

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

งานพัฒนาระบบศูนย์กลางการจัดเก็บข้อมูลการรักษาทางทันตกรรม
ผ่านเครือข่ายอินเทอร์เน็ต

Application Development of a Central Dental Clinical Data System
via the Internet



วัน เดือน ปี.....	19	ม.ค.	2550
เลขทะเบียน.....	01914		
เลขเรียกหนังสือ.....	ศษ.	๑223ง	๒5๕๕
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."			

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2545
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวเรื่อง	งานพัฒนาระบบศูนย์กลางการจัดเก็บข้อมูลการรักษาทางทันตกรรมผ่าน ผ่านเครือข่ายอินเทอร์เน็ต
นักศึกษา	นายอนุวัฒน์ กนกวิรุพท์
อาจารย์ที่ปรึกษา	ผศ. ดร. ประจวบ วาณิชชวาล
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

การรักษาโรคทางด้านทันตกรรมเป็นสิ่งจำเป็นสำหรับการดำรงชีวิตในปัจจุบันสำหรับผู้ที่ต้องการมีสุขภาพฟันที่ดี โดยจะมีการรักษาเกือบตลอดอายุของคนไข้ ซึ่งนับเป็นระยะเวลาที่ยาวนาน ในการรักษาโรคนั้นสามารถเลือกเข้ารับการรักษาได้จากทั้งจากโรงพยาบาลรัฐบาล โรงพยาบาลเอกชน สาธารณสุขจังหวัด คลินิก เป็นต้น ข้อมูลของผู้ที่เข้ารับการรักษาจะถูกเก็บไว้ที่หน่วยงานที่คนไข้เข้ารับการรักษา จากการศึกษาที่ต้องมีการรักษาอยู่เป็นประจำ จึงมักมีเหตุให้ต้องเปลี่ยนสถานที่รักษาอยู่เสมอ เป็นต้นว่า มีการเปลี่ยนแปลงที่อยู่อาศัยทำให้ไม่สะดวกต่อการเดินทาง ราคาการรักษา ความพึงพอใจในผลการรักษา เป็นต้น เมื่อมีการย้ายไปรับการรักษาในสถานที่ใหม่ ข้อมูลการรักษาเดิมก็จะถูกทิ้งเอาไว้ไม่สามารถนำมาใช้ประโยชน์ได้ ทั้งนี้เป็นข้อมูลสำคัญสำหรับคนไข้ ดังนั้นหากสามารถเก็บรักษาข้อมูลของคนไข้เอาไว้ให้สามารถเข้าถึงได้ตลอดเวลา ก็จะเกิดผลดีทั้งต่อตัวคนไข้ และทันตแพทย์ผู้ให้การรักษา รวมถึงเป็นแหล่งข้อมูลสำหรับงานวิจัยทางด้านทันตกรรมต่อไปในอนาคต

ด้วยการนำเทคโนโลยีทางด้านคอมพิวเตอร์และอินเทอร์เน็ต (Internet) เข้ามาใช้ในการบันทึกข้อมูลจากหน่วยงานอิสระต่างๆ เช่น คลินิก การกำหนดวิธีการบันทึกข้อมูลการรักษาให้มีมาตรฐาน จะช่วยให้การรวบรวมข้อมูลที่กระจัดกระจายเข้ามาอยู่ที่ศูนย์กลางเป็นแหล่งเดียวกัน คนไข้จะสามารถทราบประวัติการรักษาของตัวเองได้ทั้งหมด ในขณะที่ทันตแพทย์ทราบถึงประวัติการรักษาที่ผ่านมาของคนไข้ทำให้สามารถวินิจฉัยและตัดสินใจรักษาได้อย่างถูกต้องมากยิ่งขึ้น ในโครงการนี้จะพัฒนาเว็บแอปพลิเคชัน (Web-based Application) เพื่อรับการบันทึกข้อมูลจากเครื่องคอมพิวเตอร์ลูกข่าย (Client) ที่อยู่ตามคลินิกต่างๆ ไปจัดเก็บไว้ในเครื่องคอมพิวเตอร์แม่ข่ายที่ให้บริการ (Web Server) และสามารถทำการเรียกค้นข้อมูลการรักษาจากเครื่องคอมพิวเตอร์แม่ข่ายได้ตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Application Development of a Central Dental Clinical Data System via the Internet
Student	Mr. Anuwat Kanokwiroon
Advisor	Asst. Prof. Dr. Prachuab Vanitchatchavan
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2002

ABSTRACT

Dental care is vital to people who wish to maintain healthy teeth. There are many places that provide dental care and treatment such as the hospital, the dental clinic and so on. Patient's dental history will be recorded for every treatment. However, if the patients switch to different clinics, then the dental information will be lost or no transferred. After service, the dentists must be recorded the treatment into the patient's history in their service place. In fact, the patients may be changed to cure in the new clinic from many factors. If they changed, the treatment information was lost. In the same way, the clinics have been keeping the patients' records that no use in the future.

Therefore, if we can keep the scatter records into the central data system, the dental record could be retrieved in different clinics. Moreover, it could be an important source dental research.

To overcome the problem, internet technology and web-based application are introduced to record information from each clinic's system and keep all records to the database server. In this way, patient can access their treatment record from home and dentists can trace back patient's dental history for a more accurate diagnosis.

กิตติกรรมประกาศ

โครงการพัฒนาระบบศูนย์กลางการจัดเก็บข้อมูลการรักษาทันตกรรม ผ่านเครือข่าย อินเทอร์เน็ต สำเร็จลงได้ก็ด้วยความอนุเคราะห์จากบุคคลหลายท่าน ที่ให้ความสนับสนุนโดยตรง และโดยอ้อม ความสนับสนุนดังกล่าวประกอบไปด้วยความรู้ทางวิชาการที่นำมาจัดทำ คำแนะนำ ต่างๆ ตลอดจนถึงกำลังใจที่ได้รับ ทั้งหลายเหล่านี้เป็นสิ่งที่ช่วยผลักดันให้เกิดความสำเร็จขึ้น ผู้เขียน รู้สึกขอบคุณในผู้มีส่วนเกี่ยวข้องทั้งหมดเป็นอย่างยิ่ง

1. ผศ. ดร. ประจวบ วานิชชัชวาล ในฐานะอาจารย์ที่ปรึกษาโครงการ ที่ได้ให้ความรู้ คำปรึกษาในการจัดทำโครงการ และคำเตือนต่างอันสำคัญ โดยเฉพาะอย่างยิ่งในเรื่อง เวลา และเป็นไปได้ว่าท่านคงเห็น CFF ในโครงการล่วงหน้าก็เป็นได้ จึงต้องออกแรง เตือนบ่อยๆ
2. คุณสุรวัส ธาราสมบัติ เพื่อนร่วมรุ่นที่รีบจบไปก่อน เป็นผู้คอยแนะนำและให้ความรู้ในการพัฒนาระบบในส่วนที่ผู้จัดทำไม่เข้าใจ
3. ทพ. ญัฐพงศ์ กนกวิรุฬห์ และ ทพ.ญ. จรีวรรณ พลชัย ผู้ให้ความรู้ เอกสารต่างๆ เกี่ยวกับงานทางด้านทันตกรรมที่จำเป็น
4. ภรรยาที่น่ารักผู้คอยดูแลเอาใจใส่ และยอมให้ผู้จัดทำเป็นคุณชายประจำบ้านได้ในช่วง ระยะเวลาหนึ่ง
- 5.ญาติมิตรทั้งหลาย รวมถึงหลานๆที่น่ารัก 2 คนที่มักคอยถามเสมอว่า เมื่อไรโรงงานของผู้จัดทำจะเสร็จสักที เพราะถามทีไรก็ไม่ว่างเสมอ ถ้าว่างเมื่อไรก็จะได้ไปเล่นด้วยกัน

ทั้งหมดที่กล่าวมาข้างต้นเป็นผู้ที่มีส่วนเกี่ยวข้องโดยตรงในช่วงระยะเวลาที่ผู้จัดทำได้ทำ โครงการเท่านั้น ยังมีผู้อยู่เบื้องหลังที่ไม่ได้กล่าวถึงชื่อของท่านอีกเป็นจำนวนมากที่ได้ให้ความรู้ จนกระทั่งผู้จัดเติบโตมาและมีโอกาสจัดทำโครงการนี้ ประกอบไปด้วยพ่อแม่ซึ่งเป็นครูสองคนแรก ของผู้จัดทำ และเป็นตัวอย่างแห่งความเข้มแข็ง อดทน ครูบาอาจารย์ที่ให้ความรู้แก่ผู้จัดทำมานับแต่ เริ่มจำความได้ คณาจารย์ผู้ให้ความรู้ทั้งจากสถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือและ ลาดกระบัง ผู้ทำให้ผู้จัดทำเข้ามาอยู่ในวงการดิจิทัลและสามารถดำรงชีวิตต่อไปในวงการอย่างไม่ ขัดเงิน พระอุปฌาย์อาจารย์ของผู้จัดทำที่ให้ความรู้อันสำคัญในการดำเนินชีวิต ให้มีความสุขอยู่ได้ ในโลกอันเร่งรีบดังเช่นในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII

บทที่

1. บทนำ.....	1
1.1 ปัญหาของข้อมูลพันธุกรรมที่พบในปัจจุบัน.....	1
1.2 จุดมุ่งหมายของ โครงการ.....	1
1.3 ประโยชน์ที่จะได้รับจากโครงการพัฒนาระบบงาน.....	2
1.4 ขอบเขตของโครงการพัฒนาระบบงาน.....	2
1.5 ข้อจำกัดของระบบงาน.....	2
1.6 เนื้อหาของเอกสาร โครงการพัฒนาระบบงาน.....	3
2. เทคโนโลยีที่เกี่ยวข้อง.....	4
2.1 หลักการทำงานของระบบ.....	4
2.2 เครือข่ายอินเทอร์เน็ต.....	4
2.3 เทคโนโลยีการเชื่อมต่อแบบ World Wide Web.....	4
2.4 รูปแบบของสถาปัตยกรรมที่ใช้.....	5
3. Object-Oriented Programming.....	7
3.1 ลักษณะของ Object.....	7
3.2 คำศัพท์และความหมาย.....	7
3.3 คุณสมบัติที่สำคัญของ Object.....	8
3.4 Object-Oriented Methodology.....	8
4. ขั้นตอนการพัฒนาระบบงานและ Unify Modeling Language (UML).....	9
4.1 ภาษา UML.....	9
4.2 ประวัติของ UML.....	9

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
ประเภทของ UML Diagram.....	10
Iteration Model.....	16
5. Object-Oriented CASE Tool: Ration Rose.....	19
5.1 Rational Rose 2000 Enterprise Edition.....	19
5.2 Rational Rose J.....	19
5.3 Rational Rose Oracle8.....	25
6. การวิเคราะห์ระบบงาน.....	26
6.1 ความต้องการของระบบงาน.....	26
6.2 Use Case Diagram.....	27
6.3 Use Case Description.....	28
6.4 Preliminary Analysis Classes.....	38
6.5 Architecture Analysis.....	38
6.6 Analysis Classes.....	39
6.7 Analysis Sequence Diagram.....	42
7. การออกแบบและพัฒนาระบบ.....	46
7.1 Architectural Design.....	46
7.2 Design Sequence Diagram.....	46
7.3 Design Classes.....	50
7.4 ตารางฐานข้อมูลของระบบงาน.....	50
7.5 Source Code.....	56
8. การใช้งานระบบ.....	66
8.1 การลงทะเบียนเพื่อเริ่มต้นใช้งาน.....	66
8.2 การใช้งานระบบ.....	68
9. สรุปการพัฒนาโครงการ.....	78
บรรณานุกรม.....	80
ประวัติผู้เขียน.....	81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

7.1 ตารางฐานข้อมูล Dentist.....	52
7.2 ตารางฐานข้อมูล Patient.....	52
7.3 ตารางฐานข้อมูล Clinic.....	54
7.4 ตารางฐานข้อมูล Treatment.....	54
7.5 ตารางฐานข้อมูล User.....	55
7.6 ตารางฐานข้อมูล Trustee.....	55
7.7 ตารางฐานข้อมูล Diagnosis.....	55
7.8 ตารางฐานข้อมูล Service.....	55
7.9 ตารางฐานข้อมูล Clinic_Dentist.....	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
2.1 สถาปัตยกรรมในแบบ Multi-tier.....	5
3.1 การติดต่อสื่อสารระหว่างออบเจกต์.....	7
4.1 แผนภาพประวัติของ UML	10
4.2 ไคอะแกรมทั้งหมดใน UML.....	10
4.3 ตัวอย่างของ Use Case Diagram.....	11
4.4 สัญลักษณ์ของ Actor.....	11
4.5 สัญลักษณ์ของ Use Case.....	11
4.6 สัญลักษณ์ของ Communication.....	12
4.7 สัญลักษณ์ของ Relationship.....	12
4.8 Use Case Relationship.....	12
4.9 สัญลักษณ์ของ Class.....	13
4.10 สัญลักษณ์ของ Association Relationship.....	13
4.11 สัญลักษณ์ของ Aggregation Relationship.....	14
4.12 สัญลักษณ์ของ Depend on Relationship.....	14
4.13 ตัวอย่างความสัมพันธ์แบบ Generalization.....	14
4.14 ตัวอย่างของ Class Diagram.....	14
4.15 ตัวอย่างของ Sequence Diagram.....	15
4.16 องค์ประกอบของออบเจกต์.....	15
4.17 สัญลักษณ์ของ Message ทั้ง 3 แบบ.....	16
4.18 ความสัมพันธ์ระหว่าง Phase Iteration และ Workflow.....	18
5.1 วิธีการเข้าสู่หน้าจอเซต Classpath.....	20
5.2 หน้าจอขณะทำการเซต Classpath ที่เก็บของไฟล์ .java, .class, .jar, .zip.....	21
5.3 Class User ใน Rational Rose Model.....	21
5.4 วิธีการสั่งให้สร้าง Source Code จาก Class ที่ต้องการ.....	22
5.5 วิธีการดูผลลัพธ์ Source Code จาก Class ที่ต้องการ.....	22
5.6 ผลลัพธ์ Source Program.....	23

เอกสารนี้เป็นทรัพย์สินทางปัญญาของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตจากทางสถาบันฯ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

5.7 วิธีการเลือกเมนูเพื่อทำ Reverse Engineering.....	24
5.8 ตัวอย่างการเลือกไฟล์จาก Directory มาทำ Reverse Engineering.....	24
6.1 Use Case Diagram ของการลงทะเบียน.....	27
6.2 Use Case Diagram ของการใช้งานระบบบันทึกข้อมูลการรักษาทางทันตกรรม.....	28
6.3 Key Abstractions ของระบบ.....	38
6.4 Initial Architecture ของระบบ.....	39
6.5 Boundary Class, Control Class และ Entity Class เมื่อแสดงแบบ icon.....	39
6.6 Boundary Class, Control Class และ Entity Class เมื่อแสดงแบบ label.....	40
6.7 Boundary Classes ของระบบงาน.....	40
6.8 Entity Class ของระบบงาน.....	41
6.9 Control Classes ของระบบงาน.....	42
6.10 Analysis Sequence Diagram ของ Use Case – Dentist Registration.....	43
6.11 Analysis Sequence Diagram ของ Use Case – Add Trustee.....	43
6.12 Analysis Sequence Diagram ของ Use Case – Add New Treatment.....	44
6.13 Analysis Sequence Diagram ของ Use Case – Cancel Treatment.....	44
6.14 Analysis Sequence Diagram ของ Use Case – Add New Diagnosis.....	45
6.15 Analysis Sequence Diagram ของ Use Case – Cancel Diagnosis.....	45
7.1 Design Sequence Diagram ของ Use Case – Dentist Registration.....	47
7.2 Design Sequence Diagram ของ Use Case – Add Trustee.....	47
7.3 Design Sequence Diagram ของ Use Case – Add New Treatment.....	48
7.4 Design Sequence Diagram ของ Use Case – Cancel Treatment.....	48
7.5 Design Sequence Diagram ของ Use Case – Add New Diagnosis.....	49
7.6 Design Sequence Diagram ของ Use Case – Cancel Diagnosis.....	49
7.7 Class ที่ได้จากการออกแบบ.....	50
7.8 Database Schema ของระบบ.....	51

เอกสารนี้ 7.9 ตัวอย่าง Source code ของ Class Controller.....

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.10 ตัวอย่าง Source code ของ PatientMainPage.jsp.....	57
8.1 หน้าจอแรกเมื่อเข้าใช้งานสำหรับ Login หรือลงทะเบียนเข้าสู่ระบบ.....	66
8.2 ตัวอย่างหน้าจอสำหรับกรอกข้อมูลลงทะเบียนของผู้ใช้ – คนไข้.....	67
8.3 ตัวอย่างหน้าจอสำหรับกรอกข้อมูลลงทะเบียนของผู้ใช้ – ทันตแพทย์.....	68
8.4 หน้าจอหลักเมื่อคนไข้ Login เข้าสู่ระบบ.....	69
8.5 หน้าจอแสดงประวัติการรักษาเมื่อเลือกเมนูประวัติการรักษา.....	70
8.6 หน้าจอแสดงประวัติการรักษาในแบบละเอียด.....	71
8.7 หน้าจอแสดงประวัติการวินิจฉัย.....	71
8.8 หน้าจอแสดงประวัติการวินิจฉัย โดยละเอียด.....	72
8.9 หน้าจอหลักเมื่อ Dentist Login เข้าสู่ระบบ.....	73
8.10 หน้าจอสำหรับกรอกรหัสของ Patient เมื่อ Dentist เรียกดูประวัติของคนไข้.....	74
8.11 หน้าจอข้อมูลของคนไข้ถูกเรียกโดย Dentist.....	74
8.12 หน้าจอแสดงข้อมูลการรักษาของคนไข้ถูกเรียกโดย Dentist.....	75
8.13 หน้าจอแสดงการบันทึกข้อมูลการรักษาของคนไข้ถูกเรียกโดย Dentist.....	76
8.14 หน้าจอแสดงข้อมูลการวินิจฉัยของคนไข้ถูกเรียกโดย Dentist.....	77
8.15 หน้าจอแสดงการบันทึกข้อมูลการวินิจฉัยของคนไข้ถูกเรียกโดย Dentist.....	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ปัญหาของข้อมูลทันตกรรมที่พบในปัจจุบัน

การรักษาทางทันตกรรมในปัจจุบันมีให้เลือกทั้งจากหน่วยงานของรัฐบาลประกอบด้วย โรงพยาบาล สาธารณสุข อนามัย เป็นต้น หรือทางด้านเอกชน เช่น โรงพยาบาลเอกชน คลินิกทันตกรรม ตลอดจนถึงองค์กรมูลนิธิต่างๆ ที่ออกให้บริการทางด้านทันตกรรมแก่ประชาชนทั่วไป ในการเลือกเข้ารับการรักษานั้นก็ขึ้นอยู่กับปัจจัยหลายอย่างของแต่ละบุคคล อันได้แก่ ที่ตั้งของสถานพยาบาล ที่ตั้งบ้านเรือน ความสะดวกในการเดินทาง รายได้ของค่าใช้จ่าย ความรู้เกี่ยวกับสุขภาพ ความเชื่อถือในตัวทันตแพทย์ ความพอใจในบริการ เป็นต้น

เมื่อเข้ารับการรักษาจากสถานพยาบาลแต่ละแห่ง ในเบื้องต้นจำเป็นต้องมีการตรวจร่างกายก่อนการรักษาในครั้งแรก และทำการบันทึกอาการที่ตรวจพบลงในประวัติของคนไข้ อาการแพ้ยาต่างๆ ซึ่งรายละเอียดจะมีความแตกต่างกันออกไปตามมาตรฐานของแต่ละสถานพยาบาล พร้อมทั้งบันทึกการรักษา ซึ่งประวัติเหล่านี้จะถูกเก็บเอาไว้ ณ สถานพยาบาลแห่งนั้นๆ เพื่อใช้สำหรับเป็นข้อมูลการรักษาในคราวถัดไป

ในการเลือกสถานพยาบาลที่จะเข้ารับการรักษานั้น ย่อมขึ้นกับความสมัครใจของคนไข้ อันเกิดจากปัจจัยต่างๆอันได้กล่าวมาแล้ว ดังนั้นการเปลี่ยนแปลงสถานพยาบาลที่รักษาจึงเกิดขึ้นอยู่เสมอ อีกทั้งในชีวิตคนเราจำเป็นต้องเกี่ยวข้องข้องกับการรักษาฟันอยู่เกือบตลอดชีวิต แต่เมื่อมีการเปลี่ยนแปลงสถานพยาบาล โอกาสในการนำข้อมูลการรักษาที่เกิดขึ้นมาใช้มีน้อยมาก อันอาจเนื่องมาจากความไม่สะดวกในการไปขอข้อมูลจากสถานพยาบาลเหล่านั้น หรืออาจไม่ได้รับความร่วมมือจากสถานพยาบาล ทำให้สูญเสียข้อมูลที่สำคัญในการรักษาไป สิ่งที่ได้คือทันตแพทย์จะใช้วิธีการซักถามจากคนไข้ ซึ่งผลที่ได้จากการสอบถามมักคลาดเคลื่อนจากความเป็นจริง เช่น วันเวลาที่เคยรับการรักษา จำนวนครั้งที่เคยรักษา และบ่อยครั้งที่ไม่ทราบว่าทันตแพทย์ได้รักษาอะไรไปบ้างเนื่องจากปัญหาการสื่อสารหรือไม่เข้าใจในสิ่งที่ทันตแพทย์ได้อธิบาย

1.2 จุดมุ่งหมายของโครงการ

โครงการนี้จัดทำขึ้นเพื่อเป็นการทดลองนำเทคโนโลยีทางด้านคอมพิวเตอร์ เครือข่ายสื่อสารทางอินเทอร์เน็ต มาช่วยแก้ปัญหาที่เกิดขึ้นในระบบข้อมูลการรักษาทางทันตกรรม โดยใช้วิธีเก็บข้อมูลของคนไข้ การรักษา และทันตแพทย์เอาไว้ในระบบโปรแกรมที่อยู่บนเครือข่าย (Web-เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

based Application) ทำให้ข้อมูลที่กระจัดกระจายตามสถานที่ต่างๆถูกรวบรวมอยู่ในที่แห่งเดียวกัน ทำให้คนไข้สามารถทราบข้อมูลส่วนตัว และประวัติการรักษาจากที่ได้ก็ผ่านทางเครือข่าย อินเทอร์เน็ต และทันตแพทย์ก็สามารถที่จะนำข้อมูลของคนไข้รวมถึงประวัติการรักษาทั้งหมดที่ อาจจะรักษาโดยทันตแพทย์ผู้อื่นมาประกอบการรักษาได้

1.3 ประโยชน์ที่จะได้รับจากโครงการพัฒนาระบบงาน

ผลประโยชน์ที่ได้ก็เป็นผลดีต่อการรักษาทั้งระบบอันได้แก่ ทางด้านทันตแพทย์ผู้ทำการรักษาที่จะมีข้อมูลของประวัติของคนไข้และผลการรักษาในอดีตอย่างละเอียด ส่วนทางด้านคนไข้ก็สามารถตรวจสอบย้อนหลังได้จากบันทึกว่าเคยเข้ารับการรักษาอะไรบ้าง เกิดขึ้นเมื่อไร และหากมีความสงสัยเกี่ยวกับการรักษาก็สามารถที่จะนำผลการรักษานั้นไปสอบถามจากทันตแพทย์รายอื่น เพื่อขอความเห็น และจะเป็นผลดีในการตรวจสอบการทำงานของทันตแพทย์เช่นกัน ในอนาคตหากระบบนี้สามารถรวบรวมข้อมูลของประชากรส่วนใหญ่ได้ก็จะเป็นประโยชน์ต่องานทันตกรรมที่จะนำข้อมูลเหล่านี้ไปทำการค้นคว้าวิจัย เพื่อความก้าวหน้าทางด้านวิชาการต่อไป

1.4 ขอบเขตของโครงการพัฒนาระบบงาน

ระบบงานนี้เป็นการทดลองนำแนวคิดในการสร้างระบบงานบนเว็บมาใช้แก้ไขปัญหา ขั้นตอนต่างๆในการพัฒนาจะประกอบไปด้วย

- 1 การศึกษาข้อมูลจากการปฏิบัติงานของทันตแพทย์ในการรักษาและบันทึกข้อมูลการรักษาของคนไข้ เพื่อจัดทำเป็นความต้องการมาใช้แก้ปัญหา
- 2 วิเคราะห์และออกแบบระบบงานโดยใช้แนวคิดเชิงวัตถุ (Object-Oriented System) และ UML Diagram โดยใช้เครื่องมือ Rational Rose 2000 Enterprise Edition
- 3 พัฒนาระบบด้วยการ mapping จากแบบจำลอง UML Diagram ไปสู่โปรแกรมในลักษณะ Web-based application ด้วยภาษา Java โดยเลือกใช้ Java API (Java Application Programming Interface) ที่เหมาะสม
- 4 ออกแบบสร้างฐานข้อมูล และจัดสร้างโดยใช้โปรแกรมจัดการฐานข้อมูล MySQL 3.23.47

1.5 ข้อจำกัดของระบบงาน

ระบบข้อมูลในการรักษานี้จะเกี่ยวข้องกับตรงกบสิทธิต่างๆของคนไข้ในเปิดเผยข้อมูล การรักษา ซึ่งจะต้องตั้งอยู่ภายใต้ข้อกำหนดของทันตแพทย์สภา โดยจะต้องไม่มีการละเมิดสิทธิขั้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นฐานต่างๆ รวมถึงจะไม่ให้ผู้มีส่วนเกี่ยวข้องสามารถนำข้อมูลไปใช้สำหรับหาประโยชน์ ยกเว้น เฉพาะงานวิจัยเท่านั้น ดังนั้นจึงต้องมีการจัดระบบรักษาความปลอดภัยให้สอดคล้องกับข้อกำหนด ดังกล่าว

1.6 เนื้อหาของเอกสารโครงการพัฒนาระบบงาน

เนื้อหาในโครงการพัฒนาระบบงาน บทแรกจะกล่าวถึงความเป็นมา ความสำคัญ ประโยชน์ที่จะได้รับ และจุดมุ่งหมายรวมของโครงการพัฒนาระบบงาน บทที่ 2 กล่าวถึงเทคโนโลยีต่างๆที่เกี่ยวข้องทางด้านเครือข่ายและการเชื่อมต่อ รวมถึงสถาปัตยกรรมที่นำมาใช้โครงการ บทที่ 3 กล่าวถึงแนวคิดเชิงวัตถุ และส่วนประกอบต่างๆที่สำคัญของ Object-Oriented Methodology บทที่ 4 ได้อธิบายถึงภาษา UML และการใช้งานในขั้นตอนต่างๆ การวิเคราะห์และออกแบบระบบด้วย Iterative model บทที่ 5 เป็นเรื่องการใช้งานโปรแกรม Rational Rose 2000 พร้อมตัวอย่างในการใช้งาน การทำ Forward/Reverse Engineering การสร้าง Database Schema ด้วย Rational Rose Oracle8 การ mapping จาก Class ไปเป็น Relational Database Schema บทที่ 6 กล่าวถึงการวิเคราะห์ระบบ โดยใช้ขั้นตอนจาก Rational Unified Process บทที่ 7 แสดงถึงการออกแบบระบบโดยมีการออกแบบเพิ่มเติมที่ได้จากขั้นตอนการวิเคราะห์ระบบงานประกอบด้วย Class diagram, Sequence diagram และตารางฐานข้อมูลที่น่าไปพัฒนาระบบ บทที่ 8 ได้กล่าวถึงการใช้งานระบบและตัวอย่างหน้าจอการทำงาน บทที่ 9 เป็นบทสรุปการพัฒนาระบบงาน

บทที่ 2

เทคโนโลยีต่างๆที่เกี่ยวข้อง

2.1 หลักการทำงานของระบบ

ระบบงานในโครงการนี้จัดทำเป็นแอปพลิเคชันบนเว็บ (Web-based Application) ซึ่งต้องทำงานเชื่อมโยงผ่านเครือข่ายอินเทอร์เน็ต ทำให้การติดต่อระหว่างคนไข้และทันตแพทย์เป็นไปได้ด้วยความสะดวก ผู้ใช้จะติดต่อกับระบบงานผ่านทางเว็บไซต์ (Website) ที่กำหนดขึ้น ดังนั้นในการวิเคราะห์และออกแบบระบบงานจึงต้องคำนึงถึงการแยกส่วนต่างๆออกจากกันประกอบด้วย ส่วนติดต่อกับผู้ใช้ระบบ (User Interface) ส่วนตรรกะ (Business Logic) ทั้งนี้เพื่อให้เกิดความสะดวกในการพัฒนาและการบำรุงรักษา รวมถึงความยืดหยุ่นต่อการขยายระบบเพิ่มเติมในอนาคต

ในการแยกส่วนตรรกะออกจากส่วนติดต่อกับผู้ใช้ในระบบงานนี้จะใช้เทคโนโลยีของภาษาจาวา ซึ่งประกอบไปด้วยเซิร์ฟเล็ต (Servlet) จาวาเซิร์ฟเวอร์เพจ (Java Server Page: JSP) และจาวาบีนส์ (Java Beans) โดยใช้เซิร์ฟเล็ตทำหน้าที่ในส่วนควบคุมการทำงาน JSP ใช้ในส่วนที่ติดต่อกับผู้ใช้งานร่วมกับภาษา HTML และจาวาบีนส์รับหน้าที่ในการประมวลผลงานย่อยๆ

2.2 เครือข่ายอินเทอร์เน็ต

อินเทอร์เน็ตเป็นเครือข่ายคอมพิวเตอร์ขนาดใหญ่ที่ต่อเชื่อมเครื่องคอมพิวเตอร์ต่างๆในโลกเข้าด้วยกัน และนับวันอินเทอร์เน็ตได้ขยายออกไปมากขึ้นเรื่อยๆ ทำให้ผู้ใช้คอมพิวเตอร์สามารถติดต่อถึงกันได้ไม่ว่าจะอยู่ที่ใดที่สามารถติดต่อกับเครือข่ายได้ ผู้ใช้บริการอินเทอร์เน็ตส่วนใหญ่จะมักใช้บริการในรูปแบบของอีเมล (E-mail) เครือข่าย WWW เป็นต้น

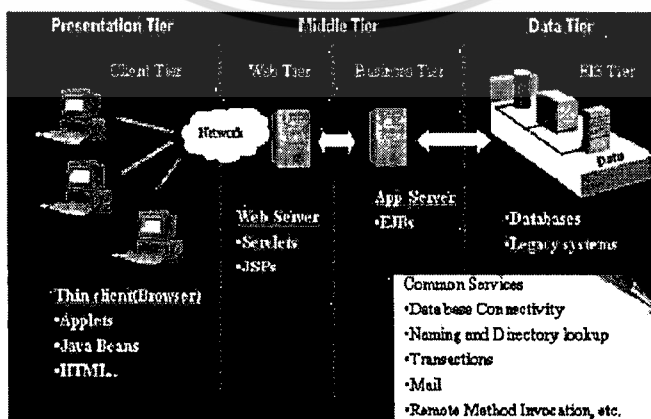
2.3 เทคโนโลยีการเชื่อมต่อแบบ World Wide Web

- **เว็บเซิร์ฟเวอร์ (Web Server)** ทำหน้าที่ให้บริการข้อมูลในรูปแบบของเว็บเพจที่ได้รับการร้องขอจากทาง Client ซึ่งอาจมีการประมวลผลก่อนที่จะส่งผลลัพธ์กลับไปให้ทาง Client หรือไม่ขึ้นกับการทำงานของเว็บเพจนั้นๆ
- **ไคลเอนต์บราวเซอร์ (Client Browser)** ทำหน้าที่ในการติดต่อร้องขอข้อมูลจากเซิร์ฟเวอร์ที่ให้บริการ (Web Server) ผ่านทางโปรโตคอล TCP/IP เมื่อติดต่อได้ทางเซิร์ฟเวอร์จะส่งข้อมูลกลับมาให้ไคลเอนต์โดยบราวเซอร์จะแปลข้อมูลที่ได้รับซึ่งส่วนใหญ่จะเป็นภาษา HTML แล้วทำการแสดงผลให้ผู้ใช้ต่อไป

- เซิร์ฟเวอร์เพจ (Server Page) เป็นเว็บเพจที่ทำงานอยู่ทางฝั่ง Server ก่อนที่จะส่งผลมายัง Client โดยมีการเขียนเป็นภาษาสคริปต์ (Script) ฝังอยู่ในเพจ เช่น ASP (Active Server Page), JSP (Java Server Page) เป็นต้น เพจเหล่านี้จะมีทำงานตามคำสั่งและติดต่อกับฐานข้อมูลหรือทรัพยากรอื่นๆทางฝั่งเซิร์ฟเวอร์
- โพรโทคอล HTTP (Hyper Text Transfer Protocol) เป็นโพรโทคอลที่ใช้ติดต่อกันระหว่างเว็บเบราว์เซอร์ทางไคลเอนต์กับเว็บเซิร์ฟเวอร์ โดยที่เมื่อมีการติดต่อและส่งข้อมูลถึงกันจะมีการสร้างเส้นทางการติดต่อขึ้นใหม่ทุกครั้ง เป็นโพรโทคอลที่ไม่เก็บสถานะการติดต่อเอาไว้ ในกรณีที่ต้องการเพิ่มความปลอดภัยก็สามารถสร้างการติดต่อผ่านทาง SSL (Secure Socket Layer) เพื่อให้มีการเข้ารหัสก่อนที่จะส่งผ่านข้อมูลออกไปทางเครือข่าย
- แอปพลิเคชันเซิร์ฟเวอร์ (Application Server) เป็นส่วนที่เซิร์ฟเวอร์นำเอาคำสั่งทางตรรกะมาประมวลผล โดยรับผิดชอบในการรันโค้ดที่อยู่ในเซิร์ฟเวอร์เพจ ซึ่งแอปพลิเคชันอาจทำงานอยู่บนเครื่องเดียวกันหรือคนละเครื่องกับเว็บเซิร์ฟเวอร์ก็ได้

2.4 รูปแบบของสถาปัตยกรรมที่ใช้

สถาปัตยกรรมที่ใช้จะเป็นแบบ Multi-tier ซึ่งแยกส่วนต่างๆออกจากกัน ดังในรูปที่ 2.1 โดยส่วนเครื่องลูกข่าย (Thin Client: Presentation Tier) มีการทำงานผ่านทางเว็บเบราว์เซอร์ (Web Browser) ทำหน้าที่ในการร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์ (Web Server) ผ่านทางโพรโทคอล HTTP เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอก็จะส่งข้อมูลกลับไปให้ Client ตามที่ต้องการ เบราวเซอร์ก็จะนำข้อมูลที่ได้ไปแปลงแล้วแสดงให้ผู้ใช้ดูในส่วนของ Client Tier จะมีหน้าที่ติดต่อกับผู้ใช้และแสดงค่าที่ได้รับมาเท่านั้น



รูปที่ 2.1 สถาปัตยกรรมในแบบ Multi-tier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของเว็บเซิร์ฟเวอร์และแอปพลิเคชันเซิร์ฟเวอร์ที่ให้บริการจะจัดอยู่ในส่วนของ Middle Tier โดยเว็บเซิร์ฟเวอร์จะให้บริการเว็บเพจ ส่วนแอปพลิเคชันเซิร์ฟเวอร์จะทำการประมวลผลหากได้รับการร้องขอจากเว็บเซิร์ฟเวอร์ ในกรณีที่การประมวลผลต้องมีการติดต่อกับฐานข้อมูลก็จะมีคำสั่งส่งเพื่อกระทำการใดๆในส่วนของ Data Tier ซึ่งในส่วนนี้จะเก็บข้อมูลต่างๆของระบบงานเอาไว้ทั้งหมด จากสถาปัตยกรรมแบบนี้ทำให้ง่ายต่อการดูแลรักษา และการปรับปรุงในอนาคต



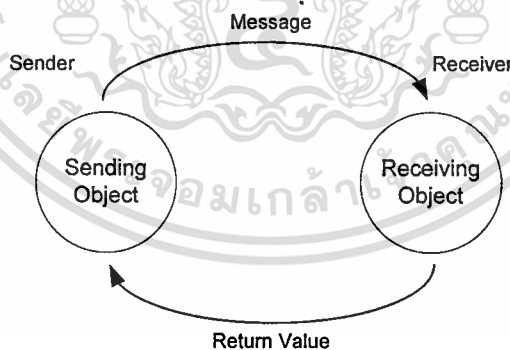
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

Object-Oriented Methodology

3.1 ลักษณะของออบเจกต์

OOP (Objected-Oriented Programming) เป็นวิธีการเขียนโปรแกรมคอมพิวเตอร์โดยใช้แนวคิดเสมือนการมองในเชิงวัตถุ (Object) ในโลกแห่งความเป็นจริง (Real-world) เพื่อจัดโครงสร้างของโปรแกรมให้มีระเบียบมากยิ่งขึ้น เอื้อประโยชน์ต่อการพัฒนาโปรแกรมในยุคต่อไปที่นับวันความซับซ้อนของโปรแกรมนิ่งเพิ่มมากขึ้น แนวคิดเชิงวัตถุของ OO นั้นจะมีการจัดแบ่งประเภทของวัตถุในทางนามธรรม (Abstract) ออกเป็นคลาส (Class) ซึ่งแต่ละคลาสจะมีสถานะ (State) และพฤติกรรม (Behavior) เฉพาะของตัวเอง เพื่อเป็นต้นแบบให้แก่วัตถุอื่นที่จะถูกสร้างขึ้นใหม่ในอนาคต ข้อมูลหรือคุณสมบัติ (Characteristic) จะถูกเก็บซ่อนไว้ภายในคลาสไม่ให้ปะปนกับคลาสอื่น ออบเจกต์จะมีการติดต่อสื่อสารระหว่างกันด้วยการส่งข้อความ (Message) ดังรูปที่ 3.1 ตัวออบเจกต์เองสามารถที่จะสืบทอดคุณสมบัติ (Inheritance) คุณสมบัติจากบรรพบุรุษไปยังลูกหลานได้



รูปที่ 3.1 การติดต่อสื่อสารระหว่างออบเจกต์

3.2 คำศัพท์และความหมาย

Object คือ วัตถุ หรือสิ่งที่ประกอบด้วยสถานะ (State) หรือข้อมูล (Information) และฟังก์ชัน

การทำงาน (Operation) หรือพฤติกรรม (Behavior) (Sibershatz, A. et.Al. 1997)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 คุณสมบัติที่สำคัญของออบเจกต์

3.3.1 Encapsulation และ Information Hiding

ข้อมูลที่อยู่ภายในออบเจกต์จะถูกห่อหุ้มรวมอยู่ด้วยกัน (Encapsulation) ทำให้เกิดการซ่อนข้อมูลเอาไว้ภายใน (Information Hiding) ในการเข้าถึงข้อมูลในออบเจกต์จึงต้องกระทำผ่านทาง Operation ของออบเจกต์ ซึ่งผลดีของการมี Encapsulation นี้ทำให้โปรแกรมลดความซับซ้อนลง เนื่องจากไม่จำเป็นต้องทราบรายละเอียดภายในออบเจกต์นั้นๆ รู้แต่เพียงคุณลักษณะภายนอกและ Operation ที่มีให้เรียกใช้เท่านั้น

3.3.2 Inheritance เป็นการสืบทอดคุณสมบัติทั่วไปที่เหมือนกันระหว่างคลาส ข้อดีของ Inheritance คือ

1. สามารถสิ่งที่สร้างเอาไว้แล้วนำกลับมาใช้ใหม่ได้ (Reuse common description)
2. การปรับปรุง แก้ไข ทำเพียงที่เดียวเท่านั้น (Modification can be made in only one place)
3. ลดความซ้ำซ้อนสำหรับ โปรแกรมย่อยเพื่ออำนวยความสะดวกทำความเข้าใจ (Redundancy can be reduced leading to smaller models that are easier to understand)

“Overriding” คือคุณสมบัติที่สามารถนิยามพฤติกรรม (Behavior) หรือ Information ใหม่ได้สำหรับคลาสที่สืบทอดมาจาก Super Class

3.3.3 Polymorphism

โดยความหมายจากตัวอักษรคือ “Having many forms” ซึ่งก็คือการส่งข้อความเดียวกันให้กับออบเจกต์ที่ต่างกันจะแสดงพฤติกรรมที่แตกต่างกัน เช่น การบวกของ String และ Int จะกำหนดเพียงสัญลักษณ์เดียว แต่สามารถทำกับข้อมูลชนิดต่างกันก็ได้ ข้อดีของ Polymorphism คือ สนับสนุนให้สามารถนำกลับมาใช้ใหม่ และมีการเปลี่ยนแปลงแก้ไขได้ง่าย

3.4 Object-Oriented Methodology

การพัฒนาระบบด้วย Object-Oriented Methodology นักวิเคราะห์ระบบจะต้องมีแนวคิดที่สอดคล้องกับวิธีคิดในลักษณะเชิงวัตถุตั้งที่ได้กล่าวมาแล้ว นับตั้งแต่การวิเคราะห์ระบบ การออกแบบ การเขียนโปรแกรม ตลอดจนถึงการทดสอบการทำงานของระบบ เช่น การวิเคราะห์ระบบก็ต้องวิเคราะห์ออกมาให้ได้ว่ามีคลาสอะไรบ้าง แต่ละคลาสมีพฤติกรรมอย่างไร และจะเกิดเหตุการณ์อะไรขึ้นได้บ้าง เป็นต้น โดยที่คลาสจะมีจัดการข้อมูลภายในตัวเอง ต่างจากการวิเคราะห์ในแบบเดิมที่มีการแยกส่วนของข้อมูลและการทำงานออกจากกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ขั้นตอนการพัฒนาระบบงาน

และ Unified Modeling Language (UML)

4.1 ภาษา UML

UML (Unified Modeling Language) เป็นภาษาชนิดหนึ่งที่ใช้อธิบายใช้สำหรับอธิบายแบบจำลอง (Model) ในแบบต่างๆ เพื่ออธิบายถึงระบบงานซึ่งเป็นสิ่งจำเป็นอย่างยิ่งในการจัดทำซอฟต์แวร์ โดย UML เป็นภาษาที่ถูกนำไปใช้ในการสร้างแบบพิมพ์เขียว (Blueprint) ให้กับระบบงาน เพื่อจะได้เห็นมุมมอง กำหนดรายละเอียด และสร้างระบบงาน UML เป็นภาษาที่เหมาะสมสำหรับระบบงานต่างๆ ในกิจการ ระบบงานแอปพลิเคชันบนเว็บ (Web-based Application) ตลอดจนถึงระบบงานแบบเรียลไทม์ (Real time system)

4.2 ประวัติของ UML

การออกแบบและพัฒนาโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ได้เริ่มได้รับความสนใจเนื่องจากสาเหตุของแอปพลิเคชันและระบบงานเริ่มมีความซับซ้อนมากขึ้นเรื่อยๆ นักวิเคราะห์ นักออกแบบซึ่งเกี่ยวข้องกับการพัฒนาระบบเริ่มสังเกตเห็นถึงวิธีการที่ใช้อยู่ในขณะนั้นว่าไม่สามารถรองรับความซับซ้อนในอนาคตได้ วิธีการเชิงวัตถุ (Object-Oriented Methodology) จึงได้เป็นตัวเลือกที่ทำการนำมาใช้และเป็นที่แพร่หลายอยู่ในปัจจุบัน

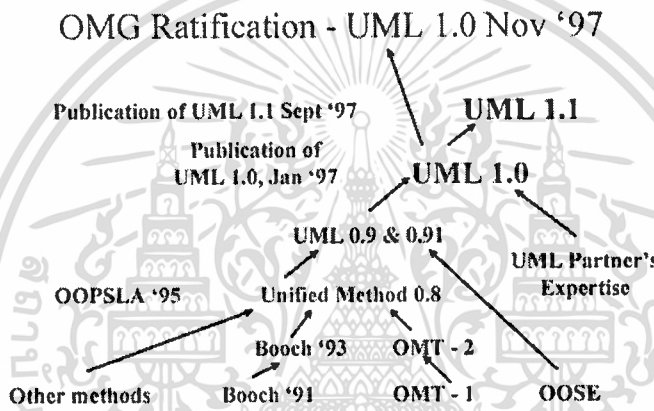
ในช่วงเวลานั้นก็ได้มีแนวคิดในการกำหนดภาษาสำหรับสร้างแบบจำลองเชิงวัตถุมีขึ้นในยุคกลางทศวรรษที่ 1970 ถึงปลายยุค 1980 โดยในยุคแรกๆ ได้มีผู้คิดค้นวิธีการเชิงวัตถุขึ้นมาอยู่ไม่เกิน 10 วิธี ต่อจากนั้นจำนวนก็เพิ่มขึ้นมากกว่า 50 วิธี ทำให้ผู้ใช้ต่างประสบปัญหาในการหาภาษาที่จะใช้สร้างโมเดลให้ได้ตรงตามความต้องการอย่างครบถ้วน ซึ่งเป็นที่มาของคำว่า Method War

จนกระทั่งกลางทศวรรษที่ 1990 ได้เริ่มมีการพัฒนา UML ขึ้น และในเดือนตุลาคม ปี ค.ศ. 1994 ก็เริ่มประกาศตัวอย่างเป็นทางการ โดย Jim Rumbaugh ได้เข้ามาทำงานร่วมกับ Grady Booch ที่บริษัท Rational ถัดจากนั้นอีก 1 ปีคือในเดือนตุลาคมปี ค.ศ. 1995 ได้มีการประกาศ Unified Method 0.8 ซึ่งเป็นฉบับร่างออกมาแต่ก็ยังไม่ประสบความสำเร็จ ต่อมาในปีเดียวกัน Ivar Jacobson ก็ได้เข้ามาทำงานในบริษัท Rational และได้นำเอาวิธีOOSE (Object-oriented Software Engineering) มารวมกับวิธีการของ Unified Method ทำให้ได้ผลลัพธ์ออกเป็น UML เวอร์ชัน 0.9 ซึ่งประกาศตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเดือนมิถุนายน ปีค.ศ. 1996 จนกระทั่งในช่วงปลายปีค.ศ. 1996 ได้มี UML เวอร์ชัน 1.0 ออกมาเป็นมาตรฐานโดยความร่วมมือจากหลายบริษัท เช่น Digital Equipment Corp., HP, i-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Rational Software, TI และ Unisys

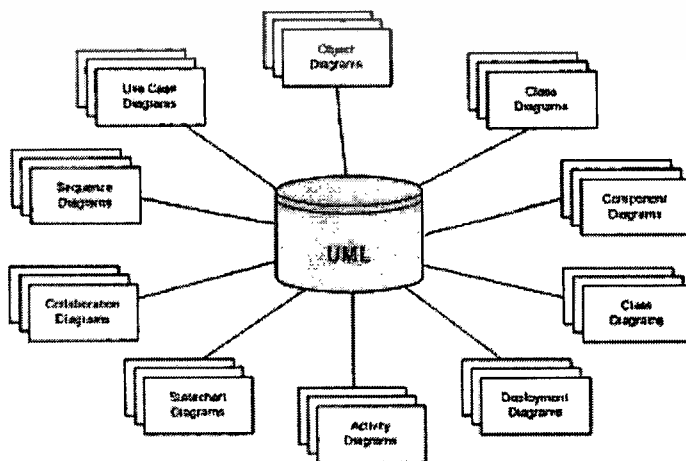
เมื่อมีการพัฒนาจนถึงเวอร์ชัน 1.1 ได้มีการยื่นขอกำหนดมาตรฐานไปยัง OMG (Object Management Group) และได้ประกาศเป็นมาตรฐานประมาณปี 1997 เป็น UML เวอร์ชัน 1.3 ซึ่งเป็นเวอร์ชันที่ทำให้องค์กรต่างๆเริ่มให้ความสนใจภาษานี้อย่างจริงจัง ปัจจุบัน UML ได้มีการพัฒนาจนถึงเวอร์ชัน 1.4



รูปที่ 4.1 แผนภาพประวัติของ UML

4.3 ประเภทของ UML Diagram

UML ได้แบ่งไดอะแกรมหลักออกเป็น 6 แบบและในบางแบบได้แตกย่อยออกไปรวมทั้งสิ้น 9 แบบ ดังแสดงในรูปที่ 4.2 ในการนำไปใช้งาน ผู้ใช้ไม่จำเป็นต้องใช้งานทุกไดอะแกรม แต่สามารถเลือกใช้ตามความเหมาะสมของระบบงานนั้นๆ

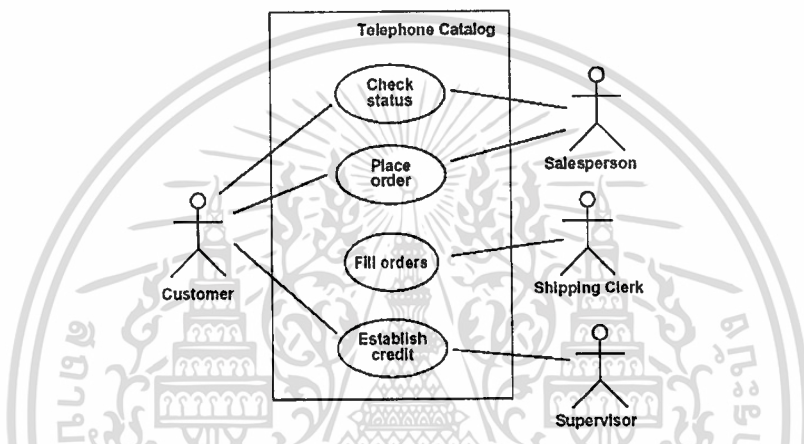


รูปที่ 4.2 ไดอะแกรมทั้งหมดใน UML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.2 ไดอะแกรมทั้งหมดใน UML เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 Use Case Diagram

Use Case Diagram เกิดจากแนวความคิดของไอวาร์ จาคอบสัน (Ivar Jacobson) มีลักษณะเป็นแผนภาพทำให้สามารถสื่อสารระหว่างผู้พัฒนาระบบและผู้ที่เกี่ยวข้องได้เป็นอย่างดีดังตัวอย่างในรูปที่ 4.3 และได้นำเข้ามารวมในมาตรฐานของ UML Use Case Diagram ถูกนำมาใช้สำหรับอธิบายความต้องการ (Requirement) ของระบบที่เราสนใจ โดยแสดงถึงความเกี่ยวข้องกันระหว่างผู้ใช้กับระบบงาน มีสัญลักษณ์ต่างๆดังนี้ คือ



รูปที่ 4.3 ตัวอย่างของ Use Case Diagram

- Actor มีสัญลักษณ์เป็นรูปคน (Stick man) ดังในรูป 4.4 อาจจะเป็นคนหรือระบบงานที่อยู่ภายนอก โดยมีความเกี่ยวข้องเป็นที่ส่งข้อมูลเข้าหรือรับข้อมูลผลลัพธ์จากระบบ โดยมากจะเป็นผู้เริ่มต้นทำงานกับ Use Case



รูปที่ 4.4 สัญลักษณ์ของ Actor

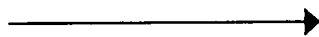
- Use Case มีสัญลักษณ์เป็นรูปวงรี แทนงานที่เกิดขึ้นในขั้นตอนต่างๆ



รูปที่ 4.5 สัญลักษณ์ของ Use Case

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Communication** มีสัญลักษณ์เป็นเส้นตรงปลายด้านหนึ่งเป็นหัวลูกศรที่บิ่สีดำ ใช้แสดงความสัมพันธ์หรือการติดต่อรับส่งข้อมูลระหว่างกันระหว่าง Actor และ Use Case โดยอาจเป็นการสื่อสารทางเดียวหรือสองทางก็ได้



รูปที่ 4.6 สัญลักษณ์ของ Communication

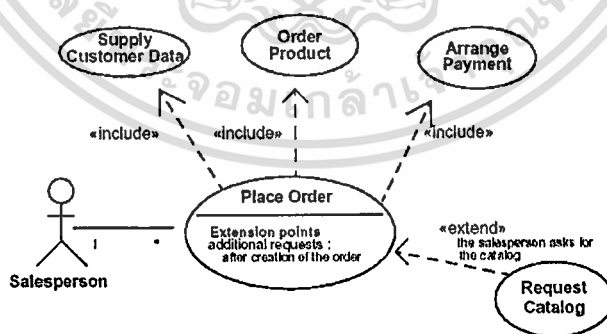
- **Relationship** มีสัญลักษณ์เป็นเส้นมีหัวเป็นลูกศรสีขาว ใช้แสดงความสัมพันธ์ระหว่าง Use Case กับ Use Case ซึ่งความสัมพันธ์ยังแบ่งได้ออกเป็นสองแบบคือ Extends และ Uses



รูปที่ 4.7 สัญลักษณ์ของ Relationship

Extends เป็นการเพิ่มการทำงานให้กับ Use Case โดยการเรียกใช้จากอีก Use Case หนึ่ง ทิศทางของลูกศรจะออกจาก Use Case ที่ถูกเรียกใช้งานยัง Use Case ที่ต้องการเพิ่มเติมการทำงาน และจะมีข้อความ <<extended>> ระบุอยู่ข้างเส้น

Uses เป็นการแสดงให้เห็นถึงการ Inherit หน้าที่การทำงานจาก Use Case หนึ่งไปอีก Use Case หนึ่ง ปลายของลูกศรจะอยู่ที่ Use Case หลักที่จะถูกถ่ายทอดความสามารถออกไป โดยมีข้อความ <<uses>> ปรากฏอยู่ข้างเส้น



รูปที่ 4.8 Use Case Relationship

4.3.2 Class Diagram

Class Diagram เป็นไดอะแกรมที่อยู่ในกลุ่ม Static ใช้แสดงถึง Class และความสัมพันธ์ระหว่าง Class ที่มีโครงสร้างของข้อมูลรวมถึงพฤติกรรมของข้อมูลที่แตกต่างกัน ซึ่งในหนึ่งคลาสสามารถที่จะกำหนดทิศทางในการนำไปสร้างโปรแกรมและการสร้างคลาสใน Object-oriented ได้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส (Class) ใช้สัญลักษณ์เป็นรูปสี่เหลี่ยม ใช้อธิบายถึงกลุ่มของอ็อบเจกต์ (Object) ต่างๆ ที่มีแอททริบิวต์ (Attribute) การทำงาน (Operation) และความสัมพันธ์ (Relationship) ขึ้นพื้นฐาน โดยในแต่ละคลาสจะประกอบไปด้วยส่วนพื้นฐาน 3 ส่วน ดังรูปที่ 4.9

Class Name
Attribute
Operator

รูปที่ 4.9 สัญลักษณ์ของ Class

- ชื่อคลาส (Class Name) แสดงถึงชื่อของคลาสที่กำหนดในระบบ
- แอททริบิวต์ (Attribute) เป็นการกำหนดคุณลักษณะทั้งหมดที่มีภายในคลาส และบอกถึงชนิดของข้อมูล ภายในคลาสจะมี Attribute ซ้ำกันไม่ได้ แต่ชื่อข้อมูลของคลาสอาจซ้ำกับชื่อข้อมูลในคลาสอื่นได้
- โอเปอเรชัน (Operation) เป็นส่วนที่ใช้อธิบายคลาสนั้นๆว่ามี Method อะไรบ้าง มีการรับค่าตัวแปร (Argument) อะไรบ้างและมีการส่งค่าอะไรออกไปบ้าง
- ความสัมพันธ์ระหว่างคลาส (Relationship) UML ได้เตรียมความสัมพันธ์ระหว่างคลาสเอาไว้ 4 แบบ คือ
 1. Association เป็นการแสดงความสัมพันธ์ระหว่างคลาส ได้ทั้งแบบทางเดียวและสองทาง โดยมีสัญลักษณ์ดังนี้



รูปที่ 4.10 สัญลักษณ์ของ Association Relationship

2. Aggregation เป็นรูปแบบพิเศษของ Association คือเป็นความสัมพันธ์ระหว่าง Whole และ Part ของมัน คือ Whole จะประกอบด้วย Parts ต่างๆของมัน ดังนั้นการคงอยู่ของ Parts จะต้องขึ้นกับ Whole หรือจะเรียกความสัมพันธ์แบบนี้ว่า “Whole-Part” ก็ได้ โดยมีสัญลักษณ์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

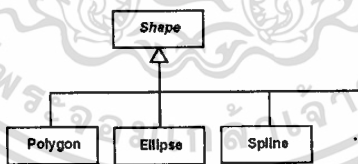
- ความสัมพันธ์แบบ By Value ◆—————
- ความสัมพันธ์แบบ By Reference ◇—————

รูปที่ 4.11 สัญลักษณ์ของ Aggregation Relationship

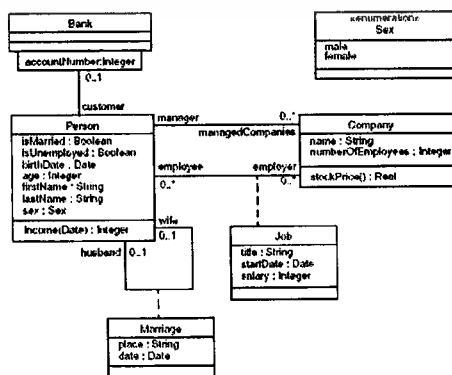
3. Depends on เป็นรูปแบบความสัมพันธ์แบบหนึ่งที่ใช้แสดงความสัมพันธ์ระหว่างคลาส 2 คลาส ในแง่ที่คลาสหนึ่งเรียกใช้บริการของอีกคลาสหนึ่ง กล่าวคือ คลาสของผู้ขอใช้บริการขึ้นอยู่กับบริการของคลาสของผู้ให้บริการ แต่ไม่มีการขึ้นต่อกันภายใน โครงสร้างของคลาสมีสัญลักษณ์ดังนี้

รูปที่ 4.12 สัญลักษณ์ของ Depend on Relationship

4. Generalization เป็นความสัมพันธ์รูปแบบที่ใช้แสดงความสัมพันธ์ระหว่างคลาสดับคลาสด ในแง่ที่คลาสหนึ่งจะถ่ายทอดคุณสมบัติและโครงสร้างจากอีกคลาสหนึ่ง โดยเรียกคลาสที่ถูกถ่ายทอดว่าซูเปอร์คลาส (Super Class) และคลาสที่ทำการถ่ายทอดว่าซับคลาส (Sub Class) มีสัญลักษณ์ดังในรูปที่ 4.13



รูปที่ 4.13 ตัวอย่างความสัมพันธ์แบบ Generalization



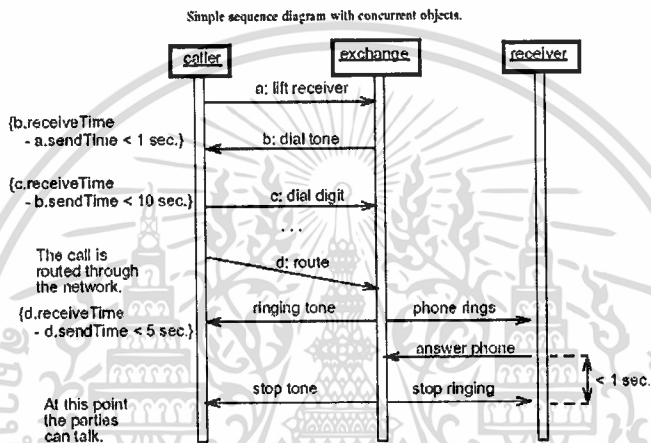
รูปที่ 4.14 ตัวอย่างของ Class Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 Sequence Diagram

เป็น Diagram ที่แสดงการทำงานระหว่างออบเจกต์ (Object) ต่างๆ ตามลำดับเหตุการณ์ที่เกิดขึ้น มีลักษณะของไคอะแกรมดังในรูปที่ 4.15 โดยมีองค์ประกอบอยู่ 3 ส่วน คือ

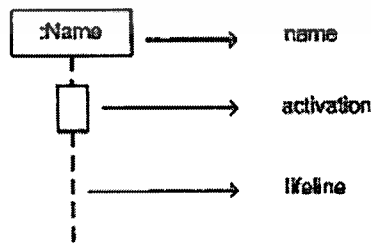
1. Object
2. Message
3. Time



รูปที่ 4.15 ตัวอย่างของ Sequence Diagram

4.3.3.1 ออบเจกต์ (Object)

ในส่วนของออบเจกต์ยังแบ่งออกได้เป็น 3 ส่วน คือ ชื่อออบเจกต์ (Object Name), Life Line และ Activation ซึ่งมีสัญลักษณ์ดังในรูปที่ 4.16

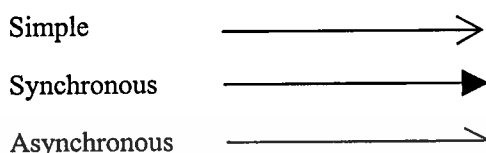


รูปที่ 4.16 องค์ประกอบของออบเจกต์

4.3.3.2 Message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Message ใน Sequence Diagram เป็นการติดต่อที่เกิดจากการส่ง Object หนึ่งไปยังอีก Object หนึ่ง ยังสามารถแบ่งชนิดของการติดต่อออกเป็นแบบย่อยๆ ได้ 3 แบบคือ Simple, Synchronous และ Asynchronous ตามลำดับ ดังรูปที่ 4.17



รูปที่ 4.17 สัญลักษณ์ของ Message ทั้ง 3 แบบ

4.3.3.3 ช่วงเวลา (Lifetime)

เป็นการแสดงเวลาโดยเริ่มต้นจากส่วนบนลงล่าง ส่วนบนจะเป็นส่วนที่เกิดขึ้นก่อน Message ที่อยู่ด้านล่าง

เนื่องจากการใช้งานไคอะแกรมของ UML สามารถที่จะเลือกใช้ได้ตามความเหมาะสม ซึ่งในระบบงานนี้จะมีการนำไคอะแกรมของ UML มาใช้เพียง 3 ไคอะแกรมเท่านั้น คือ Use Case Diagram, Class Diagram และ Sequence Diagram จึงจะไม่กล่าวถึงไคอะแกรมอื่นๆ ไว้ในที่นี้

4.4 Iterative model

โครงการนี้ได้นำ Iterative model ของ Rational Unified Process มาประยุกต์ใช้ ซึ่ง Rational Unified Process มีลักษณะสำคัญอยู่สองประการคือ

1. Rational Unified Process มีลักษณะเป็น Use-Case Driven ซึ่งหมายความว่า Use-Case เป็นแนวความคิดหลักภายในกระบวนการทำงานหรือ Process Use-Case จะถูกใช้ตลอดวงจรการพัฒนา เสมือนเป็นตัวขับเคลื่อนกิจกรรมอื่นๆ เป็นตัวที่ทำการส่งผ่านข้อมูลไปยังแบบจำลอง (Model) ต่างๆ และเป็นตัวช่วยในการตรวจสอบความถูกต้องตรงกัน (Consistency) ระหว่างแบบจำลองเหล่านั้น
2. Rational Unified Process มีลักษณะเป็น Architecture-Centric ซึ่งหมายความว่า สถาปัตยกรรม (Architecture) ถูกใช้ใน Rational Unified Process เสมือนเป็นสิ่งก่อสร้างพื้นฐาน (Primary Artifact) สำหรับการรวบรวมแนวความคิด การสร้าง การจัดการ และการก่อให้เกิดระบบภายใต้การพัฒนา

นอกจากลักษณะสองประการนี้แล้ว Rational Unified Process ยังมีการจัดแบ่งกระบวนการทำงานเป็น Life Cycle Phases ออกเป็น 4 phases ด้วยกันคือ

- **Inception** เป็นช่วงของการกำหนดขอบเขตของโครงการ
 - **Elaboration** เป็นช่วงการวางแผนโครงการ กำหนดฟังก์ชันการทำงานของระบบ รวมทั้งสถาปัตยกรรมพื้นฐานที่จะใช้ในการจัดทำระบบ
 - **Construction** เป็นช่วงการสร้างโปรแกรม ซึ่งอาจมีได้หลาย iteration
 - **Transition** เป็นช่วงของการส่งมอบโปรแกรมหรือระบบงานให้กับกลุ่มผู้ใช้งานระบบ
- ภายในแต่ละ Phase จะมีจำนวนของ iterations อยู่ โดยจำนวนของ iterations ต่อ phase นั้นจะไม่แน่นอนตายตัว ในแต่ละ iteration จะได้ผลลัพธ์เป็นโปรแกรมหรือระบบงานที่สามารถทำงานได้ ซึ่งจะเป็นส่วนหนึ่งของโปรแกรมหรือระบบงานใหญ่ ซึ่งเป็นลักษณะเด่นอย่างหนึ่งของ Iterative Model

นอกจากนี้ Rational Unified Process จะมี Workflows หลักๆที่ใช้ในการพัฒนาระบบอยู่ ซึ่งแต่ละ Workflow จะมีการสร้างแบบจำลองต่างๆ ซึ่ง Workflows ดังกล่าวประกอบไปด้วย

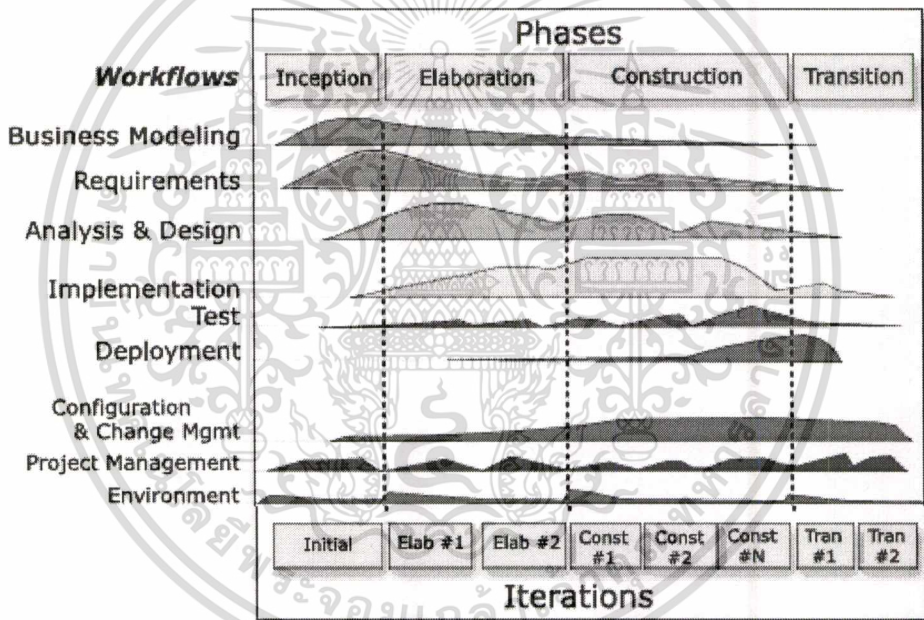
- **Business Modeling Workflow**
เป็นขั้นตอนในการทำความเข้าใจถึงโครงสร้างขององค์กร และเป็นการทำความเข้าใจร่วมกันระหว่างผู้ใช้งานกับผู้พัฒนา ซึ่งโดยทั่วไปผลลัพธ์ที่ได้มักจะเป็น Business use-case models และเอกสารประกอบอื่นๆ
- **Requirements Workflow**
เป็นขั้นตอนในการทำข้อตกลงร่วมกันว่าระบบจะสามารถทำอะไรได้บ้าง โดยอาจจัดทำเป็น Use-case model พร้อม Use-Case description หรืออาจจะมีการจัดทำ User-Interface Prototype
- **Analysis and Design Workflow**
เป็นขั้นตอนในการทำการแปลงความต้องการระบบ ไปสู่การออกแบบระบบที่ควรจะเป็น ซึ่งจะมีการกำหนดสถาปัตยกรรมของระบบ มีการปรับแต่งแบบที่ได้ทำการออกแบบไว้ ให้เหมาะสมกับสภาพแวดล้อมของการสร้างระบบ โดยผลลัพธ์ของขั้นตอนนี้อาจประกอบด้วย Design model classes, Design Packages, Software architecture document และ Data model
- **Implementation Workflow**
เป็นขั้นตอนในการกำหนดโครงสร้างของ Source code มีการสร้าง Classes และ Objects ในรูปแบบของ Components โดยผลลัพธ์ที่ได้จากขั้นตอนนี้หลักๆจะเป็น Implementation model และการกำหนด Components

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● Test Workflow

เป็นขั้นตอนของการทบทวนการโต้ตอบกันของ Objects การทบทวนการเชื่อมต่อกันที่เหมาะสมของ Component ทั้งหมด และเป็นการทบทวนความต้องการทั้งหมดว่าถูกจัดสร้างอย่างถูกต้องหรือไม่ โดยทั่วไปจะมีการจัดทำ Test model มีการกำหนด Test cases procedures และ Scripts รวมทั้งมีการทำ Test plan ไว้ด้วย

รูปที่ 4.18 แสดงถึงความสัมพันธ์ระหว่าง Phase Iteration และ Workflow ซึ่งจะเป็นลักษณะที่เรียกว่า Iterative Model โดยในแต่ละ phases จะมีได้หลาย iteration และในหนึ่ง iteration จะมีกิจกรรมที่เกิดขึ้นจากทุกๆ Workflow



รูปที่ 4.18 ความสัมพันธ์ระหว่าง Phase Iteration และ Workflow

บทที่ 5

Object-Oriented CASE Tool : Rational Rose

5.1 Rational Rose 2000 Enterprise Edition

Rational Rose 2000 Enterprise Edition เป็น Visual Modeling tool และ Object-Oriented CASE Tool ที่สามารถสร้าง UML Diagram ทั้งหมด เพื่อใช้ในการจำลองระบบงานได้ ด้วยการวาด Diagram ต่างๆ โดยมีตัวจัดการ Diagram และกำหนดคุณสมบัติต่างๆของ Diagram ที่จำเป็น การเชื่อมโยงกันระหว่าง Diagram และการจัดการเอกสารจากแบบจำลองที่ได้สร้างขึ้น รวมทั้งมีตัวช่วยอัตโนมัติในการสร้าง Class และ Add-in ต่างๆเพิ่มเติม เช่น Rational Rose J และ Rational Rose Oracle8 ที่จะกล่าวถึงต่อไป [สุรัส ธาราสมบัติ. 2544]

5.2 Rational Rose J

Rational Rose J เป็นส่วนเพิ่มเติมของ Rational Rose ที่มีคุณสมบัติดังต่อไปนี้

1. สนับสนุนวิวัฒนาการจากขั้นตอนวิเคราะห์ไปจนถึงการออกแบบ
2. สนับสนุนการทำงานเป็นทีมโดยการทำ Configuration-Management และ Version Control System
3. สามารถสร้าง Source code ภาษา Java จาก Model
4. สามารถเพิ่มเติม Classpath ภายใน Model
5. สนับสนุนไฟล์ชนิด zip, jar และ cab ใน Classpath
6. สามารถลากและวางไฟล์ชนิด java, Class, zip, jar และ cab เข้าไปใน model โดยการ ทำ Reverse Engineering อย่างรวดเร็ว
7. ขยายไฟล์แบบอัตโนมัติสำหรับไฟล์ชนิด zip, jar และ cab เมื่อขณะทำการ Reverse Engineering
8. สามารถออกแบบ จำลอง ทำให้เห็นภาพของการสร้างโปรแกรมของภาษา Java ได้แก่ package, Class, interfaces imports, inheritance, fields, methods และ modifiers
9. สามารถควบคุมคุณสมบัติต่างๆอย่างอัตโนมัติ ได้แก่ finalizers และ Static initializers และสามารถกำหนดชื่อ field-name ได้เอง
10. สนับสนุนการ Application partitioning และ distribution

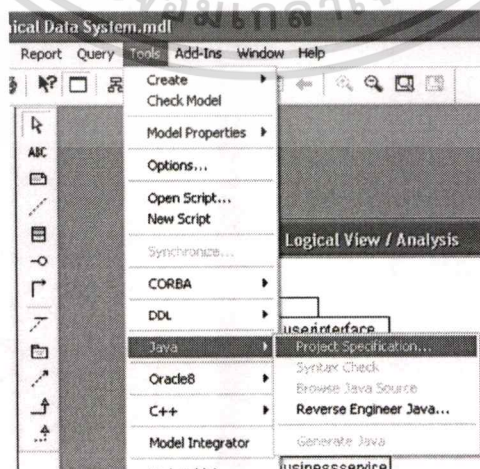
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. การ mapping อย่างฉลาดจาก model ไปสู่ code
12. สนับสนุน default package ต่างๆ
13. ปรับปรุงการสนับสนุน Class ภายในและ Classes อื่นๆที่ไม่อยู่ในระบบ
14. การสร้าง Bean (Get/Set)
15. โปรแกรม editor สามารถแสดงสีได้
16. สนับสนุน framework ขนาดใหญ่ ประกอบไปด้วย component ให้เลือกใช้งาน อาทิ
 - JDK Class Library
 - Java Database Connectivity (JDBC)
 - Java Generic Library (JGL)
 - Application Foundation Classes (AFC)
 - Windows Foundation Classes
 - Enterprise Java Frameworks (JNI, EJB, JTS)

5.2.1 ตัวอย่างการทำ Forward engineering ด้วย Rational Rose J

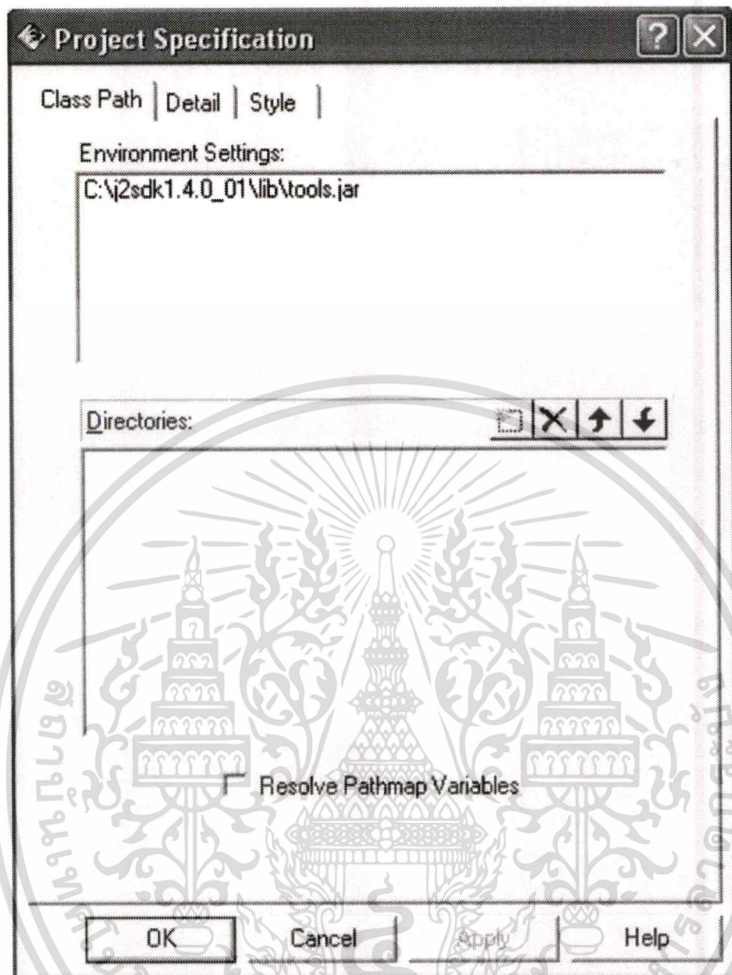
Forward engineering คือ กระบวนการสร้างโปรแกรมภาษา Java จาก Class หรือ Package หรือ component ใน Rational Rose model

ขั้นแรกจะต้องทำการเซตค่า Classpath ให้ชี้ไปยัง Directories หรือไฟล์ชนิด .jar, .cab, .zip ที่ต้องการใช้งาน โดยเพิ่มเติมในส่วนของ directories ที่เพิ่มไปจาก Classpath environment variable ที่ถูกเซตไว้ก่อนแล้ว ดังรูปที่ 4.1 และ 4.2



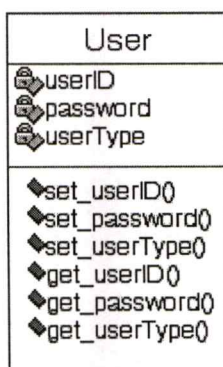
รูปที่ 5.1 วิธีการเข้าสู่หน้าจอการเซต Classpath

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 หน้าจอขณะทำการเซต Classpath ที่เก็บของไฟล์ .java, .class, .jar, .zip

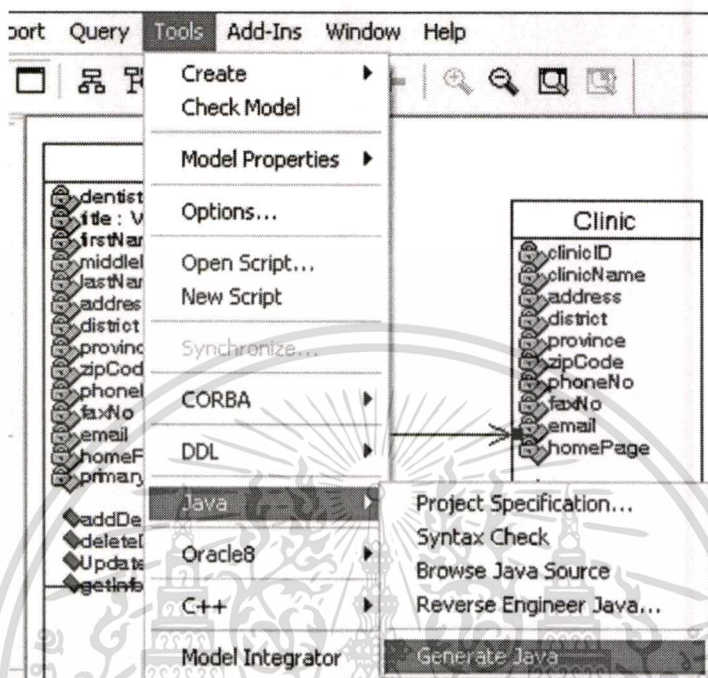
ขั้นที่ 2 ให้ทำการสร้าง Class diagram ที่ต้องการ ดังเช่นตัวอย่างต่อไปนี้ สร้าง Class User



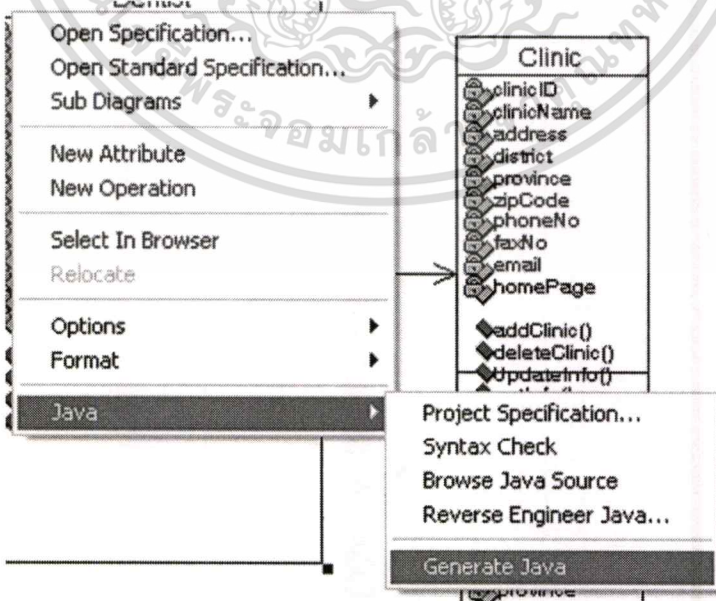
รูปที่ 5.3 Class User ใน Rational Rose Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 3 เลือก Class ที่ต้องการแล้วเลือก Tool > Java > Generate Java



รูปที่ 5.4 วิธีการสั่งให้สร้าง Source Code จาก Class ที่ต้องการ



รูปที่ 5.5 วิธีการดูผลลัพธ์ Source Code จาก Class ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

E:\TomcatServer\jakarta-tomcat-4.0.4\webapps\dental\WEB-INF\classes\User.java
File Edit Format Help
Ln1Col1
//Source file: E:/TomcatServer/jakarta-tomcat-4.0.4/webapps/dental/WEB-INF/classes/User.java

public class User
{
    private int userID;
    private String password;
    private int userType;

    public User()
    {
    }

    /**
     *roseuid 3D7F5B6A0108
     */
    public void set_userID()
    {
    }

    /**
     *roseuid 3D7F5B7D03C7
     */
    public void set_password()
    {
    }

    /**
     *roseuid 3D7F5D5101F2
     */
    public void set_userType()
    {
    }

    /**
     *roseuid 3D7F5B6D03C7
     */
    public void get_userID()
    {
    }

    /**

```

รูปที่ 5.6 ผลลัพธ์ Source Program

เมื่อทำตามขั้นตอนจะสามารถเข้าสู่ผลลัพธ์ได้ดังรูปที่ 5.5 และ 5.6 ไฟล์ .java ที่สร้างขึ้นจะบันทึกอยู่ใน Directory ที่กำหนดไว้ใน Classpath

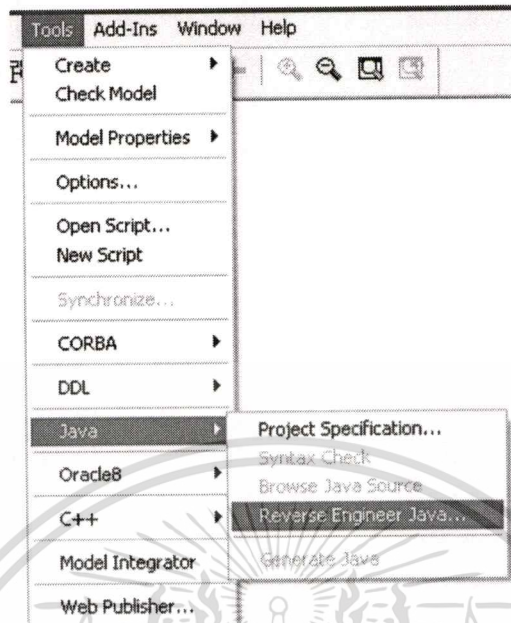
5.2.2 ตัวอย่างการทำ Reverse engineering ด้วย Rational Rose J

Reverse Engineering คือกระบวนการวิเคราะห์ Source Code ภาษา Java แล้วแปลงกลับไปเป็น Class หรือ Component Model ใน Rational Rose

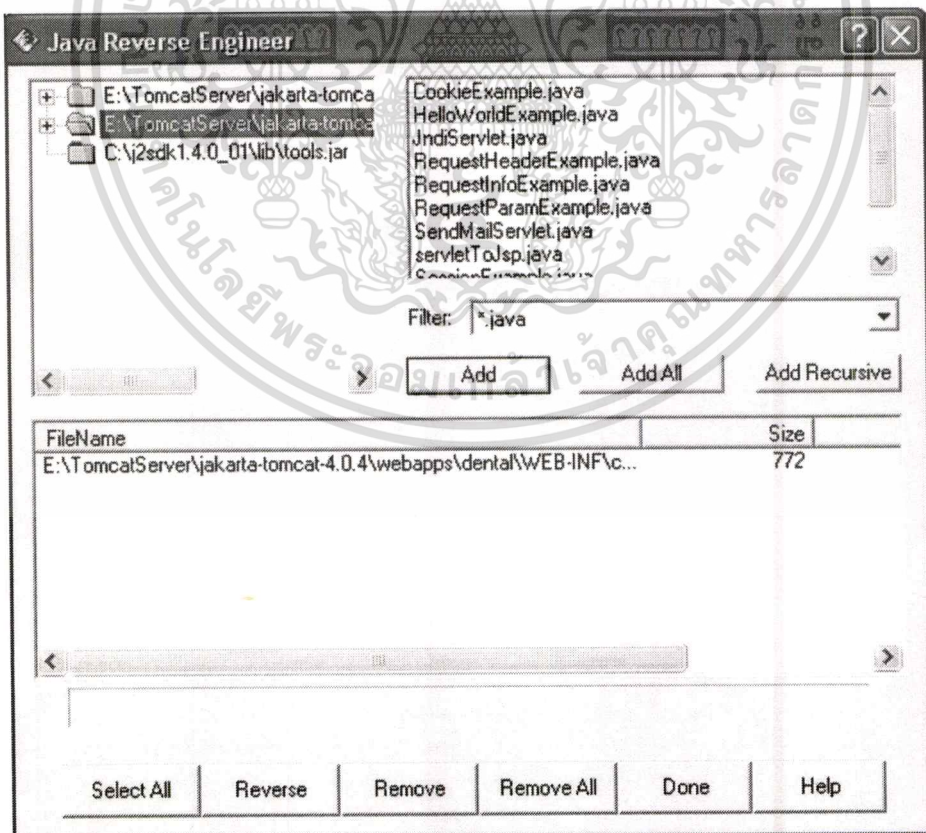
เช่นเดียวกันกับขั้นแรก จะต้องทำการเซต Classpath ให้ชี้ไปยัง Directories หรือไฟล์ชนิด .jar, .cab, .zip ที่ต้องการใช้งาน โดยเพิ่มเติมในส่วนของ Directories ที่เพิ่มไปจาก Classpath Environment Variable ที่ถูกกำหนดไว้ก่อนแล้ว ดังรูปที่ 5.1 และ 5.2

ขั้นที่ 2 เลือกไฟล์ .jar, .class ที่ต้องการ โดยเข้าที่เมนู Tool > Java > Reverse Engineering Java ดังในรูปที่ 5.7 และ 5.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 วิธีการเลือกเมนูเพื่อทำ Reverse Engineering



รูปที่ 5.8 ตัวอย่างการเลือกไฟล์จาก Directory มาทำ Reverse Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นสุดท้าย หลังจากได้เลือกไฟล์เรียบร้อยแล้ว ให้กดปุ่ม Reverse โปรแกรมจะทำการ Reverse ออกมาเป็น Class Diagram ที่ต้องการ

5.3 Rational Rose Oracle8

ในโครงการนี้มีการใช้ระบบจัดการฐานข้อมูลเป็น MySQL ซึ่งไม่ได้เกี่ยวข้องกับส่วนของ Rational Rose Oracle 8 โดยตรง แต่เนื่องจากส่วนของ Rational Rose Oracle 8 สามารถช่วยอำนวยความสะดวกในการสร้าง Database จึงได้นำมาใช้งานโดยใช้ให้โปรแกรมสร้างผลลัพธ์เป็น DDL (Database Definition Language) เพื่อนำไปใช้ในการสร้างฐานข้อมูลบน MySQL ต่อไป โดยในหัวข้อนี้จะกล่าวถึง Rational Rose Oracle เฉพาะในส่วนที่เกี่ยวข้องเท่านั้น

ส่วนของ Rational Rose Oracle 8 คือส่วนเพิ่มเติมของ Rational Rose ที่มีคุณสมบัติดังต่อไปนี้

1. สามารถสร้าง Object models จาก Oracle8 relational schema และขยายความสามารถไปสู่มุมมองเชิงวัตถุของ Oracle 8
2. สามารถทำให้มองเห็นภาพ หรือจำลอง relational database ที่มีอยู่เดิม และอำนวยความสะดวกในการค้นหาและประกอบเข้าด้วยกันกับ Business Object ซึ่งจะช่วยลดการลงทุนสำหรับการปรับปรุง relational database
3. สามารถทำให้ผู้พัฒนาระบบสร้างแบบจำลองทางธุรกิจในรูปแบบของภาษาที่ใช้ในการพัฒนาโปรแกรมและฐานข้อมูล รวมทั้งตัวช่วยอัตโนมัติที่นำทางในการสร้าง Relational table, Triggers, Object types, Object views, VARRAYs, Nested tables และคุณสมบัติเชิงวัตถุต่างๆใน Oracle8

5.3.1 การทำ Database Normalization

การจำลองแบบของข้อมูลสำหรับ Rational Rose Oracle8 ยังจำเป็นต้องผ่านกระบวนการออกแบบฐานข้อมูล และการทำ Normalize ข้อมูลจริงเสียก่อน ในระบบงานนี้ ได้ทำการ Map ข้อมูลจาก Entity Classes ซึ่งมีรายละเอียดอยู่ในบทที่ 7 แล้วทำการ Normalize ข้อมูลเพิ่มเติม จากนั้นจึงนำตารางที่ผ่านการ Normalize มาแล้วสร้างเป็น Class Diagram รวมไว้ใน Relational Databases Package แล้วจึงนำไปสร้าง Database schema

บทที่ 6

การวิเคราะห์ระบบงาน

6.1 ความต้องการของระบบงาน

จากความต้องการที่จะพัฒนาระบบงานเพื่อแก้ไขปัญหาต่างๆตามที่ได้กล่าวมาแล้วข้างต้น จำเป็นต้องนำความต้องการมาทำการวิเคราะห์เพื่อให้ทราบถึงผู้ที่เกี่ยวข้องภายในระบบงาน หน้าที่การทำงานของระบบ และเป็นแนวทางสำหรับการพัฒนาระบบงานต่อไป เครื่องมือที่จะใช้ในขั้นตอนนี้คือ Use Case Diagram เมื่อแบ่งระบบงานออกตามหน้าที่การทำงานแล้วสามารถแบ่งออกได้เป็น 2 ขั้นตอน คือ

6.1.1 การลงทะเบียนเพื่อเป็นผู้ใช้ในระบบ

ผู้ใช้ในระบบประกอบด้วยทันตแพทย์ (Dentist) คนไข้ (Patient) และเจ้าของคลินิก (Clinic's Owner) ซึ่งก่อนที่จะเริ่มใช้งานในระบบงานนี้ได้จำเป็นต้องมีการลงทะเบียนก่อนในเบื้องต้น ทั้งนี้เพื่อเก็บข้อมูลที่จำเป็นของทันตแพทย์ คนไข้ และคลินิก โดยผู้ใช้งานจะต้องกำหนดชื่อผู้ใช้และรหัสผ่านสำหรับการเข้าใช้งาน และการยืนยันไปยังผู้ลงทะเบียนเพื่อให้ผู้ใช้สามารถเข้ามาใช้งานระบบได้

6.1.2 การเข้าใช้งานระบบตามปกติ

เมื่อผ่านการลงทะเบียนแล้ว ทั้งทันตแพทย์และคนไข้สามารถที่จะเข้าใช้งานในระบบงานได้ โดยผ่านทางกลไกการล็อกอินสู่ระบบ ทันตแพทย์และคนไข้จะมีหน้าที่การทำงานงานที่แตกต่างกัน ซึ่งสามารถแยกตามหน้าที่ได้ดังนี้

ทันตแพทย์ :

- สร้างความสัมพันธ์ระหว่างทันตแพทย์และคนไข้
- ค้นหารายชื่อคนไข้ เพื่อดูประวัติส่วนตัวและประวัติการรักษา
- บันทึกข้อมูลการวินิจฉัยเบื้องต้นให้กับคนไข้
- บันทึกข้อมูลการรักษา

คนไข้ :

- การอนุญาตให้ทันตแพทย์สามารถดูข้อมูลของคนไข้ได้
- ดูประวัติส่วนตัว การวินิจฉัยโรคและประวัติการรักษา

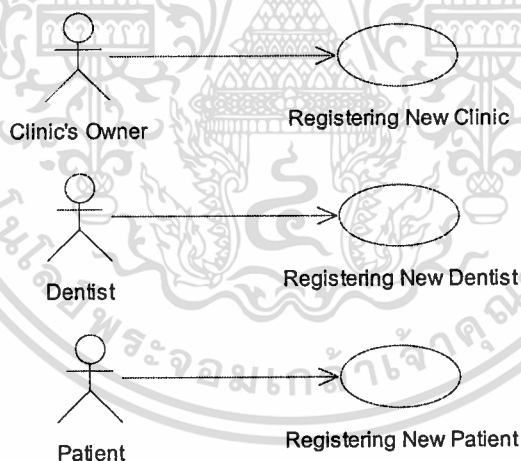
เจ้าของคลินิก :

- แก้ไขรายละเอียดของคลินิก

6.2 Use Case Diagram

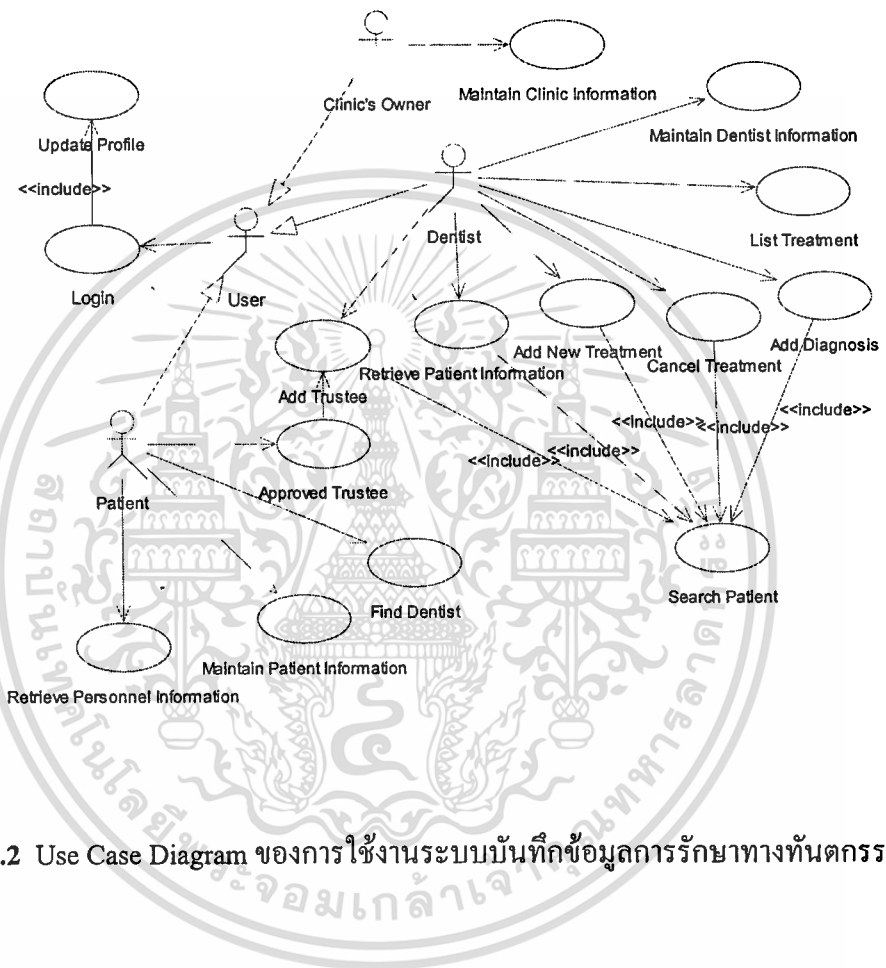
เมื่อนำความต้องการของระบบมาเขียนให้อยู่ในรูปของ Use Case Diagram สามารถแบ่งงานออกเป็นระบบย่อยดังนี้

- ระบบการลงทะเบียนผู้ใช้ (Registration)



รูปที่ 6.1 เป็น Use Case Diagram ของการลงทะเบียน

- ระบบข้อมูลประวัติการรักษาทางทันตกรรม



รูปที่ 6.2 Use Case Diagram ของการใช้งานระบบบันทึกข้อมูลการรักษาทางทันตกรรม

6.3 Use Case Description

จาก Use Case Diagram ที่ได้แสดงไว้ จะมีรายละเอียด ขั้นตอนการทำงานหรือขั้นตอนการใช้งานของแต่ละ Use Case ประกอบด้วย

6.3.1 ระบบการลงทะเบียนผู้ใช้งาน (User Registration)

6.3.1.1 Use Case – Registering New Dentist

Actors: Dentist

Description: Dentist ทำการลงทะเบียน โดยเป็นผู้กำหนดรหัสผู้ใช้ และรหัสผ่าน และรายละเอียดอื่นๆที่จำเป็น

Basic Flow:

1. Dentist เข้าสู่หน้าจอ Login ของระบบ เลือกเมนูหัวข้อ ลงทะเบียนทันตแพทย์
2. ระบบจะเปิดหน้าจอเพื่อให้กำหนดรหัสเข้าใช้งาน รหัสผ่าน พร้อมทั้งกรอกรายละเอียดอื่นๆที่จำเป็น เช่น รหัสประจำตัวทันตแพทย์, ชื่อ, นามสกุล, ที่อยู่, Email Address เป็นต้น
3. เลือกคลินิกที่สังกัด กรณีที่ยังไม่มีคลินิกให้เลือกและ Dentist เป็น Clinic's Owner ให้ไปทำการลงทะเบียนคลินิก
4. บันทึกข้อมูลประวัติส่วนตัว
5. ระบบแสดงผลของการบันทึกว่าเรียบร้อยหรือไม่ ซึ่งถ้าไม่เรียบร้อยจะแสดงข้อผิดพลาดกลับมาพร้อมทั้งรอรับรายละเอียดจากการส่งครั้งต่อไป

ขั้นตอนต่อไปผู้ใช้จะต้องรอการยืนยันจากระบบผ่านทาง E-mail โดยจะมีรหัสผู้ใช้และรหัสผ่านที่ผู้ใช้ได้ให้ไว้ในตัวจดหมาย เมื่อถึงขั้นตอนนี้ทันตแพทย์ก็สามารถเข้าใช้งานระบบได้ทันที

6.3.1.2 Use Case – Registering New Patient

Actors: Patient

Description: คนไข้ทำการลงทะเบียน โดยเป็นผู้กำหนดรหัสผู้ใช้ และรหัสผ่าน และรายละเอียดอื่นๆที่จำเป็น

Basic Flow:

1. ทันทแพทย์เข้าสู่หน้าจอ Login ของระบบ เลือกเมนูหัวข้อ ลงทะเบียนทันตแพทย์
2. ระบบจะเปิดหน้าจอเพื่อให้กำหนดรหัสเข้าใช้งาน รหัสผ่าน พร้อมทั้งกรอก รายละเอียดอื่นๆที่จำเป็น เช่น รหัสประจำตัวประชาชน, ชื่อ, นามสกุล, ที่อยู่, Email Address เป็นต้น
3. บันทึกข้อมูลประวัติส่วนตัว
4. ระบบแสดงผลของการบันทึกว่าเรียบร้อยหรือไม่ ซึ่งถ้าไม่เรียบร้อยจะแสดง ข้อผิดพลาดกลับมาพร้อมทั้งรอรับรายละเอียดจากการส่งครั้งต่อไป

ขั้นตอนต่อไปผู้ใช้จะต้องรอการยืนยันจากระบบผ่านทาง E-mail โดยจะมีรหัสผู้ใช้และ รหัสผ่านที่ผู้ใช้ได้ให้ไว้ในตัวจดหมาย เมื่อถึงขั้นตอนนี้คนไข้ก็สามารถเข้าใช้งานระบบ ได้ทันที

6.3.1.3 Use Case – Registering New Clinic

Actors: Clinic's Owner

Description: Clinic's Owner ทำการลงทะเบียน โดยเป็นผู้กำหนดรหัสผู้ใช้ และ รหัสผ่าน พร้อมทั้งรายละเอียดของคลินิกที่จำเป็น

Basic Flow:

1. ทันตแพทย์เข้าสู่หน้าจอ Login ของระบบ เลือกเมนูหัวข้อ ลงทะเบียนคลินิกทันตกรรม
2. ระบบจะเปิดหน้าจอเพื่อให้กำหนดรหัสเข้าใช้งาน รหัสผ่าน พร้อมทั้งกรอก รายละเอียดอื่นๆที่จำเป็น เช่น ชื่อสถานพยาบาล, ชื่อ-นามสกุลของเจ้าของ, ที่อยู่, Email Address ที่ใช้ติดต่อ เป็นต้น
3. ส่งข้อมูลที่บันทึกไปยังระบบ โดยกดปุ่ม Submit
4. ระบบแสดงผลของการบันทึกว่าเรียบร้อยหรือไม่ ซึ่งถ้าไม่เรียบร้อยจะแสดง ข้อผิดพลาดกลับมาพร้อมทั้งรอรับรายละเอียดจากการส่งครั้งต่อไป

ขั้นตอนต่อไป Clinic's Owner จะต้องรอการยืนยันจากระบบผ่านทาง E-mail โดยจะมีรหัส ผู้ใช้และรหัสผ่านที่ผู้ใช้ได้ให้ไว้ในตัวจดหมาย รวมถึงรหัสประจำคลินิก เมื่อถึงขั้นตอน นี้ Clinic's Owner ก็สามารถเข้าใช้งานระบบได้ที่ทันที

6.3.2 ระบบข้อมูลประวัติการรักษาทางทันตกรรม (User Registration)

6.3.2.1 Use Case – Login

Actors: User

Description: User กรอกรหัสผู้ใช้และ รหัสผ่าน ระบบจะตรวจสอบก่อนอนุญาตให้เข้าสู่ระบบ

Basic Flow:

1. ผู้ใช้เข้าสู่หน้าจอ Login ของระบบ ระบบรหัสผู้ใช้และรหัสผ่านในช่องที่กำหนด ซึ่งขึ้นกับสถานะของผู้ใช้ว่าเป็นทันตแพทย์หรือคนไข้ จากนั้นจึงกดปุ่ม Sign-in
2. ระบบนำรหัสผู้ใช้และรหัสผ่าน ไปตรวจสอบ
3. ระบบแสดงหน้าจอหลักเพื่อให้ผู้ใช้ทำงานต่อไป ซึ่งขึ้นกับสถานะของผู้ใช้ผู้นั้น

6.3.2.2 Use Case – Add Trustee

Actors: Dentist

Description: เป็นการอนุญาตให้ Dentist สามารถเข้าถึงข้อมูลของคนไข้ได้ และสามารถบันทึกการวินิจฉัย การรักษาให้กับผู้ป่วยรายนั้นๆได้

Basic Flow:

1. หลังจากผู้ใช้ Login เข้าสู่ระบบแล้ว Dentist เลือกเมนูเพื่อเพิ่มชื่อคนไข้ในความดูแล
2. ระบบแสดงหน้าจอให้ Dentist ระบบรหัสประจำตัวของคนไข้หรือให้ค้นหาด้วยชื่อ
3. เมื่อได้รหัสประจำตัวคนไข้แล้วกดปุ่ม Submit
4. ระบบทำการดึงข้อมูลคนไข้บางส่วนมาแสดง เช่น ชื่อ นามสกุล ที่อยู่ มาแสดง พร้อมทั้งรอให้ Patient กรอกรหัสสำหรับอนุญาต เมื่อกรอกแล้วกดปุ่ม Accept
5. เมื่อผ่านการอนุญาตอย่างถูกต้อง ระบบจะสร้างความสัมพันธ์ระหว่างทันตแพทย์กับคนไข้เอาไว้
6. ระบบแสดงผลลัพธ์ของการอนุญาต

เมื่อผ่านขั้นตอนการอนุญาตแล้ว ทันตแพทย์จึงมีสิทธิในการเรียกดูข้อมูลของคนไข้รายนั้นได้รวมถึงการเข้าบันทึกการวินิจฉัย บันทึกการรักษาคนไข้

6.3.2.3 Use Case – Approved Trustee

Actors: Patient

Description: เป็นการอนุญาตให้ Dentist สามารถเข้าถึงข้อมูลของคนไข้ได้ และสามารถบันทึกการวินิจฉัย การรักษาให้กับผู้ป่วยรายนั้นๆได้

Basic Flow:

1. เมื่อ Dentist ทำขั้นตอน Use Case Add Trustee
2. ระบบทำการดึงข้อมูลคนไข้บางส่วนมาแสดง เช่น ชื่อ นามสกุล ที่อยู่ มาแสดง พร้อมทั้งรอให้ Patient กรอกรหัสสำหรับอนุญาต เมื่อกรอกแล้วกดปุ่ม Accept
3. เมื่อผ่านการอนุญาตอย่างถูกต้อง ระบบจะสร้างความสัมพันธ์ระหว่างทันตแพทย์กับคนไข้เอาไว้
4. ระบบแสดงผลลัพธ์ของการอนุญาต

6.3.2.4 Use Case – Maintain Dentist information

Actors: Dentist

Description: Dentist ทำการปรับปรุงแก้ไขข้อมูลส่วนบุคคล

Basic Flow:

1. หลังจากผู้ใช้ Login เข้าสู่ระบบแล้ว Dentist เลือกเมนูเพื่อข้อมูลส่วนบุคคล
2. ระบบแสดงหน้าจอข้อมูลส่วนบุคคล ให้เลือกเมนูย่อย เปลี่ยนแปลงข้อมูล
3. ระบบแสดงข้อมูลของ Dentist เปิดให้ Dentist ทำการแก้ไขข้อมูลที่ต้องการ
4. เมื่อแก้ไขเรียบร้อยแล้วกดปุ่ม Submit
5. ระบบแสดงผลลัพธ์ของการบันทึกการแก้ไข

6.3.2.5 Use Case – Retrieve Patient Information

Actors: Dentist

Description: Dentist ทำการดึงข้อมูลของคนไข้เพื่อนำมาพิจารณาประกอบการรักษา

Basic Flow:

1. Dentist ทำการ Login เข้าสู่ระบบแล้วเลือกเมนู ประวัติการรักษาของคนไข้
2. ระบบแสดงหน้าจอประวัติการรักษาของคนไข้ โดยทันตแพทย์จะต้องกรอกชื่อนามสกุลของคนไข้ หรือรหัสประจำตัวประชาชน ถ้าหากไม่รู้สามารถกดปุ่ม Search เพื่อระบบจะได้ไปทำงานในส่วน Use Case Search Patient ต่อไป
3. ระบบทำการตรวจสอบสิทธิ์ในการเข้าถึงข้อมูลว่าทันตแพทย์มีสิทธิ์เข้าถึงข้อมูลของคนไข้รายที่ระบุไว้หรือไม่
4. ระบบแสดงข้อมูลรายละเอียดของคนไข้รายนั้น พร้อมทั้งแสดงประวัติการรักษา การวินิจฉัยที่ผ่านมา

6.3.2.6 Use Case – Add Treatment Record

Actors: Dentist

Description: Dentist บันทึกข้อมูลการรักษาของคนไข้เข้าระบบ

Basic Flow:

1. Dentist ทำการ Login เข้าสู่ระบบแล้วเลือกเมนูบันทึกการรักษา
2. ระบบแสดงหน้าจอบันทึกการรักษา โดยทันตแพทย์จะต้องกรอกชื่อนามสกุลของคนไข้ หรือรหัสประจำตัวประชาชน ถ้าหากไม่รู้สามารถกดปุ่ม Search เพื่อระบบจะได้ไปทำงานในส่วน Use Case Search Patient ต่อไป
3. ระบบทำการตรวจสอบสิทธิ์ในการเข้าถึงข้อมูลว่าทันตแพทย์มีสิทธิ์เข้าถึงข้อมูลของคนไข้รายที่ระบุไว้หรือไม่
4. ระบบจะเปิดหน้าจอบันทึกข้อมูล โดยแสดงรายละเอียดของคนไข้บางส่วน ทันตแพทย์จะระบุการรักษาลงไปในฟอร์ม เมื่อเสร็จกดปุ่ม Save เพื่อบันทึกข้อมูลเข้าสู่ระบบ
5. ระบบแสดงผลตอบรับจากการบันทึกการรักษา

6.3.2.7 Use Case – Cancel Treatment Record

Actors: Dentist

Description: Dentist บันทึกการยกเลิกข้อมูลการรักษาของคนไข้เข้าระบบ

Basic Flow:

1. Dentist ทำการ Login เข้าสู่ระบบแล้วเลือกเมนูบันทึกยกเลิกการรักษา
2. ระบบแสดงหน้าจอบันทึกการรักษา โดยทันตแพทย์จะต้องกรอกชื่อ นามสกุลของคนไข้ หรือรหัสประจำตัวประชาชน ถ้าหากไม่รู้สามารถกดปุ่ม Search เพื่อระบบจะได้ไปทำงานในส่วน Use Case Search Patient ต่อไป
3. ระบบทำการตรวจสอบสิทธิ์ในการเข้าถึงข้อมูลว่าทันตแพทย์มีสิทธิ์เข้าถึงข้อมูลของคนไข้รายที่ระบุไว้หรือไม่
4. ระบบแสดงรายการรักษาที่ผ่านมาเฉพาะที่รักษาโดย Dentist นั้นๆ
5. Dentist เลือกรายการที่ต้องการยกเลิก แล้วกดปุ่ม Cancel เข้าระบบ
6. ระบบแสดงผลตอบรับจากการบันทึกการยกเลิกรายการรักษาที่ผ่านมา

6.3.2.8 Use Case – List Treatment

Actors: Dentist

Description: Dentist ทำการดึงข้อมูลการรักษาภายในช่วงเวลาที่กำหนด

Basic Flow:

1. Dentist ทำการ Login เข้าสู่ระบบแล้วเลือกเมนูข้อมูลการรักษาตามช่วงเวลา
2. ระบบแสดงหน้าจอข้อมูลการรักษาตามช่วงเวลา รอให้ Dentist ระบุรายละเอียดช่วงเวลา โดยมีค่าปกติเป็นเวลาเริ่มต้นคือวันที่ 1 ของเดือนปัจจุบัน วันสิ้นสุดคือวันปัจจุบัน
3. Dentist ต้องระบุจุดเริ่มต้นและจุดสิ้นสุดของเวลาที่ต้องการในรูปแบบ วัน/เดือน/ปี จากนั้นกดปุ่ม Submit
4. ระบบจะแสดงข้อมูลการรักษาในช่วงที่กำหนด

6.3.2.9 Use Case – Maintain Patient Information

Actors: Patient

Description: Dentist ทำการปรับปรุงแก้ไขข้อมูลส่วนบุคคล

Basic Flow:

1. หลังจากผู้ใช้ Login เข้าสู่ระบบแล้ว Patient เลือกเมนูเพื่อแก้ไขข้อมูลส่วนบุคคล
2. ระบบแสดงข้อมูลของ Patient เปิดให้ Dentist ทำการแก้ไขข้อมูลที่ต้องการ
3. เมื่อแก้ไขเรียบร้อยแล้วกดปุ่ม Submit
4. ระบบแสดงผลลัพธ์ของการบันทึกการแก้ไข

6.3.2.10 Use Case – Retrieve Treatment Record

Actors: Patient

Description: Patient เรียกข้อมูลของประวัติการรักษา

Basic Flow:

1. Patient ทำการ Login เข้าสู่ระบบแล้วเลือกเมนูประวัติการรักษา
2. ระบบแสดงข้อมูลรายละเอียดของคนไข้รายนั้น พร้อมทั้งแสดงประวัติการรักษา การวินิจฉัยที่ผ่านมา

6.3.2.11 Use Case – Maintain Clinic information

Actors: Clinic's Owner

Description: Clinic's Owner ทำการปรับปรุงแก้ไขข้อมูลคลินิก

Basic Flow:

1. หลังจากผู้ใช้ Login เข้าสู่ระบบแล้ว Clinic's Owner เลือกเมนูเพื่อแก้ไขข้อมูลคลินิก
2. ระบบแสดงข้อมูลของคลินิกและเปิดให้ Clinic's Owner ทำการแก้ไขข้อมูลที่ต้องการ
3. เมื่อแก้ไขเรียบร้อยแล้วกดปุ่ม Submit
4. ระบบแสดงผลลัพธ์ของการบันทึกการแก้ไข

6.3.2.12 Use Case – Find dentist

Actors: Patient

Description: Patient ทำการค้นหารายละเอียดของ Dentist

Basic Flow:

1. หลังจากผู้ใช้ Login เข้าสู่ระบบแล้ว Patient เลือกเมนูเพื่อค้นหาข้อมูลทันตแพทย์
2. ระบบแสดงเปิดให้กรอกชื่อ หรือนามสกุลของทันตแพทย์ เมื่อใส่ข้อมูลแล้ว กดปุ่ม Submit
3. ระบบแสดงผลลัพธ์ของการค้นหา

6.3.2.13 Use Case – Search Patient

Actors: Use Case Add Trustee, Use Case Retrieve Patient Information, Use Case Add New Treatment, Use Case Cancel Treatment, Use Case Add Diagnosis

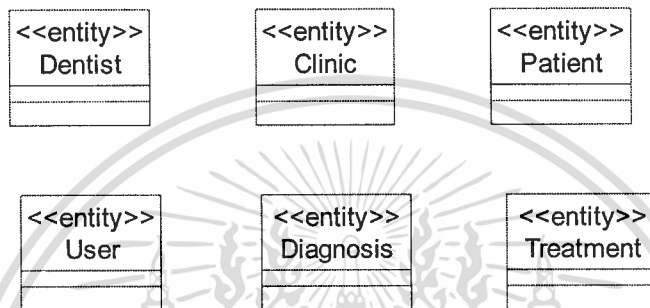
Description: Dentist ทำการค้นหารายชื่อคนไข้เมื่อไม่ทราบถึงรหัสของคนไข้ โดยผ่านทาง การเปิดสิทธิ์จากคนไข้, การเรียกข้อมูลคนไข้, การบันทึกการรักษา, การยกเลิกรายการที่ผิดพลาด, การบันทึกรายการวินิจฉัย

Basic Flow:

1. Use Case นี้เริ่มจากเมื่อผู้ใช้กดปุ่มค้นหารายชื่อคนไข้ ซึ่งกดปุ่มได้จากหน้าจอ การเปิดสิทธิ์จากคนไข้, การเรียกข้อมูลคนไข้, การบันทึกการรักษา, การยกเลิกรายการที่ผิดพลาด, การบันทึกรายการวินิจฉัย
2. ระบบแสดงหน้าจอให้ผู้ใช้ป้อนสิ่งที่ทราบคือ ชื่อหรือนามสกุลของคนไข้
3. เมื่อกรอกเรียบร้อยแล้วกดปุ่ม Submit เพื่อให้ระบบทำการค้นหา
4. ระบบแสดงผลลัพธ์ของการค้นหา กลับมายังหน้าจอ

6.4 Preliminary Analysis Classes

จากขั้นตอนนี้การทำ Use Case Model และ Use Case Description (Scenario) ซึ่งเป็นเอกสารความต้องการ (Requirement Document) นำเอกสารความต้องการดังกล่าวมาทำการหา preliminary analysis classes หรือ เรียกอีกอย่างได้ว่า Key Abstractions ซึ่งเราสามารถหาได้ดังรูปที่ 6.1



รูปที่ 6.3 Key Abstractions ของระบบ

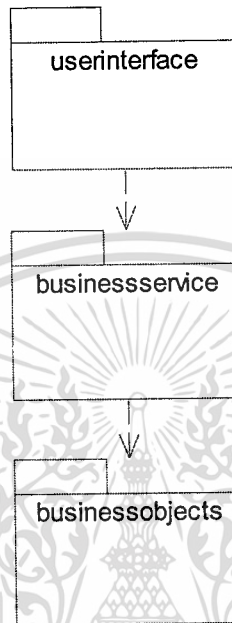
6.5 Architecture Analysis

ในขั้นตอนนี้ของการวิเคราะห์ระบบ เราควรที่จะทำการวางสถาปัตยกรรมเบื้องต้น (Initial Architecture) ของระบบ เพื่อเป็นการจัดแบ่งระบบงานออกเป็นส่วนๆไว้ และเพื่อเป็นการวางแนวทางในการวิเคราะห์ระบบในขั้นต่อไป ซึ่งจะต้องทำการวิเคราะห์และออกแบบตามแนวทางของสถาปัตยกรรมที่วางเอาไว้ (Architecture Centric) โดยเราสามารถที่จะเลือกใช้สถาปัตยกรรมที่มีผู้คิดไว้แล้ว หรือที่เรียกว่า Architecture patterns มาทำการประยุกต์ใช้กับระบบงานของเราได้ โดยในระบบนี้ได้นำ Architecture patterns มาใช้ดังนี้

- **Layers** โดย Layers pattern นั้น จะเป็นลักษณะที่ระบบงานถูกแบ่งออกเป็นส่วนๆในชั้น (Layer) ที่แตกต่างกัน โดยมีลำดับชั้นดังนี้ คือ ชั้นที่มีความเกี่ยวข้องกับแอปพลิเคชัน (Application-specific layers) จะอยู่ด้านบนสุด ไปจนถึงชั้นที่อยู่ติดกับเทคโนโลยีที่ใช้ในการสร้าง (Implementation/technology-specific layers)
- **Model-View-Controller (M-V-C)** MVC Model เป็นลักษณะการแบ่งระบบงานออกเป็นส่วนต่างๆ 3 ส่วนเพื่อแยกกันทำหน้าที่โดย Model จะเป็นส่วนของกฎเกณฑ์ทางธุรกิจ (Business Logic) รวมถึงข้อมูลต่างๆที่เกี่ยวข้อง View จะเป็นส่วนของการ

แสดงข้อมูลเพื่อติดต่อกับผู้ใช้ และ Controller เป็นส่วนที่ทำการประมวลผลข้อมูลที่ได้รับมาจากผู้ใช้

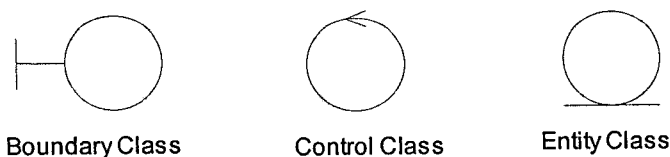
สถาปัตยกรรมของระบบงานมีลักษณะดังในรูปที่ 6.4



รูปที่ 6.4 Initial Architecture ของระบบงาน

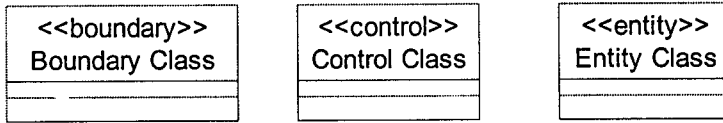
6.6 Analysis Classes

ขั้นตอนต่อไปของการวิเคราะห์ระบบงาน เป็นขั้นตอนที่จะหา และกำหนดคลาสในระดับของการวิเคราะห์ (Analysis Classes) ซึ่งเทคนิคที่ใช้ในการหา Analysis Classes คือการใช้มุมมองที่แตกต่างกัน 3 มุมมองของระบบ เป็นส่วนช่วยในการกำหนด หรือหาคลาสที่เกี่ยวข้อง ซึ่งมุมมองทั้งสามแบบ ได้แก่ ขอบเขต (Boundary) ระหว่างผู้ใช้งานกับระบบงาน ข้อมูลที่ระบบงานจะต้องใช้ และตรรกะที่ใช้ในการควบคุม (Control Logic) ของระบบงาน โดยมุมมองทั้งสามนี้จะถูกกำหนดให้เป็นประเภทของคลาสของ Analysis Classes โดยกำหนดให้เป็น Stereotype ของ Class ซึ่งสามารถแสดงสัญลักษณ์เป็นรูปได้ดังในรูปที่ 6.5 หรือแสดงเป็นแบบ Label ดังรูปที่ 6.6



รูปที่ 6.5 Boundary Class Control Class และ Entity Class เมื่อแสดงแบบ icon

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

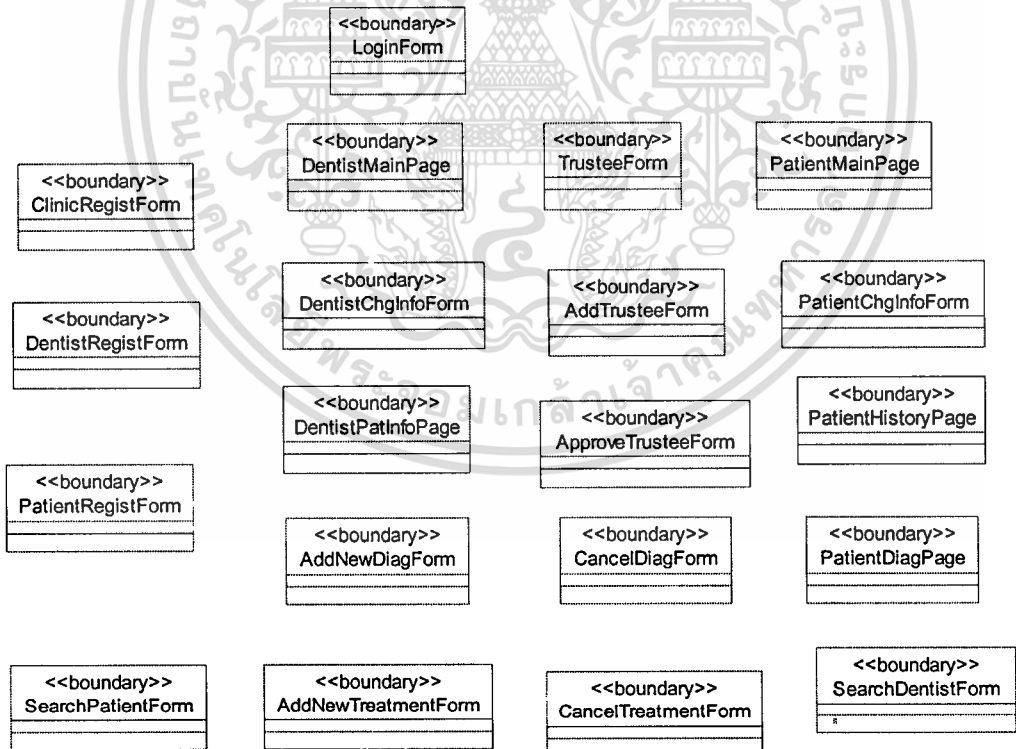


รูปที่ 6.6 Boundary Class Control Class และ Entity Class เมื่อแสดงแบบ label

6.6.1 Boundary Classes

โดยทั่วไปแล้ว ในระบบงานหนึ่งๆอาจมี Boundary class ได้หลายประเภทประกอบด้วย

- User interface classes คือ Class ซึ่งทำหน้าที่ติดต่อกับผู้ใช้ระบบงาน
- System interface classes คือ Class ที่ทำหน้าที่ติดต่อกับระบบงานภายนอก
- Device interface classes คือ Class ที่ทำหน้าที่ในการเชื่อมโยงอุปกรณ์ภายในและภายนอก

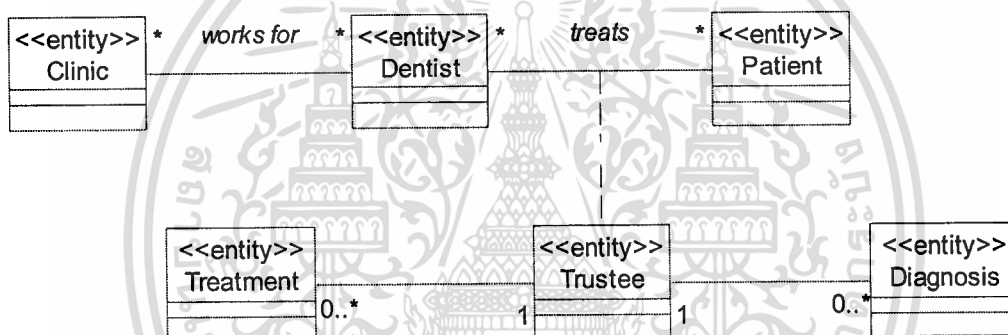


รูปที่ 6.7 เป็น Boundary classes ของระบบงาน

ในการเริ่มต้นกำหนดหรือหา Boundary Class นั้นเราสามารถที่จะกำหนดให้มี 1 Boundary Class ต่อหนึ่งคู่ของ Actor และ Use-Case ซึ่งในรูป 5.7 เป็น Boundary Classes ของระบบงาน

6.6.2 Entity Classes

Entity Class เป็นการนำเสนอโครงสร้างของข้อมูลในลักษณะ logical ซึ่งเป็นตัวแทนของข้อมูลที่จะถูกจัดเก็บในระบบ ซึ่งในการกำหนดหรือหา Entity Class มีวิธีการหาได้หลายวิธี แต่ในระบบงานนี้จะนำมาจาก Key Abstractions ที่ได้หาเอาไว้แล้วในขั้นตอนก่อนหน้านี้ (Preliminary Analysis Class) โดยในรูปที่ 6.8 แสดง Entity Class ของระบบงาน

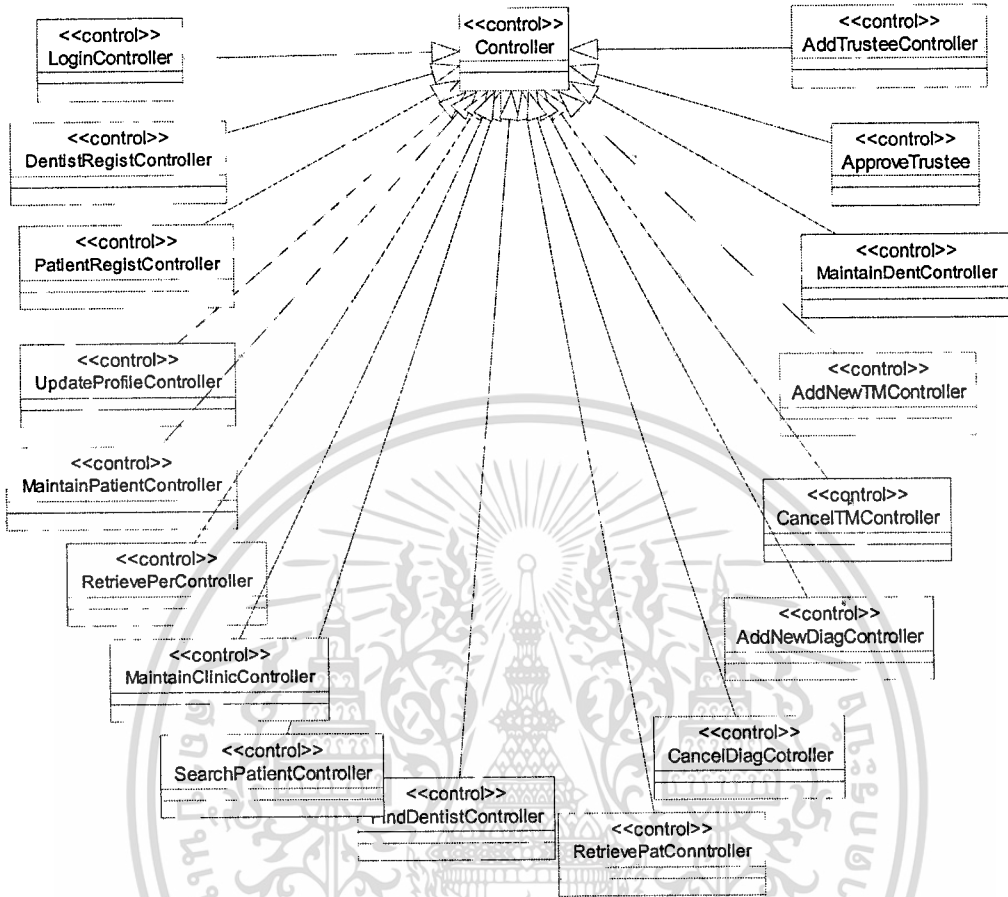


รูปที่ 6.8 เป็น Entity Class ของระบบงาน

6.6.3 Control Classes

Control Class เป็นส่วนที่จัดการประสานพฤติกรรมต่างๆที่เกิดขึ้นในระบบ โดย Control Class จะทำหน้าที่ในการแยก Boundary Objects ออกจาก หรือไม่ให้ขึ้นกับ Entity Objects ซึ่งโดยทั่วไปเรามักจะกำหนดให้มี 1 Control Class ต่อ 1 Use-Case

ใน Control Class หนึ่งๆจะถูกใช้ในการจำลองพฤติกรรมการควบคุมกระบวนการทำงาน ซึ่งเฉพาะเจาะจงไปกับ Use Case ใด Use Case หนึ่ง ซึ่งหมายความว่า Control Classes จะทำการห่อหุ้มรายละเอียดพฤติกรรมการทำงานของ Use Case เอาไว้

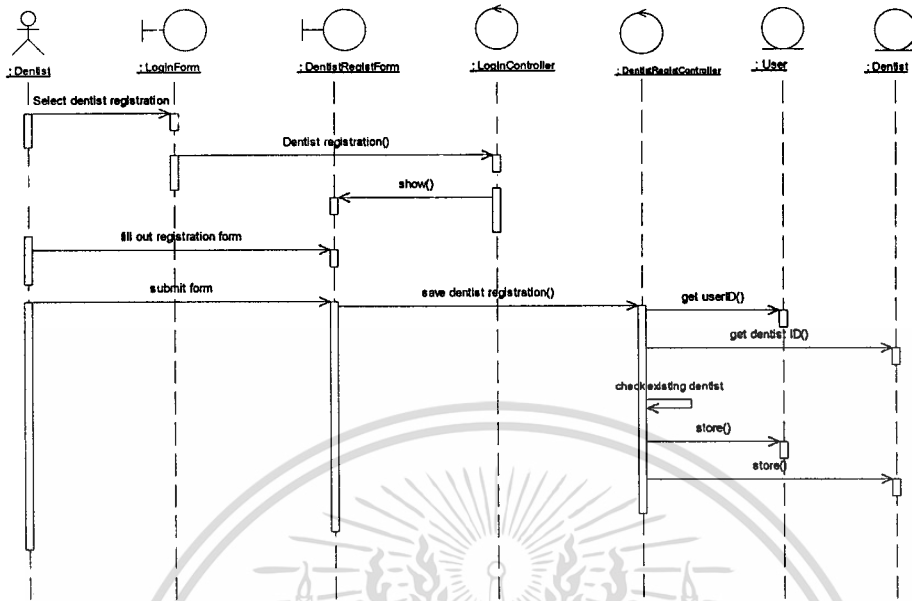


รูปที่ 6.9 Control Classes ของระบบงาน

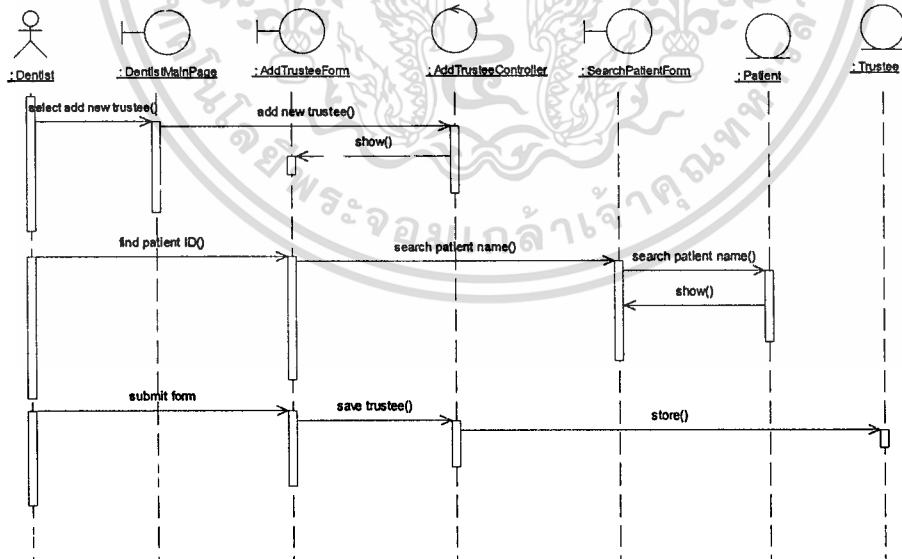
6.7 Analysis Sequence Diagram

เมื่อได้ Scenario (Use Case Description) และ Analysis Class แล้ว ขั้นตอนถัดไปเป็นการกำหนดหน้าที่ความรับผิดชอบหรือพฤติกรรมต่างๆให้กับ Classes ที่หามาได้ โดยจะพิจารณาจาก Scenario ของแต่ละ Use Case ว่าในแต่ละ Scenario ของแต่ละ Use Case นั้นมีขั้นตอนการทำงานอะไรบ้าง แต่ละขั้นตอนจะกำหนดให้ Class ใดมีหน้าที่รับผิดชอบในการทำงาน

เทคนิควิธีที่ได้นำมาใช้คือ การสร้างแบบจำลอง Sequence Diagram ขึ้นมา ซึ่งโดยทั่วไปเราจะทำการสร้าง Sequence Diagram ต่อหนึ่ง Use Case เป็นอย่างนั้น โดยในรูปที่ 6.10, 6.11, 6.12, 6.13 และ 6.14 แสดงตัวอย่าง Sequence Diagram บางส่วนของระบบงาน

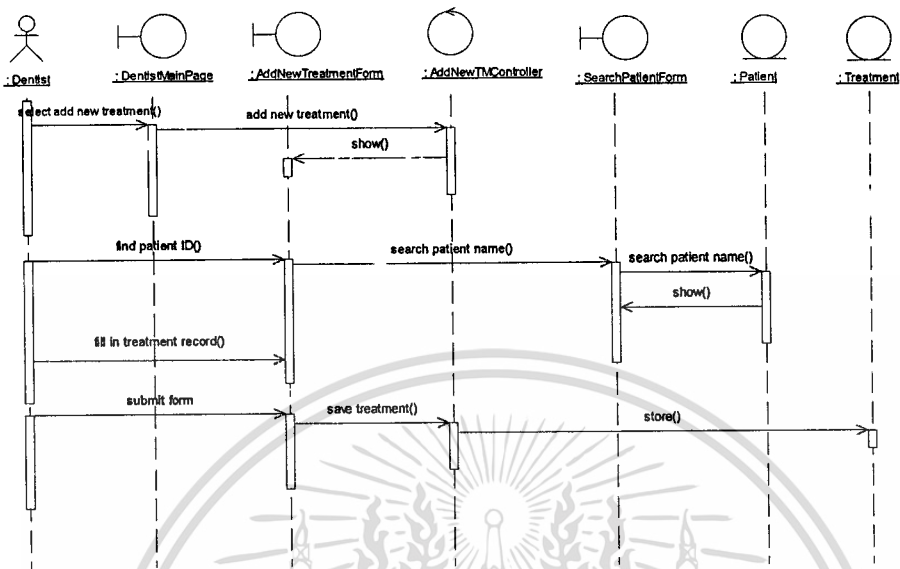


รูปที่ 6.10 Analysis Sequence Diagram ของ Use Case - Dentist Registration

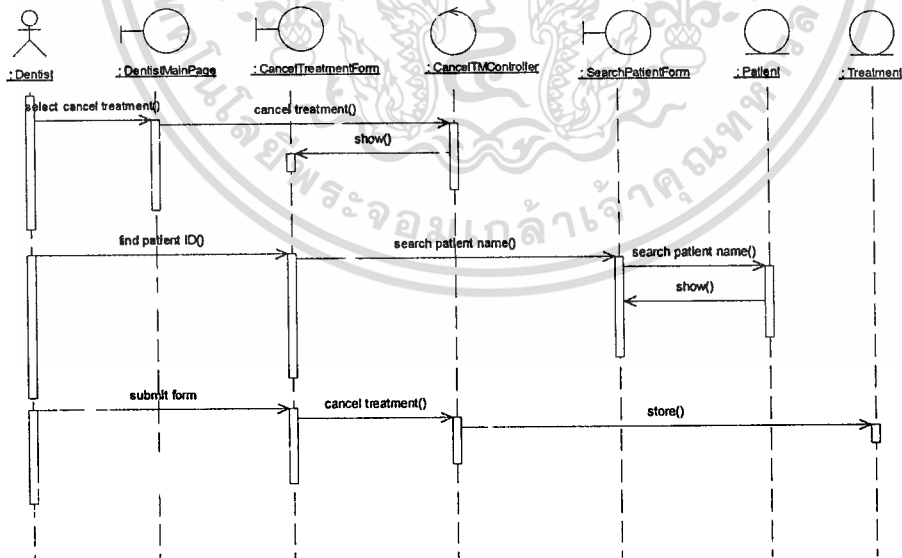


รูปที่ 6.11 Analysis Sequence Diagram ของ Use Case - Add Trustee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

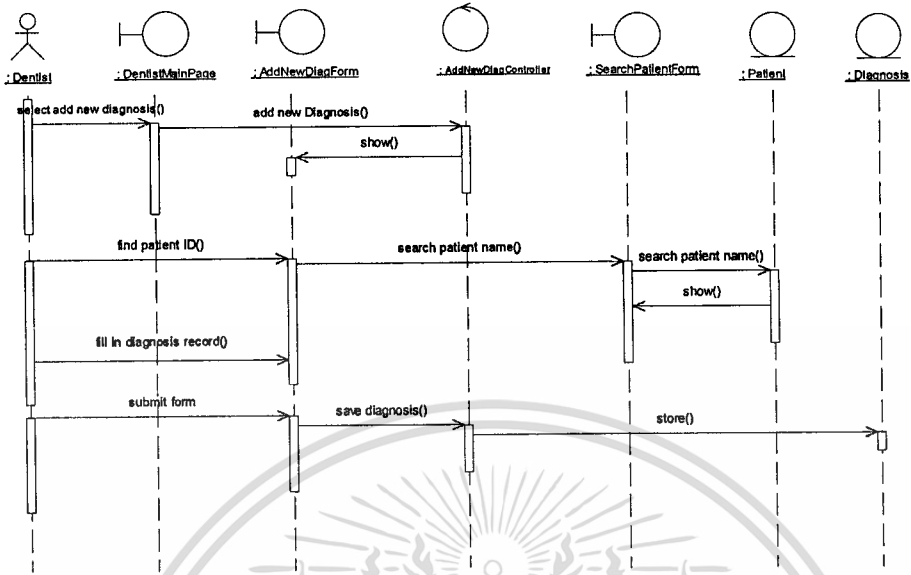


รูปที่ 6.12 Analysis Sequence Diagram ของ Use Case - Add New Treatment

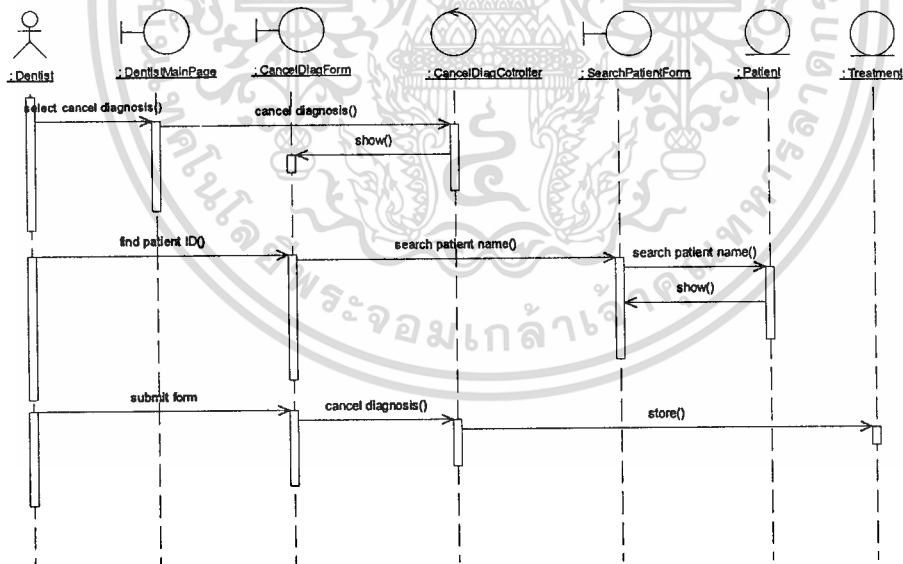


ที่ 6.13 Analysis Sequence Diagram ของ Use Case - Cancel Treatment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ที่ 6.14 Analysis Sequence Diagram ของ Use Case – Add New Diagnosis



ที่ 6.15 Analysis Sequence Diagram ของ Use Case – Cancel Diagnosis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การออกแบบและพัฒนาระบบ

7.1 Architectural Design

จากขั้นตอนของ Architecture Analysis นั้น ได้มีการกำหนดสถาปัตยกรรมเบื้องต้นเอาไว้ เมื่อถึงขั้นตอนนี้เป็นการกำหนดรายละเอียดเพิ่มเติม โดยเพิ่มกลไกในการทำงานต่างๆ โดยคำนึงถึงเทคโนโลยีที่จะนำมาใช้ในขั้นตอนของการจัดสร้าง การปรับปรุงระบบงานในอนาคต รวมถึงการขึ้นส่วนกลับมาใช้งานซ้ำในโอกาสต่อไป

กลไกการทำงานที่สำคัญอย่างหนึ่งของระบบ ที่จะต้องทำการออกแบบในรายละเอียด คือ กลไกในการจัดเก็บข้อมูลหรือที่เรียกว่า Persistent mechanism นั้น จะแตกต่างกันออกไปขึ้นกับเทคโนโลยีของระบบการจัดเก็บข้อมูลหรือระบบจัดการฐานข้อมูลว่าเป็นแบบใด เช่น เป็น RDBMS, ODBMS หรือ ORDBMS และเทคโนโลยีทางด้านภาษาของโปรแกรม ซึ่งในระบบงานนี้ได้เลือกใช้ RDBMS ของ MySQL และภาษาโปรแกรม Java และ JSP

ในการออกแบบกลไกการจัดเก็บข้อมูล จากโครงสร้างการทำงานในแบบ Object-Oriented มาเป็นระบบจัดการฐานข้อมูลในแบบ RDBMS นั้น จะมีการกำหนด Class ขึ้นมาอย่างน้อยหนึ่งคลาสเพื่อทำหน้าที่แปลง (Mapping) โครงสร้างข้อมูลที่เป็น Object Oriented ให้เป็นโครงสร้างข้อมูลที่เป็น Relational โดย Class ดังกล่าวทำหน้าที่ในการแตก Object ออกเป็นส่วนย่อยๆ แล้วทำการเขียนลงไปเป็น Table ต่างๆที่เกี่ยวข้องใน RDBMS ในทางกลับกัน การอ่าน Object จาก RDBMS นั้นก็ให้ Class ทำหน้าที่สร้าง Object ขึ้นมาจากข้อมูลใน table จาก RDBMS โดยภาษาที่ใช้ในการติดต่อเป็นภาษา SQL ซึ่งอาจมีความแตกต่างกันในแต่ละยี่ห้อของ RDBMS และภาษา Java นั้นสามารถเชื่อมต่อกับ RDBMS ผ่านทาง JDBC ได้โดยตรง แต่ใน MySQL ใน version ที่ใช้อยู่ไม่ได้สนับสนุนกลไกนี้ จึงต้องมีการเชื่อมต่อผ่านทาง JDBC-ODBC แทน

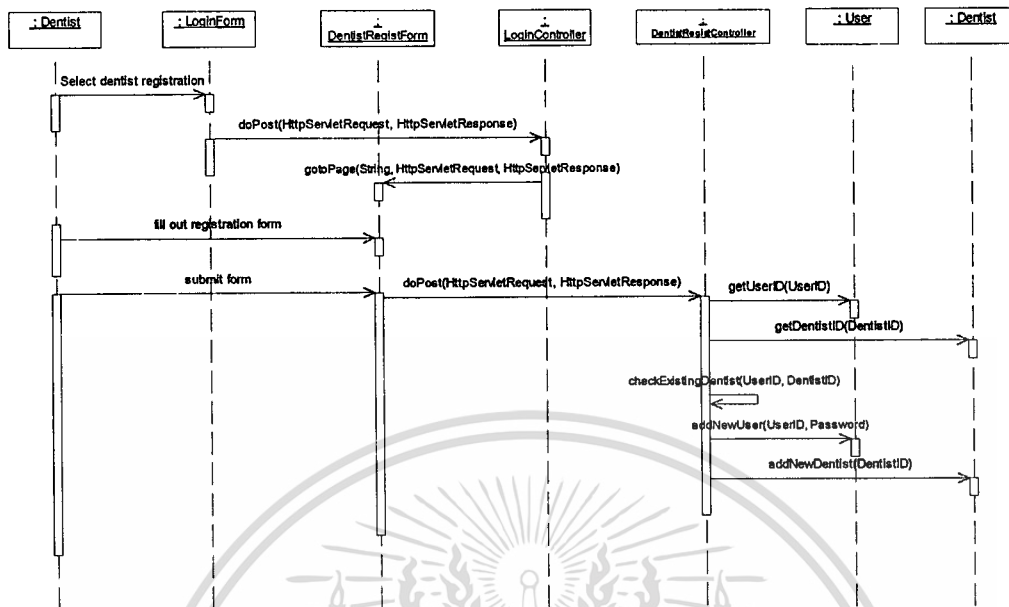
7.2 Design Sequence Diagram

เมื่อผ่านขั้นตอนการวิเคราะห์ระบบ เรานำแบบจำลอง Sequence diagram ที่ได้มาเพิ่มเติมรายละเอียดจากระดับการวิเคราะห์ (analysis) ไปสู่ระดับของการออกแบบ (design) โดยปรับปรุงรายละเอียดในการออกแบบเข้าไป เช่น เพิ่มเติมกลไกในการจัดเก็บข้อมูล กำหนดชื่อต่างๆให้ชัดเจนให้แก่ละ message ที่ object ใช้ในการติดต่อสื่อสาร ซึ่งจะกลายเป็นชื่อ method ของ object

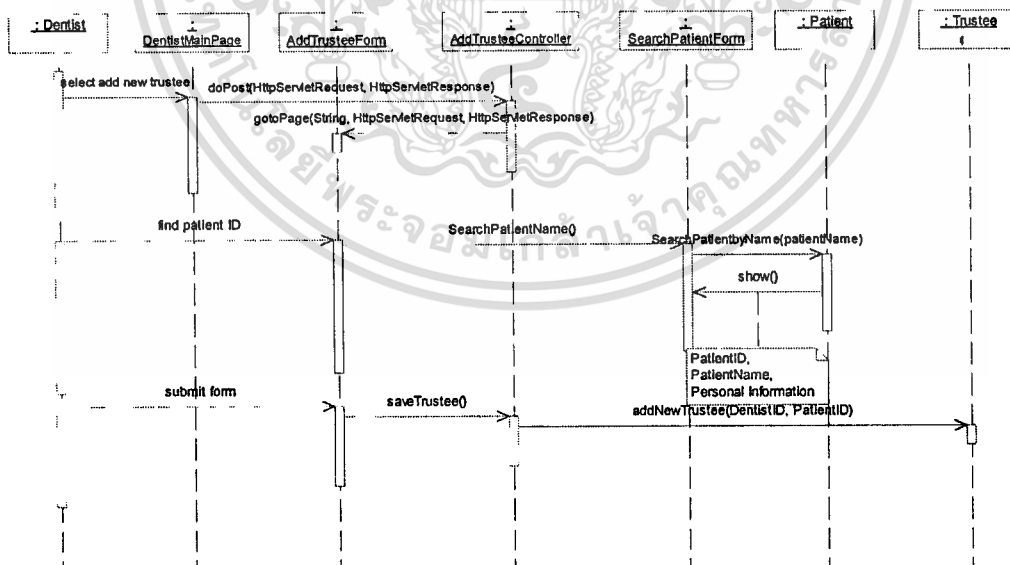
ของ class นั้นเอง ดังในรูปที่ 7.1 ถึง 7.6

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

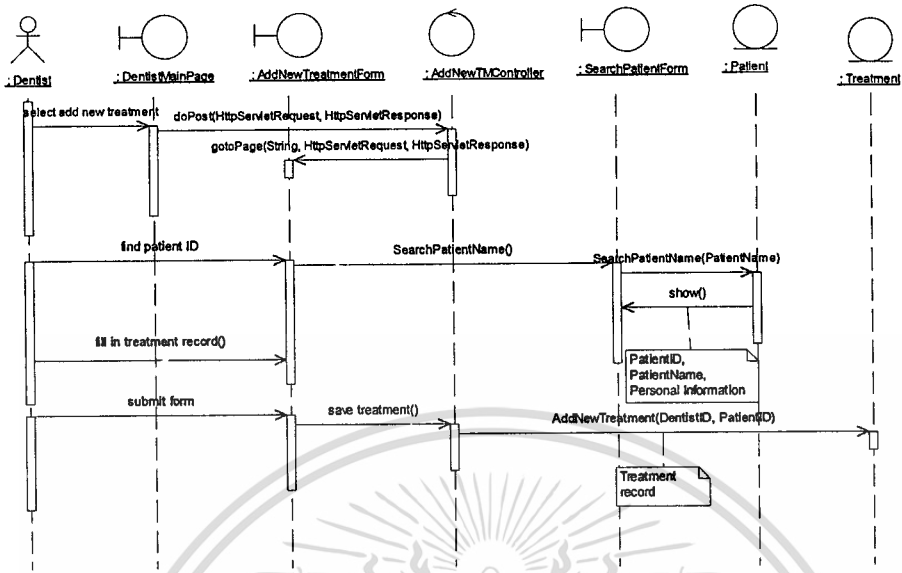


รูปที่ 7.1 Design Sequence Diagram ของ Use Case – Dentist Registration

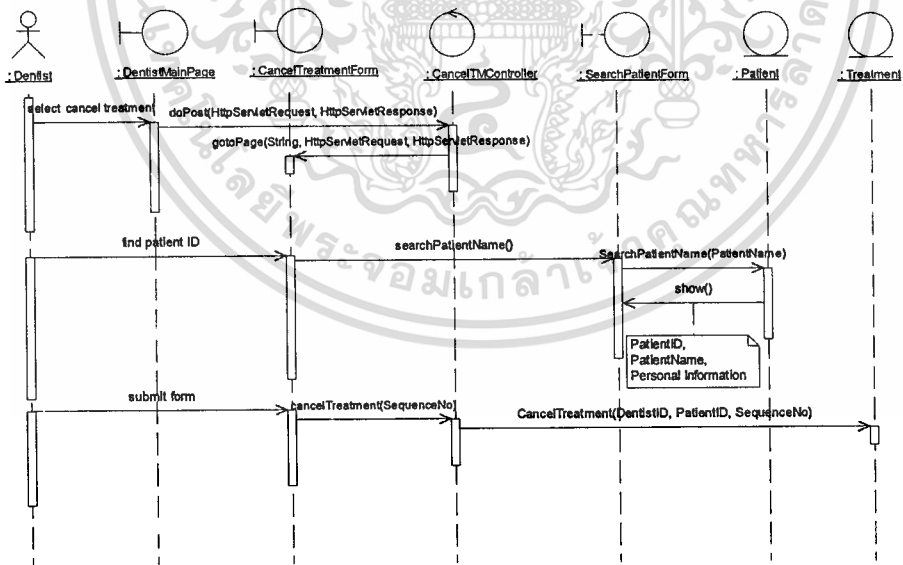


รูปที่ 7.2 Design Sequence Diagram ของ Use Case – Add Trustee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

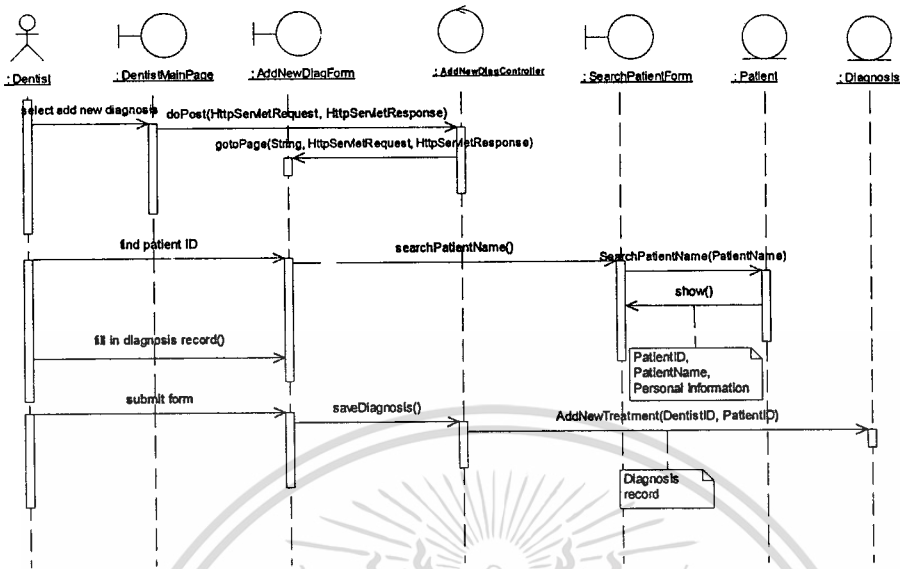


รูปที่ 7.3 Design Sequence Diagram ของ Use Case – Add New Treatment

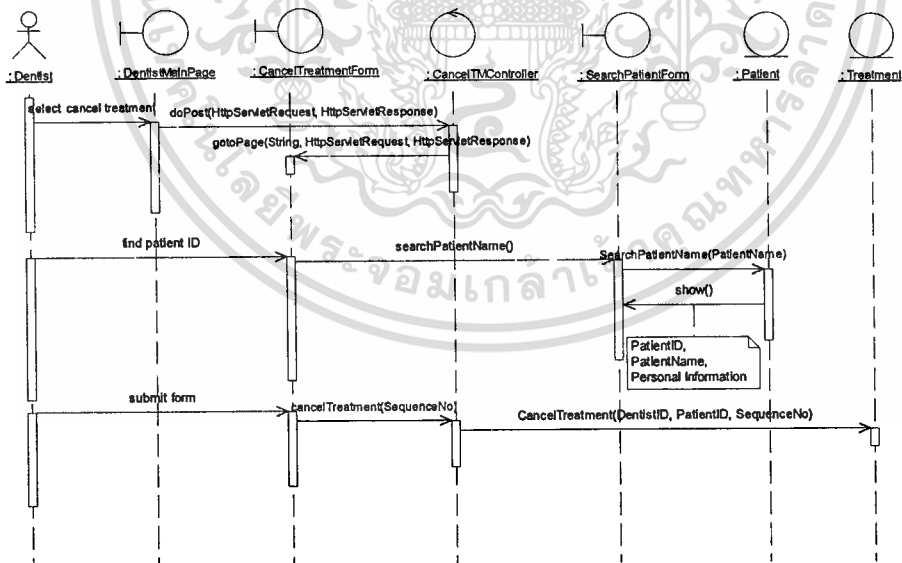


รูปที่ 7.4 Design Sequence Diagram ของ Use Case – Cancel Treatment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.5 Design Sequence Diagram ของ Use Case – Add New Diagnosis



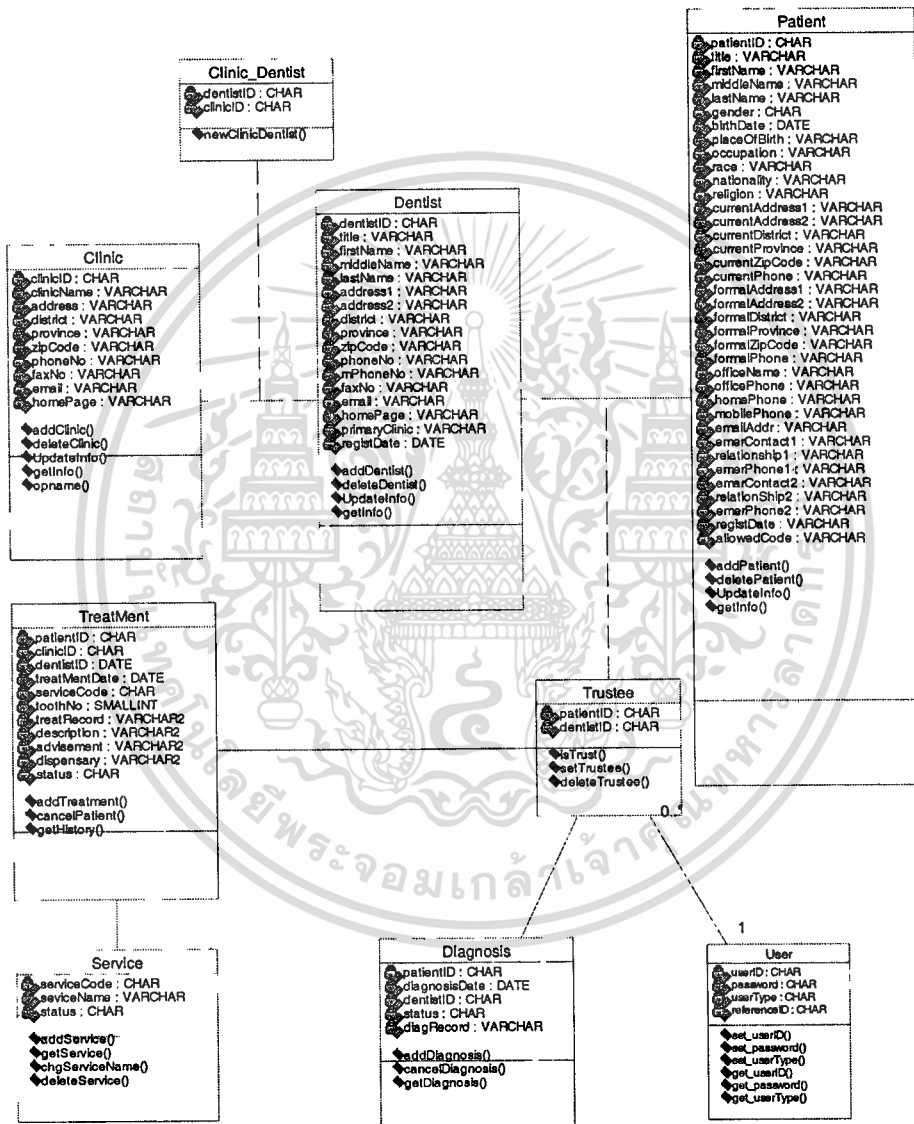
รูปที่ 7.6 Design Sequence Diagram ของ Use Case – Cancel Diagnosis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 Design Classes

เมื่อทำการสร้างแบบจำลอง Sequence Diagram ในระดับออกแบบแล้ว นำ Entity Class ที่ได้จากขั้นตอน Analysis มาทำการปรับปรุงและเพิ่มเติมรายละเอียดของ Class ซึ่ง ได้ผลดังในรูปที่

7.7



รูปที่ 7.7 Class ที่ได้จากการออกแบบ

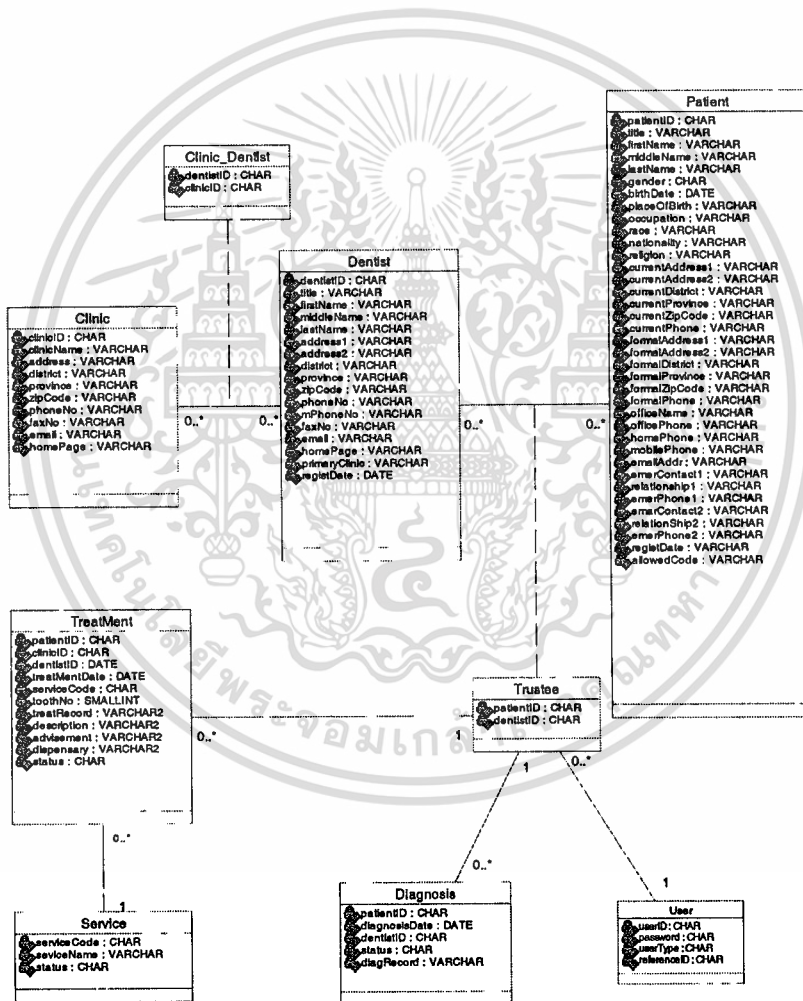
7.4 ตารางฐานข้อมูลของระบบงาน

ในระบบงานนี้มีการใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ จึงมีขั้นตอนในการออกแบบ

เพิ่มเติมขึ้นมาคือ การออกแบบโครงสร้างข้อมูลหรือ Database Schema ซึ่งสามารถทำการ map จาก เอกสาร เป็น เอกสาร ที่ส่งมอบไว้ส่วกับ การเขียนเพื่อ การศึกษา เท่านั้น เมื่อดูญาติเห็นไปใช้ประโยชน์ด้านการค้า ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class ที่อยู่ใน Entity Classes โดยมีหลักการเบื้องต้นคือ 1 Class ใน entity class จะได้เป็น 1 ตาราง ใน database schema และ attribute ของ class จะได้เป็น attribute ของ table และโดยที่ data type ระหว่าง class กับใน RDBMS อาจไม่ตรงกัน จึงจำเป็นต้องมีการแปลง data type ของ attribute ใน class ให้เหมาะสมกับ data type ที่ใช้อยู่ใน RDBMS ชนิดที่ใช้อยู่ ซึ่งในระบบงานนี้ได้มีการใช้ RDBMS ของ MySQL รวมไปถึงจำเป็นต้องกำหนดขนาดความยาวของข้อมูลที่ใช้เก็บ เช่น data type ที่เป็น char หรือ varchar เป็นต้น

ในรูปที่ 7.8 เป็น Class แสดงฐานข้อมูล โครงสร้างของระบบที่อยู่ในรูปของ Class Diagram



รูปที่ 7.8 Database Schema ของระบบ

จาก Database Schema ในลักษณะของ class diagram เมื่อนำมาแสดงในลักษณะของตาราง ประกอบไปด้วยตารางทั้งหมด 9 ตาราง ตั้งแต่ตาราง 7.1 จนถึง ตาราง 7.9 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่สามารถแก้ไขทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 7.1 Dentist

ที่	ชื่อฟิลด์	ประเภทข้อมูล	คีย์	ความหมาย
1	dentistID	Char(5)	PK	รหัสประจำตัวทันตแพทย์
2	title	Varchar(20)		คำนำหน้าชื่อ
3	firstName	Varchar(50)		ชื่อ
4	middleName	Varchar(30)		ชื่อย่อกลาง
5	lastName	Varchar(50)		นามสกุล
6	address1	Varchar(50)		ที่อยู่ #1
7	address2	Varchar(50)		ที่อยู่ #2
8	district	Varchar(30)		อำเภอ
9	province	Varchar(30)		จังหวัด
10	zipCode	Varchar(5)		รหัสไปรษณีย์
11	phoneNo	Varchar(20)		หมายเลขโทรศัพท์
12	mPhoneNo	Varchar(20)		หมายเลขโทรศัพท์พกพา
13	faxNo	Varchar(20)		หมายเลขแฟกซ์
14	email	Varchar(30)		อีเมลแอดเดรส
15	homepage	Varchar(35)		โฮมเพจ
16	primaryClinic	Char(8)		รหัสคลินิกหลัก
17	registDate	DateTime		วันที่ลงทะเบียน

ตาราง 7.2 Patient

ที่	ชื่อฟิลด์	ประเภทข้อมูล	คีย์	ความหมาย
1	patientID	Char(13)	PK	รหัสประจำตัวประชาชน
2	title	Varchar(20)		คำนำหน้าชื่อ
3	firstName	Varchar(30)		ชื่อ
4	middleName	Varchar(30)		ชื่อย่อกลาง
5	lastName	Varchar(30)		นามสกุล
6	gender	Char(1)		เพศ
7	birthDate	Date		วันเกิด
8	placeOfBirth	Varchar(20)		สถานที่เกิด
9	occupation	Varchar(20)		อาชีพ
10	race	Varchar(20)		เชื้อชาติ

เอกสารที่สงวนไว้สำหรับการศึกษาค้นคว้าวิจัย ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 7.2 Patient (ต่อ)

ที่	ชื่อฟิลด์	ประเภทข้อมูล	คีย์	ความหมาย
11	nationality	Varchar(20)		สัญชาติ
12	religion	Varchar(20)		ศาสนา
13	currentAddress1	Varchar(50)		ที่อยู่ปัจจุบัน #1
14	currentAddress2	Varchar(50)		ที่อยู่ปัจจุบัน #2
15	currentDistrict	Varchar(30)		อำเภอ
16	currentProvince	Varchar(30)		จังหวัด
17	currentZipCode	Varchar(5)		รหัสไปรษณีย์
18	currentPhone	Varchar(20)		หมายเลขโทรศัพท์
19	formalAddress1	Varchar(50)		ที่อยู่ตามทะเบียนบ้าน #1
20	formalAddress2	Varchar(50)		ที่อยู่ตามทะเบียนบ้าน #2
21	formalDistrict	Varchar(30)		อำเภอ
22	formalProvince	Varchar(30)		จังหวัด
23	formalZipCode	Varchar(5)		รหัสไปรษณีย์
24	formalPhone	Varchar(20)		หมายเลขโทรศัพท์
25	officeName	Varchar(50)		สถานที่ทำงาน
26	officePhone	Varchar(20)		หมายเลขโทรศัพท์ที่ทำงาน
27	homePhone	Varchar(20)		หมายเลขโทรศัพท์บ้าน
28	mobilePhone	Varchar(20)		หมายเลขโทรศัพท์พกพา
29	emailAddr	Varchar(30)		E-mail Address
30	emerContact2	Varchar(50)		ชื่อผู้ที่ติดต่อได้กรณีฉุกเฉิน #2
31	relationship1	Varchar(20)		ความสัมพันธ์กับคนไข้ #1
32	emerPhone1	Varchar(20)		หมายเลขโทรศัพท์ติดต่อ #1
33	emerContact2	Varchar(50)		ชื่อผู้ที่ติดต่อได้กรณีฉุกเฉิน #2
34	relationship2	Varchar(20)		ความสัมพันธ์กับคนไข้ #2
35	emerPhone2	Varchar(20)		หมายเลขโทรศัพท์ติดต่อ #2
36	registDate	DateTime		วันที่ลงทะเบียน
37	allowedCode	Varchar(16)		รหัสใบอนุญาตให้ทันตแพทย์เข้าถึงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 7.3 Clinic

ที่	ชื่อฟิลด์	ประเภทข้อมูล	คีย์	ความหมาย
1	clinicID	Char(8)	PK	รหัสประจำคลินิก
2	clinicName	Varchar(50)		คำนำหน้าชื่อ
3	address1	Varchar(50)		ที่อยู่ #1
4	address2	Varchar(50)		ที่อยู่ #2
5	district	Varchar(30)		อำเภอ
6	province	Varchar(30)		จังหวัด
7	zipcode	Char(1)		รหัสไปรษณีย์
8	phoneNo	Varchar(20)		หมายเลขโทรศัพท์
9	faxNo	Varchar(20)		หมายเลขโทรศัพท์แฟกซ์
10	email	Varchar(30)		อีเมลแอดเดรส
11	homePage	Varchar(40)		โฮมเพจ
12	registDate	DateTime		วันที่ลงทะเบียน

ตาราง 7.4 Treatment

ที่	Attribute	ประเภทข้อมูล	คีย์	ความหมาย
1	patientID	Char(13)	PK, FK	รหัสประจำตัวประชาชน
2	seqID	Int	PK, FK	ลำดับที่
3	clinicID	Char(8)	PK, FK	รหัสประจำคลินิก
4	dentistID	Char(5)	PK, FK	รหัสประจำตัวทันตแพทย์
5	treatmentDate	DateTime		วันที่ทำการรักษา
6	serviceCode	Char(4)	FK	ประเภทการรักษา
7	toothNo	Char(2)		หมายเลขซี่ฟัน
8	treatRecord	Varchar(200)		บันทึกการรักษา
9	description	Varchar(200)		รายละเอียดการรักษาสำหรับ คนไข้
10	advisement	Varchar(200)		ข้อแนะนำในการปฏิบัติตัว
11	drugDispense	Varchar(200)		การจ่ายยา
12	status	Char(1)		สถานะของรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 7.5 User

ที่	Attribute	ประเภทข้อมูล	คีย์	ความหมาย
1	userID	Char(16)	PK	รหัสผู้ใช้
2	password	Char(16)		รหัสผ่าน
3	userType	Char(1)		ประเภทผู้ใช้
4	referenceID	char(13)		รหัสคลินิก, ทันตแพทย์, คนไข้
5	lastLoginDate	DateTime		วันที่เข้ามาใช้ระบบล่าสุด

ตาราง 7.6 Trustee

ที่	Attribute	ประเภทข้อมูล	คีย์	ความหมาย
1	dentistID	Char(13)	PK, FK	รหัสประจำตัวทันตแพทย์
2	patientID	Char(5)	PK, FK	รหัสประจำตัวประชาชน
3	allowedDate	Date		วันที่กำหนดทำการอนุญาต

ตาราง 7.7 Diagnosis

ที่	Attribute	ประเภทข้อมูล	คีย์	ความหมาย
1	patientID	Char(13)	PK, FK	รหัสประจำตัวประชาชน
2	seqNo	Integer	PK	ลำดับครั้งที่ทำการวินิจฉัย
3	dentistID	Char(5)	FK	รหัสประจำตัวทันตแพทย์
4	clinicID	Char(8)	FK	รหัสประจำคลินิก
5	diagDate	DateTime		วันที่ทำการวินิจฉัย
6	diagRecord	Varchar(255)		บันทึกการวินิจฉัย
7	direction	Varchar(255)		คำแนะนำในการรักษา
8	status	Char(1)		สถานะของรายการ

ตาราง 7.8 Service

ที่	Attribute	ประเภทข้อมูล	คีย์	ความหมาย
1	serviceCode	Char(4)	PK	รหัสการให้บริการรักษา
2	serviceName	Varchar(40)		ชื่อบริการการรักษา
3	status	Char(1)		สถานะของรายการ

ตาราง 7.9 Clinic_Dentist

ที่	Attribute	ประเภทข้อมูล	คีย์	ความหมาย
1	dentistID	Char(5)	PK, FK	รหัสการให้บริการรักษา
2	clinicID	Varchar(8)	PK, FK	ชื่อบริการการรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่สามารถนำข้อมูลไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 Source Code

7.5.1 Java Servlet

Control Class ต่างๆถูกนำไปสร้างเป็น Java class ในลักษณะของ Java Servlet ซึ่งมีลักษณะดังรูปที่ 7.3 ซึ่งแสดงตัวอย่างของ Source code ของ Class controller โดยสังเกตได้ว่าลักษณะเฉพาะของ Java Servlet ก็คือมีการ import package javax.servlet.* และ package javax.servlet.http.* และมีการ extent หรือ inherit มาจาก class ที่ชื่อว่า HttpServlet

```

package com.businessservices;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

/**
 * This is the main controller servlet for the Dental Servlet relationships use case.
 */
public class DentalServlet extends HttpServlet {
    private String url;

    public void init() throws ServletException {
        url = "http://localhost:8080/dental";
    }

    public void destroy() {
    }

    /**
     * Processes a GET request by calling doPost().
     */
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws IOException, ServletException {
        doPost(request, response);
    }

    /**
     * Performs the action specified by the "action" request parameter.
     */
    public void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws IOException, ServletException {

        String queryString = request.getQueryString();
        String action = request.getParameter("SignIn");

        // Check which action to do and forward to it
        if ("Patient Login".equals(action)) {
            doPatientMain(request, response);
        }
        else if ("Dentist Login".equals(action)) {
            doDentistMain(request, response);
        }
        response.sendError(HttpServletResponse.SC_NOT_IMPLEMENTED);
    }
}

```

รูปที่ 7.9 ตัวอย่าง Source code ของ Class Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/**
 * This operation represents the relationship inquiry pathway through
 * the Maintain Relationships Use Case
 */
private void doPatientMain(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException {

    forward("patient_main.jsp", request, response);
}

/**
 * This operation represents the setup to add a customer pathway through
 * the Maintain Relationships Use Case
 */
private void doDentistMain(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException {
    logit("DentalServlet: doDentistMain");
    forward("dentist_main.jsp", request, response);
}

/**
 * Forwards the request to the specified relative URL.
 */
private void forward(String url, HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException {
    RequestDispatcher rd = request.getRequestDispatcher(url);
    rd.forward(request, response);
}

private static void logit(String s) {
    System.out.println(s);
}
}

```

รูปที่ 7.9 ตัวอย่าง Source code ของ Class Controller (ต่อ)

7.5.2 Java Server Page (JSP)

ในส่วนของ boundary classes ต่างๆถูกนำไปใช้ในการสร้าง web page ของระบบ ซึ่งโดยทั่วไปแล้ว หนึ่ง boundary class จะถูก map ไปเป็นไฟล์ html หรือ JSP หนึ่งไฟล์ ดังในรูปที่ 7-4 แสดงตัวอย่างของ source code ในรูปแบบของ Java Server Page ในไฟล์ชื่อ PatientMainPage.jsp

```

<%@ page language="java" import="java.sql.*" %>
<%@ page contentType="text/html; charset=Windows-874" %>

<html>
<head>
    <title> Login page for the calendar. </title>
    <meta http-equiv="Content-Type" content="text/html; charset=Windows-874">
</head>

<body link="#66FFFF">
<p align="left"><font face="Batang"
size="5" color="blue"><b><span lang="th">
</span></b></font><font face="Batang" color="blue" size="6">Welcome to Dental
Clinic Information System</font></p>
    <form method=POST action=login.html>

```

รูปที่ 7.10 ตัวอย่าง Source code ของ PatientMainPage.jsp

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้ในเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

<font color="#003399"><b>
<font size="2">บันทึกชื่อผู้ป่วย <%= firstName %> <%= lastName %></span>
</font></b></font></td>
</tr>
<tr>
<td width="2%" height="14" style="border-style: none; border-width: medium"
bgcolor="#003399">&nbsp;</td>
<td width="19%" height="14" style="border-style: none; border-width: medium"
bgcolor="#003399">
<p align="center">&nbsp;</p>
<td width="2%" height="14" style="border-style: none; border-width: medium"
bgcolor="#003399">&nbsp;</td>
<td width="2%" height="14" style="border-style: none; border-width: medium"
bgcolor="#C0C0C0">&nbsp;</td>
<td width="131%" height="14" style="border-style: none; border-width: medium"
bgcolor="#009933">
<table border="1" cellpadding="0" cellspacing="0" style="border-collapse: collapse; border-width:
0" bordercolor="#111111" width="674" id="AutoNumber6" height="20">
<tr>
<td width="51" bgcolor="#C0C0C0" height="17" style="border-style: none; border-width:
medium">
<p align="left"><font size="2"><b>&nbsp;</b></span></span></font></b></font></td>
medium">
<td width="153" bgcolor="#C0C0C0" height="17" style="border-style: none; border-width:
medium">
<font size="2"><b>&nbsp;</b></span></span></font></b></font></td>
medium">
<td width="117" bgcolor="#003399" height="17" style="border-style: none; border-width:
medium">
<p align="center"><span lang="th"><u>
<font size="2" color="#FFFF00"><b>
<a href="patient_chginfo.jsp?patientID=<%= patientID %>">
<font color="#FFFF00">เปลี่ยนแปลงข้อมูล</font></a></b></font></u></span></td>
medium">
<td width="132" bgcolor="#003399" height="17" style="border-style: none; border-width:
medium">
&nbsp;</td>
medium">
<td width="119" bgcolor="#003399" height="17" style="border-style: none; border-width:
medium">
<p align="center">&nbsp;</p>
medium">
<td width="100" bgcolor="#003399" height="17" style="border-style: none; border-width:
medium">
&nbsp;</td>
</tr>
</table>
</td>
</tr>
<tr>
<td width="2%" height="207" style="border-style: none; border-width: medium"
bgcolor="#003399">&nbsp;</td>
<td width="19%" height="207" style="border-style: none; border-width: medium"
bgcolor="#003399" valign="top">
<p align="left">
</p>
<p align="left"><span lang="th"><font size="2" color="#FFFF00">
<b>แผนุการทำงาน</b></span></p>
<p align="left"><span lang="th"><u>
<font size="2" color="#66FFFF"><a href="patient_main.jsp">
<font color="#0000FF">ข้อมูลส่วนบุคคล</font></a></font></u></span><font size="2"><u><font
color="#66FFFF"><BR>
<a href="patient_history.jsp?patientID=<%= patientID %>&patientName=<%= firstName
%>+<%= lastName %>"><font color="#66FFFF">ประวัติการรักษา</font></a><BR>
<span lang="th"><a href="patient_diagnosis.jsp?patientID=<%= patientID
%>&patientName=<%= firstName %>+<%= lastName %>">
<font color="#66FFFF">ประวัติการวินิจฉัย</font></a></span><BR><span lang="th">
ข้อมูลทันตแพทย์</span><BR>
<span lang="th">คลินิกในโครงการ</span></font></u></font><u><font color="#66FFFF"
size="2"><BR>

```

รูปที่ 7.10 ตัวอย่าง Source code ของ PatientMainPage.jsp (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

</center>
<p align="left"><b><font size="2" color="#008000">Last Updated:</font><font size="2" color="#FF0000">
</font>
<font size="2" color="#000080">
<!--webbot bot="Timestamp" S-Type="EDITED" S-Format="%d/%m/%Y" startspan -->25/09/2545<!--
webbot bot="Timestamp" endspan i-checksum="13286" --></font><font size="2" color="#FF0000">&nbsp;
</font><font size="2" color="#008000">By:</font></b><font size="2" color="#FF0000"> </font>
<font size="2" color="#000080">Anuwat Kanokwiroon</font></p>
<p align="left"><span lang="en-us"><font size="2"><font color="#008000"><b>
Contact:</b></font> <a href="mailto:webmaster@dentalinfo.co.th">
webmaster@dentalinfo.co.th</a> or <a href="mailto:webmaster@dentalinfo.com">
webmaster@dentalinfo.com</a></font></span></p>
</body>
</html>

```

รูปที่ 7.10 ตัวอย่าง Source code ของ PatientMainPage.jsp (ต่อ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือกหัวข้อแล้วก็ระบบจะแสดงหน้าจอสำหรับกรอกข้อมูลมาให้ ซึ่งมีรายละเอียดดังตัวอย่างในรูปที่ 8.2 และ 8.3 สำหรับผู้ใช้ที่ลงทะเบียนเป็นคนไข้ และทันตแพทย์ตามลำดับ ในส่วนของคนไข้ ระบบงานนี้ได้กำหนดให้มีการใช้รหัสประจำตัวคนไข้เป็นหมายเลขประจำตัวประชาชน 13 หลัก

Welcome to Dental Clinic Information System

คนไข้	การลงทะเบียนผู้ใช้งาน
รหัสผู้ใช้	anuwat
รหัสผ่าน
ยืนยันรหัสผ่าน

รหัสประจำตัวคนไข้	3953900069290
สาขาภาควิชา	ทันต
ชื่อ	อนุวัฒน์
นามสกุล	กานวิรุฬห์
เพศ	ชาย
เกิดเมื่อ	29/01/2510
สถานที่เกิด	ยะลา
เชื้อชาติ	ไทย
สัญชาติ	ไทย
ศาสนา	พุทธ
ที่อยู่ปัจจุบัน	111/132 ม.2 ต.บ้านใหม่ อ.ทรายทอง อ.บ้านใหม่ จ.ลำพูน
อำเภอ	เมือง
จังหวัด	ลำพูน
รหัสไปรษณีย์	11000
ที่อยู่ตามเอกสาร	111/132 ม.2 ต.บ้านใหม่ อ.บ้านใหม่ จ.ลำพูน
อำเภอ	เมือง
จังหวัด	ลำพูน
รหัสไปรษณีย์	11000
เบอร์โทรศัพท์	0-2951-3417
เบอร์ติดต่อแพทย์	0-1812-9436

รูปที่ 8.2 ตัวอย่างหน้าจอสำหรับกรอกข้อมูลลงทะเบียนของคนไข้ - คนไข้

ในรูป 8.3 เป็นหน้าจอที่พบเมื่อลงทะเบียนในสถานะของทันตแพทย์ มีการกำหนด การใช้รหัสประจำตัวทันตแพทย์ที่ออกให้โดยทันตแพทย์สภาเป็นรหัสประจำตัว ซึ่งทันตแพทย์แต่ละคนจะมีหมายเลขไม่ซ้ำกัน

Welcome to Dental Clinic Information System

ระบบการจัดเก็บข้อมูลการรักษา
พยาบาลทางคลินิกเป็นระบบผ่านระบบ
เครือข่ายอินเตอร์เน็ตเพื่อให้บริการใช้และ
ทันตแพทย์สามารถเข้าถึงข้อมูลการ
รักษาได้ตลอดเวลา สามารถไม่สุกเสีย
ประวัติการรักษาเมื่อมีการย้ายหรือ
เปลี่ยนสถานบริการทางทันตกรรม

ข้อมูลทั่วไป

[รายละเอียดแพทย์](#)
[รวมข้อมูลคลินิกในโครงการ](#)
[เกี่ยวกับเว็บไซต์](#)

เชื่อมโยง

[ทันตแพทย์สมัคร](#)
[ทันตแพทย์สมัคร](#)
[ตลับและทันตแพทย์สมัคร](#)

ทันตแพทย์	การลงทะเบียนผู้ใช้งาน
รหัสผู้ใช้	nuttapong
รหัสผ่าน	*****
ยืนยันรหัสผ่าน	*****

รหัสประจำตัวทันตแพทย์	4338
สำเนาบัตร	ทพ.
ชื่อ	ณัฐพงษ์
นามสกุล	กมลวิรุฬห์
ที่อยู่ปัจจุบัน	47/1 ม.2 ถนน สุขุมวิท 47
	แขวงคลองเตย
อำเภอ	คลองเตย
จังหวัด	กรุงเทพมหานคร
รหัสไปรษณีย์	10550
โทรศัพท์	0-9999-9999
โทรศัพท์มือถือ	1-6666-6666
Fax	1-6666-6667
email address	nuttapong@ksct.h.com
Home Page	
คลินิกสาขา	0001
	<input type="button" value="บันทึก"/>

Contact: webmaster@dentalinfo.co.th or webmaster@dentalinfo.com

English (United States)

รูปที่ 8.3 ตัวอย่างหน้าจอสำหรับกรอกข้อมูลลงทะเบียนของผู้ใช้ – ทันตแพทย์

เมื่อได้ลงทะเบียนแล้ว ก็จะสามารถเข้าใช้งานระบบงานได้ดังรายละเอียดในหัวข้อถัดไป โดยใช้รหัสผู้ใช้และรหัสผ่านที่ผู้ใช้ได้กำหนดขึ้น

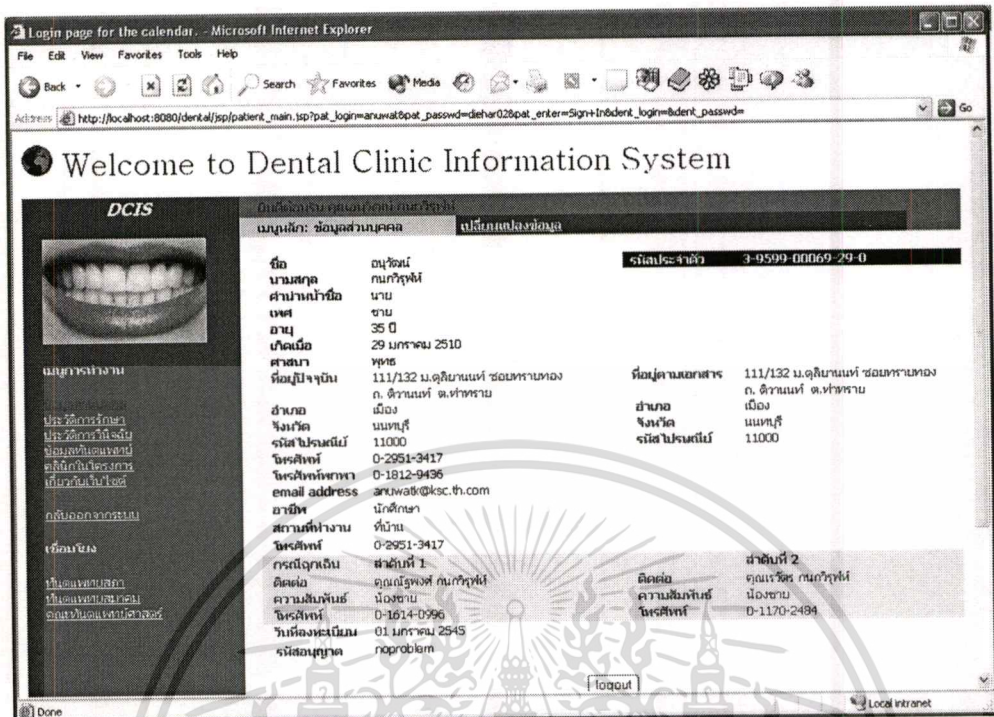
8.2 การใช้งานระบบ

เมื่อใส่รหัสผู้ใช้และรหัสผ่านในหน้าจอ login แล้วหากผู้ใช้ที่เป็นคนไข้ก็จะพบกับหน้าจอหลักของซึ่งแสดงรายละเอียดต่างๆของคนไข้ ดังในรูปที่ 8.4 สำหรับทันตแพทย์จะมีหน้าจอหลักเป็นดังในรูปที่ 8.9

8.2.1 ระบบเมนู

ระบบงานนี้มีการสั่งงานในลักษณะของเมนู คือเมนูหลักจะอยู่ทางด้านซ้ายมือของจอภาพ จากตัวอย่างในรูปที่ 8.4 เมนูหลักประกอบด้วย ข้อมูลส่วนบุคคล, ประวัติการรักษา, ประวัติการวินิจฉัย, ข้อมูลทันตแพทย์, คลินิกในโครงการ และ เกี่ยวกับเว็บไซต์ ส่วนเมนูย่อยจะอยู่ทางส่วนบนของพื้นที่แสดงผล จากในตัวอย่างรูปที่ 8.4 คือ เปลี่ยนแปลงข้อมูล โดยหัวข้อเมนูสามารถสังเกตได้จากตำแหน่งของข้อความและจะมีการขีดเส้นใต้เอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.4 หน้าจอหลักเมื่อคนใช้ Login เข้าสู่ระบบแล้ว

8.2.2 ฟังก์ชันการใช้ของผู้ใช้ที่เป็นคนใช้

ผู้ใช้ที่เป็นคนใช้สามารถดูรายละเอียดข้อมูลต่างๆ ซึ่งการใช้งานหลักจะมีอยู่สามส่วนคือ

- ข้อมูลส่วนบุคคล
- ประวัติการรักษา
- ประวัติการวินิจฉัย

8.2.2.1 ข้อมูลส่วนบุคคล

ข้อมูลส่วนบุคคลจะเป็นหน้าจอหลักหลังจากที่ได้ login เข้าสู่ระบบ ข้อมูลส่วนนี้คนใช้สามารถที่จะแก้ไขได้ โดยหากต้องการแก้ไขผู้ใช้สามารถเลือกเมนูย่อย **เปลี่ยนแปลงข้อมูล** เพื่อเข้าสู่หน้าจอการแก้ไขข้อมูลได้

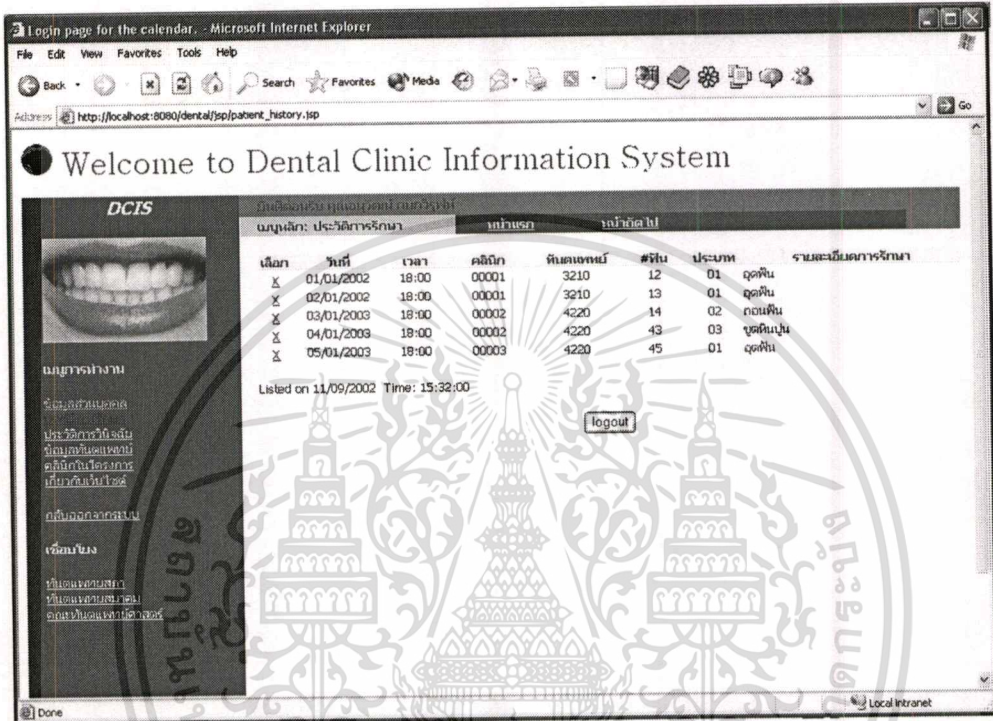
8.2.2.2 ประวัติการรักษา

เมื่อผู้ใช้ต้องการดูประวัติการรักษาก็สามารถเลือกเมนู **ประวัติการรักษา** จากเมนูหลัก หน้าจอของประวัติการรักษา จะมีรายการประวัติทั้งหมดที่เคยผ่านการรักษามาให้

โดยมีลักษณะดังในรูปที่ 8.5 ประวัติการรักษานี้ทันตแพทย์จะเป็นผู้บันทึกเท่านั้น ผู้ใช้ที่

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

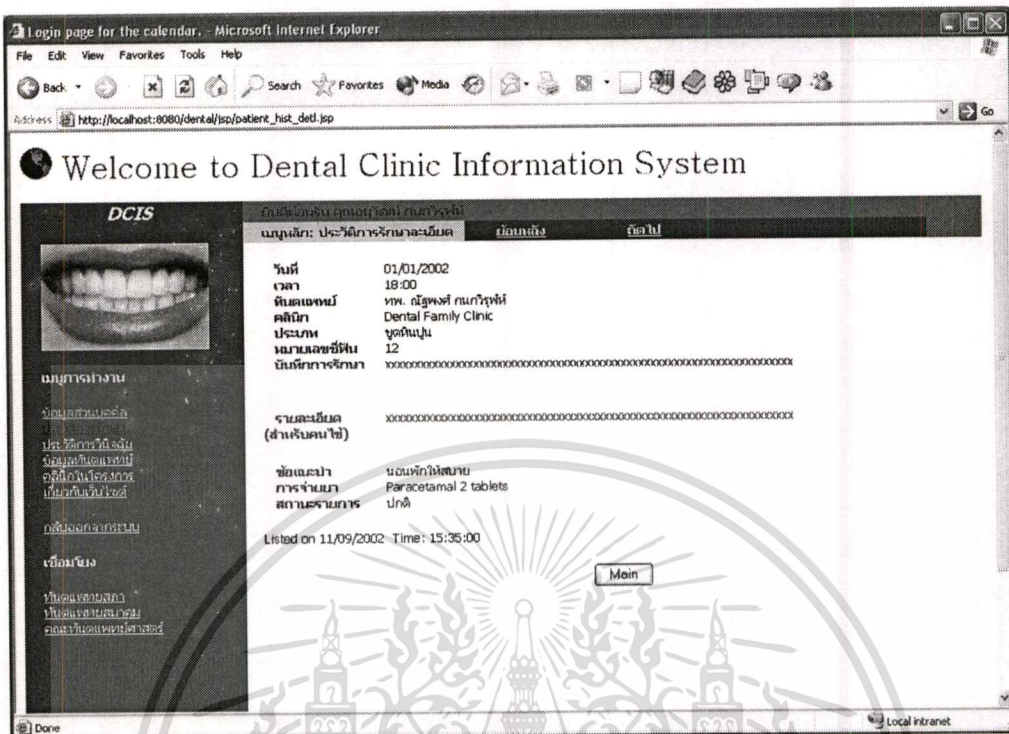
เป็นคนที่ไม่มีสิทธิ์ในการแก้ไขหรือเปลี่ยนแปลง และเมื่อผู้ใช้ต้องการทราบรายละเอียดเกี่ยวกับการรักษา สามารถเลือกเข้าไปดูรายละเอียดได้ด้วยการเลือกที่หมายเลขของรายการที่สนใจ จากนั้นก็จะพบหน้าจอดังในรูปที่ 8.6



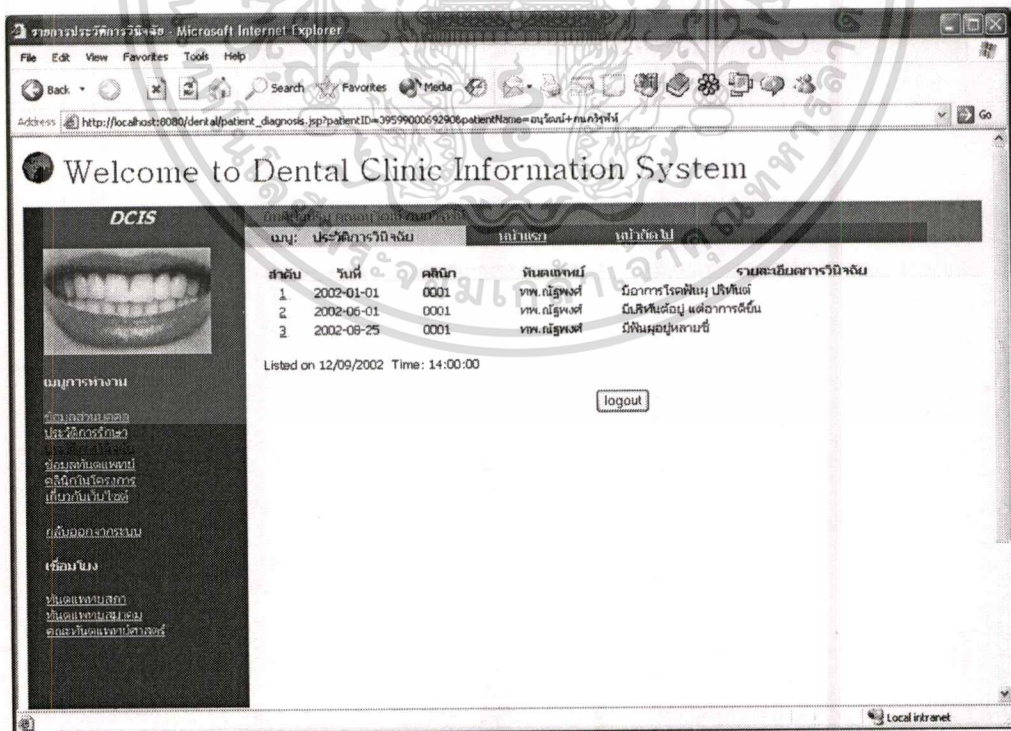
รูปที่ 8.5 หน้าจอแสดงประวัติการรักษาเมื่อเลือกเมนู ประวัติการรักษา

8.2.2.3 ประวัติการวินิจฉัย

ในส่วนของการวินิจฉัยนั้น ผู้ใช้สามารถเข้าไปดูตามที่ทันตแพทย์ได้เคยทำการวินิจฉัยและบันทึกเอาไว้ได้ และไม่สามารถแก้ไขเปลี่ยนแปลงได้เช่นเดียวกับประวัติการรักษา การใช้งานทำได้โดยการเลือกเมนูประวัติการวินิจฉัย จากนั้นระบบก็จะแสดงหน้าจอดังในรูปที่ 8.7 ซึ่งเป็นการแสดงรายการทั้งหมดที่คนไข้เคยได้รับการวินิจฉัยมา หากคนไข้ต้องการเข้าไปดูในรายละเอียดทั้งหมดของการวินิจฉัยในครั้งนั้นก็ให้เลือกไปยังหมายเลขลำดับของรายการที่ต้องการ ก็จะได้อ่านหน้าจอดังในรูปที่ 8.8

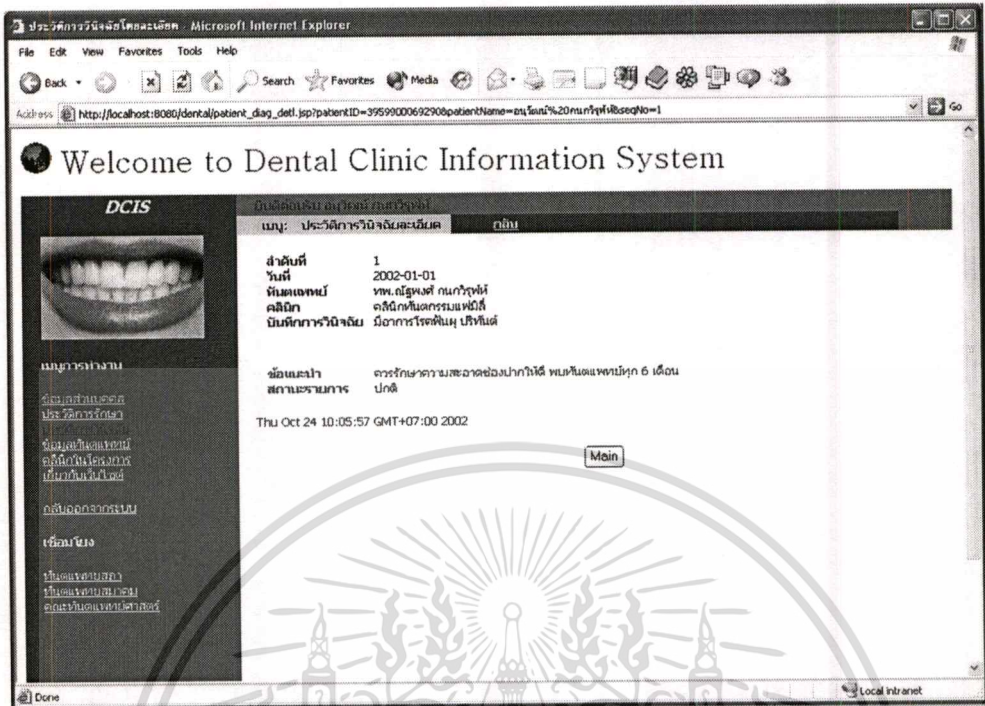


รูปที่ 8.6 หน้าจอแสดงประวัติการรักษาในแบบละเอียด



รูปที่ 8.7 หน้าจอแสดงประวัติการวินิจฉัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.8 หน้าจอแสดงประวัติการวินิจฉัยโดยละเอียด

8.2.3 ฟังก์ชันการใช้ของผู้ใช้ที่เป็นทันตแพทย์

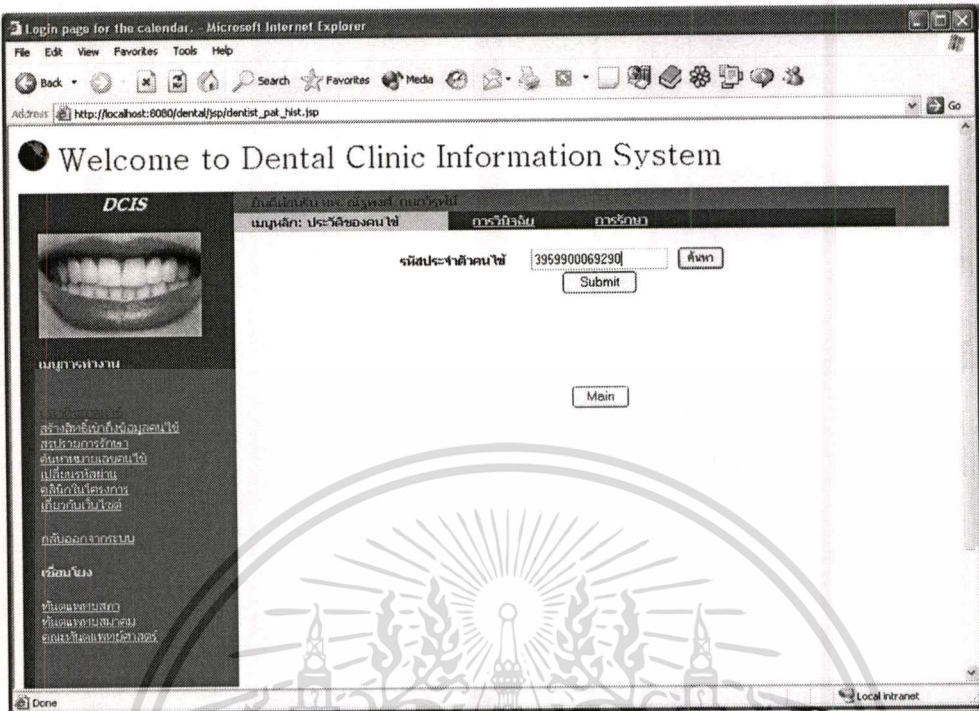
ผู้ใช้ที่เป็นทันตแพทย์สามารถดูรายละเอียดต่างๆของคนไข้ที่ต้องการ พร้อมทั้งสามารถบันทึกการวินิจฉัย การรักษาให้กับคนไข้ได้ โดยมีฟังก์ชันในการทำงานหลักๆ ดังต่อไปนี้ คือ

- ข้อมูลส่วนบุคคลของทันตแพทย์
- การสร้างสิทธิ์เข้าถึงข้อมูลคนไข้
- ข้อมูลของคนไข้
- ประวัติการรักษาของคนไข้การบันทึกการรักษา
- ประวัติการวินิจฉัยของคนไข้การบันทึกการรักษา

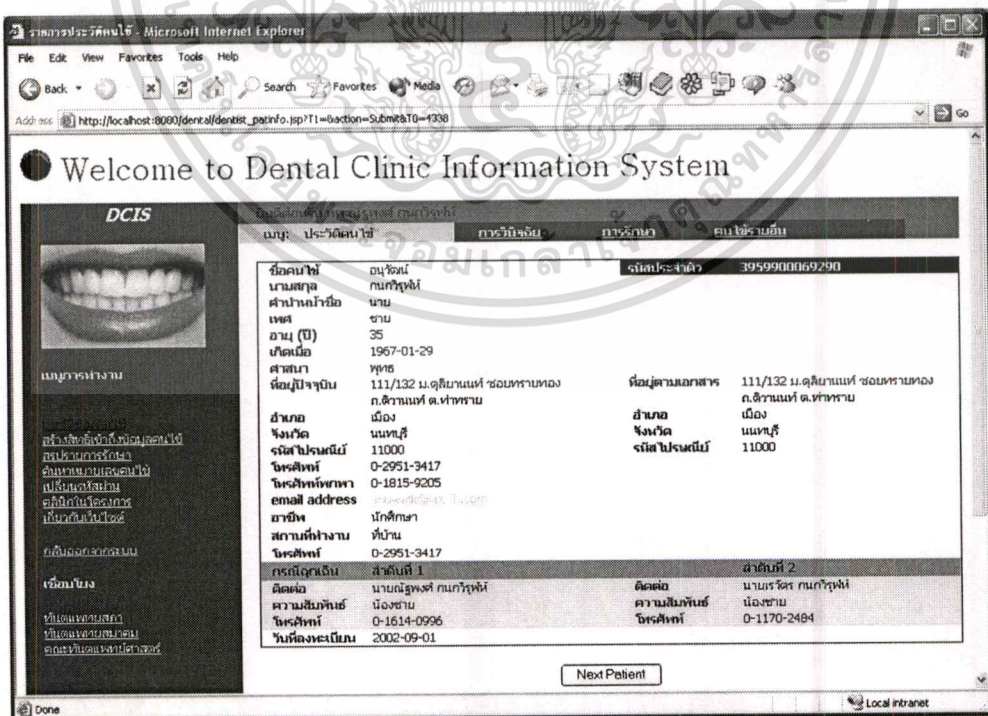
8.2.3.1 ข้อมูลส่วนบุคคลของทันตแพทย์

ข้อมูลส่วนบุคคลจะเป็นหน้าจอหลักหลังจากที่ได้ login เข้าสู่ระบบ ดังในรูปที่

8.9 ข้อมูลส่วนนี้คนไข้สามารถที่จะแก้ไขได้ โดยหากต้องการแก้ไขผู้ใช้สามารถเลือกเมนูย่อย **เปลี่ยนแปลงข้อมูล** เพื่อเข้าสู่หน้าจอการแก้ไขข้อมูลได้



รูปที่ 8.10 หน้าจอสำหรับกรอกรหัสของ Patient เมื่อ Dentist เรียกดูประวัติของคนไข้



รูปที่ 8.11 หน้าจอแสดงข้อมูลของคนไข้ถูกเรียกโดย Dentist

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2.3.4 ประวัติการรักษาของคนไข้และบันทึกการรักษา

ในส่วนของการรักษานั้น หลังจากผ่านขั้นตอนตรวจสอบสิทธิ์แล้ว ทันตแพทย์สามารถเลือกเข้าไปในเมนูย่อยหัวข้อ *การรักษา* หัวข้อนี้จะแสดงประวัติการรักษาของคนไข้ย้อนหลังกลับไปดังในรูปที่ 8.12 หากต้องการดูในรายละเอียดก็เลือกยังหมายเลขหัวข้อของรายการนั้นๆ ซึ่งมีลักษณะหน้าจอคล้ายกับในส่วนของคนไข้

สำหรับในกรณีที่ต้องการบันทึกการรักษา ทันตแพทย์สามารถคลิกปุ่ม *Add treatment* และเข้าไปกรอกข้อมูลเกี่ยวกับการรักษาในคราวนั้น ซึ่งมีหน้าจอดังในรูปที่ 8.13

DCIS

ประวัติการรักษา

ลำดับ	วันที่	เวลา	คลินิก	ทันตแพทย์	ผู้ป่วย	ประเภท	รายละเอียดการรักษา
1	2002-01-01	18:00	0001	4338	12	01	ถอนฟัน
2	2002-01-02	18:00	0001	4338	13	02	อุดฟัน
3	2002-01-08	18:00	0001	4338	11	02	อุดฟัน
4	2002-01-04	18:00	0001	4338	21	02	อุดฟัน
5	2002-01-05	18:00	0001	4338	22	02	อุดฟัน
6	2002-08-25	18:00	0001	4338	11	01	อุดฟันเดือบนอนฟัน
7	2002-08-25	18:00	0001	4338	14	01	ขัดฟันขาว

Thu Oct 24 10:23:56 GMT+07:00 2002

Add Treatment

รูปที่ 8.12 หน้าจอแสดงข้อมูลการรักษาของคนไข้ถูกเรียกโดย Dentist

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

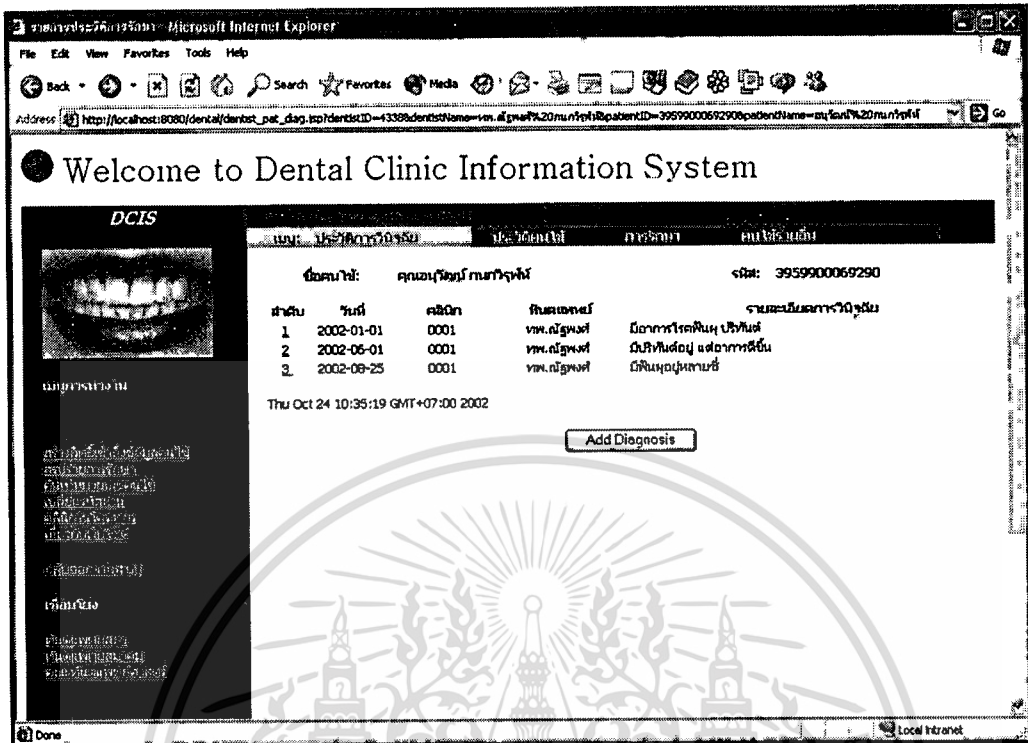
รูปที่ 8.13 หน้าจอแสดงการบันทึกข้อมูลการรักษาของคนไข้ที่ถูกเรียกโดย Dentist

เมื่อกรอกข้อมูลการรักษาเสร็จสิ้นแล้ว ให้กดปุ่ม Submit ข้อมูลจะถูกบันทึกลงในฐานข้อมูล ซึ่งจะนำไปต่อทำรายการล่าสุดที่มีการบันทึกเอาไว้

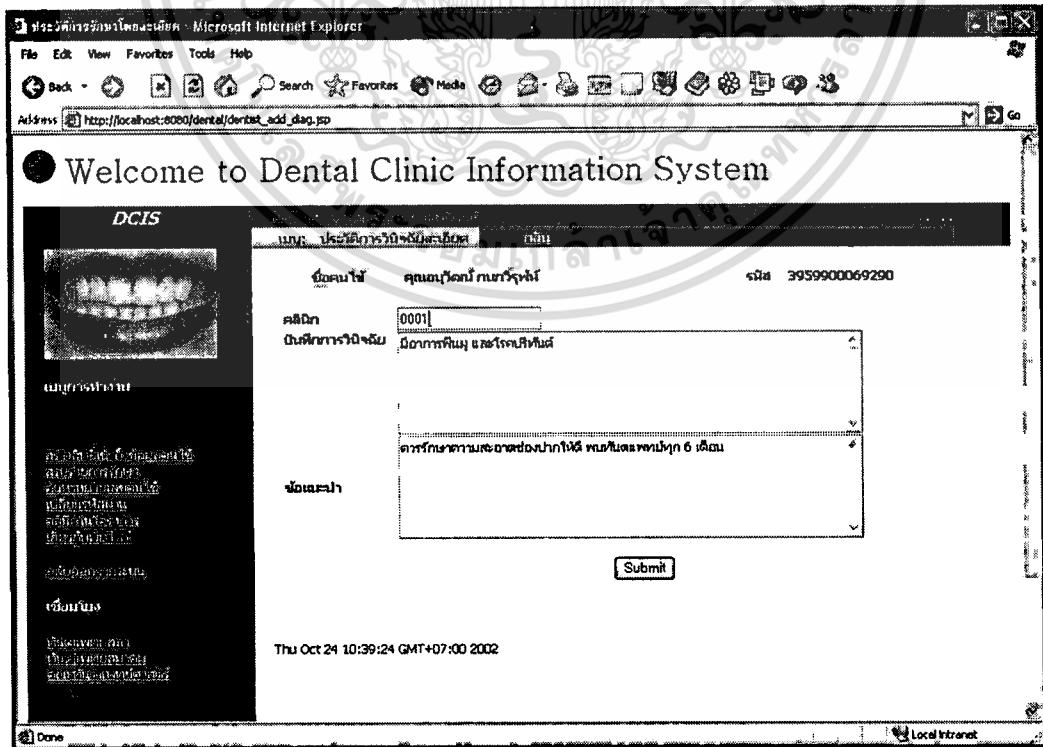
8.2.3.5 ประวัติการวินิจฉัยของคนไข้และบันทึกการวินิจฉัย

ในส่วนของการวินิจฉัยนั้น ทันตแพทย์สามารถเลือกเข้าไปในเมนูย่อยหัวข้อ การวินิจฉัย ได้เช่นเดียวกับการรักษา ทันตแพทย์จะพบกับหน้าจอประวัติรายการที่ผ่านการวินิจฉัยและบันทึกเอาไว้ดังในรูปที่ 8.14

สำหรับในกรณีที่ต้องการบันทึกการวินิจฉัยเพิ่มเติม ทันตแพทย์สามารถทำได้ด้วยการกดปุ่ม Add diagnosis และเข้าไปกรอกข้อมูลตามการวินิจฉัยในคราวนั้น ซึ่งมีหน้าจอ ดังในรูปที่ 8.15



รูปที่ 8.14 หน้าจอแสดงข้อมูลการวินิจฉัยของคนไข้ถูกเรียกโดย Dentist



เอกสารนี้เป็นเอกสารตัวอย่างของโปรแกรมนี้เอกสารบันทึกข้อมูลการวินิจฉัยของคนไข้โดย Dentist
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

สรุปผลการพัฒนาโครงการ

ในการศึกษาและการทำงานที่ผ่านมาพบว่า ในการจัดทำเว็บแอปพลิเคชัน (web-based Application) จำเป็นต้องใช้ความรู้ในหลายเรื่องมาเชื่อมโยงเข้าด้วยกันจึงจะทำให้งานประสบความสำเร็จได้ ไม่ว่าจะเป็นการออกแบบเว็บเพจ, J2EE, JSP, UML, ระบบจัดการฐานข้อมูล ภาษา SQL เป็นต้น โดยในโครงการนี้ยังต้องมีการพัฒนาต่อในส่วนของการออกแบบเพิ่มเติมให้สมบูรณ์ยิ่งขึ้น และนำไปจัดทำเป็น โปรแกรมเพื่อให้ใช้งานได้จริง สำหรับส่วนที่ควรต้องขยายและปรับปรุงต่อไปในอนาคต คือ ระบบจัดการฐานข้อมูลซึ่งในระบบงานนี้ใช้ MySQL ซึ่งเหมาะกับระบบงานที่มีขนาดไม่ใหญ่นัก หากข้อมูลภายในระบบงานมีจำนวนมาก จำเป็นต้องเปลี่ยนไปใช้ระบบจัดการที่มีความสามารถมากขึ้น เช่น DB2, Oracle หรือ Informix เป็นต้น ส่วนของการรักษาความปลอดภัยเพื่อเพิ่มความเชื่อมั่นให้กับผู้ใช้ระบบว่าจะสามารถป้องกันไม่ให้ข้อมูลส่วนบุคคลถูกนำไปใช้อย่างไม่ถูกต้อง

การนำระบบงานจัดเก็บข้อมูลทางทันตกรรมไปใช้งานนั้น ถ้าหากสามารถนำไปใช้งานได้จริงก็จะเกิดประโยชน์อย่างยิ่งแก่ทุกฝ่าย ไม่ว่าจะเป็นคนไข้ หรือทันตแพทย์ และเจ้าของคลินิก ตลอดจนถึงวงการวิชาการในสาขาวิชาทันตกรรม แต่ในความเป็นจริงแล้ว การนำไปใช้งานจริง จำเป็นต้องได้รับความร่วมมือทุกฝ่าย ทั้งจากคลินิก คนไข้ และทันตแพทย์ ซึ่งน่าจะเป็นปัญหาในทางปฏิบัติ ทั้งนี้ปัญหาประกอบขึ้นจากหลายสาเหตุ ทั้งในเรื่องการเห็นความสำคัญของการจัดเก็บรักษาข้อมูลของคนไข้ของทุกฝ่าย ความไม่พร้อมของทันตแพทย์ผู้ที่จะต้องเป็นผู้บันทึกข้อมูลคนไข้เข้าระบบ ความไม่พร้อมทางด้านเครื่องคอมพิวเตอร์และระบบสื่อสารที่จะต้องมีการลงทุนติดตั้งเพิ่มเติม รวมถึงคนไข้ที่มีความรู้และความสนใจในสุขภาพของตนเองแตกต่างกัน ในบางกลุ่มอาจมีความสนใจในด้านสุขภาพมาก ต้องการทราบการรักษาและวินิจฉัยต่างๆที่เกิดขึ้นทั้งหมด ในขณะที่คนไข้บางกลุ่มก็ไม่ได้ให้ความสนใจ และเข้าพบทันตแพทย์เมื่อเกิดปัญหาทางด้านฟันขึ้นแล้วเท่านั้น

อย่างไรก็ตาม ระบบงานนี้ยังเหมาะกับคลินิกทันตกรรมที่มีลักษณะจัดการเป็นเครือข่าย ซึ่งจะช่วยให้เกิดประโยชน์และเกิดความสะดวกต่อคลินิกที่เข้าร่วมโครงการเป็นอย่างมาก เนื่องจากสามารถเรียกดูข้อมูลคนไข้ได้ตลอดเวลา และในอนาคตผู้จัดทำเชื่อว่าลักษณะการรวมกลุ่มของคลินิกต่างๆจะต้องเกิดมากขึ้นๆ ทั้งนี้เนื่องจากสภาวะการแข่งขันที่จะต้องแข่งกับคู่แข่งที่เป็นศูนย์ทันตกรรมขนาดใหญ่ หรือโรงพยาบาลที่มีแผนกทันตกรรมให้บริการอยู่ด้วย คลินิกขนาดเล็กต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำเป็นต้องมีการปรับตัว นำข้อมูลต่างๆที่เกี่ยวข้องมาใช้ให้เป็นประโยชน์ ด้วยทำการบันทึกข้อมูลของคนไข้และการรักษาให้มากที่สุด เพื่อนำมาใช้ประโยชน์ ไม่ว่าจะนำไปใช้ในทางการตลาด การนำข้อมูลมาช่วยในการรักษา การติดต่อดนัดหมาย และการติดต่อกับคนไข้เมื่อถึงระยะเวลาที่คนไข้ควรพบทันตแพทย์ สิ่งต่างๆเหล่านี้จะเกิดขึ้นเพื่อสร้างความประทับใจและดึงดูดคนไข้ไม่ให้เปลี่ยนไปใช้บริการจากคลินิกอื่นในที่สุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- สุรัส ธาราสมบัติ. 2544. “ระบบงานควบคุมวัสดุอะไหล่เครือข่ายโทรศัพท์มือถือผ่านเว็บ” โครงการพัฒนาระบบงาน วิทยาศาสตร์มหาบัณฑิต(เทคโนโลยีสารสนเทศ) คณะเทคโนโลยีสารสนเทศ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- สุนทริน วงศ์ศิริกุล. 2543. พัฒนาโมเดลยุคใหม่ UML มาตรฐานการสร้างโมเดลระบบงาน. กรุงเทพฯ.: บริษัท ซัคเซสมิเดีย จำกัด.
- Apache. 2002. The Jakarta Project. [Online]. Available : <http://jakarta.apache.org>.
- Khawar Ahmed and Loic Julien. 2000. **Enterprise JAVA and Rational Rose**. Rational Software White Paper.
- Martin Fowler and Kendall Scott. 2000. **UML Distilled Second Edition**. New Jersey : Addison-Wesley.
- OMG. 2001. **OMG-Unified Modeling Language, v1.4 Specification**. [Online] Available : <http://www.omg.org/technology/documents/formal/uml.htm>
- Paul R. Reed, Jr. 2001. **Developing Applications with JAVA and UML**. Boston : Addison-Wesley.
- Pressman, S. 1997. **Software Engineering : A Practitioner’s Approach**. 4th edition. Singapore : McGraw-Hill.
- Sibershatz, A. et.Al. 1997. **Digital Libraries: design, development and management**. [Online]. Available: <http://www.digitallarningspace.com>

ประวัติผู้เขียน

ชื่อผู้เขียน

นายอนุวัฒน์ กนกวิรุพท์

วันเดือนปีเกิด

29 มกราคม พ.ศ. 2510

สถานที่เกิด

จังหวัดยะลา

วุฒิการศึกษาระดับปริญญาตรี

วท.บ. (วิทยาการคอมพิวเตอร์ประยุกต์)

สถานที่สำเร็จการศึกษา

คณะวิทยาศาสตร์ประยุกต์

สถาบันเทคโนโลยีพระจอมเกล้า พระนครเหนือ

ปีที่สำเร็จการศึกษา

ปีการศึกษา 2533



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้