดพลาดและชุดข้อมูลที่อยู่นอกเหนือขอบเขตที่กำหนดได้ โดยจะไม่แสดงผลลัพธ์ที่ไม่มีความหมายหรือหยุดทำงาน

5.2 การทดสอบระบบ โดยจะทำการทดสอบเป็นส่วน ๆ ถ้าพบข้อผิดพลาด จะแสดงข้อความบอกข้อผิดพลาดที่เกิดขึ้นทีละขั้นตอน เพื่อให้ทราบจุดที่ผิดพลาดสำหรับการแก้ไข

ข้อผิดพลาดสามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ คือ

1. รูปแบบภาษาผิดพลาด (Grammatical error) คือ การเขียนโปรแกรมไม่ถูกต้องตามโครงสร้างของภาษาคอมพิวเตอร์

2. ความผิดพลาดเชิงตรรกะ (Logical error) คือ การทำงานของโครงสร้างโปรแกรมผิดพลาด ซึ่งจะเป็นผลให้ผลลัพธ์ที่ได้จากการทำงานของโปรแกรมไม่ถูกต้อง โดยคอมพิวเตอร์จะไม่แสดงข้อผิดพลาดที่เกิดขึ้น

6. การจัดทำเอกสาร

โดยทั่วไปเราสามารถแบ่งชนิดของเอกสารของโปรแกรมได้เป็น 2 ชนิด คือ

1. เอกสารทางเทคนิค (Technical document) จะถูกใช้โดยผู้วิเคราะห์ระบบ และผู้พัฒนาโปรแกรม ซึ่งจะประกอบด้วย ชนิดของปัญหา ความต้องการ ผังงาน และองค์ประกอบอื่น ๆ ที่ใช้ระหว่างการกำหนดปัญหา การวิเคราะห์และออกแบบ

2. เอกสารสำหรับผู้ใช้ (User document) จะถูกใช้โดยผู้ใช้โปรแกรม ดังนั้นรูปแบบที่ใช้จัดทำต้องเป็นรูปแบบที่เข้าใจง่าย โดยทั่วไปจะประกอบด้วยสิ่งต่าง ๆ ดังต่อไปนี้

- ความสามารถในการทำงานของโปรแกรม
- อธิบายวิธีการใช้งาน
- สิ่งที่ได้จากโปรแกรม
- ข้อมูลที่ต้องการในการใช้งาน
- วิธีการเตรียมข้อมูล เพื่อนำเข้าสู่โปรแกรม
- ปัญหาที่อาจเกิดขึ้น และสามารถแก้ไขได้โดยผู้ใช้
- รายละเอียดของวิธีการแก้ปัญหาที่เกิดขึ้น เมื่อประสบปัญหาขณะทำงาน

7. การติดตั้ง

เป็นการนำระบบงานที่พัฒนาไปใช้ควบคู่กับระบบงานเดิมที่มีอยู่จนเกิดความมั่นใจว่าไม่มีข้อผิดพลาด จึงจะนำระบบใหม่เข้าใช้งานแทนระบบงานเดิม

8. การบำรุงรักษา

เมื่อเริ่มใช้งานแล้ว อาจจะมีความต้องการของผู้ใช้เพิ่มเติม หรือมีการปรับเปลี่ยนระบบวิธีการทำงานใหม่ ดังนั้นจะต้องมีการปรับปรุงแก้ไขโปรแกรมให้เข้ากับความต้องการหรือวิธีการทำงานใหม่

สำหรับขั้นตอนที่สำคัญที่สุดในการออกแบบและพัฒนาโปรแกรมประยุกต์ ได้แก่ การออกแบบ ซึ่งประกอบไปด้วยการกำหนดปัญหา การวิเคราห์และออกแบบ ทั้งนี้เนื่องจากการพัฒนาโปรแกรมประยุกต์ที่มีการออกแบบที่ดี จะช่วยให้การพัฒนามีความรวดเร็ว ถูกต้อง ประหยัดเวลาและค่าใช้จ่าย

2.2 การออกแบบฐานข้อมูล (Database Design)

การออกแบบฐานข้อมูลมีขั้นตอนดังนี้

1. ศึกษาและวิเคราะห์ความต้องการของผู้ใช้ (Requirements Collection and Analysis) เพื่อให้ทราบรายละเอียดและจุดประสงค์การใช้งาน เพื่อให้สามารถกำหนดโครงสร้างและความต้องการของฐานข้อมูลให้รองรับกับความต้องการของผู้ใช้ในปัจจุบันและอนาคต

2. การออกแบบฐานข้อมูลเชิงมโนภาพ (Conceptual Database Design) เป็นการอธิบายความต้องการของผู้ใช้ รายละเอียดของชนิดข้อมูล ความสัมพันธ์ และข้อกำหนดต่าง ๆ ด้วยแบบจำลองเชิงตรรกะ ซึ่งจะแสดงเอนทิตี (Entity) ลักษณะประจำ (Attribute) และความสัมพันธ์ระหว่างกัน (Relationship) ซึ่งจะมีกระบวนการหลักดังต่อไปนี้

2.1 สร้างมุมมองของผู้ใช้ (User view) โดยการกำหนดเอนทิตีหลักและพิจารณาหาความสัมพันธ์ระหว่างกันของเอนทิตีหลักเหล่านั้น แล้วกำหนดคีย์หลัก (Primary Key) คีย์สำรอง(Alternate Key) ฟอเรจ์นคีย์ (Foreign Key) และกฎธุรกิจ (Business Rules)

2.2 เพิ่มรายละเอียดให้กับมุมมองของผู้ใช้ ได้แก่ การเพิ่มลักษณะประจำที่ไม่ใช่คีย์ (Non-key Attribute) ซึ่งเป็นรายละเอียดของเอนทิตี

2.3 ตรวจสอบความถูกต้องด้วยการนอร์มัลไลเซชัน (Normalization) โดยการวิเคราะห์แยกโครงสร้างของข้อมูล เพื่อให้แน่ใจว่าแบบจำลองข้อมูลอยู่ในรูปแบบบรรทัดฐาน (Normal form) ไม่มีความซ้ำซ้อนของข้อมูล ซึ่งบางครั้งอาจจะต้องสร้างเอนทิตีใหม่ หรือเพิ่มลักษณะประจำเพิ่มเติมให้กับเอนทิตีเดิม

2.4 รวมมุมมองของผู้ใช้แต่ละกลุ่มเข้าด้วยกัน เพื่อให้เป็นแบบจำลองของทั้งระบบ โดยจะต้องพิจารณาถึงส่วนที่แตกต่างและเหมือนกัน รวมทั้งการเชื่อมโยงเกี่ยวข้องกันระหว่างมุมมองด้วย

3. การแปลงส่งแบบจำลองข้อมูล (Data Model Mapping) เป็นการติดตั้งฐานข้อมูลในระบบจัดการฐานข้อมูลที่ต้องการ ดังนั้นจะต้องแปลงแบบจำลองข้อมูลเชิงมโนภาพให้เป็นแบบจำลองข้อมูลเชิงกายภาพ

4. การออกแบบฐานข้อมูลเชิงกายภาพ (Physical Database Design) เป็นการระบุโครงสร้างหน่วยเก็บภายใน วิธีการเข้าถึงข้อมูล และการจัดระเบียบแฟ้มสำหรับฐานข้อมูล เพื่อให้ฐานข้อมูลมีประสิทธิภาพในการทำงาน

2.3 การพัฒนาระบบฐานข้อมูลที่ทำงานผ่านระบบเครือข่ายเวิลด์ไวด์เว็บ

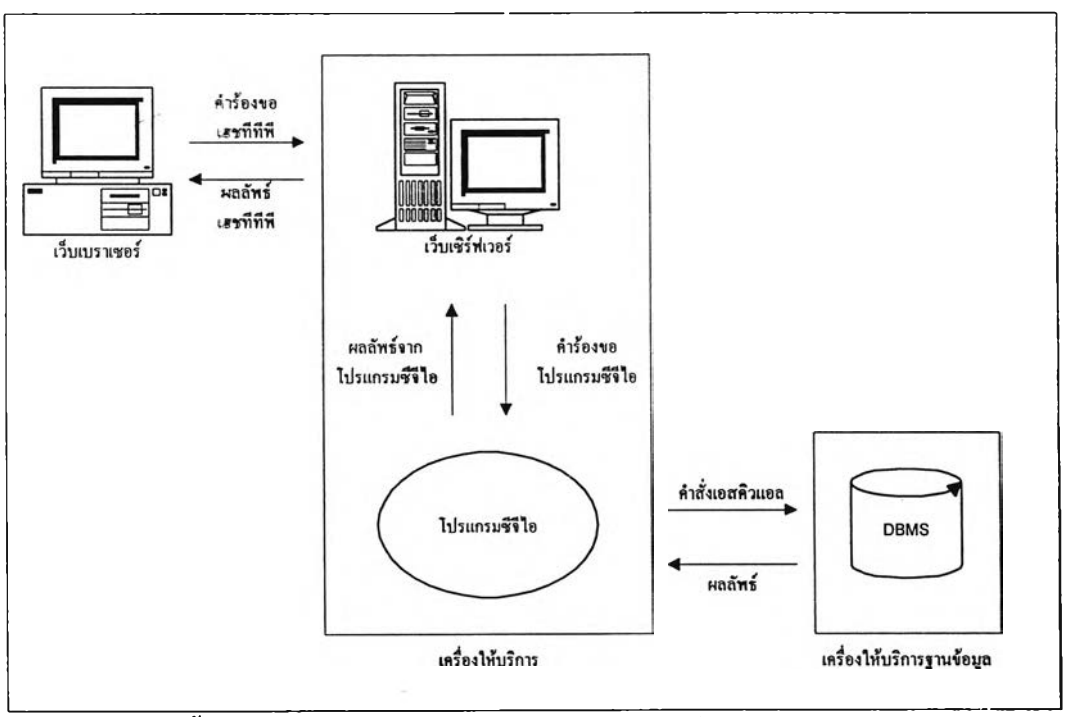
เวิลด์ไวด์เว็บเป็นวิธีการหนึ่งที่ใช้ในการติดต่อแลกเปลี่ยนข้อมูลในเครือข่ายอินเทอร์เน็ต (Internet) และได้รับความนิยมจนมีการนำมาใช้ในเครือข่ายอินทราเน็ต (Intranet) ซึ่งมีความ

สามารถใช้งานได้ทั้งในข่ายงานบริเวณเฉพาะที่ (LAN : Local Area Network) และข่ายงานบริเวณกว้าง (WAN : Wide Area Network) โดยมีองค์ประกอบที่สำคัญได้แก่ เว็บเซิร์ฟเวอร์ (Web Server) เว็บเบราเซอร์ (Web Browser) และโปรโตคอลเฮชทีทีพี (HTTP Protocol) ที่ใช้ติดต่อระหว่างเว็บเซิร์ฟเวอร์และเว็บ เบราเซอร์

โดยปกติแล้ว เว็บเซิร์ฟเวอร์จะให้บริการข้อมูลของเว็บเพจ (Web page) ที่ได้มีการจัดเตรียมไว้ล่วงหน้าแล้ว การนำเว็บเซิร์ฟเวอร์มาให้บริการข้อมูลที่เก็บอยู่ในระบบฐานข้อมูลจะต้องพัฒนาโปรแกรมเพื่อติดต่อกับระบบจัดการฐานข้อมูลเพื่อขอข้อมูลมาประมวลผล แล้วส่งผลลัพธ์ที่ได้ให้กับเว็บเซิร์ฟเวอร์เพื่อส่งผ่านให้กับเว็บเบราเซอร์ที่ร้องขอข้อมูล

2.4 ซีจีไอ (CGI : Common Gateway Interface)

ซีจีไอ (CGI : Common Gateway Interface) [2] เป็นมาตรฐานของการพัฒนาโปรแกรมที่ติดต่อกับเว็บเซิร์ฟเวอร์ ส่วนโปรแกรมที่พัฒนาขึ้นมาตามมาตรฐานของซีจีไอ จะเรียกว่าโปรแกรมซีจีไอ การพัฒนาโปรแกรมซีจีไอจะพัฒนาด้วยภาษาคอมพิวเตอร์อะไรก็ได้ แต่รูปแบบของข้อมูลนำเข้าและกฎเกณฑ์ของการแสดงผลลัพธ์จะต้องเป็นไปตามมาตรฐานของซีจีไอ ภาษาที่เหมาะสมสำหรับการเขียนโปรแกรมซีจีไอจะต้องมีความสามารถในการจัดการเกี่ยวกับข้อความและสามารถติดต่อกับไลบรารีซอฟต์แวร์และโปรแกรมอรรถประโยชน์ได้ง่าย



รูปที่ 2.2 ขั้นตอนการติดต่อระหว่างเว็บเซิร์ฟเวอร์กับเครื่องให้บริการฐานข้อมูลของโปรแกรมซีจีไอ

ขั้นตอนการทำงานของโปรแกรมซีจีไอที่แสดงในรูปที่ 2.2 มีดังนี้

1. เว็บเบราเซอร์ร้องขอบริการโปรโตคอลเฮชทีทีพี (HTTP request) ไปยังเว็บเซิร์ฟเวอร์โดยระบุยูอาร์แอล (URL : Uniform Resource Locators) ที่ต้องการ

2. เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอและพบว่า ยูอาร์แอลที่ร้องขอเป็นโปรแกรมซีจีไอ เว็บเซิร์ฟเวอร์จะทำการประมวลผลโปรแกรมซีจีไอนั้น แล้วส่งผลลัพธ์กลับมาให้เว็บเบราเซอร์

สำหรับการติดต่อระหว่างเว็บเซิร์ฟเวอร์กับเครื่องบริการฐานข้อมูล (Database Server) โปรแกรมซีจีไอจะใช้ฟังก์ชันเพิ่มเติมหรือโปรแกรมสนับสนุนอื่น เช่น ออราเพิร์ล (oraperl) , ออราเคิลเว็บเอเจนต์ (Oracle Web Agent) หรือเจดีบีซี (JDBC : Java Database Connectivity)

2.5 ภาษาจาวา

ภาษาจาวาถูกพัฒนาโดยบริษัท ซันไมโครซิสเต็ม อิงค์ ในปี พ.ศ. 2534 เพื่อให้เป็นภาษาที่ไม่ยึดติดกับแพลทฟอร์ม และสามารถใช้ในการพัฒนาซอฟต์แวร์เพื่อใช้สำหรับอุปกรณ์ไฟฟ้า เช่น เตาไมโครเวฟ หรือรีโมทคอนโทรล ที่ใช้ไมโครโพรเซสเซอร์ที่แตกต่างกันได้

ภาษาจาวาถูกพัฒนาโดยมีพื้นฐานมาจากภาษาซี มีไวยากรณ์ที่เหมือนกับภาษาซี และมีหลักการเชิงวัตถุ (object-oriented) เหมือนภาษาซีพลัสพลัส โดยมีคุณสมบัติพิเศษ ได้แก่ การ์เบจคอลเล็คชั่น (Garbage collection) ช่วยในการจัดการบริหารหน่วยความจำ และมีชุดคำสั่งที่สนับสนุนการเขียนโปรแกรมให้ทำงานเป็นเทรด (Thread) ที่ไม่ยึดติดกับระบบปฏิบัติการ

2.6 จาวาเซิร์ฟเล็ท (Java Servlet)

จาวาเซิร์ฟเล็ท [3] หรือเซิร์ฟเล็ทเป็นโปรแกรมภาษาจาวาที่ทำงานอยู่ในเครื่องให้บริการเพื่อทำงานตอบสนองการเรียกใช้บริการของผู้ขอใช้บริการผ่านระบบเครือข่ายเวิลด์ไวด์เว็บ โดยมีเซิร์ฟเล็ทเอ็นจิน (Servlet Engine) ที่ทำงานเป็นโมดูลหนึ่งของเว็บเซิร์ฟเวอร์ทำหน้าที่ในการประมวลผลคำสั่งในโปรแกรมเซิร์ฟเล็ท และส่งผลลัพธ์ในรูปแบบของโปรโตคอลเฮชทีทีพีกลับไปยังผู้ขอใช้บริการ โปรแกรมเซิร์ฟเล็ทจะมีลักษณะในการติดต่อกับเว็บเซิร์ฟเวอร์และเครื่องให้บริการฐานข้อมูลเหมือนกับโปรแกรมซีจีไอ

การทำงานของโปรแกรมเซิร์ฟเล็ท จะมีขั้นตอนดังนี้

1. การเริ่มต้น (Initialization) เซิร์ฟเล็ทเอ็นจินที่ทำงานอยู่ในเว็บเซิร์ฟเวอร์จะบรรจุ (Load) โปรแกรมเซิร์ฟเล็ทเข้ามาในหน่วยความจำและทำงานวิธีการอินิท (method init) เพื่อกำหนดค่าเริ่มต้น และปรับแต่งการทำงานของโปรแกรมเซิร์ฟเล็ท โดยปกติการเริ่มต้นจะทำเพียงครั้งเดียวเมื่อเริ่มใช้งานโปรแกรมเซิร์ฟเล็ทครั้งแรก

2. การประมวลผลคำร้องขอใช้บริการ (Request Processing) เมื่อเว็บเซิร์ฟเวอร์ได้รับคำขอใช้บริการที่เรียกใช้โปรแกรมเซิร์ฟเล็ท เว็บเซิร์ฟเวอร์จะทำงานวิธีการเซอร์วิส (method service) ของโปรแกรมเซิร์ฟเล็ทที่ผู้ขอใช้บริการร้องขอมา โดยจะประมวลผลคำสั่งในวิธีการเซอร์วิส และอาจจะเรียกวิธีการอื่นช่วยในการประมวลผล แล้วส่งผลลัพธ์ที่ได้จากการประมวลผลกลับคืนให้กับเว็บเบราเซอร์

3. การเลิก (Termination) เมื่อเว็บเซิร์ฟเวอร์ไม่ต้องการใช้โปรแกรมเซิร์ฟเล็ทอีก หรือเมื่อโปรแกรมเซิร์ฟเล็ทถูกบรรจุใหม่ เว็บเซิร์ฟเวอร์จะทำงานวิธีการเดสทรอย (method destroy) ของโปรแกรมเซิร์ฟเล็ทนั้น

ความแตกต่างระหว่างเซิร์ฟเล็ทและโปรแกรมซีจีไอ แสดงในตารางที่ 2.1

โปรแกรมซีจีไอ	โปรแกรมเซิร์ฟเล็ท
1. มักจะเขียนด้วยภาษาแปลคำสั่ง (Interpreted language) ซึ่งทำงานได้ค่อนข้างช้าในขั้นตอนการดำเนินงาน	1. เขียนด้วยภาษาแปลโปรแกรม (Compiled language) ซึ่งสามารถทำงานได้เร็วในการขั้นตอนการดำเนินงาน
2. ทำงานเป็นคนละกระบวนการกับกระบวนการของเว็บเซิร์ฟเวอร์	2. ทำงานเป็นโมดูลหนึ่งในกระบวนการของเว็บเซิร์ฟเวอร์
3. เมื่อสิ้นสุดการทำงานของโปรแกรมแต่ละครั้งจะถูกลบออกจากหน่วยความจำ ทำให้ต้องบรรจุโปรแกรมเข้าไปในหน่วยความจำใหม่ทุกครั้งที่ต้องการใช้โปรแกรม	3. เมื่อโปรแกรมถูกเรียกใช้งานแล้ว จะยังคงอยู่ในหน่วยความจำ เมื่อมีการเรียกใช้งานซ้ำโปรแกรมจะสามารถทำงานได้ทันที ไม่ต้องรอบรรจุโปรแกรมเข้าไปในหน่วยความจำ
4. ไม่สามารถเก็บสถานะการทำงานของโปรแกรมได้ เนื่องจากโปรแกรมถูกลบออกจากหน่วยความจำเมื่อสิ้นสุดการทำงาน	4. สามารถเก็บสถานะการทำงานของโปรแกรมได้เนื่องจากโปรแกรมยังคงอยู่ในหน่วยความจำ

ตารางที่ 2.1 ความแตกต่างระหว่างโปรแกรมซีจีไอและโปรแกรมเซิร์ฟเล็ท

2.7 เจดีบีซี (JDBC)

เจดีบีซี [4] เป็นส่วนเพิ่มเติมของภาษาจาวาที่ใช้งานร่วมกับไดร์เวอร์เจดีบีซี เพื่อเพิ่มความสามารถของภาษาจาวาในการติดต่อกับระบบจัดการฐานข้อมูล ลักษณะการติดต่อระหว่างไดร์เวอร์

เจดีบีซีกับระบบจัดการฐานข้อมูลสามารถแบ่งตามวิธีของบริษัทผู้ผลิตซอฟแวร์ จาวาซอฟต์ ซันไมโครซิสเต็ม อิงค์ ได้เป็น 4 รูปแบบ คือ

1. เจดีบีซี-โอดีบีซีบริดจ์ (JDBC-ODBC Bridge) รูปแบบนี้เจดีบีซีจะติดต่อฐานข้อมูลผ่านไดร์เวอร์โอดีบีซี การติดต่อโดยวิธีนี้จำเป็นต้องติดตั้งไดร์เวอร์หรือโปรโตคอลเครือข่าย (network protocol) ของระบบจัดการฐานข้อมูลเพิ่มเติมในเครื่องรับบริการเพื่อใช้ในการติดต่อกับเครื่องให้บริการฐานข้อมูล

2. ไดรเวอร์เนทีฟเอพีไอที่สนับสนุนจาวาเพียงแค่บางส่วน (Native-API partly Java technology-enabled driver) รูปแบบนี้จะเปลี่ยนคำสั่งของเจดีบีซีให้ทำงานผ่านโปรแกรมผู้ขอใช้บริการของระบบจัดการฐานข้อมูล ซึ่งจำเป็นต้องติดตั้งไดร์เวอร์เพิ่มเติมในเครื่องรับบริการเหมือนกับวิธีเจดีบีซี-โอดีบีซีบริดจ์

3. ไดร์เวอร์ของโปรโตคอลเครือข่ายที่สนับสนุนภาษาจาวาทั้งหมด (Net-protocol all-Java driver) รูปแบบนี้จะเปลี่ยนคำสั่งของเจดีบีซีไปเป็นโปรโตคอลที่ไม่ยึดติดกับระบบจัดการฐานข้อมูลก่อน จากนั้นจึงแปลผลเป็นโปรโตคอลของระบบจัดการฐานข้อมูล ซึ่งจะทำให้เครื่องรับบริการสามารถติดต่อกับฐานข้อมูลหลายฐานข้อมูลที่แตกต่างกันได้

4. ไดร์เวอร์เนทีฟโปรโตคอลที่สนับสนุนจาวาทั้งหมด (Native-protocol all-Java driver) รูปแบบนี้จะเปลี่ยนคำสั่งเจดีบีซีไปเป็นโปรโตคอลเครือข่ายของระบบจัดการฐานข้อมูลนั้น ๆ เช่น เอสคิวแอลเน็ต (SQL*NET), ไอคอนเน็ค (I-Connect), โอเพินไคลเอ็นท์ (OpenClient) ซึ่งจะเป็นการติดต่อจากเครื่องรับบริการไปยังเครื่องให้บริการฐานข้อมูลโดยตรงโดยไม่จำเป็นต้องติดตั้งไดร์เวอร์อื่นเพิ่มเติม