ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

ระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ

Information System for Medical Clinic using Object Oriented Techniques

โดย

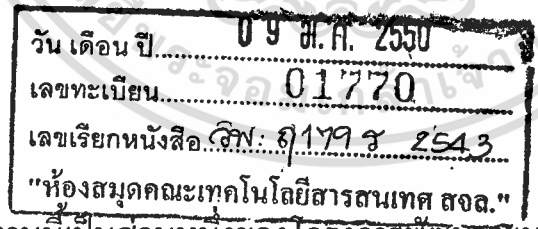
นางสาว ฤทัยรัตน์ พัดฉวีวรรณ

รหัส 42067084

H001770

อาจารย์ที่ปรึกษา

ดร. ประจวบ วานิชชัชวาล



รายงานนี้เป็นส่วนหนึ่งของโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2543

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ
นักศึกษา	นางสาว ฤทัยรัตน์ พัดฉวีวรรณ
อาจารย์ที่ปรึกษา	ดร..ประจวบ วานิชชัชวาล
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

ระบบจัดการข้อมูลภายในคลินิก นี้ จะเป็นการศึกษาถึงปัญหาและความต้องการของเจ้าหน้าที่ หรือผู้ที่เกี่ยวข้องกับการดำเนินงานภายในคลินิก ซึ่งในปัจจุบันจะเก็บข้อมูลโดยการใช้แฟ้มเอกสารบันทึกข้อมูล อันก่อให้เกิดปัญหาในเรื่องความผิดพลาด ล่าช้า ดังนั้น จึงควรนำคอมพิวเตอร์เข้ามาใช้งานมาในระบบนี้

ระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ จึงถูกพัฒนาขึ้นเพื่อแก้ปัญหาดังกล่าว และเพิ่มประสิทธิภาพในการทำงาน เพื่อตอบสนองตามความต้องการของผู้ใช้งาน โดยในการพัฒนาระบบนี้ จะนำเอาแนวความคิดเชิงวัตถุ (Object-Oriented Concept) เข้ามาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Information System for Medical Clinic using Object-Oriented Techniques
Student	Miss Ruetairut Pudchawewon
Advisor	Prachuab Vanitchatchavan, Ph.D.
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

ABSTRACT

This development project is to studying the problems of the clinic management system, requirement of the clinic personal and to solve the problems and redesign procedure to help the system increase efficient correctness and speed than the system which is still using manual system to a more efficient computerize system.

Information System for Medical Clinic using Object-Oriented Techniques was developed to solve the problems in management of clinic and make satisfied to the person with involve the system. This system develops by using Object-Oriented Methodology.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการศึกษาและพัฒนาระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุนี้ ผู้จัดทำขอขอบพระคุณ ท่าน อาจารย์ ดร.ประจวบ วานิชชัชวาล ที่ได้ให้คำปรึกษาและแนะนำแนวทางในการพัฒนาระบบจนเสร็จสมบูรณ์ และ โครงงานนี้ได้มีผู้เกี่ยวข้อง ที่สนับสนุนและให้ความช่วยเหลือหลายท่าน ดังนี้

- ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และน้องสาว ที่คอยดูแลเป็นกำลังใจ และสนับสนุนในการทำงานเสมอมา
- ขอขอบคุณเพื่อนๆ IS7 สมทบ ที่คอยช่วยเหลือ ให้คำแนะนำ ถามไถ่
- ขอบคุณเพื่อนๆชาวธรรมศาสตร์ที่ให้ความช่วยเหลือและคำแนะนำ
- และขอขอบคุณ พี่ป๊อบ ที่คอยช่วยเหลือ และ สนับสนุน ทุกๆอย่างตลอดมา

ฤทัยรัตน์ พัดฉวีวรรณ

11 กุมภาพันธ์ 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
บทที่	
1. บทนำ	1
1.1 ความเป็นมาของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	2
1.3 ขอบเขตของโครงงาน	2
1.4 หลักการในการพัฒนาระบบ	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
2. หลักการและทฤษฎี	5
2.1 แนวความคิดเชิงวัตถุ	5
2.2 ภาษาแบบจำลองสำหรับการพัฒนาโปรแกรมเชิงวัตถุ (Unified Modeling Language)	9
2.3 การวิเคราะห์และออกแบบระบบเชิงวัตถุ	23
3. วิธีการดำเนินการและรวบรวมข้อมูล	30
3.1 กรณีศึกษา	30
3.2 เครื่องมือที่ใช้ในโครงการ	30
3.3 การเก็บรวบรวมข้อมูล	31
3.4 การวิเคราะห์และขั้นตอนการพัฒนาระบบ	31
4. การวิเคราะห์และออกแบบระบบงาน	33
4.1 การดำเนินการภายในคลินิกในปัจจุบัน	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การวิเคราะห์และออกแบบระบบจัดการข้อมูลภายในระบบคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ 34

5. การดำเนินการพัฒนาระบบ 53

5.1 Rational Rose 2000 Enterprise Edition 54

5.2 Microsoft Visual Basic 6 54

5.3 กระบวนการในการวิเคราะห์ ออกแบบ และพัฒนาระบบจัดการข้อมูลภายในระบบคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ 55

5.4 จอภาพและรายงานของระบบจัดการข้อมูลภายในคลินิกโดนหลักการพัฒนาระบบเชิงวัตถุ 55

6. บทสรุปและข้อเสนอแนะ

6.1 บทสรุป 70

6.2 ข้อเสนอแนะ 71

บรรณานุกรม 72

ประวัติผู้เขียน 73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่		หน้า
2.1	ตัวอย่าง Class Car และ Object Car	6
2.2	แสดงการส่ง Message ระหว่าง Object A	7
2.3	ตัวอย่างการ Inheritance ของ Class Vehicle	8
2.4	ตัวอย่างแบบจำลอง Use Case	13
2.5	ตัวอย่างแบบจำลอง Class	15
2.6	ตัวอย่างแบบจำลอง Object	16
2.7	สัญลักษณ์แบบจำลอง State	17
2.8	ตัวอย่างแบบจำลอง State	18
2.9	ตัวอย่างแบบจำลอง Sequence	19
2.10	ตัวอย่างแบบจำลอง Collaboration	20
2.11	ตัวอย่างแบบจำลอง Activity	21
2.12	ตัวอย่างแบบจำลอง Component	22
2.13	ตัวอย่างแบบจำลอง Deployment	23
2.14	การเพิ่ม Class จาก ขั้นตอน Analysis ในขั้นตอนการ Design	27
4.1	Use Case Diagram ของระบบคลินิก	35
4.2	Class Diagram ที่ได้จากขั้นตอนการวิเคราะห์ระบบคลินิก	37
4.3	Activity Diagram ที่ได้จากขั้นตอนการวิเคราะห์ระบบคลินิก	39
4.4	Class Diagram ที่ได้จากการออกแบบระบบคลินิก	41
4.5	องค์ประกอบของ Class Address	42
4.6	องค์ประกอบของ Class Person	42
4.7	องค์ประกอบของ Class PR	43
4.8	องค์ประกอบของ Class Nurse	43
4.9	องค์ประกอบของ Class Doctor	43
4.10	องค์ประกอบของ Class Patient	44
4.11	องค์ประกอบของ Class Contact Person	45
4.12	องค์ประกอบของ Class ProgramUser	45
4.13	องค์ประกอบของ Class Queue	46
4.14	องค์ประกอบของ Class PayQueue	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่

4.15 องค์ประกอบของ Class Diagnosis 47

4.16 องค์ประกอบของ Class PrescriptionLine 48

4.17 องค์ประกอบของ Class Medicine 48

4.18 องค์ประกอบของ Class DIsease 49

4.19 Sequence Diagram แสดงกระบวนการจัดการข้อมูลคนไข้ที่มารับการรักษา 50

4.20 Sequence Diagram แสดงกระบวนการรับการตรวจจากแพทย์ 51

4.21 Sequence Diagram แสดงกระบวนการชำระเงินและรับยา 52

5.1 จอภาพแสดงการ Login เข้าสู่ระบบ 55

5.2 จอภาพหลักของระบบ 56

5.3 จอภาพแสดงข้อมูลคนไข้ แสดง Tab ของข้อมูลที่อยู่ของคนไข้ 57

5.4 จอภาพแสดงข้อมูลคนไข้ แสดง Tab ของข้อมูลญาติคนไข้ 58

5.5 จอภาพแสดงข้อมูลยา แสดง Tab ของ ข้อมูลการใช้ยา 59

5.6 จอภาพแสดงข้อมูลยา แสดง Tab ของ ข้อมูล Stock 60

5.7 จอภาพแสดงข้อมูลยา แสดง Tab ของ ข้อมูลผลิต 60

5.8 จอภาพแสดงการรับยาเข้า Stock 61

5.9 จอภาพแสดงข้อมูลโรค 62

5.10 จอภาพแสดงข้อมูลพนักงาน 63

5.11 จอภาพแสดงลำดับฉารเข้ารับการตรวจ 64

5.12 จอภาพแสดงข้อมูลการตรวจโรค แสดง Tab การตรวจเบื้องต้น 65

5.13 จอภาพแสดงข้อมูลการตรวจโรค แสดง Tab การวินิจฉัยของแพทย์ 66

5.14 จอภาพแสดงข้อมูลการตรวจโรค แสดง Tab การจ่ายยา 66

5.15 จอภาพแสดงลำดับการชำระเงิน 67

5.16 จอภาพแสดงการรับชำระเงิน 68

5.17 บัตรคนไข้ 68

5.18 รายงานสรุปปริมาณยาคงเหลือ 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 ความเป็นมาของโครงงาน

ในปัจจุบันได้มีการพัฒนาเทคโนโลยีให้มีความก้าวหน้าเพื่ออำนวยความสะดวกแก่มนุษย์มากขึ้น เทคโนโลยีคอมพิวเตอร์ เป็นส่วนหนึ่งของเทคโนโลยีที่ได้มีการพัฒนา และได้เข้ามาเป็นส่วนหนึ่งในการดำเนินชีวิตของมนุษย์ โดยได้มีการนำคอมพิวเตอร์เข้ามาใช้งานในระบบงานต่างๆ เพื่อเพิ่มประสิทธิภาพในการทำงาน ทั้งในด้านความรวดเร็วถูกต้อง แม่นยำ ลดความผิดพลาด และประหยัดงบประมาณอีกด้วย ทั้งนี้จะต้องมีโปรแกรมที่เหมาะสมกับการใช้งานในด้านต่างๆ โดยอาจจะเป็นโปรแกรมสำเร็จรูป หรือเป็นโปรแกรมที่พัฒนาขึ้นเพื่อใช้ในงานเฉพาะด้านนั้นๆ

แนวคิดการพัฒนาโปรแกรมเชิงวัตถุ เป็นเทคโนโลยีการเขียนโปรแกรมที่กำลังได้รับความสนใจอีกแนวคิดหนึ่ง เนื่องจากเป็นแนวคิดที่เพิ่มประสิทธิภาพในการเขียนโปรแกรม ลดความซ้ำซ้อน และทำให้โปรแกรมน่าเชื่อถือขึ้น โดยมีแนวความคิดที่มองระบบเป็นวัตถุ (Real World) หรือการมองทุกอย่างที่เกี่ยวข้องกับระบบเป็นวัตถุ (Object) ทั้งหมด โดยที่วัตถุจะมีคุณสมบัติ (Properties) และพฤติกรรม (Method และ Behavior) ของวัตถุนั้นๆ นอกจากนี้ระบบเชิงวัตถุยังมีคุณสมบัติที่สำคัญอีกประการหนึ่งคือ คุณสมบัติการนำกลับมาใช้ใหม่ (Reusable) ซึ่งจากคุณสมบัตินี้ทำให้การพัฒนาโปรแกรมสามารถทำได้อย่างรวดเร็วขึ้น ทั้งยังเป็นการลดค่าใช้จ่ายในการบำรุงรักษาและพัฒนาโปรแกรม

ระบบการจัดการข้อมูลภายในคลินิก เป็นระบบหนึ่งที่ควรจะนำคอมพิวเตอร์เข้ามาใช้เพื่อเพิ่มประสิทธิภาพในการทำงาน เนื่องจากระบบดังกล่าว จำเป็นที่จะต้องจัดเก็บข้อมูลเป็นจำนวนมาก ไม่ว่าจะเป็น ข้อมูลบุคลากรของคลินิก ข้อมูลประวัติคนไข้ ข้อมูลยา และ ข้อมูลการตรวจ ซึ่งข้อมูลที่กล่าวมานั้นล้วนแล้วแต่มีความสำคัญยิ่ง โดยเฉพาะข้อมูลประวัติคนไข้ และข้อมูลการตรวจ ซึ่งถ้าเก็บในรูปของเอกสาร อาจยากต่อการค้นหาและเกิดการชำรุด สูญหายได้ง่าย รวมทั้งข้อมูลคลังยาจากการที่ใช้การจดบันทึกข้อมูลการจ่ายยา ถ้าไม่มีการจัดการที่ดี อาจจะไม่ทราบได้ว่ายาชนิดใดเหลือจำนวนที่แน่นอนเท่าใด ซึ่งอาจทำให้ยาที่ต้องการจ่ายให้แก่คนไข้นั้นหมดจากคลังยา อันจะทำให้คนไข้เกิดความไม่พอใจในบริการของคลินิกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ (Information System Of Clinic Using Object-Oriented Techniques) เป็นระบบหนึ่งซึ่งนำคอมพิวเตอร์เข้ามาใช้ในการจัดการข้อมูลในด้านต่างๆ อันได้แก่ การเก็บข้อมูลบุคลากรของคลินิก ข้อมูลคนไข้ ข้อมูลยา ข้อมูลโรคและข้อมูลการตรวจ นอกจากนี้ยังช่วยในด้านการออกเอกสารต่างๆ จากเดิมซึ่งต้องบันทึกด้วยมือ ซึ่งช่วยอำนวยความสะดวกและเพิ่มประสิทธิภาพให้แก่กระบวนการทำงานภายในคลินิกมากยิ่งขึ้น รวมทั้งสร้างความประทับใจให้แก่คนไข้ที่มาใช้บริการของคลินิก นอกจากนี้ระบบดังกล่าว จะนำเอาแนวความคิดการพัฒนาโปรแกรมเชิงวัตถุมาใช้ในการพัฒนาระบบ

สำหรับระบบงานนี้ในขั้นตอนของ การวิเคราะห์และออกแบบระบบเชิงวัตถุ จะเลือกใช้ Rational Rose เป็น Case Tool, ใช้ภาษาวิชวลเบสิก (Microsoft Visual Basic 6) เป็นภาษาในการพัฒนาโปรแกรม และใช้ Cache' Object-Oriented Database เป็นฐานข้อมูลเชิงวัตถุ

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อพัฒนาระบบการจัดการข้อมูลในคลินิกให้มีประสิทธิภาพ และรวดเร็วยิ่งขึ้น
2. เพื่อช่วยในการสืบค้นข้อมูล ลดความผิดพลาดในการจัดเก็บข้อมูล
3. เพื่อตรวจสอบชนิดและปริมาณยาที่มีอยู่ และเตือนเมื่อยาเหลือจำนวนน้อยว่าจำนวนยาคงเหลือขั้นต่ำที่กำหนดไว้
4. เพื่อป้องกันปัญหายาขาดคลัง อันก่อให้เกิดปัญหายาไม่เพียงพอจ่ายให้แก่คนไข้
5. เพื่อลดปัญหาการอ่านลายมือของแพทย์ในการสั่งยาและวินิจฉัยโรค
6. เพื่อเก็บข้อมูลโรคต่างๆ

1.3 ขอบเขตของโครงงาน

ระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ เป็นการพัฒนาระบบโดยใช้แนวความคิดเชิงวัตถุ (Object-Oriented Technology) ซึ่งการทำงานของระบบจะแบ่งส่วนของการทำงานเป็นดังนี้

1. ระบบจัดการฐานข้อมูลเชิงวัตถุ เก็บรวบรวมข้อมูลต่างๆที่เกี่ยวข้องกับระบบในรูปแบบเชิงวัตถุ
2. การเข้าใช้งานระบบจะมีการกำหนดสิทธิ์ ผู้เข้าใช้งานระบบ เพื่อป้องกันไม่ให้ผู้ที่ไม่เกี่ยวข้องเข้าถึงข้อมูลต่างๆของระบบ
3. ออกรายงาน และเอกสารต่างๆ ที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 หลักการในการพัฒนาระบบงาน

ในการพัฒนาระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ ให้มีประสิทธิภาพที่ดี จำเป็นที่จะต้องศึกษาหลักการและทฤษฎีต่างๆ ดังนี้

1. ศึกษาหลักการชิงวัตถุ (Object-Oriented Technique)
2. ศึกษากระบวนการวิเคราะห์และออกแบบระบบเชิงวัตถุ โดยนำเสนอแนวความคิดในการวิเคราะห์และออกแบบระบบ ในรูปแบบของ UML (Unified Modeling Language) และใช้ Rational Rose เป็น Case Tool
3. ศึกษาวิธีการพัฒนาโปรแกรมเชิงวัตถุ โดยใช้ Microsoft Visual Basic 6
4. ศึกษาวิธีการทำงานของระบบฐานข้อมูล Cache'
5. ศึกษาการสร้างรายงานจากฐานข้อมูลเชิงวัตถุ โดยใช้ Crystal Report 6

1.1 องค์ประกอบของระบบงาน

ระบบจะประกอบไปด้วยองค์ประกอบดังนี้คือ

- เครื่องคอมพิวเตอร์ Server
  - ระบบปฏิบัติการ Windows 98 / Windows NT
  - ระบบฐานข้อมูลที่ใช้เป็น Object-Oriented Database System โดยใช้ซอฟต์แวร์การจัดการฐานข้อมูล Cache' โดยใช้เก็บฐานข้อมูลต่างๆ ภายในระบบคลินิก
- เครื่องคอมพิวเตอร์ Client
  - เป็นระบบปฏิบัติการ Windows98
  - ลงโปรแกรมระบบการจัดการข้อมูลภายในคลินิก
- เครื่องพิมพ์ เพื่อใช้ในการพิมพ์บัตรคนไข้ และรายงานต่างๆ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการพัฒนาระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ จำแนกออกเป็น 3 หัวข้อหลัก ๆ ดังนี้

1. ประโยชน์ต่อผู้ทำการพัฒนาระบบ
   - ได้พัฒนาความรู้และความสามารถในการวิเคราะห์ ออกแบบ และพัฒนาระบบด้วยหลักการแนวความคิดเชิงวัตถุ
   - เพื่อประโยชน์ต่อการทำงานในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ประโยชน์ต่อองค์กร

- ปรับปรุงการทำงานจากเดิมที่เป็น Manual โดยการนำเอาคอมพิวเตอร์เข้ามาใช้ เพื่อเพิ่มประสิทธิภาพในการทำงาน
- ลดความผิดพลาด บกพร่องในการทำงาน เช่น มีการเตือนเมื่อจำนวนยาน้อยกว่าจำนวนยาคงเหลือขั้นต่ำที่กำหนดไว้
- สร้างความน่าเชื่อถือต่อคนไข้ที่มารับการรักษาจากคลินิก
- เป็นแนวทางในการปรับปรุงขั้นตอนการทำงานในระบบเดิมให้มีประสิทธิภาพมากยิ่งขึ้น

3. ประโยชน์ทั่วไป

- สามารถนำไปศึกษาเพื่อเป็นแนวทางในการพัฒนาระบบงานอื่น ๆ ได้ต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
แนวความคิดและทฤษฎีที่เกี่ยวข้อง

2.1 แนวความคิดเชิงวัตถุ (Object-Oriented Concept)

การพัฒนาโปรแกรมเชิงวัตถุ (Object-Oriented Software Development) เป็นแนวคิดในการพัฒนาโปรแกรมรูปแบบหนึ่ง ที่ผนึกเอาข้อมูลและฟังก์ชันการทำงานเข้าด้วยกัน เรียกว่า วัตถุ (Object) และสามารถกำหนดอิน-เตอร์เฟส (Iinterface) ระหว่างวัตถุต่างๆ โดยถ้าวัตถุหนึ่งต้องการจะติดต่อกับอีกวัตถุหนึ่ง ต้องติดต่อผ่าน Interface ที่กำหนดไว้เท่านั้น ซึ่งจะติดต่อการด้วยการส่งข้อความ (Message Passing) ระหว่าง Object จากหลักการนี้จะพบว่าการเปลี่ยนแปลงแก้ไขฟังก์ชันการทำงาน และข้อมูลใดๆ ในวัตถุหนึ่ง จะมีผลกระทบต่อวัตถุอื่นน้อยมาก ซึ่งจะทำให้การแก้ไขโปรแกรมทำได้สะดวกและรวดเร็วยิ่งขึ้น เป็นการลดต้นทุนในการบำรุงรักษาโปรแกรม นอกจากนี้ยังมีความสามารถในการถ่ายทอดคุณสมบัติ (Inheritance) จากวัตถุหนึ่งไปอีกวัตถุหนึ่งได้ ทำให้สามารถนำโปรแกรม บางส่วนที่มีอยู่เดิมกลับมาใช้ใหม่ได้ โดยผู้พัฒนาไม่จำเป็นต้อง ทราบว่าโปรแกรมบางส่วนที่นำกลับมา ใช้นั้นเขียนขึ้นมาอย่างไร (How does it work?) เพียงแต่ทราบว่าโปรแกรมส่วนนั้นทำอะไร (What does it work?) และทำการเพิ่มคุณสมบัติอื่นๆ ที่ผู้พัฒนา ต้องการเข้าไป ทำให้การพัฒนาโปรแกรมใหม่ๆ ทำได้รวดเร็วขึ้น

คำว่า "Object" (Real World Object) โดยทั่วไปจะหมายถึง วัตถุใดๆจะมีสิ่งสำคัญคือ คุณสมบัติ (Properties) พฤติกรรม(Method หรือ Behavior) และ สถานะ (State) ส่วน Software Object จะพยายามเลียนแบบ Real World Object นั่นคือ การพัฒนาซอฟต์แวร์ที่จำลองโลกแห่งความเป็นจริงในมุมมองเชิงวัตถุ

2.1.1 Object Methodology

1) Objects

Object หนึ่งๆจะเป็น Collection ของ Data (Attribute หรือ properties) และ Function logic (Method) ซึ่ง data จะบอกถึงคุณสมบัติหรือสถานะของ Object และ Method จะบอกถึงพฤติกรรมต่างๆของ Object นั้นๆ โดย

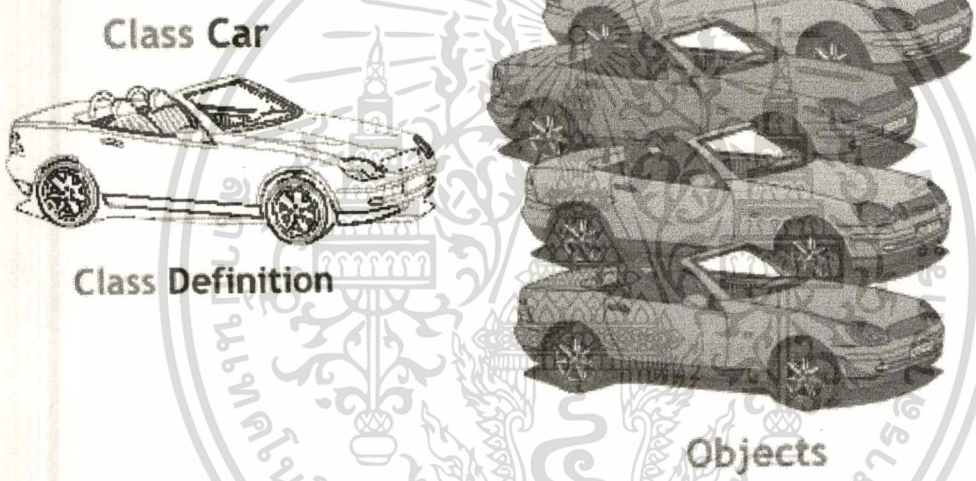
- Attribute คือข้อมูลที่เราสนใจเกี่ยวกับ Object นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Method จะแบ่งเป็น 2 ประเภทคือ Interface Method (เป็น Method ที่ถูกใช้ได้จาก Object อื่น) และ Internal Method (เป็น method ที่จะถูกเรียกใช้ได้เฉพาะภายใน Object ที่เป็นเจ้าของเท่านั้น)

2) Class

เป็นพิมพ์เขียวของ Object ที่ไม่สามารถนำมาใช้ได้โดยตรง โดยจะมีการบอกถึง Method ที่ใช้ได้โดย Object และมีการแสดง data type ที่บอกถึงสถานะของ Object โดยยังไม่ระบุค่าใน Attribute แต่ละตัวซึ่งถ้าเป็น Object จะมีการระบุค่าของ Attribute ที่แน่นอน ถ้า Object เป็น Instance ของ Class ใด Object นั้นจะต้องมี Attribute และ Method เหมือน Class ของมัน



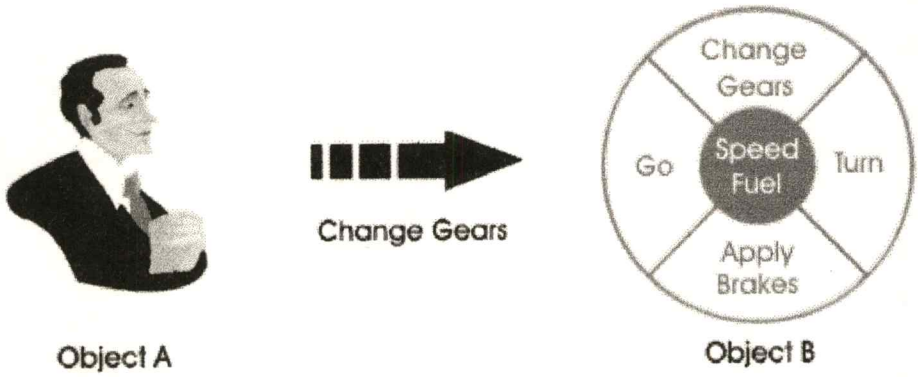
รูปที่ 2.1 ตัวอย่าง Class Car และ Object Car

3) Message Passing

คือการที่ Object ติดต่อกันด้วยการส่งข้อความ (Message) ถึงกันและกันซึ่งข้อความจะประกอบด้วยจุดหมายปลายทาง (Destination) ของข้อความนั้นและข้อมูลที่สำคัญ (Argument หรือ Parameter)

Message Passing เปรียบได้กับ Function Call หรือ Procedure Call ที่มีใน Structured Programming โดยผ่าน Interface Method ของ Object นั้นๆ มีผลทำให้Object ที่เป็นผู้รับข้อความ (Received Object) นั้นกระทำการอย่างใดอย่างหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการส่ง Message ระหว่าง Object A

2.1.2 การที่จะเป็น Object ได้ต้องมีคุณสมบัติดังนี้

1) Encapsulation/Information Hiding

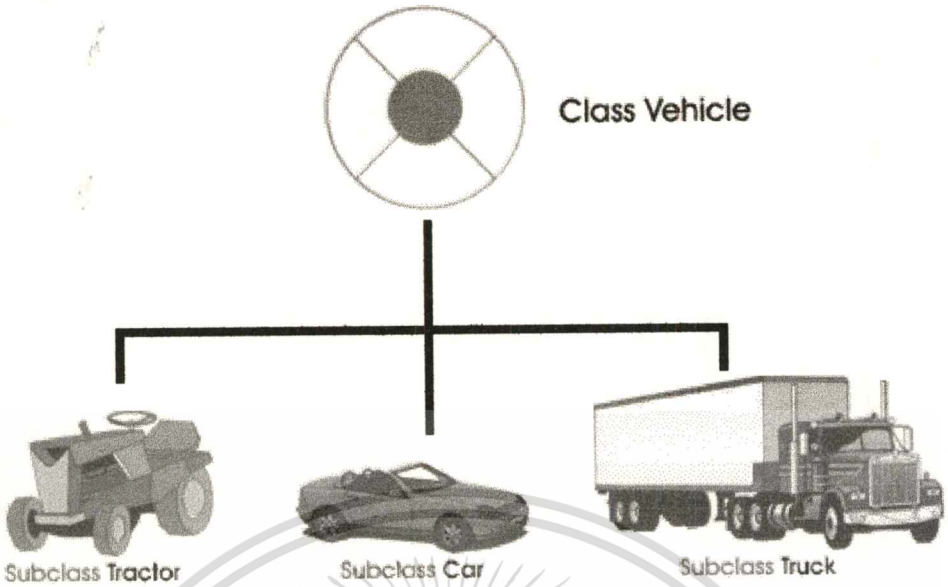
ทฤษฎีนี้ถูกคิดค้นโดย James Rambaugh ซึ่งหมายถึง การแยกลักษณะภายนอก (Interface) ซึ่งสามารถติดต่อกับ Object อื่นๆ ออกจากส่วนที่ Implement ภายในของ Object ซึ่งจะถูกใช้ได้เฉพาะตัว Object นั้นๆ มีข้อดีคือ เพิ่มความสามารถในการปกป้องข้อมูลโดยไม่ให้ Object อื่นเข้ามาทำการเปลี่ยนแปลงข้อมูลก่อนได้รับอนุญาติทั้งนี้ตัว Object ควรรู้และทำการเปลี่ยนค่าด้วยตนเองซึ่งเป็นผลให้ Maintain Object ง่ายขึ้น

2) Inheritance

คือหลักการในการที่ Object ทุกๆตัวใน Generalized Collection ได้มีการใช้ข้อมูล (Structure) และพฤติกรรม(Behavior) ร่วมกันซึ่งจะมีความสัมพันธ์แบบ IS-A Relationship โดยเรียก Class ที่อยู่เหนือกว่าว่า Parent Class ซึ่งจะถ่ายทอดคุณสมบัติทั้ง Attribute และ Method มายัง Class ที่ต่ำกว่าเรียกว่า Subclass ซึ่งจะมี Attribute และ Method เพิ่มเติมจาก Parent class มีข้อดีคือ

- เพิ่ม Consistancy เพียงแค่เปลี่ยน Method หรือ Attribute ที่ Super class ซึ่งเป็นผลให้ค่าที่ Subclass เปลี่ยนไปด้วย
- เป็นการส่งเสริมการนำ Object กลับมาใช้ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ตัวอย่างการ Inheritance ของ Class Vehicle

3) Polymorphism

อาจหมายถึง "Having Many Forms" โดยการส่ง Message เดียวกันให้กับ Object ที่ต่างกัน เป็นผลทำให้ Object แสดงพฤติกรรมที่แตกต่างกัน มีข้อดีคือ เป็นการสนับสนุนการนำโปรแกรมกลับมาใช้ใหม่และสามารถเปลี่ยนแปลงซอฟต์แวร์ได้ในระหว่างการพัฒนา

2.1.3 หลักการพื้นฐานของ Object-Oriented Programming

1) Object: Object เป็นกล่องๆ หนึ่ง ซึ่งข้างในกล่องมี Data (Instance Variable) และ Method รวมกันอยู่ภายใน และ Object นี้จะทำการรับและการส่ง Message ระหว่าง Object เพื่อทำการใช้งาน Data และ Method ที่อยู่ภายใน โดยการรับและการส่ง Message นี้จึงเปรียบเสมือน Interface ของ Object นั้นๆ ด้วย

2) Class: ไม่ใช่ Object แต่เป็นการจัดกลุ่มของ Object ตามProperties และผู้ใช้สามารถติดต่อกับคลาสโดยผ่านทาง Method ภายในClass หรือ Method ใน Super Class (Parent Class) เท่านั้น

3) Method: เป็นการทำงาน (Operation) ที่สำคัญหรือเป็นวิธีการกระทำที่สามารถทำงานกับ Object ได้ แต่ละคลาสจะมี Method ของตัวเอง โดยการทำงาน จะเริ่มจากการส่ง message ไปยัง Object ที่ต้องการ (Receiver Object) เพื่อเรียกใช้ Method ที่อยู่ภายใน Object นั้น

4) Encapsulation: คือ การรวมกันของโครงสร้างข้อมูล (Data Structure) กับฟังก์ชัน (Method, Action) เกิดเป็นวัตถุใหม่ที่มีความสามารถในการซ่อนข้อมูลจากระบบภายนอกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(เหมือนเป็นยาเม็ดแคปซูลที่เราทานแก้หวัดซึ่งจะมองไม่เห็นยาภายในรู้แต่เพียงว่ายาเม็ดนี้ใช้รักษาโรคอะไรเท่านั้น) ทำให้ข้อมูลมีความมั่นคงขึ้น

5) Inheritance: คือ การสร้างคลาสใหม่ขึ้นมา โดยมีการสืบทอดคุณสมบัติพื้นฐานที่มาจากคลาสเดิม แต่จะมี ข้อมูล หรือ Method เพิ่มขึ้นมาจากคลาสเดิม ถ้าคลาส B เป็นSub Class ของ Class A ดังนั้น Object ที่เป็น Instant ของ Class B จะมีคุณสมบัติพิเศษกว่า Instant ของ Class A แต่อย่างน้อยที่สุด จะต้องมีคุณสมบัติเหมือน Instant ของ Class A

การถ่ายทอดคุณสมบัติ นี้จะรวมถึง Data และ Method ของ Class A และClass A จะถูกเรียกว่าเป็น Super Class ของ Class B

6) Polymorphism: คือ การที่เราส่ง Message ที่เหมือนกัน ไปใน Object ที่ต่างกัน แต่ละ Object จะตอบสนองออกมาไม่เหมือนกัน ตามแต่ชนิดและหน้าที่ของ Object ความสามารถที่ใช้ Message เหมือนกัน สำหรับการกระทำที่เหมือนกัน ไปยัง Object ต่างชนิดกัน มีลักษณะเหมือนกับการที่มนุษย์คิดในการแก้ปัญหาหนึ่ง ๆ

ตัวอย่าง การ Polymorphism

- AB + CD ได้ผลลัพธ์ABCD(Concat)
- 5 + 3 ได้ผลลัพธ์ 8 (Add)

ทั้ง 2 แบบ เป็นการส่ง Message '+' เข้าไปยัง Object ภายใน Class String และ Integer ตามลำดับ

7) Dynamic Binding: คือ การนำโปรแกรมย่อยๆ มาประกอบให้ใช้งานในขณะ Run Time โดยในขณะ Compile Time นั้นจะเก็บโปรแกรมในรูปแบบของ Class ต่างๆ ไว้ เพื่อไม่ให้เกิดความยุ่งยาก ซับซ้อน ต่อจากนั้น เมื่อนำมาใช้ในขณะ Run Time เมื่อมีการเรียกใช้ Class นั้นๆ จะทำการนำ Class นั้นมาไว้ในส่วนของโปรแกรม และเมื่อใช้งานเสร็จแล้วจะถูกลบออกจากหน่วยความจำ

8) Override: คือ การที่เราสร้าง Subclass ขึ้นมาใหม่ และมีการสร้าง Method ที่ซ้ำกับ Method ที่เป็นของ Super Class

9) Overload: คือ การที่เราสร้าง Method ชื่อเหมือนกัน แต่รับ Parameter ต่างกัน ภายใน Class เดียวกัน

2.2 ภาษาแบบจำลองสำหรับการพัฒนาโปรแกรมเชิงวัตถุ (Unified Modeling Language)

การที่จะนำวิธีการพัฒนาโปรแกรมเชิงวัตถุมาใช้งานได้อย่างมีประสิทธิภาพนั้น (Object-Oriented Programming) จะต้องมีกระบวนการวิเคราะห์และออกแบบระบบเชิงวัตถุ (Object-Oriented Analysis and Design: OOAD) เพื่อที่จะสามารถพัฒนาโปรแกรมได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังสามารถเอื้ออำนวยให้มีการพัฒนาโปรแกรมได้อย่างรวดเร็ว และง่ายต่อการบำรุงรักษา จึงจำเป็นต้องกำหนดภาษาในการสร้างแบบจำลองเชิงวัตถุ

เนื่องจากแบบจำลองอื่นๆ เช่น Data Flow Diagram, Context Diagram เป็นแบบจำลองที่นำเสนอในรูปแบบของการส่งผ่านการทำงานภายในระบบ ซึ่งเป็นแบบจำลองที่เหมาะสมในการนำเสนอแนวความคิดการออกแบบระบบเชิงโครงสร้างมากกว่าระบบเชิงวัตถุ ประกอบกับแนวคิดการวิเคราะห์และออกแบบระบบเชิงวัตถุเป็นที่นิยมเพิ่มขึ้นอย่างแพร่หลาย จึงทำให้ภาษาแบบจำลองเชิงวัตถุ มีมากมายแตกต่างกันไป และผู้ใช้จำนวนมากที่ใช้วิธีดังกล่าวก็ยังไม่สามารถพบวิธีที่ตนเองพอใจได้ ดังนั้นจึงต้องมีการคิดค้นภาษารูปแบบจำลองขึ้นมาเป็นมาตรฐาน ได้แก่ แบบจำลอง Unified Modeling Language (UML)

2.2.1 ความเป็นมาของ UML

UML เป็นภาษาภาพเพื่อเป็นสื่อแสดงถึง ความคิด,ระบุรายละเอียด และ วางโครงสร้าง ในการออกแบบระบบเชิงวัตถุ เริ่มมีการคิดค้นเมื่อปลายศตวรรษที่1980 โดย Grandy Booch, James Rumbaugh และ Ivar jarcobson ซึ่ง UML1.0 เป็นเวอร์ชันที่สามารถใช้ได้และมีเสถียรภาพ ได้รับการรับรองจาก Object Management Group (OMG) เมื่อเดือนมกราคม 1997 ว่าเป็นภาษาสากลที่เปรียบเสมือนภาพพิมพ์เขียวของโปรแกรม หลายๆองค์กรได้มีการสนับสนุนมาตรฐานของ UML ตั้งแต่พื้นฐานรูปแบบทางภาษา (Modeling Language) ซึ่งเป็นวิธีการเชิงวัตถุ ในปัจจุบันเวอร์ชั่นล่าสุดของ UML คือ UML 1.3 ซึ่งได้มีการพัฒนา UML เพื่อแก้ไขข้อผิดพลาด และเพิ่มเติมในส่วนที่ขาดหายไป ให้ตรงกับความต้องการของผู้ใช้งานมากที่สุด

สิ่งที่ทำให้ UML ประสบความสำเร็จคือ

1. UML เป็นสิ่งที่ได้ผลในหลายๆวิธี ที่เป็นภาษาแบบจำลองเชิงวัตถุที่ได้เสนอมาในอดีต เนื่องจากเป็นรูปแบบที่แสดงให้เห็นถึงการวิเคราะห์และออกแบบระบบเชิงวัตถุอย่างชัดเจนและ เป็นขั้นตอน

2. UML เป็นการรวมรูปแบบที่ใช้ได้ในหลายๆระบบ (Business Versus Software) ,ขั้นตอนการพัฒนาระบบ (Development Phase)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 แบบจำลอง ต่างๆ ใน UML

มีแบบจำลอง(Diagrams) ต่างๆ ดังนี้

1. Use Case Diagram
2. Class Diagram
3. Object Diagram
4. State Diagram
5. Sequence Diagram
6. Collaboration Diagram
7. Activity Diagram
8. Component Diagram
9. Deployment Diagram

เราสามารถแบ่งแบบจำลอง ใน UML ออกเป็น Model ต่างได้ดังนี้

1. **Functional Model**

เป็น Model ที่ใช้แสดงความต้องการของระบบทั้งหมด ช่วยในการอธิบายรายละเอียดหลักๆภายใน Object แสดงถึงการส่งผ่านข้อมูลในการทำงาน โดยจะสนใจเพียงแค่ว่าต้องทำงานอะไรบ้าง แต่ยังไม่ต้องสนใจว่าจะทำงานนั้นอย่างไร โดยแบบจำลองที่จะนำเสนอความต้องการของระบบทั้งหมดคือ Use Case Diagram โดยจะถูกนำไปในการวิเคราะห์และออกแบบระบบใน Phase ต่อๆไป

2. **Object Model**

เป็น Model ที่ใช้แสดงโครงสร้างของระบบ ดังนั้นจึงเป็นModel ที่คงที่ จึงสามารถเรียกได้อีกอย่างหนึ่งว่า Static Model ซึ่งพิจารณาจากความต้องการของระบบจาก Functional Model นำเสนอในรูปของ Class ต่างๆ โดยจะนำเสนอด้วยแบบจำลอง Class Diagram และ Component Diagram

3. **Dynamic Model**

เป็น Model ที่ใช้แสดงถึงการทำงานระหว่าง วัตถุ(Object) ต่างๆ ตามการส่งข้อความ (Message) หรือเมื่อมีเหตุการณ์ (Event) ต่างๆเกิดขึ้น โดย Object ในที่นี้หมายถึงตัวอย่าง (Instance) ของ Class ที่ได้ออกแบบไว้ใน Class Model ดังนั้น Object จะมีคุณลักษณะ (Attribute) และ พฤติกรรม(Method) เช่นเดียวกับ Class ที่เป็นต้นแบบของ Object นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของระบบเกิดจากการส่งMessage ระหว่าง Object เหล่านั้น เมื่อมีการทำงาน Object จะเปลี่ยนสถานะ (State) ไปตามเหตุการณ์ที่เกิดขึ้นได้ เพื่อให้เป็นไปตามที่กำหนดไว้ใน Functional Modelซึ่งแบบจำลองที่จะใช้นำเสนอคือ Sequence Diagram และ State Diagram

2.2.3 ความหมายของแบบจำลองต่างๆใน UML

2.2.3.1 แบบจำลอง Use Case (Use Case Diagram)

เป็นแบบจำลองที่แสดงถึงความสัมพันธ์ ระหว่างผู้ใช้ กับระบบต่างๆ โดยนำเสนอถึงความต้องการของระบบทั้งหมดในรูปแบบที่เข้าใจได้ง่าย

1) Actor ใช้สัญลักษณ์รูปคน แสดงถึงบทบาท(Roles) ที่ผู้ใช้จะเกี่ยวข้องกับระบบ ซึ่งผู้ใช้แต่ละคนอาจจะมีได้หลายบทบาท แต่ละบทบาทจะมีความสัมพันธ์กับกิจกรรม(Use Case)ได้เพียงหนึ่งกิจกรรมเท่านั้น

2) Use Case ใช้สัญลักษณ์ วงรี แสดงถึง แต่ละกิจกรรมที่ทำในระบบ ซึ่งจะเกิดจากการแลกเปลี่ยนข้อความ (Message)ระหว่างผู้ใช้กับระบบ หรือระหว่างระบบด้วยกันเอง

3) ความสัมพันธ์ระหว่าง Actor กับ Use Caseหรือ ความสัมพันธ์ระหว่าง Use Case ด้วยกันเอง ใช้สัญลักษณ์ลูกศรในการเชื่อมโยงโดยหัวลูกศรจะชี้ตามทิศทางของการส่งข้อความ(Message) โดยมีลักษณะความสัมพันธ์คือ

- Association คือการเชื่อมโยงความสัมพันธ์ระหว่าง Actor กับ Use Case ซึ่งแสดงถึงการติดต่อสื่อสารกัน อาจจะเป็นทางเดียวหรือ สองทางก็ได้ โดยหัวลูกศรแสดงถึงทิศทางการส่งข้อความ
สัญลักษณ์ที่ใช้คือ

- Extend คือการเชื่อมโยงความสัมพันธ์ระหว่าง Use Case ด้วยกันเอง โดยความสัมพันธ์จะเป็นแบบการขยายหรือเพิ่มการทำงานให้กับ Use Case หัวลูกศรจะชี้ไปที่ Use Case ที่ถูกเรียกใช้งานไปยัง Use Case ที่ต้องการเพิ่มเติมการทำงาน และมีข้อความ <<extend>> ระบุอยู่เหนือเส้น
สัญลักษณ์ที่ใช้คือ

- Generization คือความสัมพันธ์ระหว่าง Use Case ด้วยกันเอง โดยเป็นความสัมพันธ์ในลักษณะการกำหนดขอบเขต เช่น Use Case A เป็นตัวกำหนดขอบเขต Use Case B หัวลูกศรจะชี้จาก Use Case A ไปที่ Use Case B
สัญลักษณ์ที่ใช้คือ

- Include คือความสัมพันธ์ระหว่าง Use Case ด้วยกันเอง โดยจะเป็นการถ่ายทอดคุณสมบัติ

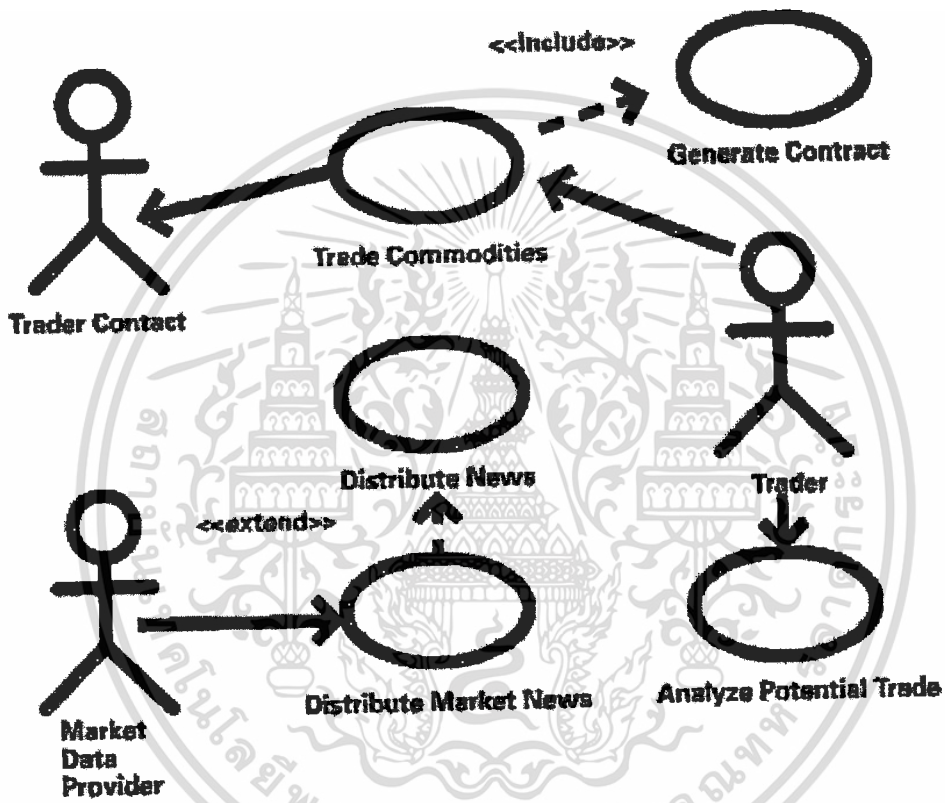
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Inherit) จาก Use Case หนึ่ง ไปยังอีก Use Case หนึ่ง ซึ่งหมายถึงความสัมพันธ์แบบ Uses ใน UML เวอร์ชัน 1.1 โดยหัวลูกศรจะชี้ไปยัง Use Case หลักที่จะถูกถ่ายทอดคุณสมบัติออกไป และมีข้อความ <<include>> อยู่เหนือเส้น

สัญลักษณ์ที่ใช้คือ ---------->

4) ความสัมพันธ์ระหว่าง Actor ด้วยกันเอง มีลักษณะความสัมพันธ์คือ

- Generalization คือ ความสัมพันธ์ระหว่าง Actor ด้วยกันเองโดยเป็นความสัมพันธ์ในลักษณะที่ Actor หนึ่ง สามารถสื่อสารกับ Use Case ได้ในลักษณะเดียวกันกับอีก Actor หนึ่ง



รูปที่ 2.4 ตัวอย่างแบบจำลอง Use Case

(Rational Software Corporation. 1999)

2.2.3.2 แบบจำลอง Class (Class Diagram)

จัดว่าเป็นแบบจำลองที่สำคัญที่สุดในการออกแบบระบบเชิงวัตถุ แสดงให้เห็นถึง

1) คุณลักษณะ (Attribute) คือ ข้อมูล(Data) หรือ ตัวแปร(Variable)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) หน้าที่การทำงาน (Operation) คือ วิธีการ (Method) หรือ พฤติกรรม (Behavior) ภายใน Class และ

3) ความสัมพันธ์ของ Class ต่างๆ ภายในระบบ

สัญลักษณ์ของ Class ที่ใช้คือ รูปสี่เหลี่ยมซึ่งอธิบายถึงกลุ่ม Object ต่างๆ โดยมีรายละเอียดพื้นฐาน 3 ส่วนคือ

Name Compartment: แสดงชื่อ Class ภายในระบบ

Attribute: แสดงถึงโครงสร้างและคุณลักษณะทั้งหมดที่มีภายใน Class ซึ่งภายใน Class เดียวกันจะไม่สามารถมีคุณลักษณะที่ซ้ำกันได้ แต่ถ้าต่าง Class กันจะสามารถมีคุณลักษณะที่ซ้ำกับคุณลักษณะใน Class อื่นได้

Operation: เป็นส่วนที่ใช้อธิบายถึงวิธีการ(Method) หรือพฤติกรรม(Behavior) ภายใน Class รวมทั้งการรับค่าตัวแปร(Argument) หรือส่งค่าตัวแปรออกไป

การเชื่อมโยงความสัมพันธ์ภายใน Class จะนำเสนอโดย

1) บทบาทของความสัมพันธ์ (Role Names) แสดงถึงบทบาทที่ Class หนึ่ง สัมพันธ์กับอีก Class หนึ่ง ตัวอย่างเช่น จากรูปที่ 2 Class Trade Administrator มีความสัมพันธ์กับ Class Trade โดยมีบทบาทคือ เป็น Manager

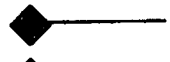
2) รูปแบบของความสัมพันธ์ แบ่งออกเป็น 4 แบบ คือ

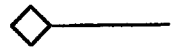
- Association: เป็นการแสดงความสัมพันธ์ระหว่าง Class โดยอาจจะเป็นทางเดียวหรือ สองทาง สัญลักษณ์ที่ใช้ มีดังนี้

ความสัมพันธ์แบบทางเดียว

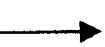
ความสัมพันธ์แบบสองทาง

- Aggregation : เป็นรูปแบบพิเศษของ Association คือเป็นความสัมพันธ์ที่แสดงว่า Class หนึ่ง เป็น ส่วนหนึ่งของอีก Class หนึ่งเช่น จากรูปที่ 2 Class Administrator Position(ตำแหน่งผู้จัดการฝ่ายขาย) เป็นส่วนหนึ่งของ Class Slate(ผู้ที่ได้รับการเสนอชื่อ) สัญลักษณ์ที่ใช้ มีดังนี้

ความสัมพันธ์แบบ By Value 

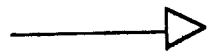
ความสัมพันธ์แบบ By Reference 

-Depends on : เป็นรูปแบบความสัมพันธ์แบบหนึ่งที่ใช้แสดงความสัมพันธ์ระหว่าง 2 Class ในแง่ที่Class หนึ่งเรียกใช้บริการของอีก Class หนึ่ง กล่าวคือClass ของผู้ขอบริการ ขึ้นอยู่กับบริการของ Class ของผู้ให้บริการ แต่ไม่มีการขึ้นต่อกันภายในโครงสร้างของ Class

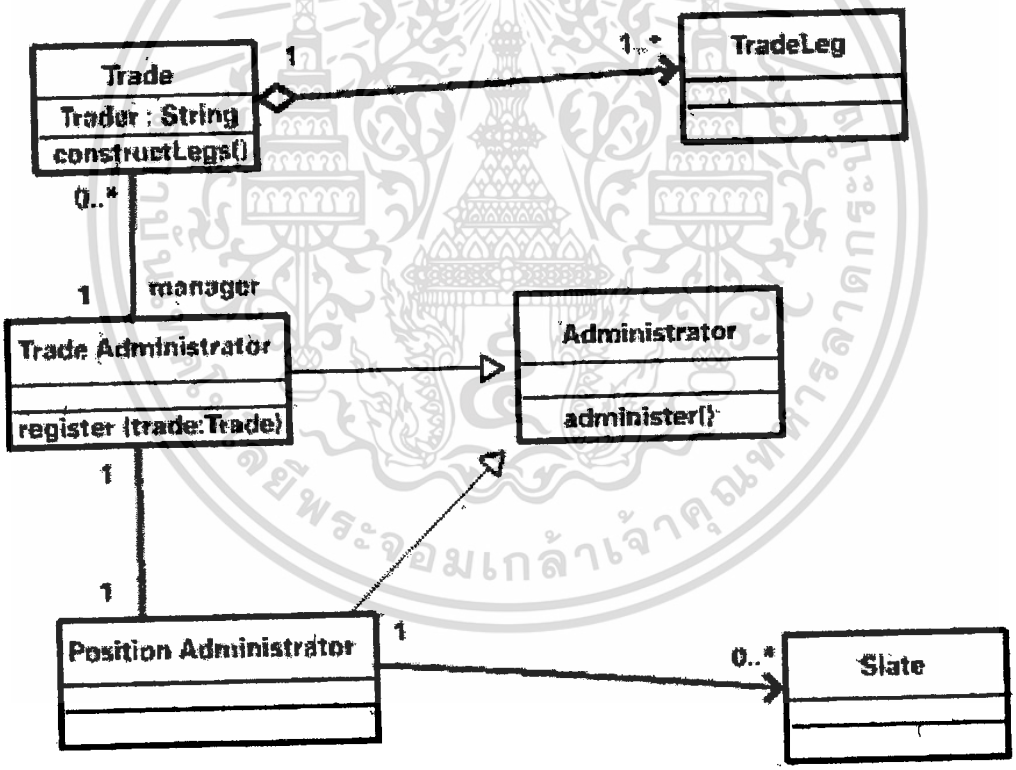
สัญลักษณ์ที่ใช้คือ 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Generalization : เป็นความสัมพันธ์ระหว่าง Class กับ Class ในแง่ของการถ่ายทอดคุณสมบัติ(Inherit) จาก Class หนึ่งไปยังอีก Class หนึ่ง โดย Class ที่เป็นต้นแบบเรียกว่า Super Class และ Class ที่ได้รับการถ่ายทอด เรียกว่า Sub Class เช่น จากรูปที่ 2 Class Trade Administrator และ Class Position Administrator เป็นClass ที่ได้รับการถ่ายทอดคุณสมบัติจาก Class Administrator โดยหัวลูกศรชี้ไปที่ Class ต้นแบบ

สัญลักษณ์ที่ใช้คือ 

3) จำนวนความสัมพันธ์ (Multiplicity) แต่ละ Class จะมีความสัมพันธ์ระหว่างกันโดยสามารถกำหนดจำนวนความสัมพันธ์เป็นตัวเลข หรือช่วงของตัวเลข ในรูปแบบค่าต่ำสุด...ค่าสูงสุด ไว้ที่ปลายของเส้นแสดงความสัมพันธ์ ถ้าเป็นเลขจำนวนเดียว หมายถึงค่าที่แน่นอน ถ้าเป็นช่วงหมายถึงค่าที่เป็นไปได้ ถ้าหมายถึงจำนวนใดก็ได้จะใช้เครื่องหมายดอกจัน (*) เช่น จากรูปที่ 2 Class Trade มีความสัมพันธ์กับ Class Trade Administrator ซึ่งมีบทบาทเป็น Manager โดยที่ 1 Trade Administrator จะทำหน้าที่เป็น Manager แก่ Class Trade เป็นจำนวน 0 หรือมากกว่า



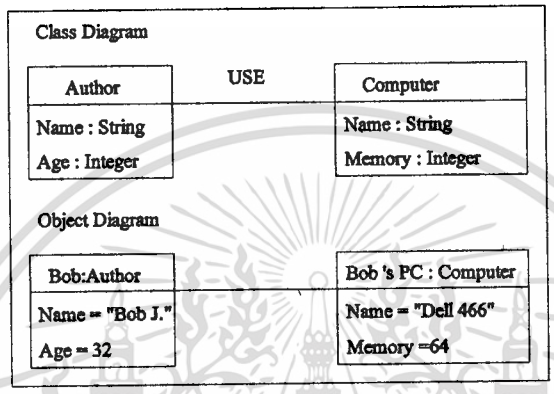
รูปที่ 2.5 ตัวอย่างแบบจำลอง Class

(Rational Software Corporation. 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.3 แบบจำลอง Object (Object Diagram)

เป็นแบบจำลองที่ขึ้นอยู่กับ Class Diagram ที่มีโครงสร้างมาก่อนแล้ว โดย Object Diagram จะนำเสนอในเชิงการบันทึกข้อมูลตามโครงสร้างของ Class Diagram ทำให้เราสามารถมองเห็นภาพรวมอย่างกว้างๆของ Class Diagram ได้

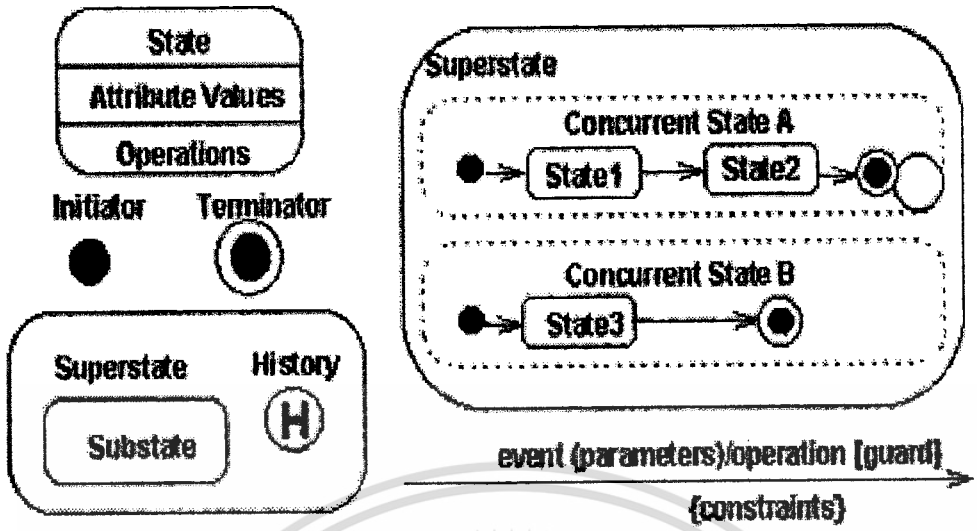


รูปที่ 2.6 ตัวอย่างแบบจำลอง Object

2.2.3.4 แบบจำลอง State (State Diagram)

เป็นแบบจำลองที่อธิบายถึงวงจรชีวิต(Life Cycle)ของ Object ซึ่งจะเป็นตัวบอกว่าเกิดอะไรขึ้นบ้าง State diagram จะเป็นตัวเชื่อมทุก Class ที่มีพฤติกรรมซับซ้อนให้ชัดเจนขึ้นโดยมีสัญลักษณ์ต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 สัญลักษณ์แบบจำลอง State

(Rational Software Corporation. 1999)

State Diagram จะมีส่วนประกอบอยู่ 3 ส่วน

-State 's Name : ชื่อของ State

-Optional State : เป็นคุณสมบัติของคลาสใน State Diagram

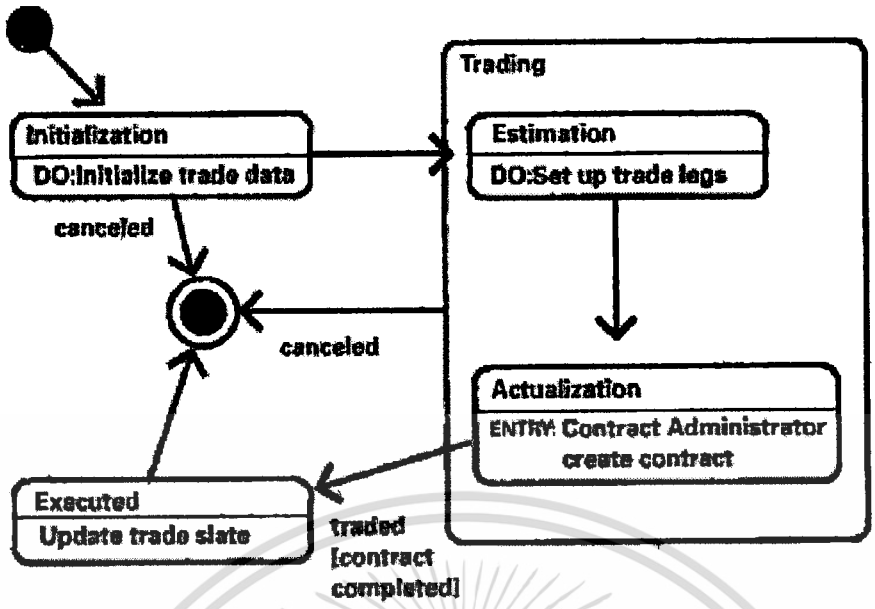
-Optional Activity State จะเป็นรายละเอียดของเหตุการณ์และกิจกรรม (Action) ต่างๆ State ที่ใช้เป็นมาตราฐาน ในกิจกรรมต่างๆมี 3 สถานะคือ

-Entry คือการเข้าไปทำกิจกรรมต่างๆภายในState

-Exit จะใช้ในกรณีที่ต้องการออกจาก State

-Do ใช้ในการปฏิบัติงานต่างๆใน State เช่น การส่งข้อความ (message), การรอหรือการคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 ตัวอย่างแบบจำลอง State

(Rational Software Corporation. 1999)

2.2.3.5 แบบจำลอง Sequence (Sequence Diagram)

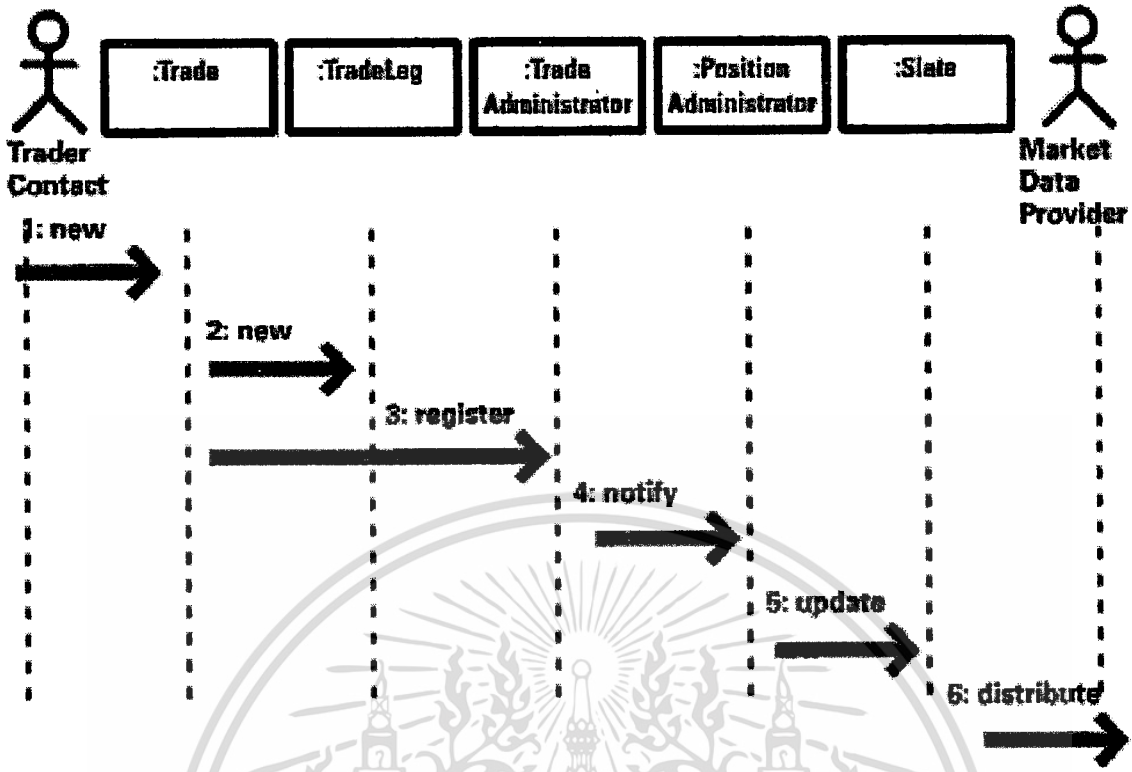
เป็นแบบจำลองที่แสดงให้เห็นถึงการทำงานระหว่างObject ต่างๆตามการส่งข้อความ และเมื่อเหตุการณ์ต่างๆได้เกิดขึ้น ซึ่งผู้เขียนโปรแกรมจะใช้Diagram ตัวนี้เป็นตัวช่วยเพื่อที่จะได้เขียนโปรแกรมให้ได้ตามที่ได้ออกแบบไว้ สัญลักษณ์ที่ใช้คือ

Object

ช่วงเวลา

ข้อความระหว่าง Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ตัวอย่าง แบบจำลอง Sequence

(Rational Software Corporation. 1999)

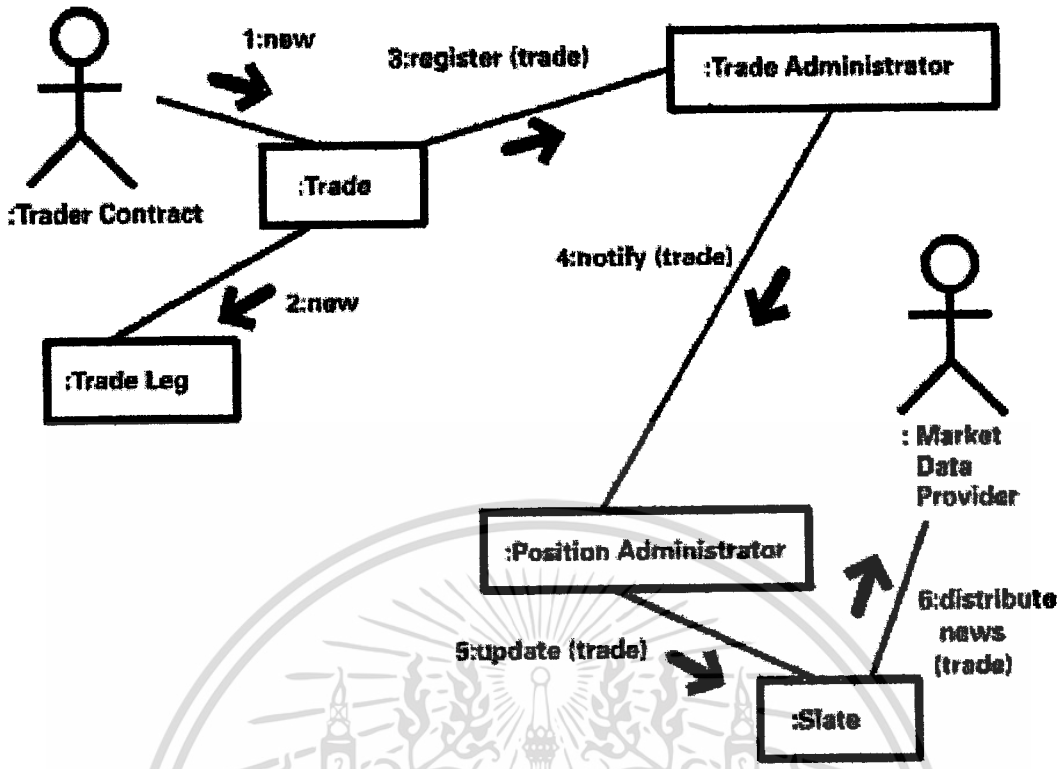
2.2.3.6 แบบจำลอง Collaboration (Collaboration Diagram)

เป็นแบบจำลองแสดงการติดต่อสื่อสารระหว่างObject ต่างๆที่เกี่ยวข้องกับแต่ละ Object ที่สนใจ รวมทั้งความสัมพันธ์ระหว่างแต่ละ Object ในการติดต่อสื่อสารด้วย สัญลักษณ์ที่ใช้จะเหมือนกับSequence Diagram ซึ่งสามารถเลือกแสดงโดยแบบจำลองอันใดอันหนึ่งได้ กล่าวคือทั้งสองแบบสามารถปรับเปลี่ยนรูปแบบระหว่างกันได้ มีหลักเกณฑ์ในการพิจารณาเลือกว่าจะใช้แบบจำลองใดดังต่อไปนี้

1) ถ้าเป็นการกำหนดช่วงเวลาที่แน่นอนและใช้เวลาเป็นสิ่งสำคัญควรเลือกใช้ Sequence Diagram

2) ถ้าให้ความสำคัญกับ Context แล้ว ควรเลือกใช้ Collaboration Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แบบจำลองCollaboration(Collaboration Diagram)

(Rational Software Corporation. 1999)

2.2.3.7 แบบจำลอง Activity (Activity Diagram)

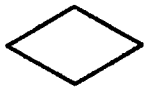
เป็นแบบจำลองที่อธิบายถึงกิจกรรม (Action) และผลลัพธ์ของกิจกรรมนั้น เพื่อแสดงถึงความเปลี่ยนแปลงที่เกิดจากการประมวลผลภายใน ซึ่งสถานะการทำงานนั้นเมื่อเริ่มต้นทำก็ต้องทำให้เสร็จก่อนที่จะเริ่มทำงานในสถานะต่อไปได้ ดังนั้นสิ่งที่จะใช้ในการควบคุมการทำงานในขั้นตอนต่างๆ ก็คือการติดต่อสื่อสารกันหรือการเชื่อมโยงกันของ State ต่างๆ ซึ่งสามารถกระทำได้ทั้งการส่งและรับข้อมูลข่าวสารได้พร้อมๆ กัน โดยนำมาใช้ประโยชน์ในการเชื่อมต่อกับ flow และรายละเอียดของพฤติกรรมที่มีการประมวลผลแบบขนาน(Parallel) คือสามารถทำงานได้หลายอย่างในเวลาเดียวกันได้เป็นจำนวนมาก ลักษณะของ Activity Diagram จะคล้ายกับ State Diagram แต่ Activity Diagram จะใช้แสดงเหตุการณ์ต่างๆอันเป็นผลมาจากการกระทำกิจกรรมภายในระบบเอง ในขณะที่ State Diagram จะใช้ในการแสดงการกระทำที่เกิดขึ้นในลักษณะที่ไม่ต่อเนื่องกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์ที่ใช้ใน Activity Diagram

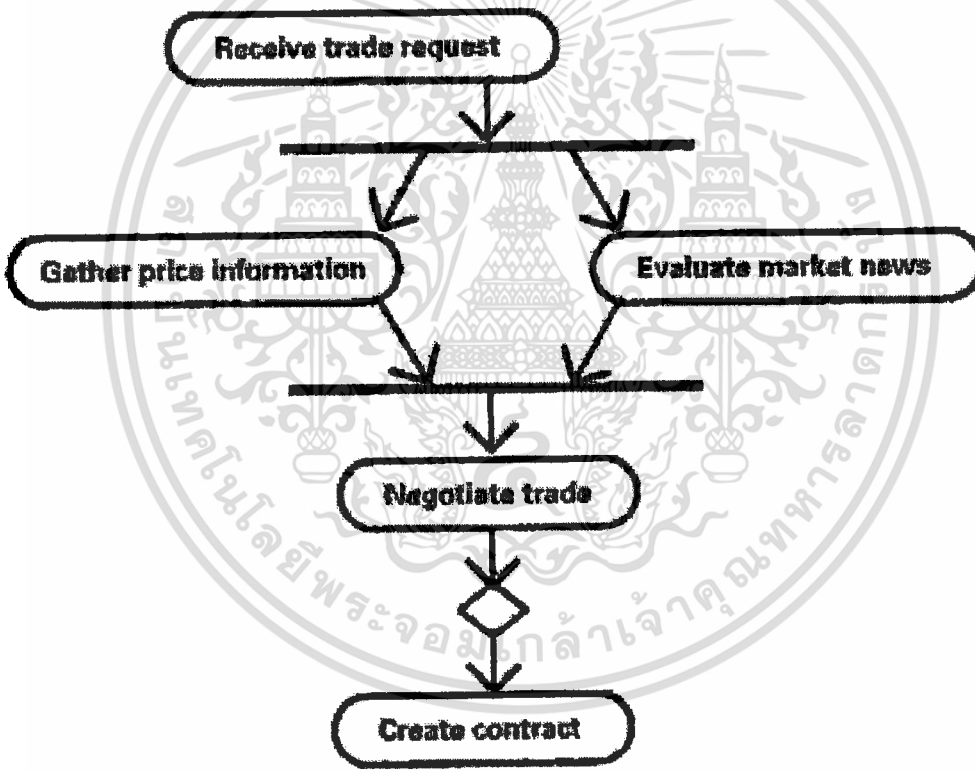
- กิจกรรม (Action)

- การตัดสินใจ(Decision)



- ลู่ขอบเขตความรับผิดชอบ (Swimlane)

- จุดแยกให้กิจกรรมก่อนหน้าเสร็จก่อน (Sync States)



รูปที่ 2.11 ตัวอย่างแบบจำลอง Activity

(Rational Software Corporation. 1999)

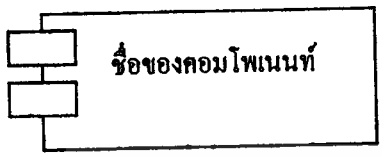
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.8 แบบจำลอง Component (Component Diagram)

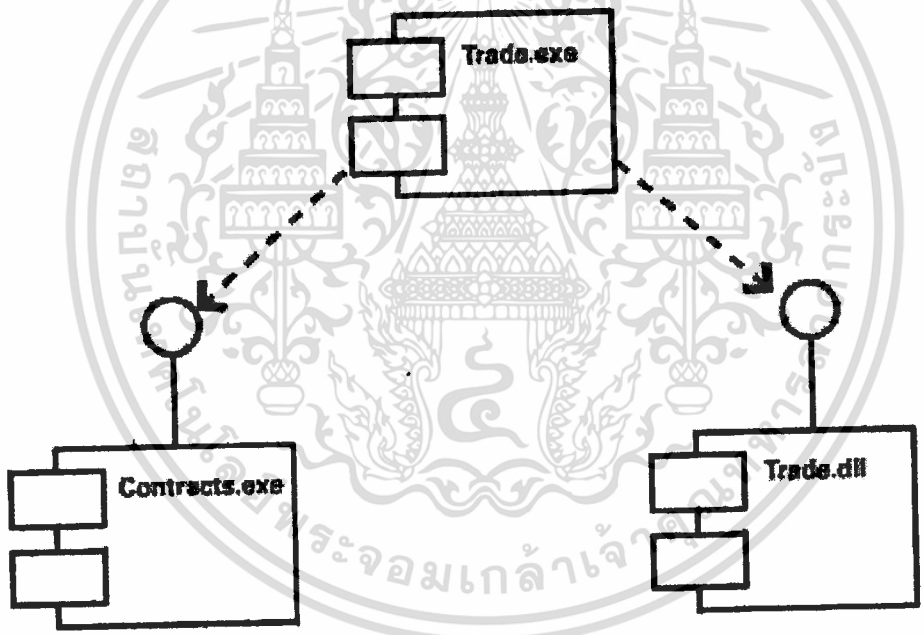
เป็นแบบจำลองที่แสดงถึงโครงสร้างภายนอกและความสัมพันธ์ระหว่างองค์ประกอบต่างๆ ของโปรแกรม File ที่มีในระบบในสภาวะแวดล้อมของการพัฒนาระบบ

สัญลักษณ์ที่ใช้ใน Component Diagram

- คอมโพเนนท์ต่าง ๆ ของซอฟแวร์



- อินเตอร์เฟซ (Interface)



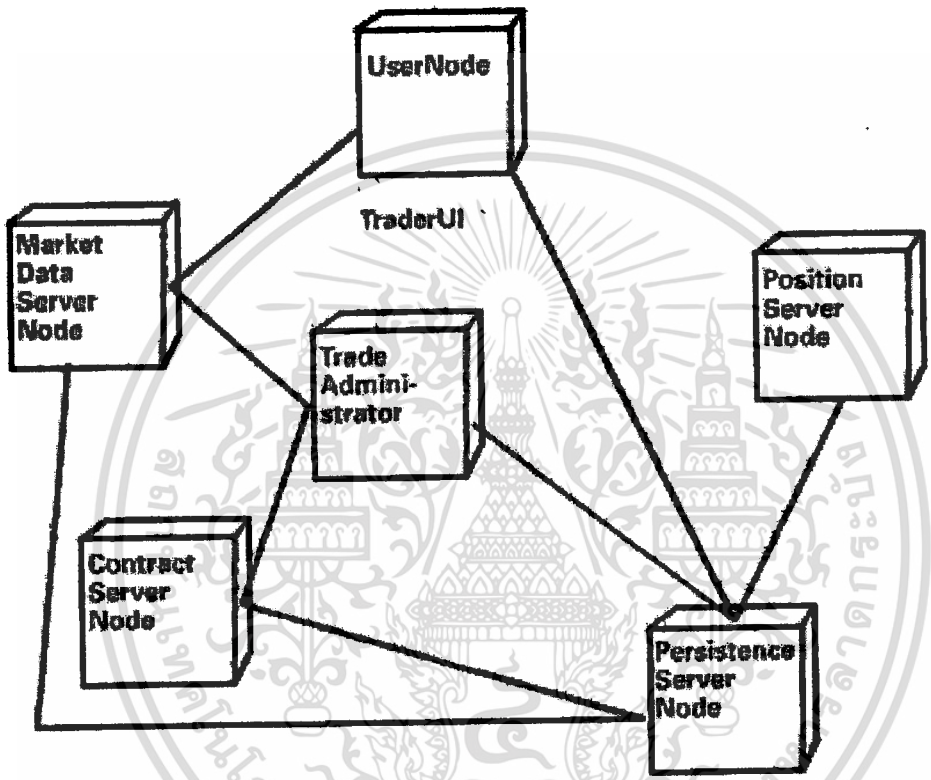
รูปที่ 2.12 ตัวอย่างแบบจำลอง Component

(Rational Software Corporation. 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.9 แบบจำลอง Deployment (Deployment Diagram)

เป็นแบบจำลองที่แสดงถึงส่วนประกอบต่างหรือสถาปัตยกรรมของระบบขณะโปรแกรมทำงาน รวมถึงโปรโตคอล (Protocol) ในการเชื่อมต่อระหว่างโนด (Node) โดยที่ Node แสดงถึงหน่วยประมวลผล อันได้แก่อุปกรณ์คำนวณต่างๆ หรือมนุษย์ รวมถึงอุปกรณ์ต่างๆ ที่ใช้ในการประมวลผลด้วย



รูปที่ 2.13 ตัวอย่างแบบจำลอง Deployment

(Rational Software Corporation. 1999)

2.3 การวิเคราะห์และออกแบบระบบเชิงวัตถุ

จุดประสงค์ที่สำคัญในการวิเคราะห์และออกแบบระบบเชิงวัตถุ คือ การที่จะมองระบบเป็นวัตถุให้ใกล้เคียงกับวัตถุในโลกแห่งความเป็นจริง เพื่อที่จะนำมาแปลงเป็นแบบจำลองของระบบซึ่งจะเป็นแนวทางในการพัฒนาโปรแกรมต่อไป โดยแบบจำลองที่ได้จะเสมือนการสื่อสารระหว่างสิ่งที่ผู้วิเคราะห์ระบบ(System Analyst) มีความเข้าใจระบบตรงกับผู้ใช้งานระบบ (User) หรือไม่ อย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และปรับปรุงให้มีความเข้าใจตรงกัน จากนั้นแบบจำลองนี้จะนำเสนอแนวความคิดของผู้ออกแบบระบบ เพื่อให้ผู้พัฒนาระบบ (Programmer) สามารถดำเนินการพัฒนาระบบได้ต่อไป

การวิเคราะห์ระบบเชิงโครงสร้าง(Structure Analysis and Design) จะนำเสนอโดย Entity Relationship Diagram (ERDs) และ Data Flow Diagram (DFDs) ซึ่งในส่วนของ DFDs จะเป็นแบบจำลองที่นำเสนอถึงกระบวนการส่งผ่านข้อมูลภายในระบบ และ ERDs จะเป็นแบบจำลองที่แสดงถึงโครงสร้างของข้อมูลหรือสิ่งต่างๆภายในระบบ โดยที่ DFDs กับ ERDs จะเชื่อมโยงกันโดยแฟ้มข้อมูล (Data Store) จาก DFDs โดยที่หลายๆกระบวนการใน DFDs อาจเกี่ยวข้องกับ Data Store เดียวกัน ซึ่งก่อให้เกิดความซ้ำซ้อน นอกจากนี้การวิเคราะห์และออกแบบระบบเชิงโครงสร้าง ไม่สามารถนำเสนอถึงคุณสมบัติเชิงวัตถุ เช่น การนำกลับมาใช้ใหม่ เป็นต้น

จากข้อจำกัดของการวิเคราะห์และออกแบบระบบเชิงโครงสร้างที่ได้กล่าวมาแล้วข้างต้น จะเห็นว่าการวิเคราะห์และออกแบบระบบเชิงวัตถุจึงเป็นวิธีการที่เหมาะสมกว่าในการวิเคราะห์และออกแบบระบบเพื่อพัฒนาระบบเชิงวัตถุ กล่าวคือ การวิเคราะห์และออกแบบระบบเชิงวัตถุจะนำเสนอในรูปแบบที่เสมือนโลกแห่งความเป็นจริงมากกว่าการวิเคราะห์ระบบเชิงโครงสร้าง ดังนั้น User จึงสามารถทำความเข้าใจได้ง่าย นอกจากนี้การวิเคราะห์และออกแบบระบบเชิงวัตถุนี้ยังสนับสนุนหลักการทำงานและคุณสมบัติเชิงวัตถุ จึงทำให้แบบจำลองที่ได้จากการวิเคราะห์และออกแบบระบบสามารถแสดงถึงแนวความคิดเชิงได้อย่างชัดเจน ทำให้มองเห็นเค้าโครงของระบบและพัฒนาโปรแกรมเชิงวัตถุได้โดยสะดวกขึ้น

2.3.1 ขั้นตอนในการพัฒนาระบบเชิงวัตถุ

ในการพัฒนาระบบเชิงวัตถุนี้ จะแบ่งขั้นตอนในการทำงานเป็น Phase หลัก 6 Phase คือ

1) Requirement Analysis
2) Analysis
3) Design
4) Programming
5) Testing
6) Maintenance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.1 ขั้นตอนการวิเคราะห์ความต้องการของระบบ (Requirement Analysis)

เป็นขั้นตอนที่พยายามหาความต้องการของผู้ใช้งานและวัตถุประสงค์ของระบบ อันจะเป็นตัวช่วยในการกำหนดว่าระบบที่จะพัฒนาขึ้นนี้ต้องมีอะไรบ้าง โดยจะอธิบายความต้องการของผู้ใช้งานว่าอะไรคือสิ่งที่ระบบควรจะเป็น เพื่อที่จะเรียบเรียงเป็นความต้องการของระบบ และทำให้สามารถพัฒนาระบบที่บรรลุตามวัตถุประสงค์ที่วางไว้ได้ ดังนั้นจึงควรอธิบายความต้องการของระบบให้ชัดเจน เพื่อที่จะพัฒนาระบบได้ถูกต้อง ตรงตามความต้องการต่อไป

โดยขั้นตอนนี้จะนำเสนอด้วย แบบจำลอง Use Case ซึ่งจะนำเสนอให้เห็นว่าระบบมีความเกี่ยวข้องกับสิ่งใดๆ หรือ วัตถุใดๆบ้าง และระบบต้องมีกิจกรรมอะไรบ้าง สิ่งที่ได้จากการวิเคราะห์ความต้องการของระบบ นอกจากจะเป็นการทำความเข้าใจระบบเพื่อนำไปพัฒนาโปรแกรมเท่านั้น แต่ยังเป็นการทำความเข้าใจเกี่ยวกับระบบการทำงานในเชิง Business Process ด้วย

2.3.1.2 ขั้นตอนในการวิเคราะห์ระบบเชิงวัตถุ (Analysis)

ขั้นตอนการวิเคราะห์ระบบถือว่าเป็นส่วนที่มีความสำคัญ เนื่องจากเป็นขั้นตอนที่ช่วยในการกำหนดว่าระบบที่จะพัฒนาขึ้นนี้ต้องมีอะไรบ้าง เพื่อที่จะพัฒนาระบบได้ตรงตามความต้องการของผู้ใช้งานระบบ โดยจะนำเอาความต้องการของผู้ใช้งานที่ได้ศึกษา มาทำการวิเคราะห์ว่าเราจะพัฒนาระบบได้อย่างไร

ในขั้นตอนนี้ จะเป็นการระบุถึง Class ของวัตถุ (Object) ที่มีอยู่ในระบบ โดยรวมถึง คุณสมบัติ (Attribute) และพฤติกรรม (Operation) ของ Class รวมทั้งระบุความสัมพันธ์ระหว่าง Class ว่าเป็นแบบ One-to-One, One-to-Many หรือ Many-to-One และแต่ละ Class มีความสัมพันธ์แบบใด นอกจากนี้ จะนำเสนอถึงขั้นตอนการทำงานของระบบเพื่อเป็นแนวทางในการพัฒนาระบบต่อไป ซึ่งในขั้นตอนนี้จะเป็นการวิเคราะห์ปัญญาหา และหาทางเพื่อให้บรรลุตามวัตถุประสงค์ เป็นการกำหนดว่าระบบที่จะพัฒนาขึ้นมา จะต้องมีอะไรบ้าง โดยที่ยังไม่อธิบายถึงเทคนิคในการที่จะนำมาใช้

ขั้นตอนนี้จะนำเสนอด้วยแบบจำลอง UML คือ

1) Class Diagram เพื่อนำเสนอถึง Class ต่างๆภายในระบบ

2) Activity Diagram หรือ State Diagram เพื่อนำเสนอถึงขั้นตอน กระบวนการทำงานของระบบ ซึ่งทั้ง 2 Diagram สามารถนำเสนอได้เหมือนกัน แต่ Activity Diagram จะแสดงกระบวนการซึ่งเป็นการกระทำที่ต่อเนื่องกันและสามารถแสดงให้เห็นว่าในกระบวนการทำงานนั้นจำเป็นที่จะต้องให้ขั้นตอนใดเสร็จก่อนจึงจะทำขั้นตอนต่อไปได้ โดยใช้เส้นทึบเป็นตัวแบ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.3 ขั้นตอนในการออกแบบระบบเชิงวัตถุ (Design)

ในขณะที่การวิเคราะห์ระบบจะเป็นการกำหนดว่าระบบที่จะพัฒนาขึ้นนั้นต้องมีอะไรบ้าง การออกแบบระบบจะเป็นการกำหนดว่าจะทำการพัฒนาระบบได้อย่างไร

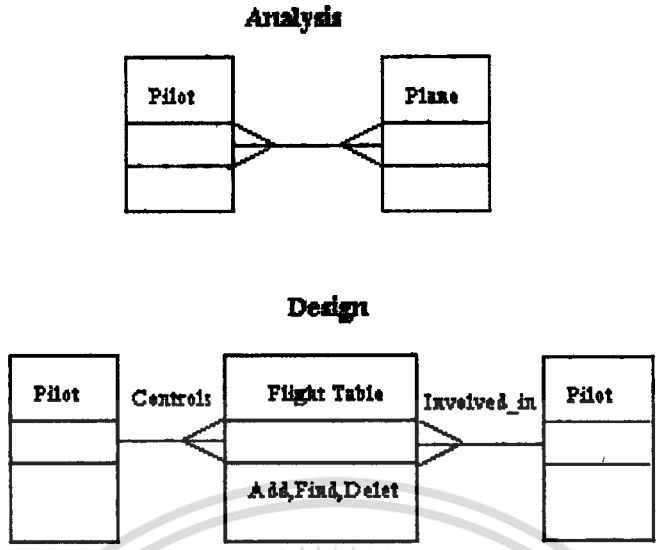
ในมุมมองเชิงวัตถุ การวิเคราะห์และการออกแบบระบบจะมีความแตกต่างน้อยมาก เนื่องจากความสัมพันธ์ของคุณสมบัติและพฤติกรรมของวัตถุ และแบบจำลองที่ใช้ในขั้นตอนของการวิเคราะห์และออกแบบจะใช้หลักการเดียวกัน ดังนั้นกระบวนการของการออกแบบจึงสามารถกระทำต่อได้จากการวิเคราะห์ระบบได้อย่างราบรื่น

ช่วงของการออกแบบระบบจะเริ่มเมื่อได้ข้อมูลจากการวิเคราะห์ระบบเพียงพอที่จะพัฒนาระบบได้แล้ว จาก Classes ต่างๆ ที่ได้จากช่วงของการวิเคราะห์ระบบ อาจจะต้องมีการเพิ่มเติมบาง Classes เพื่อทำให้ระบบสมบูรณ์ขึ้น และสามารถพัฒนาระบบต่อไปได้ตาม Hardware และ Software ที่ได้พิจารณาเลือกไว้ ในขั้นนี้จะมอง Classes เป็นกลุ่มของวัตถุ (Objects) ซึ่งจะทำให้เห็นมุมมองในการสร้างหน้าจอของโปรแกรมเพื่อผู้ใช้งานระบบได้อย่างชัดเจนขึ้น

ในขั้นตอนของการออกแบบระบบนี้ อาจจะแบ่งเป็น ส่วนในการพิจารณาเลือกเทคนิคเพื่อใช้พัฒนาระบบออกเป็น 2 ส่วนคือ การเลือก Hardware และ Software ที่จะนำมาพัฒนาระบบ โดยอาจมีเกณฑ์ในการพิจารณาดังต่อไปนี้

1) การพิจารณาในส่วนของ Hardware
   - ระบบควรจะเป็นแบบ Centralized หรือ Distributed
   - ถ้าเป็นระบบ Distributed การจะจัดตำแหน่ง Object ควรจะนำ Object ไปไว้ที่ใดจึงจะสามารถทำงานได้ตรงตามความต้องการและได้ประสิทธิภาพสูงสุดและจะใช้อะไรเป็นตัวติดต่อสื่อสาร
   - พิจารณาประสิทธิภาพของเครื่องที่จะนำมาใช้ในการทำงานของโปรแกรม
2) การพิจารณาในส่วนของ Software
   - ระบบที่จะพัฒนาขึ้นนี้ ควรที่จะแยกออกเป็นโปรแกรมมากกว่า 1 โปรแกรมหรือไม่
   - Database ใดที่ควรจะนำมาใช้ในระบบ
   - ควรจะเพิ่ม Classes ใดขึ้นมาเพื่อทำให้ระบบสมบูรณ์ขึ้น และสนับสนุนการสร้างหน้าจอโปรแกรมใช้งานตามที่ต้องการได้ หรือไม่
   - เพิ่ม operation ใน Class เพื่อเพิ่มความเข้าใจ และสะดวกในการเขียนโปรแกรม
   - ควรเพิ่มเติม Classes เพื่อทำให้ความสัมพันธ์ระหว่าง Classes เข้าใจได้ง่ายขึ้นหรือไม่ เช่น จากการวิเคราะห์ อาจจะได้ Classes ซึ่งมีความสัมพันธ์แบบ Many-to-Many ดังนั้นอาจจะเพิ่มอีก Class หนึ่งเพื่อทำให้เป็นความสัมพันธ์แบบ One-to-Many ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 การเพิ่ม Classes จาก ขั้นตอน Analysis ในขั้นตอนการ Design

ขั้นตอนนี้จะนำเสนอด้วยแบบจำลอง UML คือ

1) Class Diagram ซึ่งจะนำ Class ที่ได้ในขั้นตอนของการ วิเคราะห์ระบบเชิงวัตถุ มาเพิ่มเติม เพื่อให้เข้าใจการทำงานของระบบง่ายขึ้นและสามารถพัฒนาโปรแกรมได้ง่ายขึ้นด้วย

2) Sequence Diagram หรือ State Diagram โดยจะเป็นการแสดงให้เห็นถึงการทำงานระหว่าง Object ต่างๆ ตามการส่งข้อความ (Message) และเมื่อเหตุการณ์ (Event) ต่างๆ ได้เกิดขึ้น

จะแสดงถึงการส่ง Message เพื่อเป็นการติดต่อสื่อสาร ระหว่าง Object ต่างๆที่เกี่ยวข้อง ซึ่งสามารถนำเสนอได้เช่นเดียวกับ Collaboration Diagram แต่การนำเสนอด้วย Sequence Diagram นี้จะแสดงให้เป็นถึงลำดับช่วงเวลาที่แน่นอนของการส่ง Message

2.3,1.4 ขั้นตอนการเขียนโปรแกรม (Programming)

ในขั้นตอนของการเขียนโปรแกรมเชิงวัตถุ จะเป็นการนำเอา Class และ Model ต่างๆ ที่ได้จากขั้นตอนของการวิเคราะห์และออกแบบระบบเชิงวัตถุ มาเขียนเป็น Code Program

จาก UML ซึ่งเป็น Model ที่ใช้เป็นสื่อกลางในการนำเสนอแนวความคิดในการวิเคราะห์และออกแบบระบบ จะทำให้เราสามารถเขียนโปรแกรมได้สะดวกขึ้น โดยที่ผู้เขียนโปรแกรมไม่ต้องกังวลในการแปลงจาก Model มาเป็น Code Program เนื่องจาก รูปแบบของ UML ที่กำหนดนั้น มีประโยชน์ในการช่วยให้ผู้ใช้งานได้ทำความเข้าใจถึงโครงสร้างของระบบได้ดีขึ้น ดังนั้นการด่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปถึงตัวCode นั้น จะไม่เกิดประโยชน์ต่อการพัฒนา เมื่อเทียบกับวิธีแบบมาตรฐานที่ถูกต้อง นอกจากนี้ Rational Rose ซึ่งเป็น Case Tools ในการเขียน Model UML ยังมีฟังก์ชันช่วยในการสร้างเป็น Code โปรแกรมอย่างง่าย ตาม Model ที่ได้ออกแบบไว้

2.3.1.5 ขั้นตอนการทดสอบระบบ (Testing)

ในขั้นตอนของการทดสอบระบบ จะแบ่งการทดสอบออกเป็นระดับต่างๆ คือ Unit Test, Integration Test, System Test และ Acceptance Test

1) **Unit Test** จะเป็นการทดสอบการทำงานในส่วนของ Class หรือ กลุ่มของ Class ซึ่งการทดสอบนี้จะกระทำโดย Programmer

Diagram ที่เกี่ยวข้องกับการทดสอบระดับนี้คือ Class Diagram

2) **Integration Test** จะเป็นการทอดสอบการทำงานโดยรวมของ Component หรือ Class เพื่อที่จะบรรลุตามความต้องการบางอย่างของระบบ

Diagram ที่เกี่ยวข้องกับการทดสอบระดับนี้คือ Components Diagram และ Collabolation / Sequence Diagrams

3) **System Test** จะเป็นการทดสอบทั้งระบบ โดยทดสอบความถูกต้องของการทำงานของระบบ ซึ่งจะทดสอบโดย End User

Diagram ที่เกี่ยวข้องกับการทดสอบระดับนี้คือ Use Case Diagram อันแสดงถึงการทำงานของระบบโดยรวม

4) **Acceptance test** จะเป็นการทดสอบระบบโดย User ว่าระบบที่ได้นั้น ตรงตามความต้องการหรือไม่

Diagram ที่เกี่ยวข้องกับการทดสอบระดับนี้คือ Use Case Diagrams เช่นเดียวกับระดับ System Test

2.3.1.6 ขั้นตอนการบำรุงรักษา (Maintenance)

ในการ Maintenance ระบบเชิงวัตถุจะสามารถทำได้สะดวกกว่าระบบเชิงโครงสร้าง เช่น ถ้ามีการเปลี่ยนแปลงโครงสร้างของ Class ก็จะทำการเปลี่ยนแปลงเฉพาะ Class ที่ต้องการเปลี่ยนแปลงเท่านั้น โดยที่จะไม่ส่งผลกระทบต่อ Class อื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ข้อดี-ข้อเสียของการวิเคราะห์และออกแบบระบบเชิงวัตถุเปรียบเทียบกับเชิงโครงสร้าง

ข้อดี

1) ผลลัพธ์ที่ได้จากการวิเคราะห์ระบบเชิงโครงสร้าง จะได้แบบจำลอง DFDs ซึ่งสามารถนำเสนอได้เฉพาะมุมมองในแง่การทำงานของระบบ แต่ไม่ได้ระบุถึงมุมมองในแง่การวิเคราะห์เพื่อพัฒนาโปรแกรม ในขณะที่การวิเคราะห์เชิงวัตถุมองระบบในมุมมองเชิงวัตถุ จึงทำให้การวิเคราะห์และออกแบบระบบทำได้ต่อเนื่องกันโดยตรง และง่ายต่อการพัฒนาระบบโดยใช้ภาษาโปรแกรมเชิงวัตถุ (Object-Oriented Programming Language) และ/หรือระบบจัดการฐานข้อมูลเชิงวัตถุ

2) จากหลักการของการวิเคราะห์ระบบเชิงวัตถุที่มองระบบในแง่เชิงวัตถุเสมือนโลกแห่งความจริง จึงทำให้สามารถอธิบายถึงภาพรวมของระบบได้โดยง่าย และสามารถสื่อสารระหว่างผู้วิเคราะห์ระบบกับผู้ใช้งานระบบได้เข้าใจดีกว่าการวิเคราะห์ระบบเชิงโครงสร้าง

3) จากข้อดีต่างๆของการพัฒนาระบบเชิงวัตถุ เช่นการนำกลับมาใช้ใหม่ (Reusability) เป็นต้น จึงทำให้การพัฒนาระบบเชิงวัตถุเริ่มได้รับความนิยมมากกว่าการพัฒนาระบบเชิงโครงสร้าง ซึ่งการวิเคราะห์ระบบเชิงวัตถุก็สนับสนุนการพัฒนาระบบเชิงวัตถุมากกว่าการวิเคราะห์ระบบเชิงโครงสร้าง

4) การมองระบบในมุมมองเชิงวัตถุช่วยให้การสร้างหน้าจอใช้งานสามารถทำได้สะดวกขึ้น เนื่องจากการมองระบบเชิงวัตถุทำให้ทราบว่าแต่ละวัตถุจะต้องมีคุณสมบัติและพฤติกรรมอะไรบ้าง ในขณะที่การวิเคราะห์และออกแบบระบบเชิงโครงสร้างสามารถช่วยให้เห็นมุมมองของหน้าจอการใช้งานระบบได้เพียงเล็กน้อย

ข้อเสีย

1) ผู้วิเคราะห์ระบบจะต้องมีความชำนาญในหลักการเชิงวัตถุเป็นอย่างดี

2) ภาษาโปรแกรมเชิงวัตถุ เป็นภาษาที่ค่อนข้างซับซ้อนในการที่จะสร้างระบบเชิงวัตถุให้บรรลุได้ตามวัตถุประสงค์

3) เนื่องจากระบบเชิงวตถุค่อนข้างมีหลักการที่ซับซ้อนจึงทำให้การทดสอบระบบทำได้ยากด้วยเช่นกัน

ขั้นตอนการวิเคราะห์ระบบเชิงวัตถุอาจจะต้องใช้เวลานานกว่าการวิเคราะห์ระบบเชิงโครงสร้าง เนื่องจากจะต้องเก็บรายละเอียดให้ได้ทั้งหมด หรือมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีดำเนินการและการรวบรวมข้อมูล

3.1 กรณีศึกษา

โครงการระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ (Information System of Clinic Using Object-Oriented Techniques) จะเป็นการศึกษาถึงกระบวนการทำงานภายในคลินิกตั้งแต่การเก็บข้อมูลบุคคลากรในคลินิก ข้อมูลประวัติคนไข้ ข้อมูลการตรวจรักษา ข้อมูลยา ข้อมูลโรค รวมทั้งการตรวจสอบปริมาณยาในคลังยา และการออกรายงานเอกสารต่างๆ ซึ่งการศึกษาการทำงานของระบบนี้จะทำการสอบถามข้อมูลจากความต้องการของผู้ใช้งานระบบอันได้แก่ แพทย์และพยาบาลในคลินิก เพื่อที่ทำให้เราสามารถดำเนินการวิเคราะห์และออกแบบระบบได้ตรงตามความต้องการของผู้ใช้งาน และมีประสิทธิภาพยิ่ง

3.2 เครื่องมือที่ใช้ในโครงการ

ในโครงการระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ เราจะใช้เครื่องมือช่วยในการทำงาน ซึ่งประกอบด้วย

1. อุปกรณ์ Hardware ดังนี้
   - 1.1 คอมพิวเตอร์จำนวน 1 เครื่อง แป้นพิมพ์ข้อมูล (Keyboard) พร้อม Mouse จำนวน 1 ตัว
   - 1.2 เครื่องพิมพ์ (Printer) 1 เครื่อง
2. Software ในการพัฒนาระบบ ดังนี้
   - 2.1 ระบบปฏิบัติการ (Operating System) จะใช้ Window 98 Thai Edition
   - 2.2 Rational Rose 2000 เป็น Case Tools ในการนำเสนอแนวความคิดการออกแบบและวิเคราะห์ระบบเชิงวัตถุ
   - 2.3 ใช้ Cache' Database เป็นระบบจัดการฐานข้อมูลเชิงวัตถุ
   - 2.4 ใช้ Microsoft Visual Basic ในการพัฒนาระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเก็บรวบรวมข้อมูล

เก็บรวบรวมข้อมูลทั้งหมดที่ใช้ในระบบงาน โดยจะศึกษาถึงกระบวนการขั้นตอนต่างๆในการดำเนินการภายในคลินิก และข้อมูลต่างๆที่จำเป็นต้องจัดเก็บในระบบ เช่น ข้อมูลประวัติคนไข้ จะต้องมีรายละเอียดของ ชื่อคนไข้ ที่อยู่ วัน/เดือน/ ปีเกิด เป็นต้น และจะต้องดูว่าความต้องการในระบบที่จะพัฒนามีอะไรบ้าง เช่น จากเดิมที่บันทึกการจ่ายยาแบบ Manual ซึ่งอาจมีข้อผิดพลาดในการเตือนเมื่อยาใกล้หมดจากคลังยา ก็อาจจะต้องการระบบที่สามารถเตือนได้เมื่อจำนวนยาเหลือน้อยกว่าที่กำหนด รวมทั้งเก็บข้อมูลว่าข้อมูลใดจะต้องเป็นข้อมูลนำเข้าหรือข้อมูลออกจากกระบวนการใดๆ และต้องมีเอกสารหรือรายงานใดบ้างจากระบบ

ในขั้นตอนของการเก็บรวบรวมข้อมูลนี้จะศึกษาจากระบบเก่าที่ทางคลินิกดำเนินการอยู่และสอบถามความต้องการเพิ่มเติมจากผู้ใช้งานระบบอันได้แก่ แพทย์และพยาบาลในคลินิกว่าอะไรคือสิ่งที่ระบบควรจะเป็น เพื่อที่จะเรียบเรียงเป็นความต้องการของระบบ และทำให้สามารถพัฒนาระบบที่บรรลุตามวัตถุประสงค์ที่วางไว้ได้ ดังนั้นจึงควรอธิบายความต้องการของระบบให้ชัดเจนเพื่อที่จะพัฒนาระบบได้ถูกต้อง ตรงตามความต้องการต่อไป โดยในขั้นตอนนี้ อาจจะนำเสนอด้วยแบบจำลอง Use Case Diagram เพื่อแสดงถึงแนวความคิดตามที่ผู้รวมรวบข้อมูลเข้าใจ ว่าถูกต้องตรงกับการทำงานของระบบหรือไม่

3.4 การวิเคราะห์และขั้นตอนการพัฒนาระบบ

การวิเคราะห์และการพัฒนาระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุจะนำเอาข้อมูลที่ได้จากขั้นตอนของการรวบรวมข้อมูลมาวิเคราะห์และออกแบบระบบ เพื่อให้ระบบเป็นไปตามความต้องการของผู้ใช้งาน มีประสิทธิภาพ ลดความซ้ำซ้อน ข้อมูลปราศจากความขัดแย้งซึ่งกันและกัน

การวิเคราะห์สามารถแบ่งออกเป็นขั้นตอนดังนี้

3.4.1 การวิเคราะห์ขั้นตอนและวิธีการในการดำเนินการ

1) ศึกษาและวิเคราะห์ข้อมูลทางด้านการจัดการข้อมูลภายในคลินิก โดยรวบรวมข้อมูลการทำงานของระบบเดิม และสำรวจความต้องการของผู้ใช้งานระบบ
2) วิเคราะห์ถึงปัญหาที่มีในระบบเดิม รวมทั้งข้อกำหนดต่างๆ และวิเคราะห์ถึงโครงสร้างของข้อมูลที่ได้ทำการรวบรวมมา เพื่อเป็นแนวทางในการออกแบบระบบที่จะพัฒนาขึ้นมาใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ออกแบบและพัฒนาระบบใหม่ โดยนำข้อมูลที่ได้จากการวิเคราะห์มาทำการออกแบบระบบให้ได้ตรงตามความต้องการ และออกแบบให้ข้อมูลที่รวบรวมมานั้นมีความสัมพันธ์กันอย่างถูกต้อง โดยใช้หลักการของ การออกแบบระบบเชิงวัตถุ

3.4.2 การพัฒนาระบบสารสนเทศเพื่อจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ

เมื่อได้ทำการออกแบบระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุได้ตรงตามความต้องการแล้ว จะดำเนินการพัฒนาโปรแกรมโดยใช้เทคโนโลยีเชิงวัตถุ (Object-Oriented Programming) ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การวิเคราะห์และออกแบบระบบงาน

4.1 การดำเนินการภายในคลินิกในปัจจุบัน

ระบบการดำเนินการในปัจจุบันจะเป็นการดำเนินการจัดการข้อมูลโดยใช้การจดบันทึกทั้งสิ้น โดยจะมีผู้ที่เกี่ยวข้องกับระบบคือ คนไข้ที่มารับการรักษา แพทย์ พยาบาล และ พนักงานต้อนรับหรือประชาสัมพันธ์

กระบวนการทำงาน จะเริ่มตั้งแต่ เมื่อมีคนไข้เข้ามารับการรักษาจากคลินิก พนักงานต้อนรับจะทำการค้นหาข้อมูลประวัติของคนไข้ในกรณีที่เป็นคนไข้เก่าซึ่งเคยได้รับการรักษาจากคลินิกแล้ว ถ้าหากเป็นคนไข้ใหม่ จะทำการให้คนไข้กรอกประวัติข้อมูลของตัวเองใหม่ เมื่อคนไข้กรอกข้อมูลประวัติของตัวเองลงในแบบฟอร์มแล้ว พนักงานต้อนรับจะทำการคัดลอกข้อมูลดังกล่าวลงในแฟ้มประวัติคนไข้ ซึ่งแฟ้มประวัติคนไข้นี้ จะจัดเก็บในตู้เอกสารตามลำดับตัวอักษรนำหน้าชื่อ ในกรณีที่คนไข้เก่า ต้องการจะเปลี่ยนประวัติตัวเอง เช่น อาจต้องการเปลี่ยนชื่อหรือนามสกุล พนักงานต้อนรับจะทำการเปลี่ยนข้อมูลที่แฟ้มประวัติคนไข้ให้ โดยใช้วิธีการลบและเขียนข้อมูลใหม่แทน หลังจากได้ทำการบันทึกหรือค้นหาประวัติคนไข้เรียบร้อยแล้ว พนักงานต้อนรับจะสอบถามถึงอาการที่ต้องการมาตรวจรักษา และบันทึกลงในกระดาษที่เตรียมไว้สำหรับการบันทึกข้อมูลการตรวจซึ่งจะแนบกับแฟ้มประวัติคนไข้ จะนำแฟ้มข้อมูลดังกล่าวจัดใส่ตะกร้ารอคิว เพื่อให้พยาบาลมารับแฟ้มประวัติดังกล่าวละทำการตรวจเบื้องต้นต่อไป

เมื่อพยาบาลมารับแฟ้มประวัติคนไข้ตามคิวแล้วจะดำเนินการตรวจเบื้องต้น เช่นการชั่งน้ำหนัก วัดส่วนสูง วัดความดัน วัดชีพจร เป็นต้น และจะบันทึกข้อมูลดังกล่าวลงในกระดาษในแฟ้มประวัติคนไข้ซึ่งพนักงานต้อนรับจัดเตรียมไว้ จากนั้นจะจัดคิวเพื่อรอรับการตรวจรักษาจากแพทย์ต่อไป โดยที่เมื่อคนไข้ถึงคิวพบแพทย์แล้ว แพทย์จะทำการตรวจวินิจฉัยโรคและบันทึกข้อมูลการวินิจฉัย , ข้อเสนอแนะ และ รายการยาที่สั่ง เป็นลายมือของแพทย์ ลงในแฟ้มข้อมูลประวัติคนไข้ และจะเขียนใบสั่งยาเพื่อให้พยาบาลดำเนินการจัดยาให้ตามที่แพทย์สั่ง ซึ่งจุดนี้มักจะเป็นปัญหามาเนื่องจากในบางครั้งพยาบาลอาจจะจะอ่านลายมือของแพทย์ผิดพลาดได้ง่าย ประกอบกับ ในกรณีที่คนไข้มารับการรักษาอีกครั้งแต่เป็นแพทย์อื่น อาจจะทำให้แพทย์ที่มาทำการรักษาต่ออ่านข้อมูลการตรวจรักษาก่อนหน้าผิดพลาดได้

หลังจากได้รับการตรวจรักษาเรียบร้อยแล้ว พยาบาลจะเป็นผู้นำแฟ้มประวัติคนไข้ไปจัดเก็บที่ตู้เอกสารและจะนำใบสั่งยาไปดำเนินการจ่ายยาต่อไป โดยการจ่ายยานี้ พยาบาลจะต้องอ่านจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลายมือแพทย์ที่เขียนชื่อยา ซึ่งอาจจะเกิดข้อผิดพลาดได้ง่าย และเมื่อจ่ายยาไป ก็ไม่ได้เป็นการลดจำนวนยาจากคลังยาเลย พยาบาลอาจจะต้องมาสรุปจำนวนยาที่จ่ายไปในแต่ละวันและปริมาณยาที่คงเหลืออยู่ ในเวลาปิดทำการในแต่ละวัน โดยเช็คจำนวนจากใบสั่งยาที่จ่ายไปในวันนั้นๆ แล้วบันทึกเป็นเอกสารเอง ซึ่งอาจจะเกิดข้อผิดพลาดได้สูง ทำให้บางครั้งจำนวนยาไม่เพียงพอต่อความต้องการ อาจทำให้คนไข้ไม่ได้รับยาตามที่แพทย์สั่ง และเกิดความไม่ประทับใจในการบริการของคลินิกได้

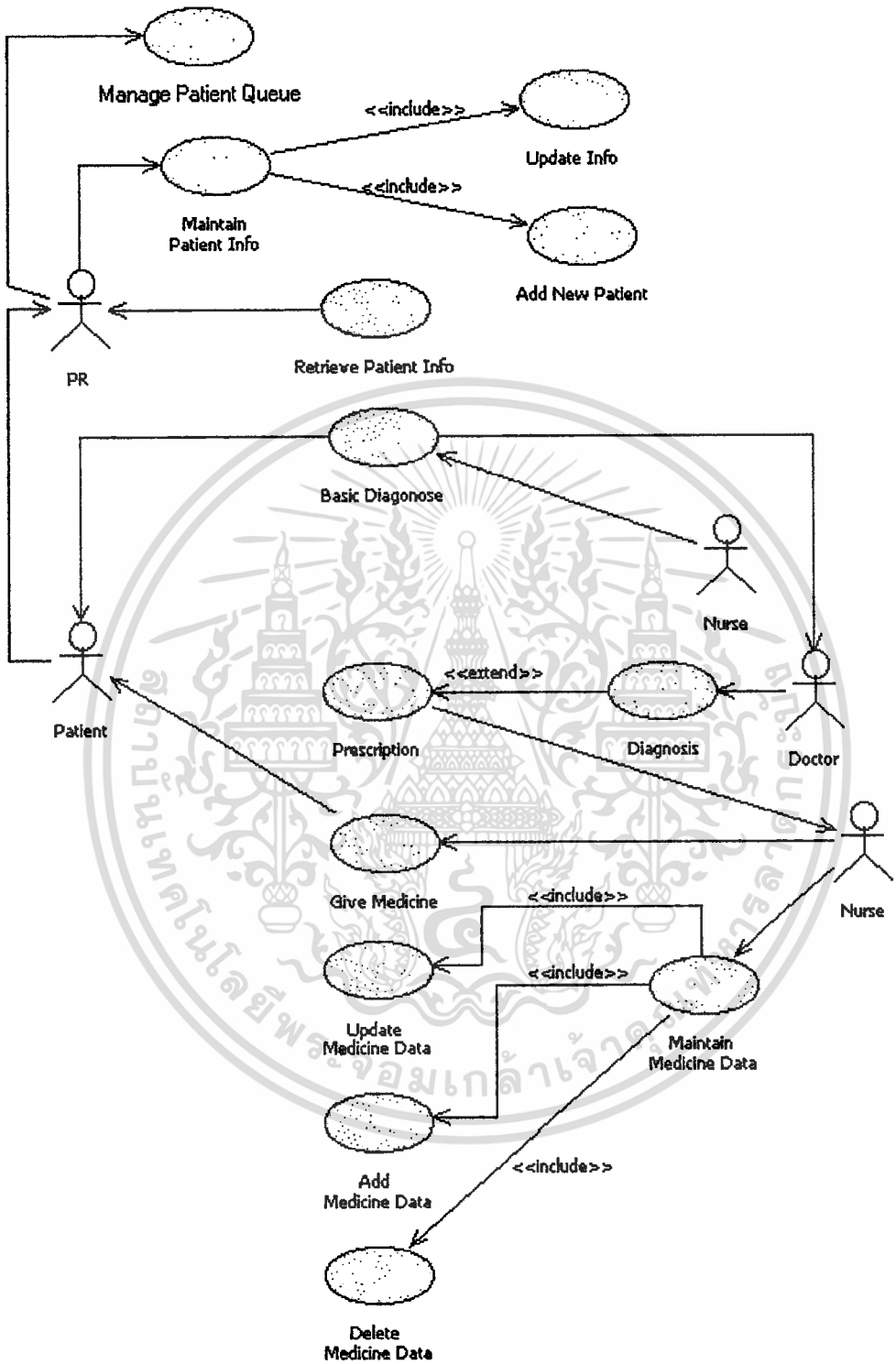
4.2 การวิเคราะห์และออกแบบระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ จะแบ่งขั้นตอนในการดำเนินการเป็นขั้นตอนดังนี้

4.2.1 ขั้นตอนการวิเคราะห์ความต้องการของระบบ (Requirement Analysis)

จะเป็นการศึกษาถึงความต้องการของระบบว่าระบบการจัดการข้อมูลภายในคลินิกที่จะพัฒนาขึ้นมานั้นต้องมีอะไรบ้าง ระบบมีความเกี่ยวข้องกับสิ่งใดๆบ้าง โดยจะเป็นการรวบรวมข้อมูลต่างๆ ซึ่งในขั้นตอนนี้จำนำเสนอด้วย Use Case Diagram เพื่อเป็นการแสดงแนวคิดของผู้วิเคราะห์ว่ามีแนวคิดที่ถูกต้องตรงกับกระบวนการต่างๆภายในระบบ และความต้องการของผู้ใช้งานหรือไม่ ซึ่งในขั้นตอนนี้จะต้องศึกษาความต้องการของระบบอย่างชัดเจนเพื่อที่จะสามารถพัฒนาระบบที่มีประสิทธิภาพ ถูกต้องตามความต้องการต่อไปได้

จาก Use Case Diagram ใน รูปที่ 4.1 จะแสดงให้เห็นถึงผู้ที่เกี่ยวข้องกับระบบและกิจกรรม กระบวนการ ต่างๆภายในระบบ โดยผู้ที่เกี่ยวข้องกับระบบ(Actor)จะแสดงด้วยสัญลักษณ์รูปคน อันได้แก่ คนไข้ (Patient) แพทย์ (Doctor) พยาบาล (Nurse) และ พนักงานต้อนรับ (PR) และในส่วนของกระบวนการหรือกิจกรรมที่กระทำภายในระบบ (Use Case) จะแสดงด้วยสัญลักษณ์วงรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 Use case Diagram **ของระบบ คลินิก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ขั้นตอนในการวิเคราะห์ระบบโดยแนวความคิดเชิงวัตถุ (Object-Oriented Analysis)

หลังจากได้ศึกษาและรวบรวมข้อมูลการทำงานภายในระบบแล้ว ขั้นตอนการวิเคราะห์ระบบนี้จะเป็นการนำเอาข้อมูลที่ได้มำกี่กำหนดว่าระบบที่จะพัฒนาจะเป็นอย่างไร โดยจะระบุถึง Class ของวัตถุ (Object) ที่จะต้อมีภายในระบบ โดยรวมถึงคุณสมบัติและพฤติกรรมของ Class

การวิเคราะห์ระบบจัดการข้อมูลภายในคลินิกโดยหลักการเชิงวัตถุนี้ จะนำเสนอด้วยแบบจำลอง Class Diagram และ Activity Diagram

1) Class Diagram

จาก Use Case Diagram ที่ได้จากขั้นตอนของการวิเคราะห์ความต้องการของระบบ เราสามารถวิเคราะห์ Class ต่างๆที่จะมีในระบบการจัดการข้อมูลภายในคลินิกได้ดังนี้

- Class Patient
- Class Doctor
- Class PR
- Class Nurse
- Class Diagnose
- Class Medicine
- Class Disease

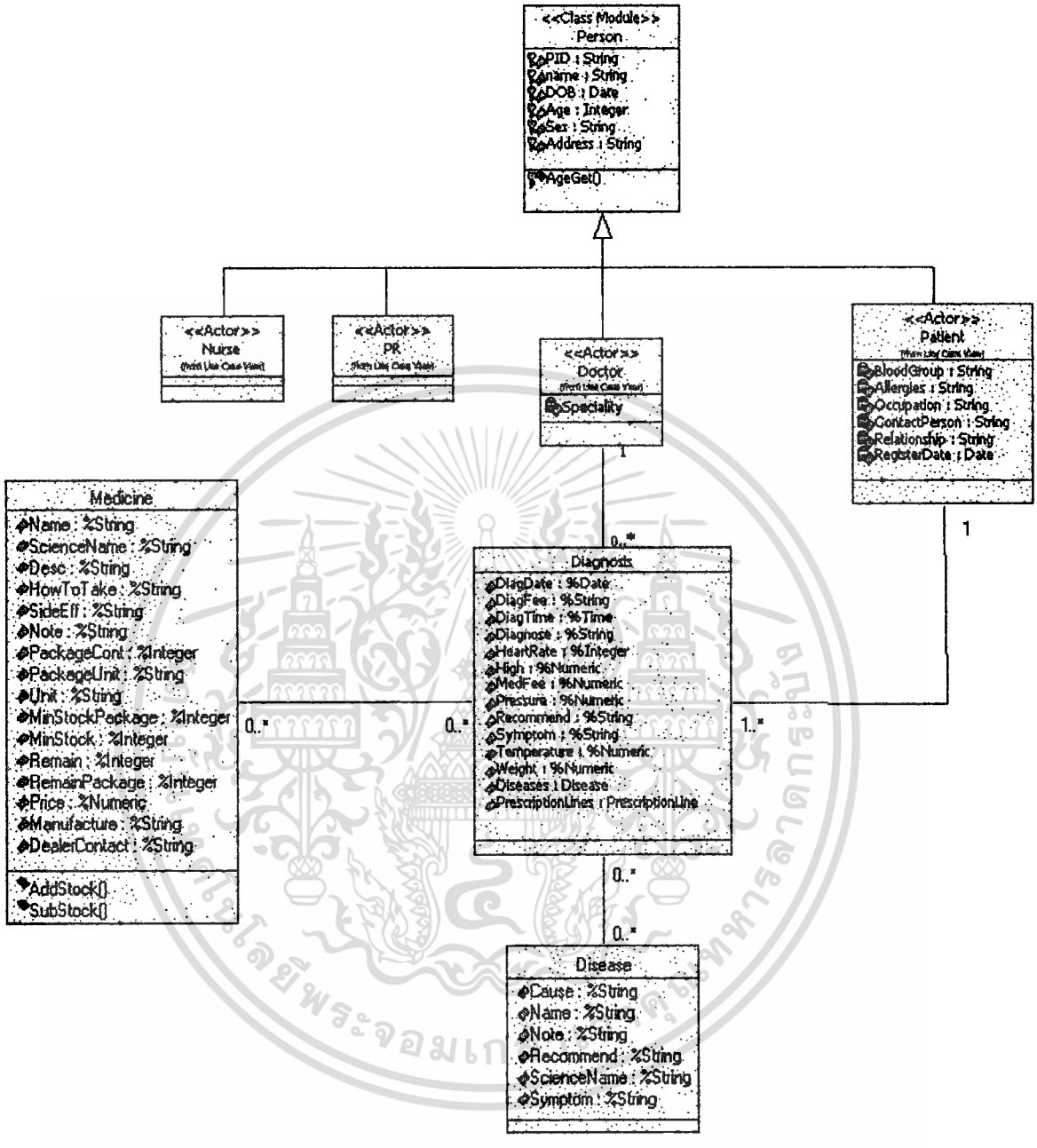
จาก Class ที่ได้พบว่า Class Patient, Class Doctor, Class PR และ Class Nurse มี Attribute และ Operation บางอย่างที่เหมือนกัน จึงจัดให้ Class Person เป็น Class แม่ และ Inherit (ถ่ายทอด) คุณสมบัติ และ พฤติกรรม มายัง Class ดังกล่าว ดังรูปที่ 4.2

Class Person จะมี Attribute ต่างๆ ซึ่งเป็นคุณสมบัติของบุคคลนั้นๆ และจะมี Operation คือ AgeGet ซึ่งเป็น Operation ที่ทำการคำนวณอายุของบุคคลนั้นๆ โดยนำค่าจาก Attribute DOB หรือ วันเกิด มาใช้ในการคำนวณ

Class Medicine จะมี Attribute ต่างๆ ซึ่งเป็นคุณสมบัติของยาชนิดนั้นๆ และจะมี Operation คือ

- AddStock เพื่อทำการเพิ่มจำนวนยาในข้อมูลยา และ
- SubStock เพื่อทำการลดจำนวนยาในข้อมูลยา เมื่อมีการจ่ายยาให้แก่คนไข้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 Class Diagram ที่ได้จากขั้นตอนการวิเคราะห์ระบบคลินิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

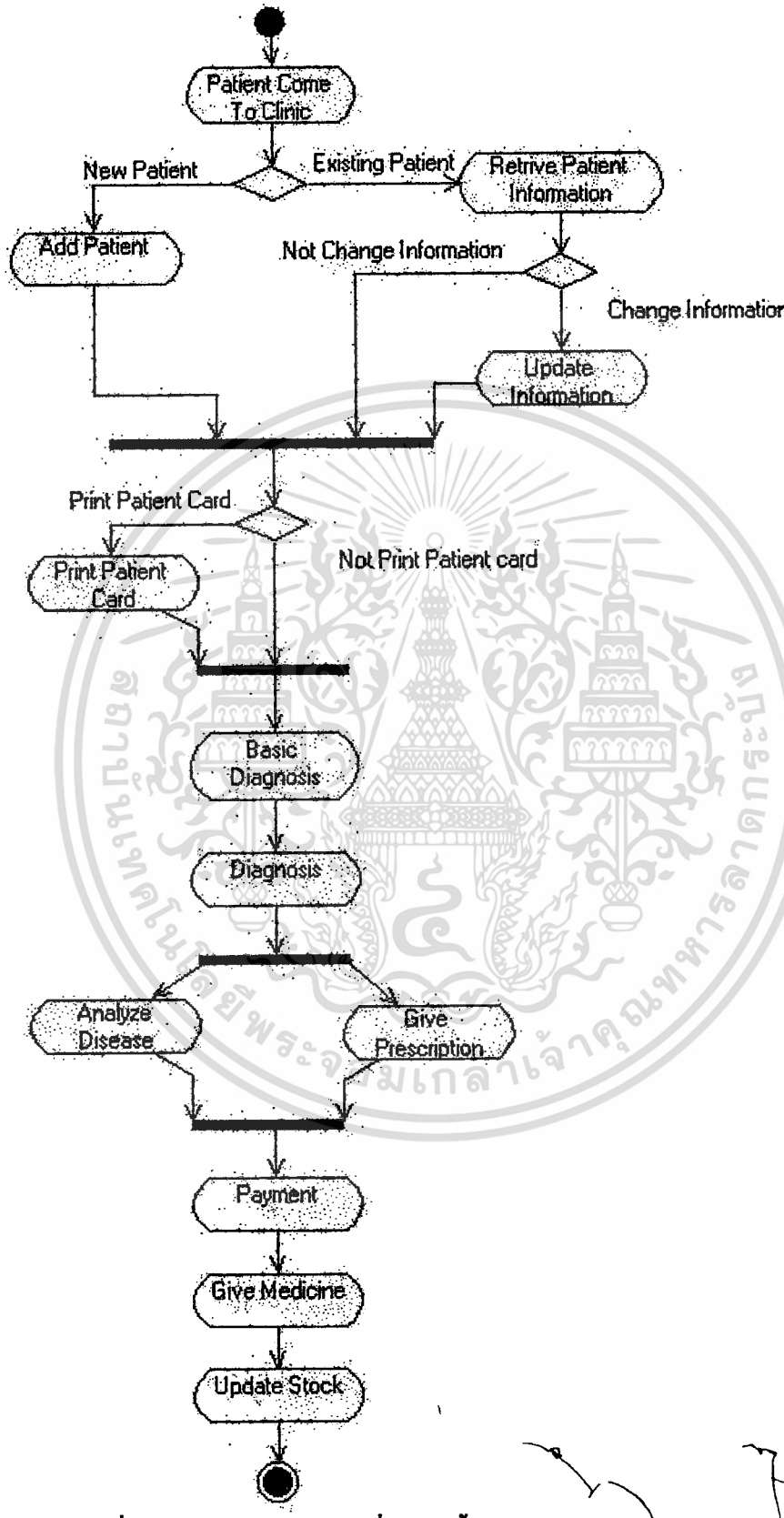
2) Activity Diagram

จากการวิเคราะห์ถึงกระบวนการ ขั้นตอนการทำงานของระบบ สามารถนำเสนอโดยใช้ Activity Diagram ซึ่งสามารถนำเสนอได้เช่นเดียวกับการใช้ State Diagram แต่ Activity Diagram จะแสดงกระบวนการซึ่งเป็นการกระทำที่ต่อเนื่องกัน และสามารถแสดงให้เห็นว่าในกระบวนการทำงานนั้น จำเป็นจะต้องให้ขั้นตอนใดเสร็จก่อน จึงจะทำขั้นตอนต่อไปได้ โดยใช้เส้นทึบเป็นตัวแบ่ง ดังนั้นการวิเคราะห์ระบบการจัดการข้อมูลภายในคลินิกนี้ จะเลือกใช้ Activity Diagram ในการนำเสนอ

ในกระบวนการทำงานของระบบคลินิกจะเริ่มจากการที่คนไข้เข้ามาติดต่อ และเสร็จสิ้นเมื่อได้รับยาและทำการลดจำนวนยาในคลังยาตามที่แพทย์สั่ง โดยมีลำดับในการทำงานคือ

- เมื่อคนไข้เข้ามาติดต่อที่คลินิก จะทำการค้นหาข้อมูลประวัติคนไข้ว่าเป็นคนไข้เก่าซึ่งเคยมารับการรักษาจากคลินิกแล้ว หรือ เป็นคนไข้ใหม่ กรณีเป็นคนไข้ใหม่จะเพิ่มข้อมูลคนไข้ แต่ถ้าเป็นกรณีคนไข้เก่าจะค้นหาข้อมูลคนไข้นั้นมา และถ้าหากคนไข้ต้องการจะเปลี่ยนข้อมูลก็จะทำการแก้ไข
- เมื่อได้ข้อมูลประวัติคนไข้แล้ว พิจารณาว่าจะต้องทำการพิมพ์บัตรให้คนไข้ใหม่หรือไม่ เช่นในกรณีที่เป็นคนไข้ใหม่ คนไข้เปลี่ยนข้อมูล หรือ กรณีบัตรหายหรือชำรุด
- หลังจากนั้นคนไข้จะได้รับการตรวจเบื้องต้นโดยพยาบาล เช่น การชั่งน้ำหนัก วัดส่วนสูง วัดความดัน เป็นต้น และจากนั้นจะเข้าสู่การตรวจวินิจฉัยจากแพทย์
- ในขั้นตอนการวินิจฉัยจากแพทย์ จะมีกระบวนการวินิจฉัยโรค และวิเคราะห์ว่าควรจะจ่ายยาให้คนไข้ อย่างไรบ้าง โดย 2 กระบวนการนี้ สามารถทำควบคู่กันไปในช่วงเวลาเดียวกันได้
- หลังจากได้รับการตรวจวินิจฉัยโรค และสั่งยาจากแพทย์เรียบร้อยแล้ว คนไข้จะทำการชำระเงินค่ายาและค่าตรวจรักษา จากนั้นจะได้รับยาตามที่แพทย์สั่ง เมื่อคนไข้ได้รับยาแล้ว ระบบจะทำการลดจำนวนยาในคลังยาตามชนิด และปริมาณที่แพทย์สั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3Activity Diagram ที่ได้จากขั้นตอนการวิเคราะห์ระบบคลินิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ขั้นตอนในการออกแบบระบบโดยแนวความคิดเชิงวัตถุ (Object-Oriented Design)

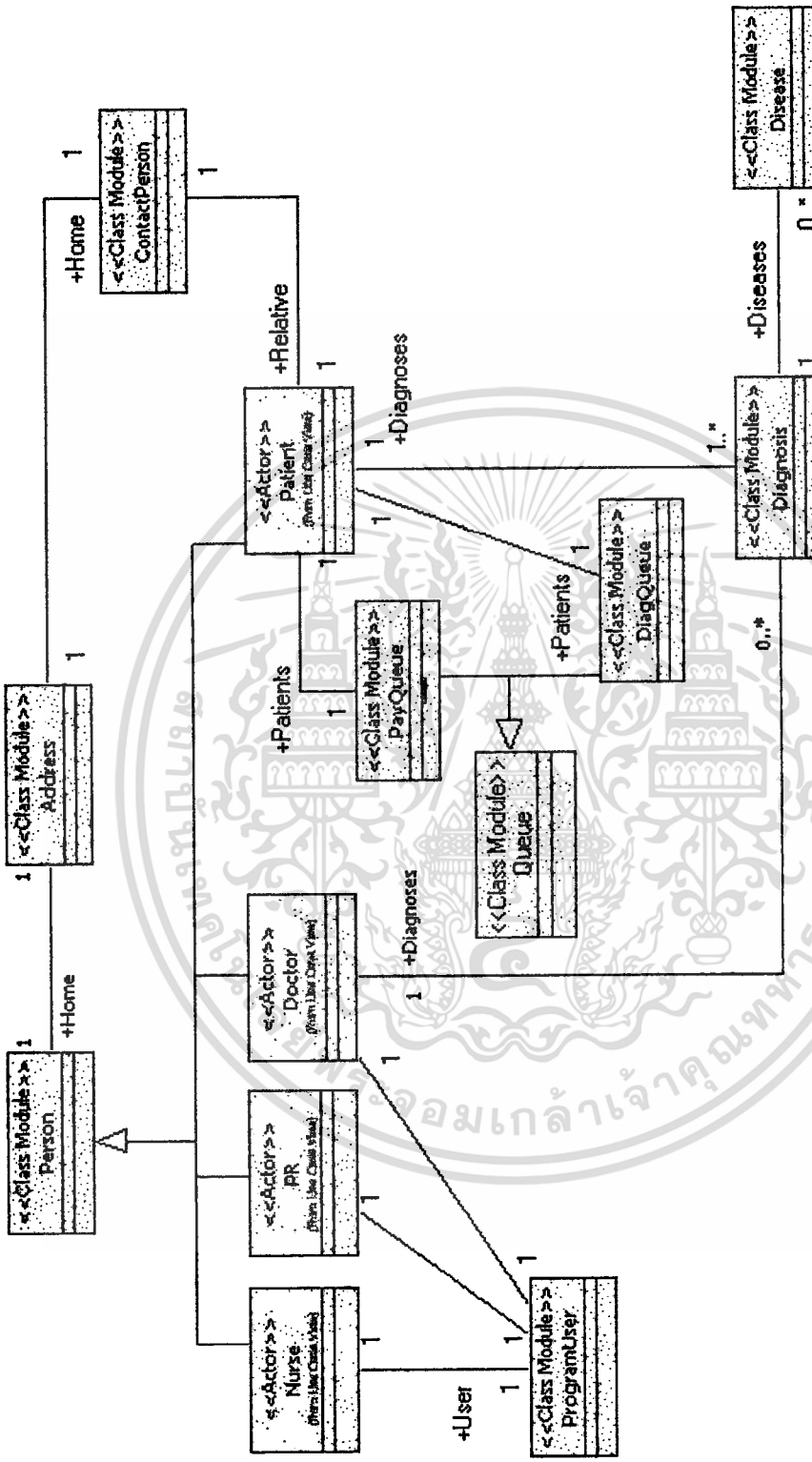
จากการออกแบบระบบเชิงวัตถุ เราจะนำเสนอด้วย Diagram ดังนี้

1) Class Diagram

จาก Class ที่ได้ในขั้นตอนของการ วิเคราะห์ระบบเชิงวัตถุ เราจะนำมาเพิ่มเติม เพื่อให้เข้าใจการทำงานของระบบง่ายขึ้นและสามารถพัฒนาโปรแกรมได้ง่ายขึ้นด้วย ดังแสดงในรูปที่ 4.4 ซึ่งประกอบด้วย Class ต่างๆ คือ

- Class Address
- Class Person
- Class PR
- Class Nurse
- Class Doctor
- Class Patient
- Class ContactPerson
- Class ProgramUser
- Class Queue
- Class DiagQueue
- Class PayQueue
- Class Diagnosis
- Class PrescriptionLine
- Class Medicine
- Class Disease

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

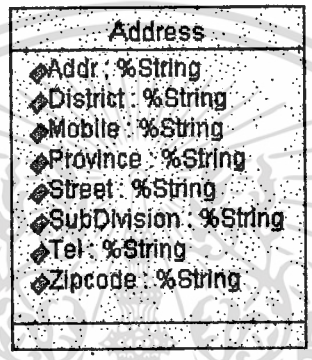


รูปที่ 4.4 Class Diagram ที่ได้จากการออกแบบระบบคลินิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

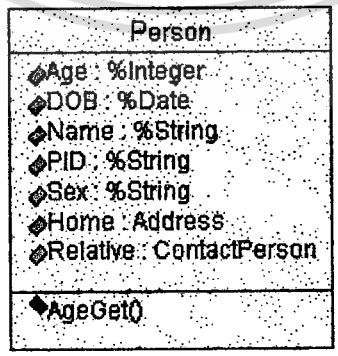
โดยที่แต่ละ Class มี Attribute และ Operation ดังต่อไปนี้

1.1) **Class Address** ในขั้นตอนของการออกแบบระบบนี้ จะแยก Class Address ออกมาจาก Class Person เนื่องจากเราสามารถมองได้ว่า Address นั้น เป็น Embedde Class คือ Class ฝังอยู่ หรือเป็นส่วนประกอบของอีก Class หนึ่ง นั้นคือ Class Address จะเป็น Class ซึ่งเป็นส่วนประกอบของ Class Person



รูปที่ 4.5 องค์ประกอบของ Class Address

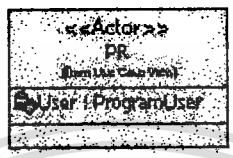
1.2) **Class Person** จะเป็น Parent Class ที่ถ่ายทอดคุณสมบัติทั้ง Attribute และ Operation มายัง Sub Class คือ Class PR , Class Nurse , Class Doctor และ Class Patient



รูปที่ 4.6 องค์ประกอบของ Class Person

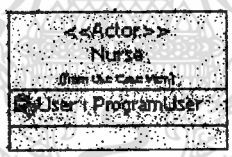
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3) Class PR เป็น Sub Class ซึ่งได้รับการถ่ายทอดคุณสมบัติจาก Class Person ดังนั้นจะมี Attribute และ Operation เช่นเดียวกับ Class Person แต่ Class PR จะมี Attribute ที่เพิ่มขึ้นมา นั่นคือ Attribute User ซึ่งมี Type เป็น ProgramUser เนื่องมาจากว่า PR หรือประชาสัมพันธ์ของคลินิกนี้ จะต้องเป็นผู้หนึ่งซึ่งใช้งานระบบด้วย ดังนั้น Class ProgramUser จึงเป็น Embedde Class ของ Class PR



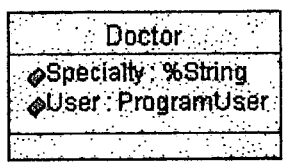
รูปที่ 4.7 องค์ประกอบของ Class PR

1.4) Class Nurse ลักษณะของ Class Nurse ซึ่งเป็น Class ของพยาบาล จะมีองค์ประกอบคุณสมบัติเช่นเดียวกับ Class PR เนื่องจากว่า ได้รับการถ่ายทอดคุณสมบัติจาก Class Person และเป็นผู้ใช้งานระบบด้วย



รูปที่ 4.8 องค์ประกอบของ Class Nurse

1.5) Class Doctor ลักษณะของ Class Doctor ซึ่งเป็น Class ของแพทย์นั้น จะมีองค์ประกอบคุณสมบัติคล้ายกับ Class PR และ Class Nurse เนื่องจากว่า ได้รับการถ่ายทอดคุณสมบัติจาก Class Person และเป็นผู้ใช้งานระบบด้วย แต่ Class Doctor จะเพิ่มในส่วนของ Attribute Speciality ซึ่งระบุถึงด้านที่เชี่ยวชาญของแพทย์



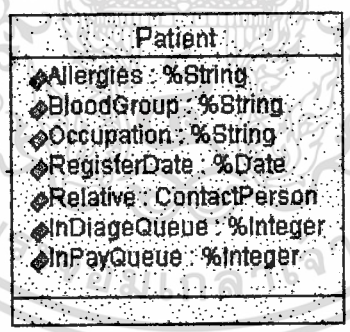
รูปที่ 4.9 องค์ประกอบของ Class Doctor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6) **Class Patient** เป็นอีก Class หนึ่งซึ่งได้รับการถ่ายทอดคุณสมบัติจาก Class Person แต่เนื่องจาก Patient หรือ คนไข้ ไม่ใช้ผู้ที่จะมาใช้งานระบบ ดังนั้น Class ProgramUser จึงไม่เป็นส่วนประกอบหนึ่งของ Class Patient

แต่ในส่วนของ Class Patient จะมี Attribute อื่นๆ เพิ่มเติมจาก Class PR, Class Nurse และ Class Doctor ได้แก่

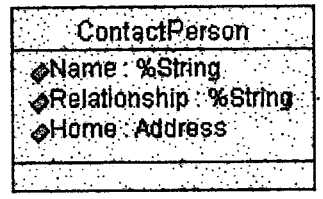
- Attribute Allergies ซึ่งจะเก็บข้อมูลว่าคนไข้แพ้ยาประเภทใด ชนิดใดบ้าง
- Attribute BloodGroup จะเก็บข้อมูลหมู่เลือดของคนไข้
- Attribute Occupation จะเก็บข้อมูลอาชพของคนไข้
- Attribute Register Date คือข้อมูลวันที่ ที่ทำการลงทะเบียนคนไข้ ของคลินิกนี้
- Attribute Relative ซึ่งมี Type เป็น ContactPerson กล่าวคือ Class ContactPerson หรือ Class ของญาติคนไข้ จะเป็น Class หนึ่งซึ่งเป็นองค์ประกอบของ Class Patient
- Attribute InDiagQueue คือข้อมูลระบุลำดับการรอรับการตรวจรักษา
- Attribute InDiagQueue คือข้อมูลระบุลำดับการรอรับการชำระเงินและรับยา



รูปที่ 4.10 องค์ประกอบของ Class Patient

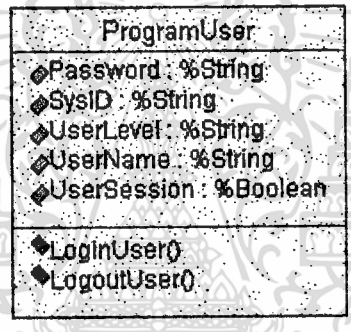
1.7) **Class ContactPerson** ในขั้นตอนของการออกแบบระบบนี้ จะมองว่า Class ContactPerson เป็น Embedde class ของ Class ContactPerson ซึ่งจะเก็บข้อมูลชื่อ , ความสัมพันธ์ และ ที่อยู่ ของญาติคนไข้ ดังนั้น Class Address จึงเป็นส่วนประกอบหนึ่ง (Embedde class) ของ Class ContactPerson นี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 องค์ประกอบของ Class ContactPerson

1.8) Class ProgramUser เป็น Class ของ กลุ่มผู้ใช้งานระบบ โดยที่ Class นี้ จะเก็บข้อมูลต่างๆซึ่งเกี่ยวกับการเข้าใช้งานระบบของ ผู้ใช้งานระบบ รวมทั้งจะเป็นตัวกำหนดสิทธิ์การเข้าใช้งานระบบ

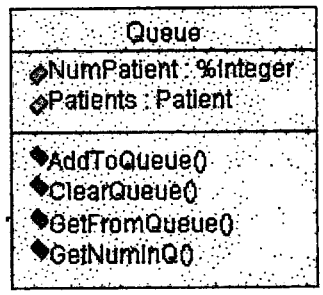


รูปที่ 4.12 องค์ประกอบของ Class ProgramUser

1.9) Class Queue เป็น Class ที่ทำหน้าที่ในการจัดการลำดับของคนไข้ในกระบวนการต่างๆ ได้แก่ การจัดลำดับเพื่อเข้ารับการตรวจรักษา และการจัดลำดับเพื่อรับยาและชำระเงิน โดยจะเป็น Parent Class ที่ถ่ายทอดคุณสมบัติทั้ง Attribute และ Operation มายัง Sub Class คือ Class DiagQueue และ Class PayQueue

- Attribute NumPatient จะเป็นข้อมูลลำดับที่ของคนไข้นั้นๆ
- Attribute Patients จะเป็นข้อมูลคนไข้ที่อยู่ในลำดับ
- Operation AddToQueue เป็นการนำข้อมูลคนไข้ใส่ในลำดับคิว
- Operation ClearQueue เป็นการนำคนไข้นั้นๆออกจากลำดับคิว
- Operation GetFromQueue เป็นการดึงข้อมูลคนไข้รายนั้นๆจากคิว
- Operation GetNumberInQueue เป็นการหาจำนวนคนไข้ที่มีอยู่ในคิวทั้งหมด ณ ขณะนั้น

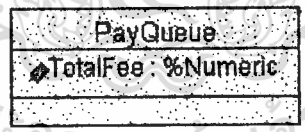
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 องค์ประกอบของ Class Queue

1.10) Class DiagQueue จะเป็น Class ซึ่งเก็บข้อมูล การจัดลำดับเพื่อเข้ารับการตรวจรักษา และเป็น Sub Class ซึ่งได้รับการถ่ายทอดคุณสมบัติจาก Class Queue ดังนั้นจะมี Attribute และ Operation เช่นเดียวกับ Class Queue

1.11) Class PayQueue จะเป็น Class ซึ่งเก็บข้อมูล การจัดลำดับเพื่อรับยาและชำระเงิน และเป็น Sub Class ซึ่งได้รับการถ่ายทอดคุณสมบัติจาก Class Queue ดังนั้นจะมี Attribute และ Operation เช่นเดียวกับ Class Queue แต่จะมี Attribute เพิ่มคือ Attribute TotalFee ซึ่งเป็นข้อมูลยอดรวมค่าใช้จ่ายที่คนไข้จะต้องชำระ



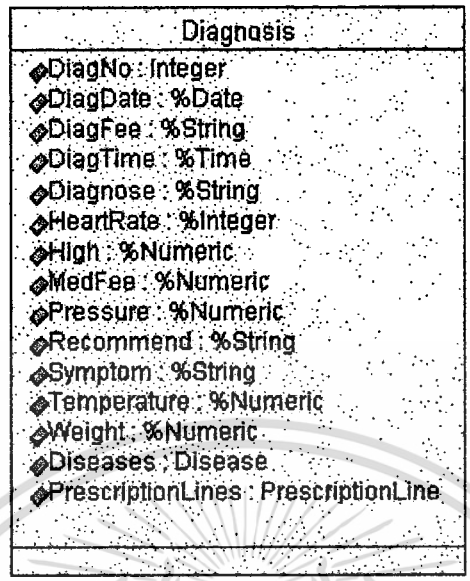
รูปที่ 4.14 องค์ประกอบของ Class PayQueue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.12) **Class Diagnosis** เป็น Class ที่เก็บข้อมูลของการตรวจรักษา โดยมี Attribute คือ

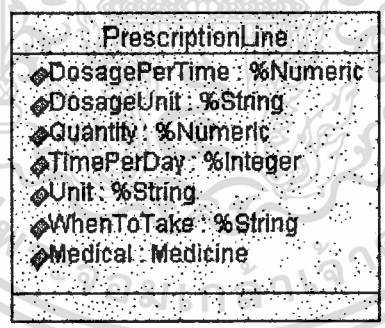
- Attribute DiagNo เป็นข้อมูลครั้งที่ของการตรวจของคนไข้
- Attribute DiagDate เป็นข้อมูลวันที่ของการตรวจรักษาครั้งนั้นๆ
- Attribute DiagFee เป็นข้อมูลค่าตรวจรักษาครั้งนั้นๆ
- Attribute DiagTime เป็นข้อมูลเวลาของการตรวจรักษาครั้งนั้นๆ
- Attribute Diagnose เป็นข้อมูลการวินิจฉัยของการตรวจรักษาครั้งนั้นๆ
- Attribute HeartRate เป็นข้อมูลชีพจรของคนไข้ในการตรวจรักษาครั้งนั้นๆ
- Attribute High เป็นข้อมูลส่วนสูงของคนไข้ในการตรวจรักษาครั้งนั้นๆ
- Attribute MedFee เป็นข้อมูลค่ายาในการตรวจรักษาครั้งนั้นๆ
- Attribute Pressure เป็นข้อมูลความดันของคนไข้ในการตรวจรักษาครั้งนั้นๆ
- Attribute Recommend เป็นข้อมูลการรักษา หรือ วินิจฉัยจากแพทย์ ในการตรวจรักษา
- Attribute Symptom เป็นข้อมูลอาการของคนไข้ในการตรวจรักษาครั้งนั้นๆ
- Attribute Temperature เป็นข้อมูลอุณหภูมิของคนไข้ในการตรวจรักษาครั้งนั้นๆ
- Attribute Weight เป็นข้อมูลน้ำหนักของคนไข้ในการตรวจรักษาครั้งนั้นๆ
- Attribute Disease เป็นข้อมูลโรคที่แพทย์วินิจฉัย ซึ่งจะเก็บเป็น List of Disease เนื่องจาก Class Disease มีความสัมพันธ์เป็น Many-to-One กับ Class Diagnoiss
- Attribute PrescriptionLines เป็นข้อมูลรายการสั่งยาโดยแพทย์ ซึ่งจะเก็บเป็น List of PrescriptionLine เนื่องจาก Class PrescriptionLine มีความสัมพันธ์เป็น Many-to-One กับ Class Diagnoiss

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 องค์ประกอบของ Class Diagnosis

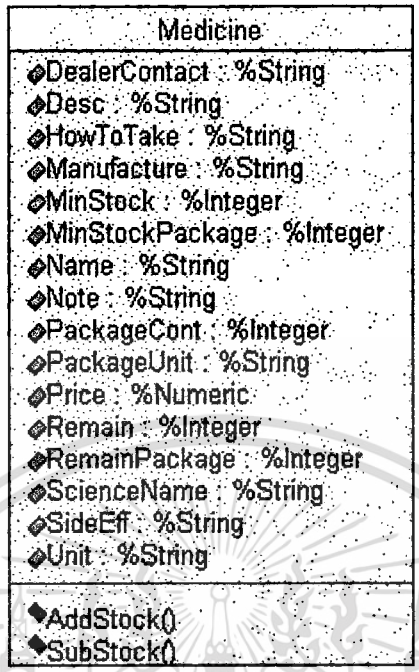
1.13) **Class PrescriptionLine** เป็น Class ที่เพิ่มขึ้นเพื่อจาก Class Diagnose และ Class Medicine ซึ่งมีความสัมพันธ์แบบ Many-to-One จึงเพิ่ม Class PrescriptionLine ซึ่งจะระบุถึงยาแต่ละชนิด ในใบสั่งยาของแพทย์ และข้อมูลการใช้ยาที่แพทย์สั่งให้กับคนไข้



รูปที่ 4.16 องค์ประกอบของ Class PrescriptionLine

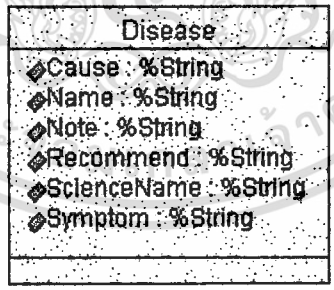
1.14) **Class Medicine** เป็น Class ซึ่งเก็บข้อมูลต่างๆที่เกี่ยวกับยา รวมถึงข้อมูล คลังยา และ Operation ในการคำนวณปริมาณยาที่เพิ่มเข้ามาหรือจ่ายให้แก่คนไข้ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 องค์ประกอบของ Class Medicine

1.15) Class Disease เป็น Class ซึ่งเก็บข้อมูลเกี่ยวกับโรคต่างๆ , อาการ , คำแนะนำเพื่อเป็นประโยชน์ต่อการรักษา



รูปที่ 4.18 องค์ประกอบของ Class Disease

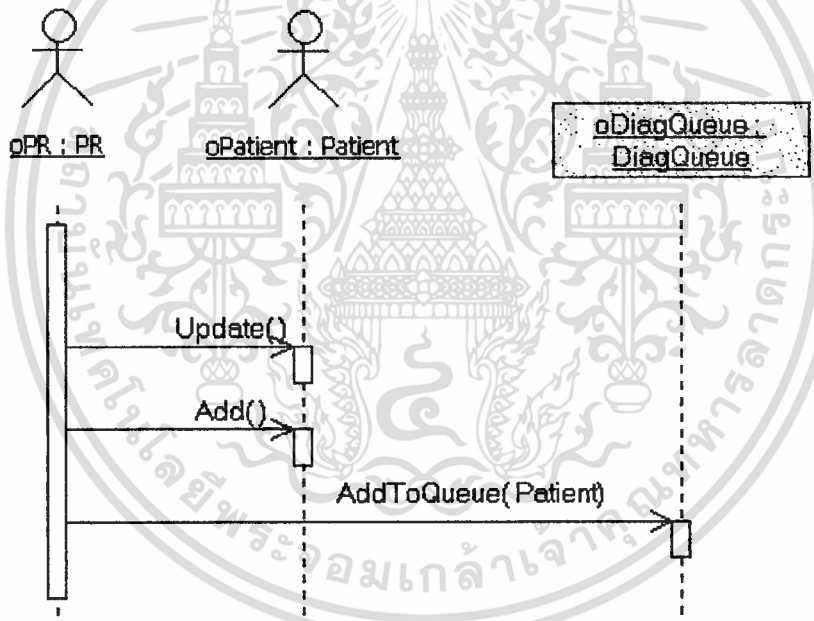
2) Sequence Diagram

จะแสดงถึงการส่ง Message เพื่อเป็นการติดต่อสื่อสาร ระหว่า ง Object ต่างๆที่เกี่ยวข้อง ซึ่งสามารถนำเสนอได้เช่นเดียวกับ Collaboration Diagram แต่การนำเสนอด้วย Sequence Diagram นี้ จะแสดงให้เป็นถึงลำดับช่วงเวลาที่แน่นอนของการส่ง Message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1) กระบวนการจัดการข้อมูลคนไข้ที่มารับการรักษา

- เมื่อมีคนไข้มาติดต่อเพื่อรับการรักษา ประชาสัมพันธ์ (Object PR) จะทำการค้นหาข้อมูลคนไข้ว่าเคยมีข้อมูลอยู่แล้วหรือไม่ กรณีที่ยังไม่มีข้อมูลคนไข้ดังกล่าว Object PR จะทำการ ส่ง Message Add ไปยัง Object Patient เพื่อทำการเพิ่มข้อมูลคนไข้ (Add Object Patient) เข้าสู่ระบบ หรือในกรณีที่มีข้อมูลคนไข้อยู่แล้วแต่ต้องการแก้ไขปรับปรุง ก็จะส่ง message Updateไปยัง Object Patient เพื่อเป็นการปรับปรุงแก้ไขข้อมูล
- เมื่อค้นหาข้อมูลหรือเพิ่มข้อมูลคนไข้เป็นที่เรียบร้อยแล้ว Object PR จะส่ง Message AddToQueue ไปยัง Object PatientQueue เพื่อให้ Object ดังกล่าวทำการเพิ่มข้อมูลคนไข้ไปจัดลำดับการรอรับการตรวจรักษาจากแพทย์



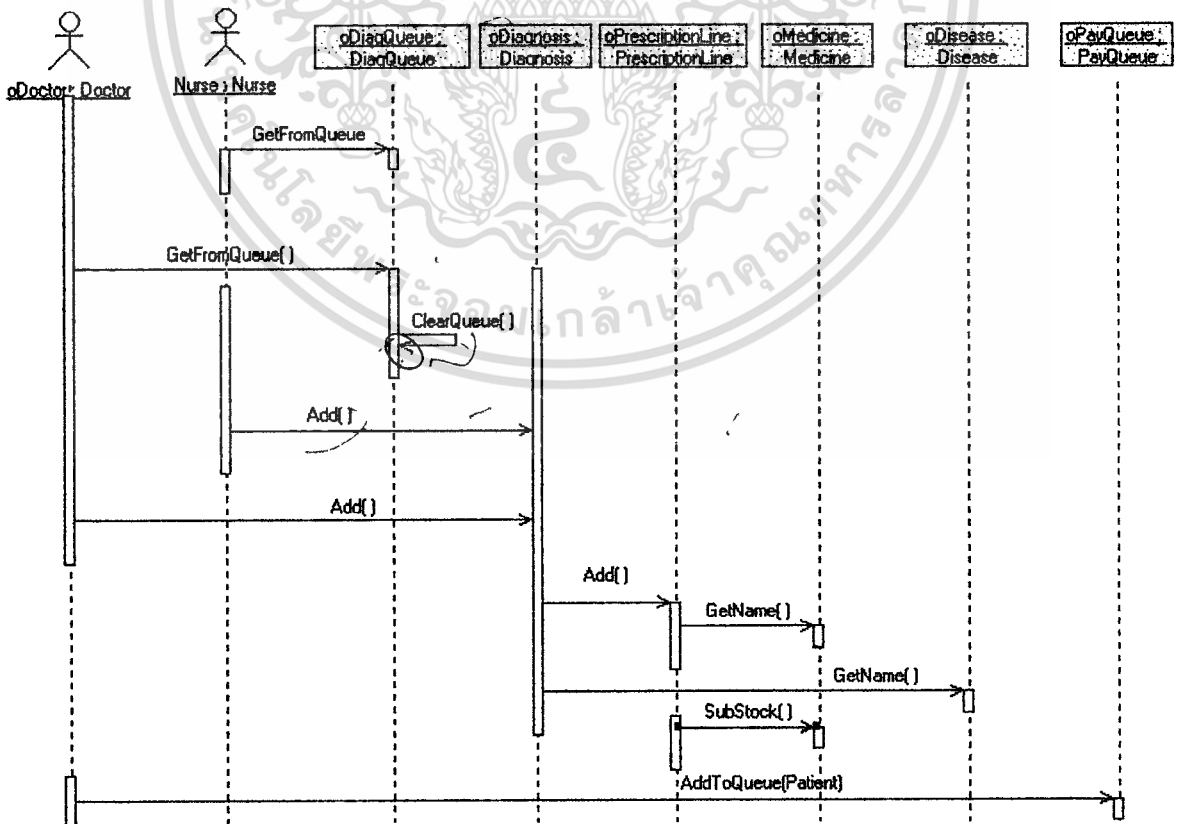
รูปที่ 4.19 Sequence Diagram แสดงกระบวนการจัดการข้อมูลคนไข้ที่มารับการรักษา

2.2) กระบวนการรับการตรวจรักษาจากแพทย์

- เมื่อพยาบาลจะทำการแพทย์จะทำการตรวจรักษาคนไข้ จะดึงข้อมูลคนไข้จากลำดับการรอรับการตรวจรักษา โดย Object Doctor จะส่ง Message GetFromQueue เพื่อดึง Object Patient ขึ้นมาว่าจะตรวจคนไข้รายใดต่อ
- จากนั้น Object DiagQueue จะส่ง Message ClearQueue เพื่อให้ตัวเองทำ การ Clear Object Patient ที่ได้รับการเรียกเพื่อรับการตรวจรักษาจากแพทย์แล้ว ออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อได้ข้อมูลคนไข้ที่จะทำการตรวจรักษาแล้ว Object Doctor จะส่ง Message Add ไปยัง Object Diagnosis เพื่อสร้าง Object ของการตรวจครั้งนั้นๆ เช่นเดียวกับที่ Object Nurse ส่งMessage Add ไป เพื่อใส่ข้อมูลการตรวจพื้นฐาน
- โดยในการตรวจจะมีการส่ง Message ไปยัง Object PrescriptionLine เพื่อเป็นการเพิ่มข้อมูลการสั่งยา
- Object PrescriptionLine จะมีการส่ง Message Get ไปยัง Object Medicine เพื่อเป็นการดึงเอาชื่อยา มาแสดงตามการตรวจวินิจฉัยนั้นๆ และจะส่ง Message SubStock เพื่อทำการลดปริมาณยาตามที่ได้สั่งจ่ายยาไปแล้วหลังจากทำการบันทึกข้อมูลการตรวจนั้น
- และในการวินิจฉัยโรคจะมีการส่ง Message Get ไปยัง Object Disease เพื่อเป็นการดึงเอาชื่อโรค มาแสดงตามการตรวจวินิจฉัยนั้นๆ
- เมื่อแพทย์ทำการตรวจวินิจฉัยเรียบร้อยแล้ว นั่นคือ Object Doctor จะมีการส่ง Message AddToQueue ไปยัง Object PayQueue เพื่อเป็นการจัดลำดับให้คนไข้รอรับยาและชำระเงินต่อไป

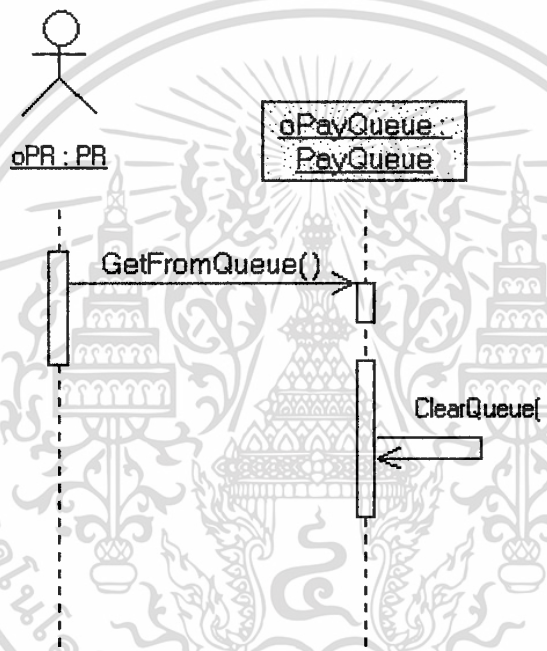


รูปที่ 4.20 Sequence Diagram แสดงกระบวนการรับการตรวจรักษาจากแพทย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3) กระบวนการชำระเงินและรับยา

- เมื่อคนไข้ได้รับการตรวจเรียบร้อยแล้วจะมารอชำนะเงินและรับยา โดยที่ PR จะเป็นผู้ทำการ ดึงข้อมูลคนไข้ในคิวการชำระเงินและจ่ายยา โดย Object PR จะส่ง Message GetFromQueue ไปยัง Object PayQueue เพื่อดึงข้อมูลคนไข้ตามลำดับคิว
- หลังจากคนไข้ได้ทำการชำระเงินและรับยาเรียบร้อยแล้ว Object PayQueue จะ ทำการลบลำดับคิวโดยเรียกใช้ Operation ClearQueue



รูปที่ 4.21 Sequence Diagram แสดงกระบวนการชำระเงินและรับยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การพัฒนาระบบงาน

เมื่อได้ทำการวิเคราะห์และออกแบบระบบงานเป็นที่เรียบร้อยแล้ว จะมีการพิจารณาถึงเครื่องมือที่จะนำมาใช้ในการพัฒนาระบบงาน โดยพิจารณาหาเครื่องมือที่มีประสิทธิภาพ เหมาะสมกับระบบที่จะพัฒนาขึ้น และต้องเป็นเครื่องมือซึ่งมีความยืดหยุ่นในการใช้งาน สะดวก แก้ไขได้โดยง่าย ในคณะเดียวการ ฐานข้อมูลที่ใช้จะต้องรองรับต่อปริมาณข้อมูลที่มีในระบบ รวมทั้งเครื่องมือต่างๆที่จะนำมาใช้ในการพัฒนาระบบนี้ควรจะใช้งานร่วมกันได้เป็นอย่างดี

เมื่อพิจารณาแล้ว ได้เลือกพัฒนาระบบโดยใช้เครื่องมือต่างๆ คือ

- Rational Rose 2000 Enterprise Edition เป็น Tools ช่วยในการสร้าง UML (Unified Modeling Language)
- Microsoft Visual Basic 6 เป็น Tools ในการพัฒนาโปรแกรม
- Cache' Database เป็นระบบจัดการฐานข้อมูลเชิงวัตถุ
- Crystal Report 6 เป็นเครื่องมือในการสร้างรายงาน

5.1 Rational Rose 2000 Enterprise Edition

เนื่องจากในการวิเคราะห์และออกแบบระบบเชิงวัตถุนั้น เพื่อที่จะสามารถสื่อถึงแนวความคิดความเข้าใจได้อย่างชัดเจน จึงควรนำเสนอด้วยแบบจำลองเชิงวัตถุ หรือ แบบจำลอง Unified Modeling Language (UML)

Rational Rose จึงเป็น Tools ในการสร้างแบบจำลองดังกล่าวที่เหมาะสมและกำลังเป็นที่นิยม ซึ่งนอกจาก Rational Rose จะมีความสามารถเป็น Drawing Tools ที่สร้างความสะดวกในการสร้างแบบจำลอง UML แล้ว Rational Rose ยังมีความสามารถที่เหมาะสมต่อการนำมาใช้พัฒนาระบบจัดการข้อมูลโดยหลักการเชิงวัตถุในครั้งนี้ กล่าวคือ

- Rational Rose สามารถ Generate Code เบื้องต้น ได้หลายภาษาโปรแกรม เช่น Visual Basic ,Java ,C++ ,Delphi เป็นต้น และยังสามารถ Update Diagram ต่างๆที่สร้างไว้ ได้ด้วย Code Program ของภาษาดังกล่าวอีกด้วย
- Rational Rose สามารถ Export Class ที่ได้จากการออกแบบ ไปยัง Database ต่างๆได้ เช่น Cache' ,Oracle เป็นต้น รวมทั้ง ยังสามารถ Import Class จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Object Oriented Database เพื่อทำการ Update Diagram ต่างๆ ที่ได้สร้างไว้ใน Rational Rose ได้ด้วย

5.2 Microsoft Visual Basic6

ในการพัฒนาระบบครั้งนี้ ได้เลือก Visual Basic6 เป็น Tools ที่ใช้ในการพัฒนาระบบเนื่องจาก

5.2.1 Visual Basic สามารถพัฒนาโปรแกรมบน Windows และมีเครื่องมือที่ใช้ในการช่วยเขียนโปรแกรมได้ง่ายขึ้น

5.2.2 สามารถทำงานร่วมกับฐานข้อมูลได้หลายชนิด รวมทั้งฐานข้อมูล Cache' ซึ่งเป็นระบบจัดการฐานข้อมูลเชิงวัตถุ (Object Oriented Database)

5.2.3 สามารถ Generate Program Code ได้บางส่วน จาก Rational Rose 2000 และสามารถนำ Program Code กลับไป Update Diagram ที่สร้างไว้โดย Rational Rose ได้

5.3 กระบวนการในการวิเคราะห์ ออกแบบ และพัฒนาระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ

5.3.1 ติดตั้งโปรแกรม Microsoft Visual Basic 6 ,Cache' ,Rational Rose 2000 Enterprise Edition ,Crystal Report 6

5.3.2 วิเคราะห์และออกแบบระบบเชิงวัตถุ โดยใช้ Rational Rose เป็น Tools ในการสร้าง Diagram ต่างๆ และ Export Class ที่ได้จาก Class Diagram มายังระบบจัดการฐานข้อมูลเชิงวัตถุ Cache' และ Generate Code Program บางส่วน เป็น Visual Basic

5.3.3 สร้างจอภาพตามที่ ออกแบบระบบ

5.3.4 เขียนโปรแกรม (Coding)

5.3.5 สร้างรายงาน

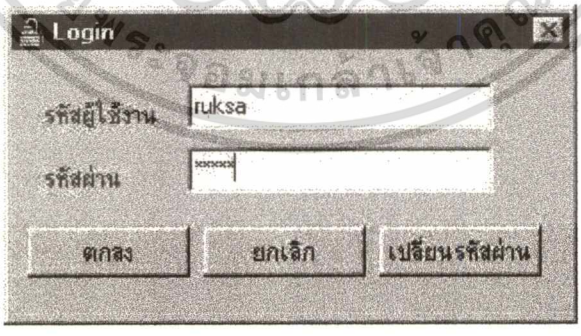
5.3.6 ทดสอบระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 จอภาพและรายงานของระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ

5.4.1. จอภาพแสดงการ Login เข้าสู่ระบบ จะเป็นจอภาพเริ่มต้นของระบบเพื่อตรวจสอบสิทธิ์ในการใช้งานระบบโดยจะมีความสัมพันธ์กับ Class ProgramUser นั่นคือ

- เมื่อทำการ Login แล้วจะไปค้นหา Object ที่มีรหัสผู้ใช้งาน (UserName) ตรงตามที่ระบุในจอภาพ
- กรณีที่พบ UserName ดังกล่าว จะทำการตรวจสอบว่า รหัสนั้นถูกต้องตรงตามข้อมูลใน Database หรือไม่
- เมื่อตรวจสอบว่า UserName และ Password ถูกต้องแล้ว จะทำการเรียก Method LoginUser เพื่อ Set ค่าให้ UserSession เป็น True อันหมายถึงกำลังใช้งานระบบอยู่ ในกรณีที่ผู้ใช้งานเป็นแพทย์จะมีผลต่อจอภาพแสดงการจัดลำดับเข้ารับการตรวจซึ่งจะกล่าวถึงต่อไป
- จะทราบว่าผู้ที่เข้าระบบมานั้นมีสิทธิ์ในการใช้ระบบอย่างไร เช่นกรณีที่เป็นแพทย์จะสามารถเข้าสู่จอภาพการตรวจรักษาได้ ในขณะที่พยาบาลจะสามารถเข้าสู่จอภาพการตรวจเบื้องต้นได้เท่านั้น ส่วนประชาสัมพันธ์ จะไม่สามารถเข้าสู่จอภาพการตรวจได้เลย หรือในกรณีที่ผู้ใช้งานเป็น Administrator ของระบบ จะสามารถเข้าสู่จอภาพข้อมูลพื้นฐานของระบบบางจอภาพได้ เช่น ข้อมูลพนักงาน ในขณะที่ผู้ใช้งานที่เป็น User ไม่สามารถเข้าไปใช้งานได้
- จากจอภาพนี้ ผู้ใช้งานระบบสามารถทำการเปลี่ยนรหัสผ่านได้เลย เมื่อใส่รหัสผู้ใช้งานและรหัสผ่านเดิมถูกต้อง

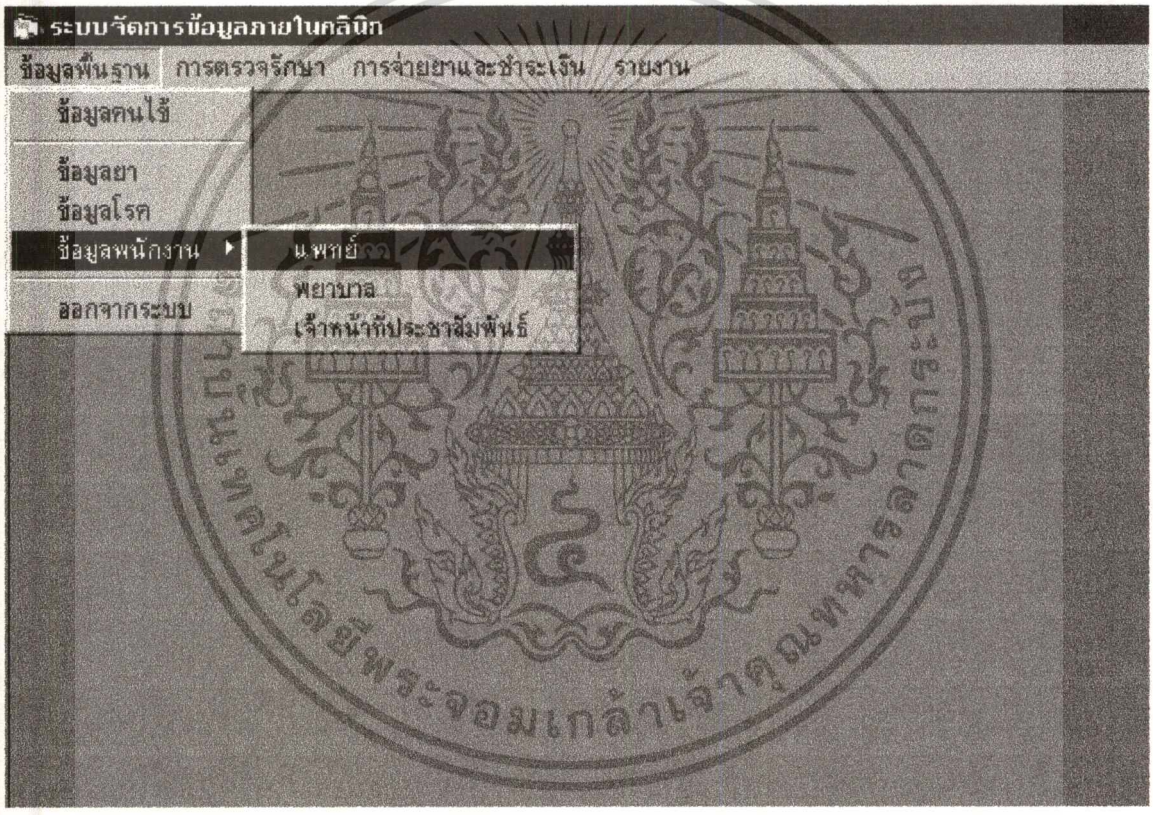


รูปที่ 5.1 จอภาพแสดงการ Login เข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2. จอภาพหลักของระบบ เป็นจอภาพหลักเพื่อเลือกเข้าไปปฏิบัติงานย่อยภายในระบบ โดยแบ่งเป็นเมนูหลักต่างๆ คือ

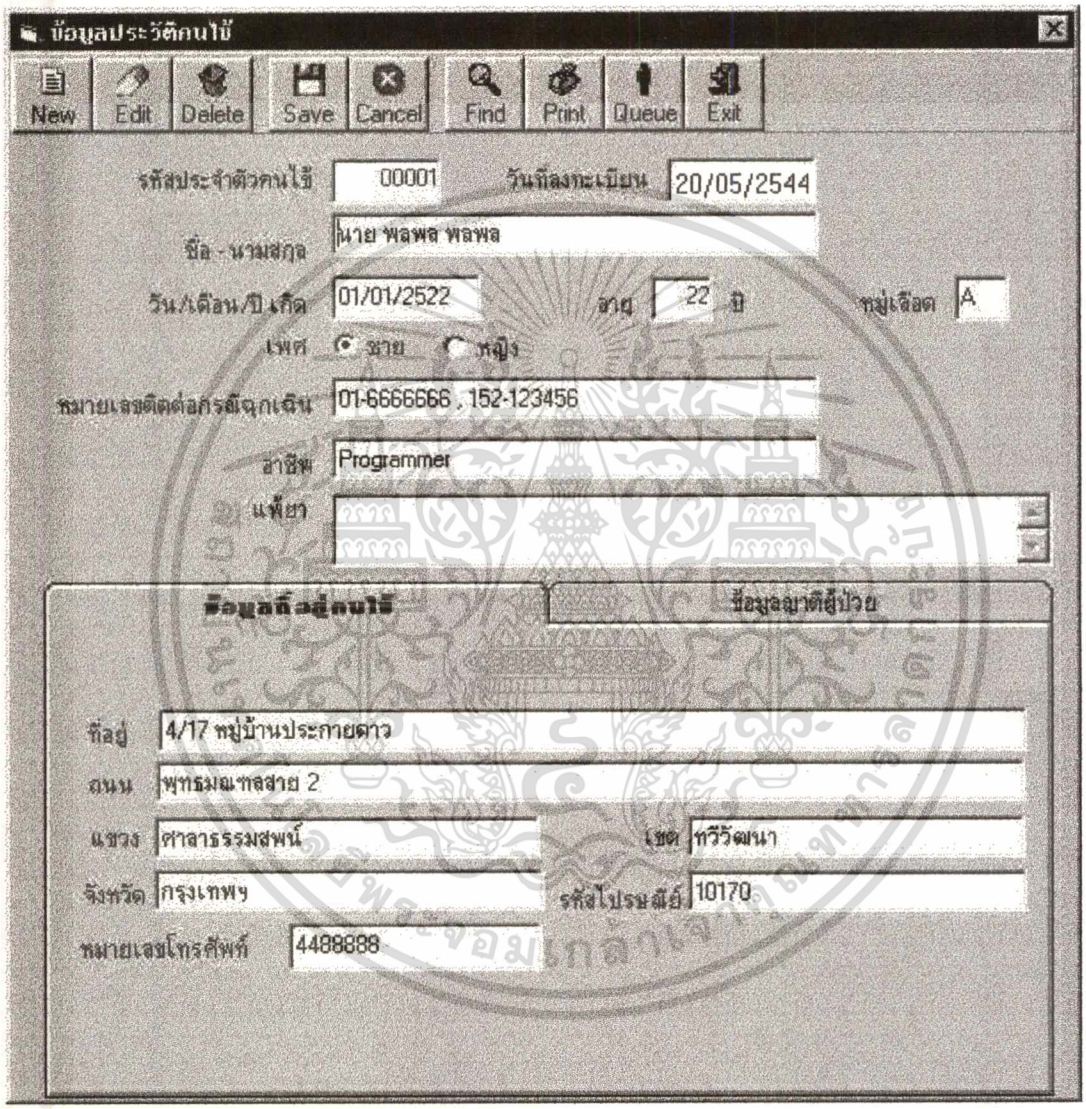
1) เมนูข้อมูลพื้นฐาน
2) เมนูการจัดคิว
3) เมนูการตรวจรักษา
4) เมนูการจ่ายยาและการชำระเงิน
5) เมนูรายงาน



รูปที่ 5.2 จอภาพหลักของระบบ

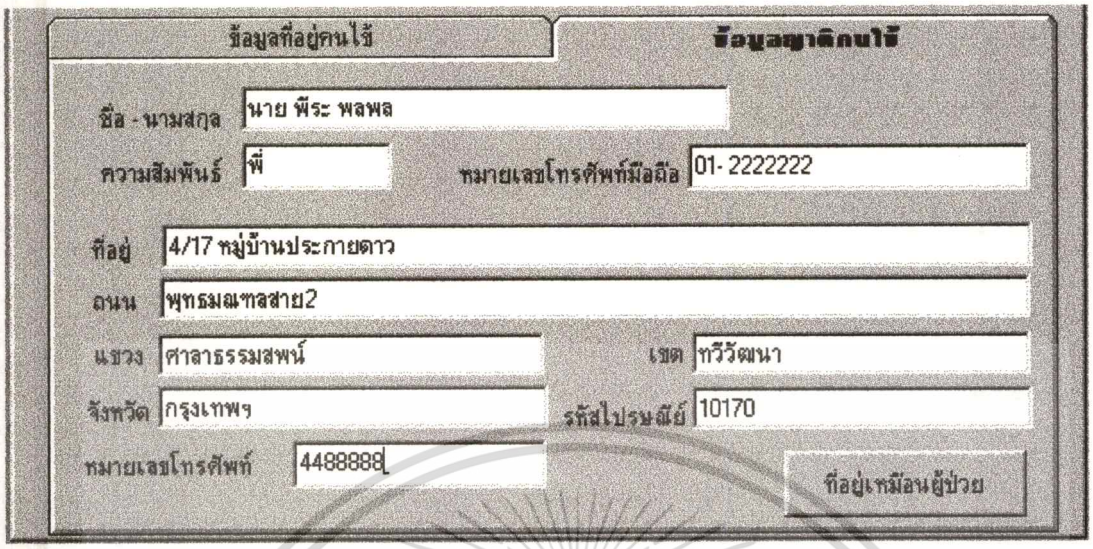
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.3 จอภาพแสดงข้อมูลคนไข้ จะแสดงข้อมูลของคนไข้ที่มีรับการรักษาที่คลินิก ซึ่งจะมีความสัมพันธ์กับ Class Patient นอกจากนี้ยังแสดงถึงข้อมูลญาติผู้ป่วย ซึ่งสัมพันธ์กับ Class ContactPerson อีกด้วย



รูปที่ 5.3 จอภาพแสดงข้อมูลคนไข้ แสดง Tab ของ ข้อมูลที่อยู่ของคนไข้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



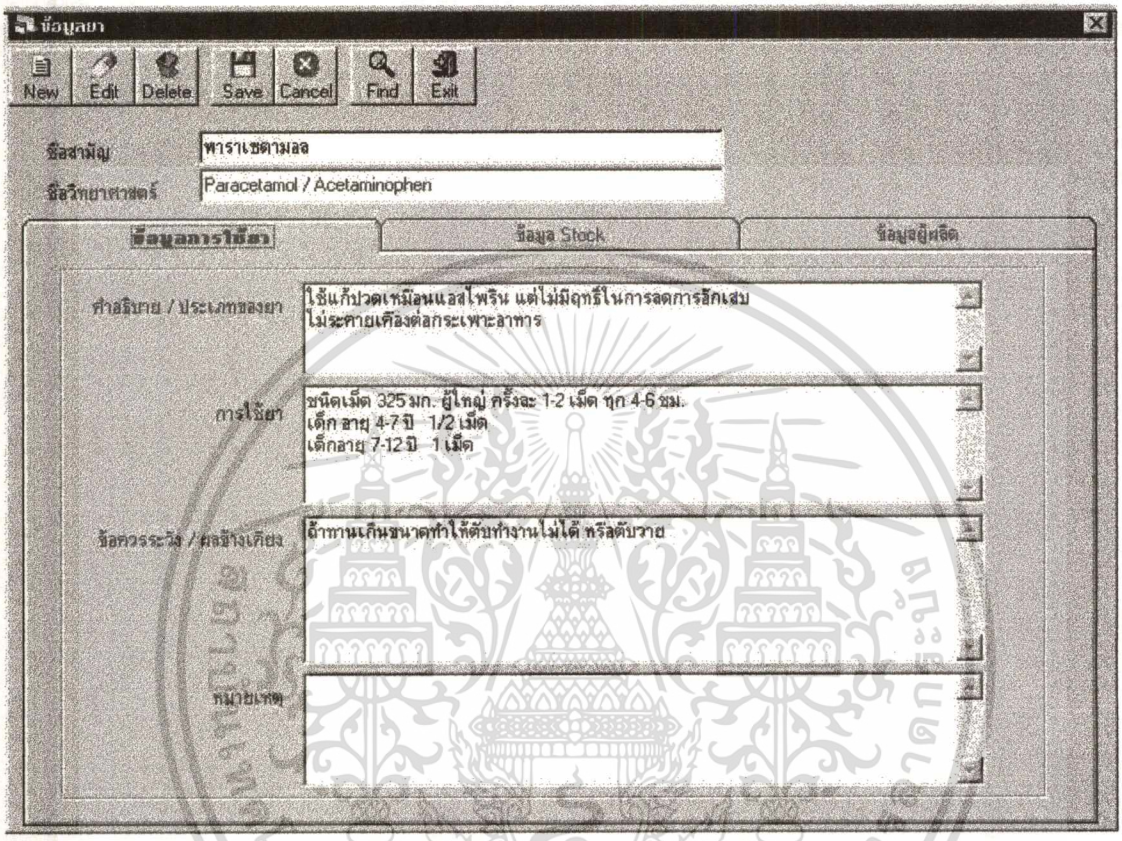
รูปที่ 5.4 จอภาพข้อมูลคนไข้ แสดง Tab ของ ข้อมูลญาติคนไข้

ปุ่มการใช้งานในจอภาพแสดงข้อมูลคนไข้

- New : เพื่อทำการเพิ่มข้อมูลคนไข้
- Edit : เพื่อทำการแก้ไขข้อมูลคนไข้
- Delete : เพื่อทำการลบข้อมูลคนไข้
- Save : เพื่อทำการบันทึกข้อมูลคนไข้
- Cancel : เพื่อทำการยกเลิกการเพิ่มหรือแก้ไขข้อมูลคนไข้
- Find : เพื่อทำการค้นหาข้อมูลคนไข้
- Print : เพื่อทำการพิมพ์บัตรคนไข้
- Queue : เพื่อทำการนำคนไข้เข้าสู่คิวรอรับการตรวจ
- Exit : ออกจากจอภาพ

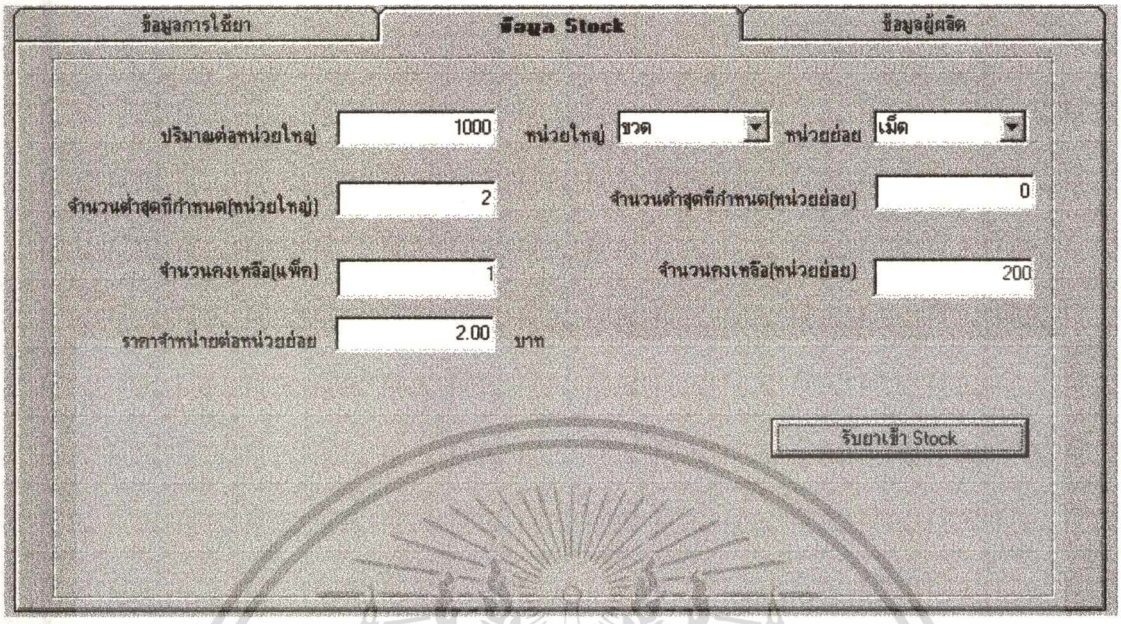
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.4 จอภาพแสดงข้อมูลยา จะแสดง ข้อมูลยาที่มีอยู่ในคลังยาของคลินิก ซึ่งมีความสัมพันธ์กับ Class Medicine

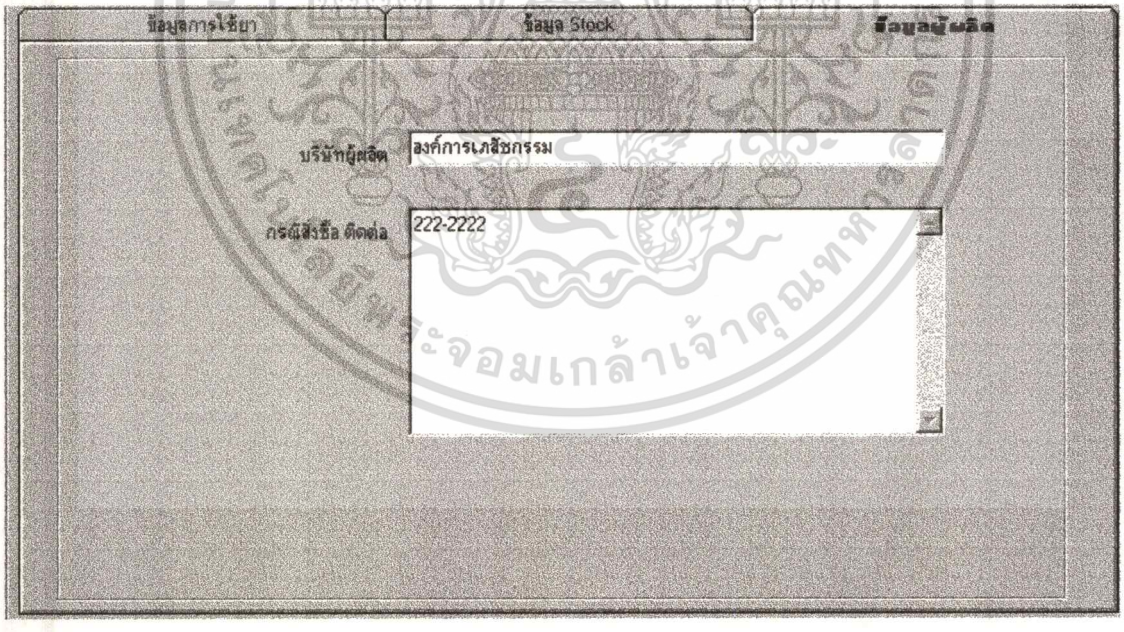


รูปที่ 5.5 จอภาพแสดงข้อมูลยา แสดง Tab ของ ข้อมูลการใช้ยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 จอภาพแสดงข้อมูลยา แสดง Tab ของ ข้อมูล Stock



รูปที่ 5.7 จอภาพแสดงข้อมูลยา แสดง Tab ของ ข้อมูลผู้ผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอภาพแสดงข้อมูลยา จะแบ่งข้อมูลเป็น 3 Tab คือ

5.4.9.1 ข้อมูลการใช้ยา : จะแสดงข้อมูลการใช้ยาโดยทั่วไป

5.4.9.2 ข้อมูล Stock : แสดงข้อมูลคลังยา และการเพิ่มยาเข้าสู่คลังยา ในกรณีที่จำนวนยาคงเหลือน้อยกว่าจำนวนขั้นต่ำที่ระบุไว้ จำนวนยาคงเหลือจะแสดงเป็นสีแดง

5.4.9.3 ข้อมูลผู้ผลิต : แสดงข้อมูลผู้ผลิตยา และข้อมูลในการติดต่อ

ปุ่มการใช้งานในจอภาพแสดงข้อมูลยา

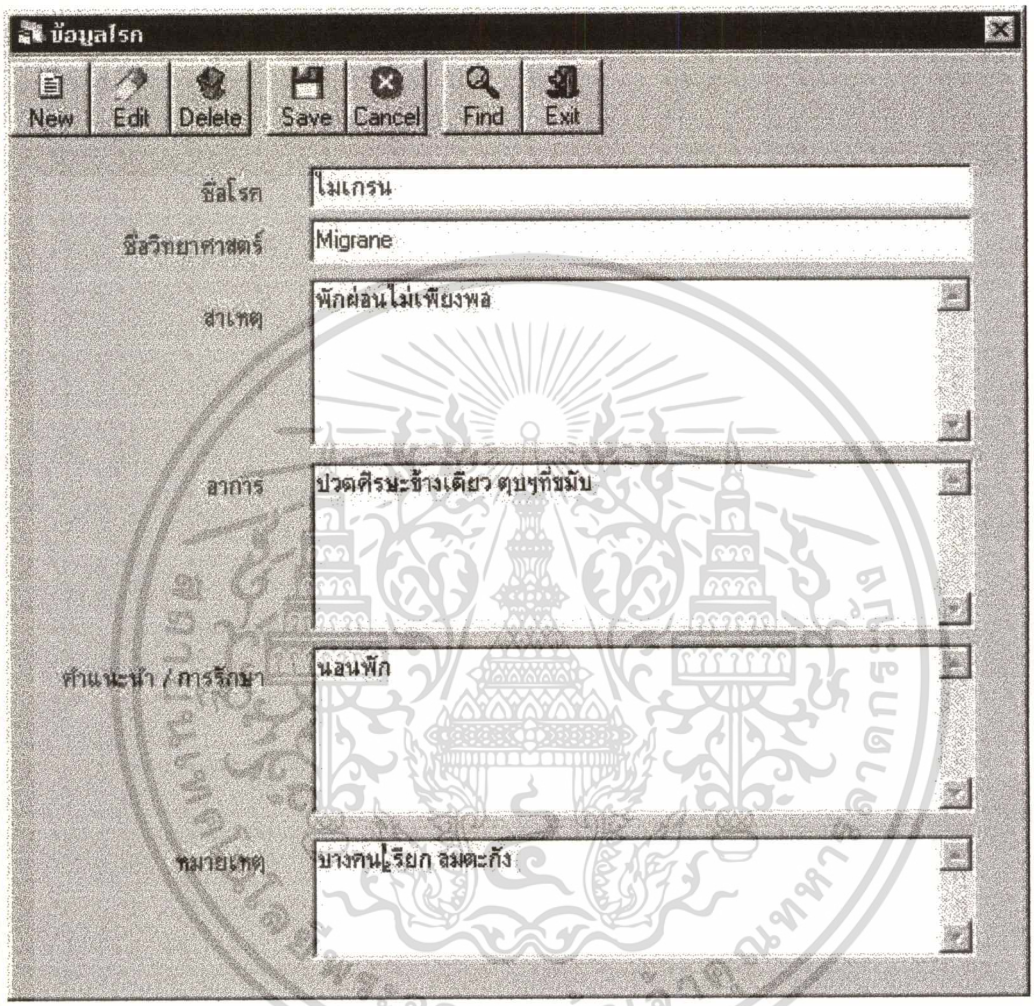
- New : เพื่อทำการเพิ่มข้อมูลยา
- Edit : เพื่อทำการแก้ไขข้อมูลยา
- Delete : เพื่อทำการลบข้อมูลยา
- Save : เพื่อทำการบันทึกข้อมูลยา
- Cancel : เพื่อทำการยกเลิกการเพิ่มหรือแก้ไขข้อมูลยา
- Find : เพื่อทำการค้นหาข้อมูลยา
- Exit : ออกจากจอภาพ
- รับยาเข้า Stock : เป็นการเรียกจอภาพแสดงการรับยาเข้า Stock เพื่อเพิ่มจำนวนยาใน Stock

5.4.5 จอภาพแสดงการรับยาเข้า Stock จะเป็นการเพิ่มจำนวนยาใน คลังยา

รูปที่ 5.8 จอภาพแสดงการรับยาเข้า Stock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

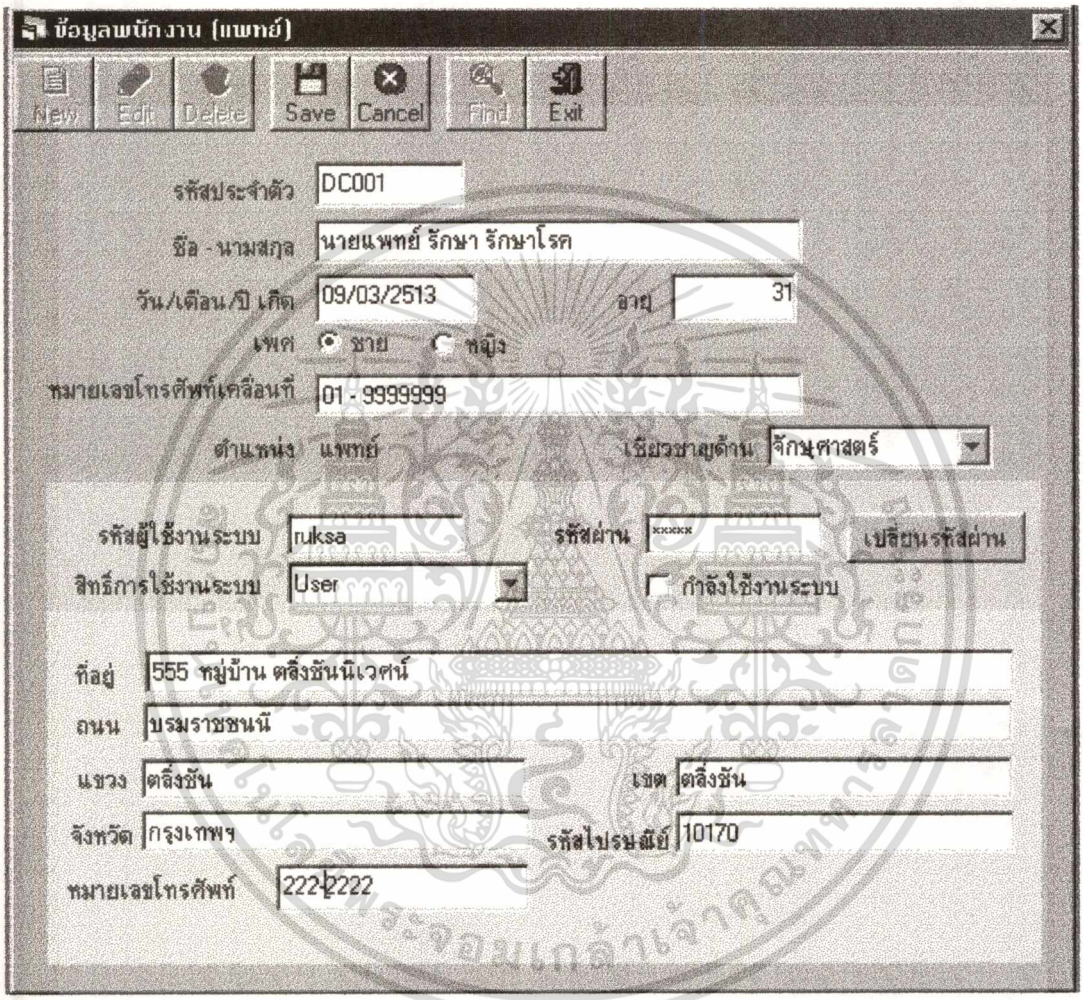
5.4.6 จอภาพแสดงข้อมูลโรค จะแสดงข้อมูลโรคซึ่งจัดเก็บไว้เผื่อการอ้างอิงในการวินิจฉัยโรคของแพทย์



รูปที่ 5.9 จอภาพแสดงข้อมูลโรค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.7 จอภาพแสดงข้อมูลพนักงาน จะแสดง ข้อมูลพนักงานในคลินิก ได้แก่ แพทย์ พยาบาล และประชาสัมพันธ์ โดยผู้ใช้งานระบบที่มีสิทธิ์ในการใช้งานเป็น Administrator เท่านั้น จึงจะสามารถเข้าสู่จอภาพนี้ได้



รูปที่ 5.10 จอภาพแสดงข้อมูลพนักงาน

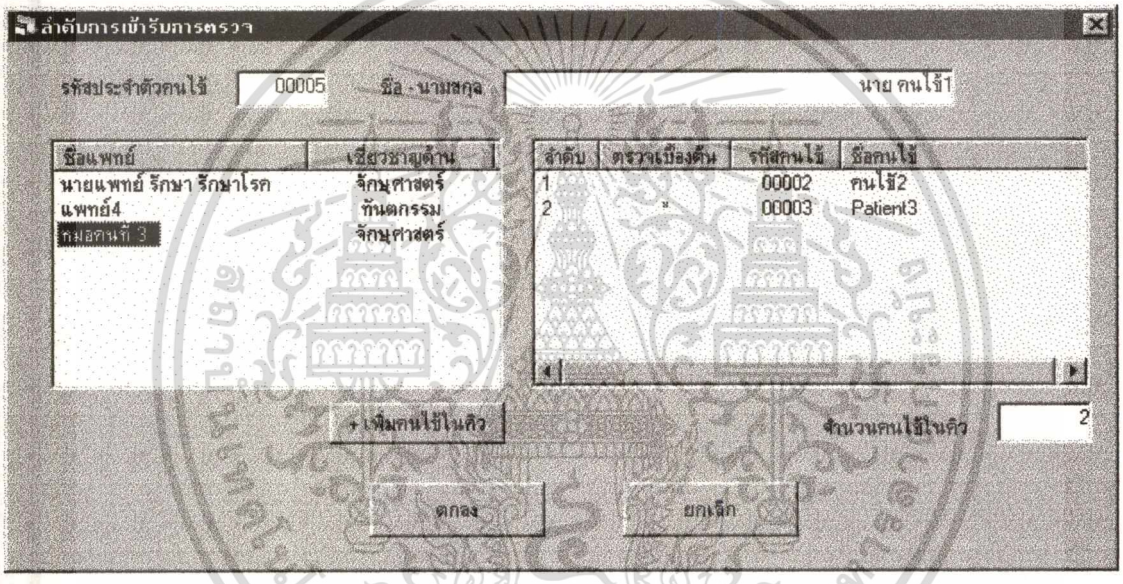
ปุ่มการใช้งานในจอภาพแสดงข้อมูลพนักงาน

- New : เพื่อทำการเพิ่มข้อมูลพนักงาน
- Edit : เพื่อทำการแก้ไขข้อมูลพนักงาน
- Delete : เพื่อทำการลบข้อมูลพนักงาน
- Save : เพื่อทำการบันทึกข้อมูลพนักงาน
- Cancel : เพื่อทำการยกเลิกการเพิ่มหรือแก้ไขข้อมูลพนักงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Find : เพื่อทำการค้นหาข้อมูลพนักงาน
- Exit : ออกจากจอภาพ
- เปลี่ยนรหัสผ่าน : เพื่อเป็นการเปลี่ยนรหัสผ่านของพนักงาน โดย Administrator ของระบบ

5.4.8 จอภาพแสดงลำดับการเข้ารับการตรวจ เพื่อเลือกว่าจะให้คนไข้ที่เพิ่งมาติดต่อนั้นพบแพทย์ท่านใด และเป็นลำดับที่เท่าไร



รูปที่ 5.11 จอภาพแสดงลำดับการเข้ารับการตรวจ

ปุ่มการใช้งานในจอภาพแสดงลำดับการเข้ารับการตรวจ

- +เพิ่มคนไข้ในคิว : เพื่อจัดคนไข้เข้าสู่ลำดับการตรวจของแพทย์ที่เลือก
- ตกลง : เพื่อยืนยันการเพิ่มข้อมูลคนไข้ในลำดับการตรวจ
- ยกเลิก : เพื่อยกเลิกการเพิ่มข้อมูลคนไข้ในลำดับการตรวจ

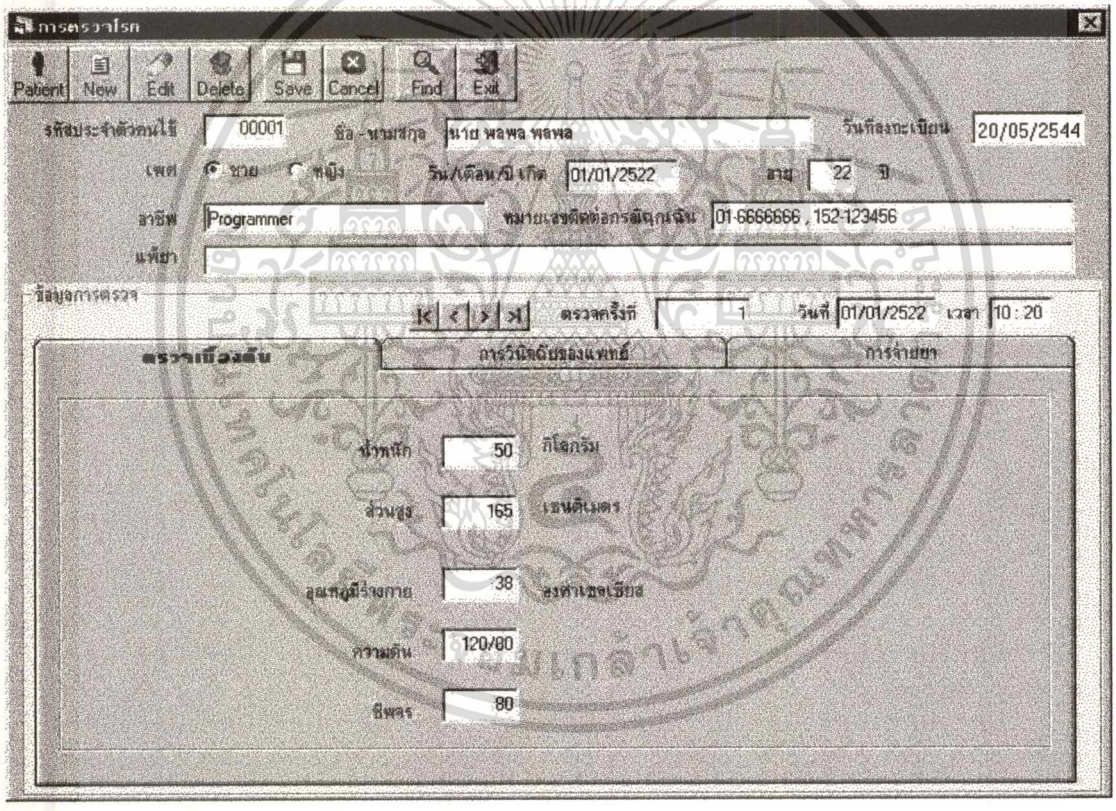
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.9 จอภาพแสดงข้อมูลการตรวจโรค เพื่อบันทึก เรียกดูข้อมูล ประวัติการตรวจโรค จอภาพแสดงข้อมูลยา จะแบ่งข้อมูลเป็น 3 Tab คือ

5.4.9.4 ข้อมูลการตรวจเบื้องต้น : เพื่อให้พยาบาลเป็นผู้ทำการตรวจเบื้องต้น

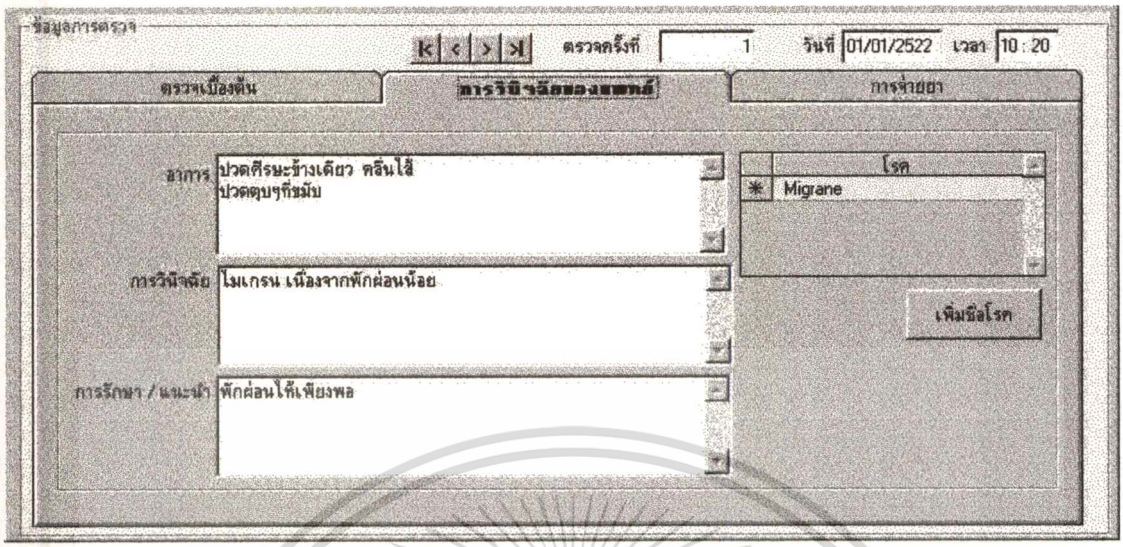
5.4.9.5 ข้อมูลการวินิจฉัยของแพทย์ : ผู้ใช้งานระบบที่เป็นแพทย์เท่านั้นจึงจะสามารถเพิ่ม แก้ไข หรือ เรียกดูข้อมูล จาก Tab นี้ได้

5.4.9.6 ข้อมูลการจ่ายยา : เป็นรายการยาที่แพทย์เป็นผู้สั่ง ผู้ใช้งานอื่นสามารถเห็น Tab นี้ได้เพื่อการจ่ายยาและชำระเงิน แต่ไม่สามารถเข้าไปแก้ไขได้



รูปที่ 5.12 จอภาพแสดงข้อมูลการตรวจโรค แสดง Tab การตรวจเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.13 จอภาพแสดงข้อมูลการตรวจโรค แสดง Tab การวินิจฉัยของแพทย์

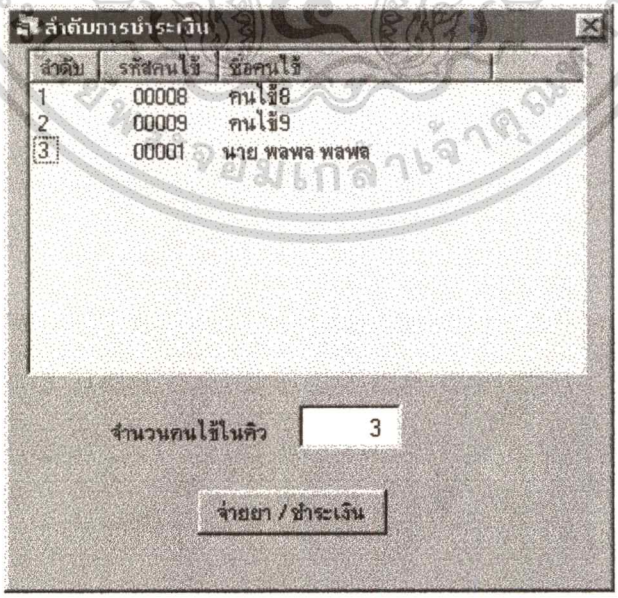
รูปที่ 5.14 จอภาพแสดงข้อมูลการตรวจโรค แสดง Tab การจ่ายยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่มการใช้งานในจอภาพแสดงข้อมูลการตรวจโรค

- Patient : เพื่อทำการนำคนไข้จากคิวรอรับการตรวจเข้ารับการตรวจ
- New : เพื่อทำการเพิ่มข้อมูลการตรวจ
- Edit : เพื่อทำการแก้ไขข้อมูลการตรวจ
- Delete : เพื่อทำการลบข้อมูลการตรวจ (Administrator สามารถทำได้เท่านั้น)
- Save : เพื่อทำการบันทึกข้อมูลการตรวจ
- Cancel : เพื่อทำการยกเลิกการเพิ่มหรือแก้ไขข้อมูลการตรวจ
- Find : เพื่อทำการค้นหาข้อมูลการตรวจ
- Exit : ออกจากจอภาพ
- เพิ่มชื่อโรค : เพื่อเพิ่มชื่อโรคตามการวินิจฉัยของแพทย์
- เพิ่มยา : เพื่อสั่งยาตามการวินิจฉัยของแพทย์
- ลบยา : เพื่อลบยาที่อยู่ในรายการสั่งยา ตามการวินิจฉัยของแพทย์
- ชำระเงิน : สำหรับพนักงานที่มีหน้าที่จัดยาและรับชำระเงินเข้ามาดูรายการยาที่สั่งและจัดยาและรับชำระเงิน

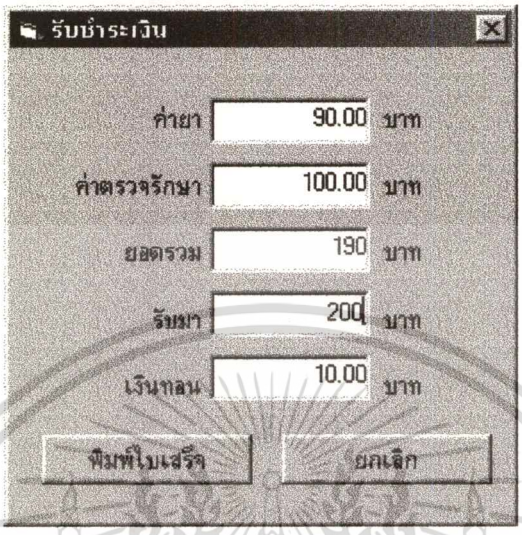
5.4.10 จอภาพแสดงลำดับการชำระเงิน เพื่อดึงข้อมูลคนไข้ตามลำดับการรอรับยาและชำระเงินข้อมูล



รูปที่ 5.15 จอภาพแสดงลำดับการชำระเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.11 จอภาพแสดงการรับชำระเงิน เพื่อรับชำระเงินจากคนไข้



รูปที่ 5.16 จอภาพแสดงการรับชำระเงิน

5.4.12 บัตรคนไข้ บัตรที่ออกให้แก่คนไข้

รูปที่ 5.17 บัตรคนไข้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.13 รายงานสรุปปริมาณยาคงเหลือ แสดงข้อมูลสรุปปริมาณยาในคลัง โดยกรณีที่ยาชนิดใดเหลือจำนวนต่ำกว่าปริมาณต่ำสุดที่กำหนด จะแสดงเครื่องหมาย "*"

รายงานสรุปปริมาณยาคงเหลือ

ประจำวันที่ 01/02/2544

	ชื่อยา	ปริมาณคงเหลือ (แพ็ค)	หน่วย (แพ็ค)	ปริมาณคงเหลือ (ย่อย)	หน่วย (ย่อย)	ปริมาณต่อแพ็ค	ปริมาณต่ำสุด (แพ็ค)	ปริมาณต่ำสุด (ย่อย)
*	พาราเซตามอล	1	ขวด	200	เม็ด	1000	2	0
	ยาธาตุน้ำขาว	30	ขวด	30	ขวด	1	20	0
	Medicine1	50	แผง	5	เม็ด	10	20	0
*	Medicine2	3	กล่อง	2	ขวด	10	5	0
*	Medicine3	2	ขวด	100	เม็ด	500	3	0
	Medicine4	30	ขวด	30	ขวด	1	20	0
	Medicine5	100	แผง	5	เม็ด	10	50	0
	Medicine6	15	กล่อง	2	ขวด	10	10	0
	Medicine7	50	แผง	5	เม็ด	10	20	0
*	Medicine8	3	กล่อง	2	ขวด	10	5	0
*	Medicine9	2	ขวด	100	เม็ด	500	3	0
	Medicine10	30	ขวด	30	ขวด	1	20	0
	Medicine11	100	แผง	5	เม็ด	10	50	0
	Medicine12	15	กล่อง	2	ขวด	10	10	0
	Medicine13	10	ขวด	100	เม็ด	500	3	0
*	Medicine14	3	กล่อง	2	ขวด	10	5	0
*	Medicine15	2	ขวด	100	เม็ด	500	3	0
	Medicine16	30	ขวด	30	ขวด	1	20	0
	Medicine17	100	แผง	5	เม็ด	10	50	0
	Medicine18	15	กล่อง	2	ขวด	10	10	0
	Medicine19	10	ขวด	100	เม็ด	500	3	0
	Medicine20	10	ขวด	100	เม็ด	500	3	0

รูปที่ 5.18 รายงานสรุปปริมาณยาคงเหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
บทสรุปและข้อเสนอแนะ

จากการพัฒนาระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุนั้น ได้ผลสรุปและแนวทางข้อเสนอแนะดังนี้

6.1 บทสรุป

การพัฒนาระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุนี้ เป็นการพัฒนาระบบที่อาศัยหลักการการพัฒนาระบบเชิงวัตถุเข้ามาประยุกต์ใช้ในขั้นตอนการวิเคราะห์ออกแบบ ซึ่งมีข้อสรุปดังนี้

- การออกแบบและพัฒนาระบบเชิงวัตถุ สามารถช่วยสะท้อนให้เห็นถึงการทำงานในโลกแห่งความจริงซึ่งต้องเกี่ยวข้องกับวัตถุต่างๆ โดยการประยุกต์ใช้ขั้นตอนเชิงวัตถุ ช่วยให้ผู้พัฒนาสามารถ ออกแบบวิเคราะห์ระบบ ได้อย่างมีประสิทธิภาพยิ่งขึ้น สามารถนำกลับมาใช้ใหม่ได้ และช่วยให้เวลาในการพัฒนาระบบเร็วขึ้นด้วย
- การประยุกต์ใช้ระบบจัดการฐานข้อมูลเชิงวัตถุ Cache' สามารถช่วยลดปัญหาในเรื่องที่ Relational Database ไม่สามารถรองรับได้ เช่น เรื่องขนาดของข้อมูลบางประเภท หรือ การเปลี่ยนแปลงข้อมูลของ Object Identity
- ในการพัฒนาระบบงานนี้ได้เลือกใช้เครื่องมือที่ช่วยในการออกแบบเชิงวัตถุ ซึ่งอาศัยหลักการ UML คือ Rational Rose ของบริษัท Rational Software โดยสามารถที่จะใช้งานร่วมกับระบบจัดการฐานข้อมูลเชิงวัตถุ Cache'
- ผลที่ได้จากการออกแบบ Class ต่างๆในระบบตามหลักการเชิงวัตถุนั้น อาจมีผลที่แตกต่างกันออกไปตามความต้องการของระบบงานที่สนใจ
- จาก ระบบจัดการข้อมูลภายในคลินิก พบว่าสามารถทำการพัฒนาได้อย่างรวดเร็วและมีประสิทธิภาพ เป็นผลมาจาก การใช้เวลาอย่างละเอียดและรอบคอบ ตามขั้นตอนการวิเคราะห์และออกแบบเชิงวัตถุ ซึ่งอาศัยหลักการของ UML และเครื่องมือที่กล่าวมาข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ข้อเสนอแนะ

จากการทดลองใช้งานระบบที่พัฒนาขึ้นนั้น พบว่ายังคงมีบางจุดที่ยังต้องปรับปรุงแก้ไขในส่วนรายละเอียดการทำงานของระบบ ไม่ว่าจะเป็นการทำหน้าจอให้มีความน่าใช้งานมากขึ้น หรือการเพิ่มความสะดวกในการใช้งาน เช่น การทำโปรแกรมติดตั้งอัตโนมัติเพื่อให้ผู้ใช้งานสามารถติดตั้ง Client Software Component ได้สะดวก

จากการออกแบบและพัฒนาระบบจัดการข้อมูลภายในคลินิกเชิงวัตถุนี้ ทำให้ได้รู้จักการนำเทคโนโลยีเชิงวัตถุมาใช้ในการแก้ปัญหาในการออกแบบ และ วิเคราะห์ระบบงาน อย่างเป็นขั้นตอน ฉะนั้น หน่วยงานหรือองค์กรต่าง ๆ จึงสมควรอย่างยิ่งที่จะนำเทคโนโลยีดังกล่าวมาประยุกต์ใช้ในการออกแบบ เพื่อประโยชน์ในการวิเคราะห์และวางแผนการพัฒนาระบบ เพื่อลดความผิดพลาดที่อาจจะเกิดขึ้น เพิ่มประสิทธิภาพการทำงานขององค์กร และความสำเร็จขององค์กรในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Booch Grady, Rumbaugh James, Jacobson Ivar. 1999. **The Unified Modeling Language User Guide :** Addison-Wesley.

Dimitroff Donal C., And Chang Ifay F.. 1990. "An Object Oriented Approach to Automating Patient Medical Record. " 82-87: **IEEE.**

Eliens Anton. 1995. **Object-Oriented Software Development :** Addison-Wesley.

Eriksson Hans-Erik and Penker Magnus. 1998. **UML Toolkit** : John Wiley & Sons.

Fowler Martin and Scott Kendall.1998 .**UML Distilled :** Addison-Wesley.

Hoffer Jeffrey A., George Joey F. and Valacich Joseph S.. 1996. **Modern System Analysis and Design :** The Benjamin/Cummings.

Kendall Kenneth E. and E. Kendall Julie. **System Analysis and Design :** Prentice-Hall.

Rational Software Corporation. 1999. **Inside The Unified Modeling Language UML.** [CDROM].

Tkach Daniel, Puttick Richard.1996. **Object Technology In Application Development :** Addison-Wesley Publishing.

Wilkie George. 1994. **Object-Oriented Software Engineering** : Riverside Printing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	น.ส. ฤทัยรัตน์ พัดฉวีวรรณ
เกิด	วันที่ 3 ธันวาคม 2522
สถานที่เกิด	กรุงเทพมหานคร
สถานที่สำเร็จการศึกษา	
ระดับมัธยม	โรงเรียนเตรียมอุดมศึกษา (พญาไท)
ระดับปริญญาตรี	ปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาวิชาเอก สถิติ สาขาวิชาโท วิทยาการคอมพิวเตอร์ และ คณิตศาสตร์ประกันภัย มหาวิทยาลัยธรรมศาสตร์
ปีที่สำเร็จการศึกษา	1999 (2542)
ประวัติการทำงาน	พ.ศ. 2542 – ปัจจุบัน นักโปรแกรม บริษัท โปรเกรสซอฟท์แวร์ จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้