

บทที่ 3

วิธีการดำเนินงาน

3.1 การศึกษาและการวิเคราะห์ระบบงานปัจจุบัน

การดำเนินงานของระบบทันตสุขภาพเด็ก โรงพยาบาลสมเด็จพระยุพราชเด่นชัย แบ่งลำดับขั้นตอนการดำเนินงานโดยเริ่มจากการศึกษาความเป็นไปได้ของโครงการ โดยได้รวบรวมข้อมูลและหลักการที่เกี่ยวข้องจากหนังสือและเว็บไซต์ต่างๆ มาวิเคราะห์และออกแบบเป็นระบบงาน และพัฒนาตามขั้นตอนต่างๆ

ในการวิเคราะห์และออกแบบระบบนั้น ผู้จัดทำได้ใช้แนวความคิดของการวิเคราะห์และออกแบบเชิงวัตถุ หรือ Objected Oriented Analysis and Design ซึ่งการออกแบบระบบนี้ประกอบไปด้วย

การวิเคราะห์และออกแบบเชิงวัตถุ

(Objected Oriented Analysis and Design)

ประกอบด้วยเอกสารที่สำคัญ ดังต่อไปนี้

1. Software Project Plan
2. Software Requirement Specification
3. Software Design
4. Test Plan

Project Plan

ระบบอาจารย์ที่ปรึกษาออนไลน์สำหรับคณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยนอร์ทเชียงใหม่

[Dental Health System for Yuparaj Denchai]

Project Name		
Dental Health System for Yuparaj Denchai		
Project Plan		
Cross Ref.	Coverage Level	Version
Cross Ref VSE-29110	Project	1.0

Process Ownership	Approving Authority
JedSadaKorn S.	Danuphon W.
Scope	Approved Date
Use in Project	

Document History				
Version Number	Record Date	Prepared/ Modified By	Reviewed By	Change Details

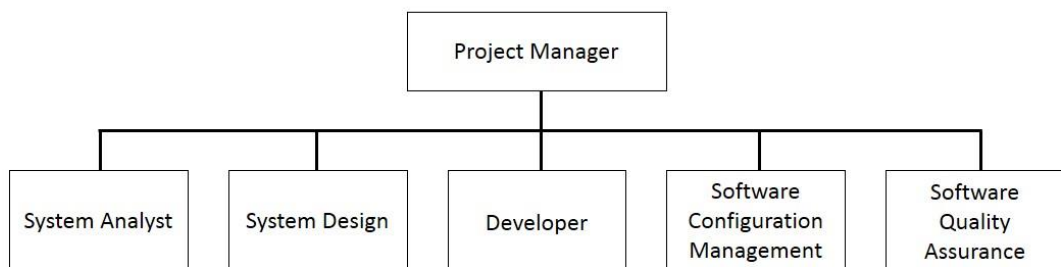
Title Page

Document Name:	Software Project Plan
Publication Date:	
Revision Date:	
Contract Number:	
Project Number:	
Prepared by:	JedSadaKorn S.
Approved by:	

Software Project Plan

1. Management Procedures

1.1. Project Team Structure



ภาพที่ 3.1 Project Team Structure

หน้าที่รับผิดชอบในตำแหน่งต่างๆ ขององค์กรที่รับผิดชอบในโครงการมีดังต่อไปนี้

ผู้จัดการโครงการ (Project Manager)

ภาระหน้าที่ของ Project Manager

- จัดทำ และนำเสนอโครงการ
- ประเมินค่าใช้จ่ายในการทำโครงการ
- วางแผน และจัดเวลาการดำเนินโครงการ
- ตรวจสอบ ควบคุม ติดตาม และทบทวนโครงการ
- รายงาน และนำเสนอโครงการ
- จัดการความเปลี่ยนแปลงในโครงการ

นักวิเคราะห์ระบบ (System Analyst)

ภาระหน้าที่ของนักวิเคราะห์ระบบ

- ศึกษา และวิเคราะห์ความต้องการของการพัฒนาระบบ
- วิเคราะห์ และออกแบบระบบ
- ติดต่อประสานงานกับผู้ใช้ และผู้เกี่ยวข้องกับการพัฒนาระบบ
- จัดทำเอกสารประกอบการวิเคราะห์ และออกแบบ

นักออกแบบระบบ (System Design)

ภาระหน้าที่ของนักออกแบบ

- ศึกษา และวิเคราะห์ความต้องการของการพัฒนาระบบ
- ออกแบบระบบ ระดับ Detail Design
- ติดต่อประสานงานกับโปรแกรมในการพัฒนาระบบ
- จัดทำเอกสารประกอบการออกแบบ

นักพัฒนาระบบ (Developer)

ภาระหน้าที่ของนักพัฒนาระบบ

- เขียนโปรแกรมตามที่วิเคราะห์ และออกแบบไว้
- พัฒนา Test Case และดำเนินการทดสอบโปรแกรม
- จัดทำเอกสารประกอบการพัฒนาโปรแกรม และการใช้โปรแกรม

Software Configuration Management

ภาระหน้าที่ของ Software Configuration Management

- จัดสรรพื้นที่ในการจัดเก็บเอกสารโครงการ
- บริหารการเข้าถึงพื้นที่ในการจัดเก็บเอกสารโครงการ
- กำหนดกฎเกณฑ์ในการระบุรุ่น (Version/Release) ของเอกสาร/ซอฟต์แวร์ โครงการ

Software Quality Assurance

ภาระหน้าที่ของ Software Quality Assurance

- พัฒนาระบบประกันคุณภาพซอฟต์แวร์
- บริหารจัดการกระบวนการผลิตซอฟต์แวร์
- ตรวจสอบติดตามกระบวนการ และการผลิตซอฟต์แวร์ทั้งระบบ อบรมกระบวนการ/

เครื่องมือที่เกี่ยวข้อง

1.2. Project Responsibility

กำหนดผู้รับผิดชอบในแต่ละหน้าที่ดังต่อไปนี้

หน้าที่ความรับผิดชอบ	ผู้รับผิดชอบ
Project Manager	นายเจษฎากร ศิริกุลพันธ์
System Analyst	นายเจษฎากร ศิริกุลพันธ์
System Design	นายเจษฎากร ศิริกุลพันธ์
Developer	นายเจษฎากร ศิริกุลพันธ์
Software Configuration Management	นายเจษฎากร ศิริกุลพันธ์
Software Quality Assurance	นายเจษฎากร ศิริกุลพันธ์

ตารางที่ 3.1 ตาราง Project Responsibility

1.3. Monitoring and Controlling Mechanisms

1.3.1. Project Meeting

กำหนดให้เมื่อแต่ละวันทำเอกสารเสร็จให้ไปเปลี่ยนแปลงที่ Status Reporting และวางแผนว่าจะทำอะไรวันครั้งต่อไป

1.3.2. Status Reporting

Tasks	%	Duration	Start Date	Finish Date	Resources Used
บทที่ 1	99	1 วัน	6 ต.ค. 2557	6 ต.ค. 2557	Document
บทที่ 2	99	4 วัน	6 ต.ค. 2557	9 ต.ค. 2557	Document
บทที่ 3	70	20 วัน	10 ต.ค. 2557	30 ต.ค. 2557	Document
บทที่ 4	0				Document
บทที่ 5	0				Document

ตารางที่ 3.2 ตาราง Status Reporting

1.3.3. Escalation Mechanisms

Project Manager จะเป็นผู้แก้ไขสถานการณ์/ปัญหาที่เกิดขึ้น ในกรณีที่ไม่สามารถแก้ไข หรือกระทำได้ ให้แจ้งแก่ อาจารย์ที่ปรึกษา เพื่อรับทราบปัญหาและแก้ไขสถานการณ์/ปัญหาที่เกิดขึ้นต่อไป

1.3.4. Change Management

ในกรณีที่มีความต้องการเปลี่ยนแปลงในโครงการ จะต้องดำเนินการดังนี้

- 1) จัดทำเอกสารร้องขอเปลี่ยนแปลง
- 2) วิเคราะห์การเปลี่ยนแปลงว่าจะกระทบต่อส่วนไหนบ้าง
- 3) ยื่นต่ออาจารย์ที่ปรึกษา ว่าอาจารย์ที่ปรึกษาจะอนุมัติให้เปลี่ยนแปลงหรือไม่
- 4) เปลี่ยนแปลงและตรวจสอบความถูกต้อง ของการสร้าง Project Baseline ใหม่ที่เกิดการเปลี่ยนแปลงขึ้น
- 5) บันทึกความเปลี่ยนแปลง Baseline ใหม่

2. Quality Planning

2.1. Review/Responsibility

Stage Exit Review			
No.	Stage	Review Item	Responsibility
1	เมื่อเสร็จสิ้นการทำ Proposal	Proposal	PM
2	เมื่อเสร็จสิ้นการทำ Requirement Specification	Software Requirement Specification (SRS)	PM
3	เมื่อเสร็จสิ้นการทำ Software Design	Software Design	SA,SD
4	เมื่อเสร็จสิ้นการทำ Project Planning	Preliminary Planning Document	PM,SA
5	เมื่อเสร็จสิ้นการพัฒนาระบบ อาจารย์ที่ปรึกษาออนไลน์สำหรับ	ระบบอาจารย์ที่ปรึกษาออนไลน์สำหรับคณะ	Developer, Tester

	คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัย	วิทยาศาสตร์และ เทคโนโลยีมหาวิทยาลัย	
--	---	--	--

ตารางที่ 3.3 ตาราง Reviews/Responsibility

2.2. Testing

3. Test Process			
No.	Test	Verification	Responsibility
1	Unit Testing	ทดสอบความถูกต้องของการ ทำงานระดับฟังก์ชัน	Developer, Tester
2	Integration Testing	ทดสอบการประกอบโมดูลย่อย ต่างๆเข้าด้วยกัน	Developer, Tester
3	System Testing	ทดสอบความถูกต้องของการ ทำงานระบบ	PM, Tester

ตารางที่ 3.7 ตาราง Testing

4. Estimated Duration of Task

ระยะเวลาที่ใช้ในการดำเนินงานแต่ละขั้นตอนสามารถประมาณได้ดังนี้

Task Name	Duration	Plan Start	Plan Finish	Responsibility
ศึกษาหัวข้อและรวบรวมข้อมูล	15 วัน	1/9/2557	19/9/2557	PM
นำข้อมูลที่ได้มาวิเคราะห์ความต้องการ	15 วัน	15/9/2557	3/10/2557	PM, SA
วางแผนโครงการ	3 วัน	4/10/2557	6/10/2557	PM
พัฒนาระบบ	32 วัน	1/11/2557	2/12/2557	D
Testing	30 วัน	5/11/2557	3/12/2557	QA
ส่งมอบโปรแกรม		8/12/2557	29/12/2557	PM

ตาราง 3.4 ตาราง ระยะเวลาที่ใช้ในการดำเนินงาน

5. Estimated Effort and Cost

การพัฒนาระบบทันตสุขภาพเด็ก โรงพยาบาลสมเด็จพระยุพราชเด่นชัย สามารถประมาณ
การต้นทุน และผลตอบแทนที่สมควรได้รับตามแผนการดำเนินงานโครงการระยะเวลา 4 เดือน ได้
ดังนี้

-ต้นทุนในการพัฒนา

No.	รายการ	ราคา/วัน	จำนวนวัน	คน	รวม
1	ค่าจ้าง Project Manager	0	0	1	0
2	ค่าจ้าง System Analyst	0	0	1	0
3	ค่าจ้าง Developer	0	0	1	0
4	ค่าจ้าง Software Quality Assurance	0	0	1	0
5	ค่าจ้าง Software Design	0	0	1	0
6	ค่าจ้าง Software Configuration Management	0	0	1	0
7	ค่าสืกรหรือและเสื่อมราคาของอุปกรณ์	0	0	1	0
8	ค่าที่พัก	80	120	1	9,600
รวมทั้งสิ้น					9,600

ตาราง 3.5 ตาราง ค่าจ้างผู้พัฒนา

-จุดคุ้มทุนและผลตอบแทนที่ควรได้รับ

No.	รายการ	รวม
1	ต้นทุนในการพัฒนา	9,600
รวมทั้งสิ้น		9,600

ตาราง 3.6 ตาราง จุดคุ้มทุนและผลตอบแทนที่ควรได้รับ

ดังนั้นในการพัฒนาในปีแรก จะต้องได้รับผลตอบแทนจากลูกค้าทั้งสิ้น 9,600 บาท

หมายเหตุ คำบำรุงรักษาซอฟต์แวร์ดังกล่าว ไม่รวมถึงการเพิ่มเติมระบบงานที่ไม่ได้อยู่ในขอบเขตของสัญญาว่าจ้างโครงการเพื่อพัฒนา ซึ่งหากต้องการนอกเหนือจากที่ระบุไว้ในสัญญานั้น ลูกค้าจะต้องชำระเพิ่มเติมโดยเป็นไปตามเงื่อนไขของผู้ว่าจ้างในขณะนั้น

6. Identification of Project Risks

ในการพัฒนาระบบทันตสุขภาพเด็ก โรงพยาบาลสมเด็จพระยุพราชเด่นชัย มีความเสี่ยงที่จะเกิดขึ้นดังต่อไปนี้

- การล้มเลิกธุรกิจของผู้ว่าจ้างระหว่างการพัฒนา อาจทำให้ผู้ว่าจ้างไม่ต้องซอฟต์แวร์ที่กำลังพัฒนาอีกต่อไป
- ผู้ใช้งานระบบมีหลายกลุ่มซึ่งความเห็นอาจไม่ตรงกัน อาจจะได้ไม่ถูกใจครบทุกคน แต่ต้องให้ผู้ใช้งานหาข้อสรุประหว่างตัวผู้ใช้งานให้ได้ โดยต้องเน้นความถูกต้องและกระบวนการทำงานเป็นหลัก
- เวลาว่างของผู้ให้ความต้องการของระบบ ในช่วงเก็บความต้องการของระบบอาจไม่มากพอที่จะทำให้เก็บรายละเอียดได้ครบถ้วน
- เนื่องจากผู้พัฒนามีเพียงคนเดียว หากผู้พัฒนาป่วยไม่สามารถทำงานได้ ก็อาจจะทำให้งานดำเนินไปไม่ตรงตามแผนที่วางไว้

- Hardware/Software ที่ใช้ในการพัฒนาระบบ เกิดความเสียหายระหว่างการพัฒนา ทำให้เกิดความล่าช้าในการพัฒนาระบบ ส่งผลให้การพัฒนาไม่เป็นไปตามแผนที่วางไว้

7. Version Control Strategy

การควบคุมเวอร์ชันของไฟล์ต่างๆที่เกี่ยวข้องกับการพัฒนาระบบทันตสุขภาพเด็ก
โรงพยาบาลสมเด็จพระยุพราชเด่นชัย สามารถอธิบายได้ดังนี้

1) รูปแบบการจัดเก็บไฟล์ลง Repository

Directory

- Directory หลักสำหรับเก็บข้อมูลของโครงการคือ Project
- Directory สำหรับจัดเก็บ Document คือ DOC
- Directory สำหรับจัดเก็บ Source Code คือ SCR
- Directory สำหรับจัดเก็บ Testing Files คือ TST
- Directory สำหรับจัดเก็บ Image Files คือ IMG

Files

- ไฟล์ที่เกี่ยวข้องกับโครงการนี้ จะมีรูปแบบการตั้งชื่อดังนี้

XxxxYyyy คือประเภทของเอกสาร

ตัวอย่างเช่น SoftwareProjectPlan

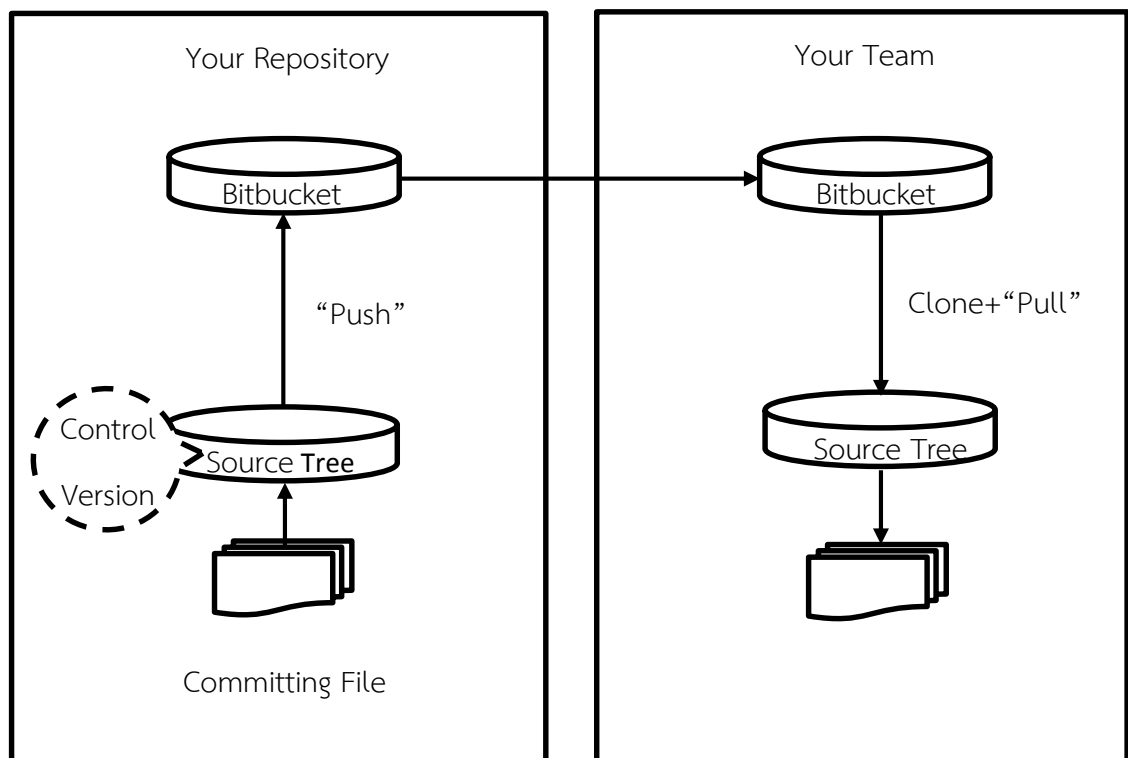
Source Code

- จัดเก็บอยู่ภายใต้ Directory SCR ซึ่ง Directory นั้นจัดเก็บ Source Code ต่างๆขึ้นอยู่กับเฟรมเวิร์คที่ใช้ในการพัฒนา ในที่นี่ใช้รูปแบบของ MVC ก็จะมีโครงสร้างของ Directory ย่อย คือ models views controller ซึ่งใน Directory models ก็จะทำหน้าที่จัดเก็บ Source Code ที่เป็นการออกแบบโมเดล ใน Directory views ก็จะทำหน้าที่จัดเก็บ Source Code ที่เกี่ยวกับการแสดงผล ส่วนติดต่อกับผู้ใช้ และใน Directory controller ก็จะทำหน้าที่จัดเก็บ Source Code ที่เกี่ยวกับส่วนควบคุมการทำงานพวก Business Logic ต่างๆ

- การตั้งชื่อไฟล์ Source Code นั้น ให้ยึดหลักคือ txxxxyy โดยที่ t คือประเภทของการทำงานของ Source Code โดยกำหนดให้ m คือ Model v คือ View c คือ Control เช่น caddcustomer.php

2) เครื่องมือที่ใช้ในการจัดการ Repository

เพื่อความสะดวกและง่ายต่อการใช้งาน จึงใช้ Source Tree เป็นเครื่องมือจัดการกับข้อมูลต่างๆร่วมกับ Bitbucket ซึ่งเป็นเว็บไซต์หนึ่งที่ทำให้บริการทำหน้าที่เป็นแม่ข่ายให้กับ Git (Git เป็น Revision control แบบกระจายศูนย์) สามารถควบคุมเวอร์ชันของเอกสารและ Source Code ได้



ภาพที่ 3.3 แผนภาพแสดงการทำงานร่วมกันระหว่าง Source Tree และ Bitbucket

Class Diagram

CD: 01 Address



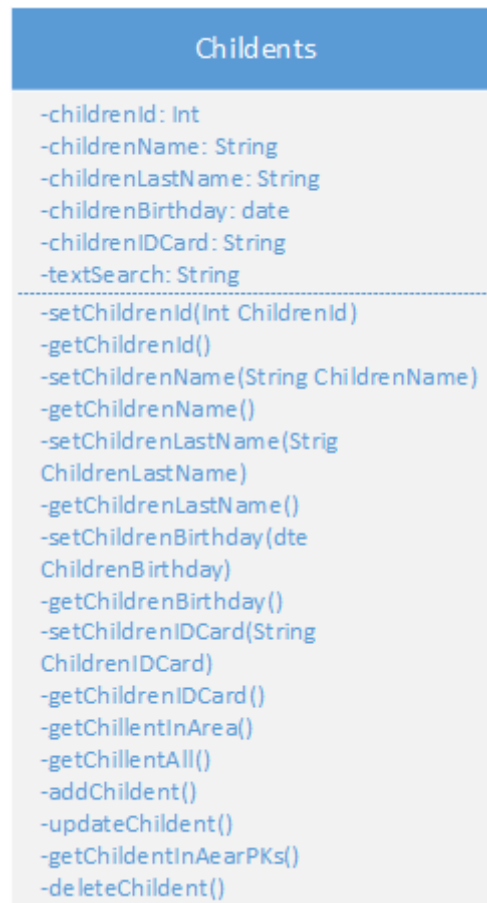
ภาพที่ 3.4 Class Diagram Address

CD: 02 Behavior

Behavior
<div><div>-behaviorId: Int</div><div>-behaviorName: String</div><div>-behaviorType: String</div><div>-behaviorTypeId: Int</div><div>-colorCode: String</div><div>-behaviorTypeName: String</div></div> <div><div>-setBehaviorId(Int BehaviorId)</div><div>-getBehaviorId()</div><div>-setBehaviorName(String BehaviorName)</div><div>-getBehaviorName()</div><div>-setBehaviorType(String BehaviorType)</div><div>-getBehaviorType()</div><div>-setBehaviorTypeId(Int BehaviorTypeId)</div><div>-getBehaviorTypeId()</div><div>-setColorCode(String ColorCode)</div><div>-getColorCode()</div><div>-setBehaviorTypeName(String BehaviorTypeName)</div><div>-getBehaviorTypeName()</div><div>-getAlldataBehavior()</div><div>-getAlldataBehaviortype()</div><div>-addBehavior()</div><div>-addBehaviorType()</div><div>-getBehaviorTypePk()</div><div>-updateBehaviorType()</div><div>-updateBehavior()</div><div>-deleteBehaviorTypeDataPk()</div><div>-deleteBehaviorDataPk()</div><div>-getBehaviorPk()</div></div>

ภาพที่ 3.5 Class Diagram Behavior

CD: 03 Childents



ภาพที่ 3.6 Class Diagram Childents

```

classDiagram
    class Members {
        -memberid int
        -memberFirstname String
        -memberLastname String
        -memberEmail String
        -memberPhoneNo String
        -memberDob Date
        -memberGender String
        -memberIsDeleted Boolean
    }
    class Labname {
        -labnameid int
        -labnameName String
        -labnameRoom String
        -labnameRoomNo String
    }
    class District {
        -districtid int
        -districtName String
        -districtProvince String
    }
    class Canton {
        -cantonid int
        -cantonName String
        -cantonDistrict String
        -cantonProvince String
    }
    class Address {
        -addressid int
        -addressName String
        -addressRoom String
        -addressRoomNo String
    }
    class Tel {
        -telid int
        -telNumber String
        -telType String
    }
    class Polityps {
        -politypid int
        -politypName String
        -politypDistance String
        -politypMeeting String
    }
    class Meetings {
        -meetingid int
        -meetingOrder String
        -meetingDate String
        -meetingPolityp String
    }
    class Distance {
        -distanceid int
        -distanceOrder String
        -distancePolityp String
    }
    class Disease {
        -diseaseid int
        -diseaseName String
        -diseaseLabname String
        -diseaseDistrict String
        -diseaseCanton String
    }
    class Childrens {
        -childrenid int
        -childrenFirstname String
        -childrenLastname String
        -childrenEmail String
        -childrenPhoneNo String
        -childrenDob String
        -childrenGender String
    }
    Members "1" -- "1" Labname
    Labname "1" -- "1" District
    District "1" -- "1" Canton
    Canton "1" -- "1" Address
    Address "1" -- "1" Tel
    Tel "1" -- "1" Polityps
    Polityps "1" -- "1" Meetings
    Meetings "1" -- "1" Distance
    Distance "1" -- "1" Disease
    Disease "1" -- "1" Childrens
    
```

The diagram illustrates the relationships between various entities in a COVID-19 data system. The entities and their attributes are as follows:

- Members**: memberid (int), memberFirstname (String), memberLastname (String), memberEmail (String), memberPhoneNo (String), memberDob (Date), memberGender (String), memberIsDeleted (Boolean).
- Labname**: labnameid (int), labnameName (String), labnameRoom (String), labnameRoomNo (String).
- District**: districtid (int), districtName (String), districtProvince (String).
- Canton**: cantonid (int), cantonName (String), cantonDistrict (String), cantonProvince (String).
- Address**: addressid (int), addressName (String), addressRoom (String), addressRoomNo (String).
- Tel**: telid (int), telNumber (String), telType (String).
- Polityps**: politypid (int), politypName (String), politypDistance (String), politypMeeting (String).
- Meetings**: meetingid (int), meetingOrder (String), meetingDate (String), meetingPolityp (String).
- Distance**: distanceid (int), distanceOrder (String), distancePolityp (String).
- Disease**: diseaseid (int), diseaseName (String), diseaseLabname (String), diseaseDistrict (String), diseaseCanton (String).
- Childrens**: childrenid (int), childrenFirstname (String), childrenLastname (String), childrenEmail (String), childrenPhoneNo (String), childrenDob (String), childrenGender (String).

The relationships between the entities are as follows:

- Members** (1) is associated with **Labname** (1).
- Labname** (1) is associated with **District** (1).
- District** (1) is associated with **Canton** (1).
- Canton** (1) is associated with **Address** (1).
- Address** (1) is associated with **Tel** (1).
- Tel** (1) is associated with **Polityps** (1).
- Polityps** (1) is associated with **Meetings** (1).
- Meetings** (1) is associated with **Distance** (1).
- Distance** (1) is associated with **Disease** (1).
- Disease** (1) is associated with **Childrens** (1).

ภาพที่ 3.7 Persistran Diagram

ชื่อตาราง : members

รายละเอียด : สำหรับเก็บข้อมูลผู้ใช้งาน

คีย์หลัก : memberId

คีย์รอง : -

ตารางชื่อ members			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>memberId</u>	int(4) คีย์หลัก	รหัสข้อมูลผู้ใช้งาน	0001
memberUsername	varchar(100)	ชื่อเข้าใช้งาน	admin
memberPassword	varchar(50)	รหัสผ่าน *เข้ารหัส MD5	21232f297a57a5a743894 a0e4a801fc3
memberName	varchar(100)	ชื่อผู้เข้าใช้งาน	ฉลวย
memberLastName	varchar(100)	นามสกุลผู้เข้าใช้งาน	ศิริกุลพันธ์
memberPosition	varchar(100)	ตำแหน่งผู้เข้าใช้งาน	เจ้าหน้าที่ ทดสธารณสุข ชำนาญการพิเศษ
memberBirthday	date	วันเกิดผู้เข้าใช้งาน	1966-08-22
memberStatus	enum('boss','officials')	สถานะบัญชีผู้เข้าใช้งาน	boss
memberForgetCode	varchar(55)	code ใช้ในการลืมรหัสผ่านของ ผู้ใช้แต่ละคน โดยนำ รหัสข้อ ผู้ใช้งานและชื่อนำมาต่อกัน ของแต่ ละคนนำมาเข้ารหัส MD5	1377c2bca036e34ce4618 0f29a487481
memberActiveStatus	enum('activated','notA ctivated')	สถานะการเปิดใช้บัญชีผู้ใช้งาน	activated

ตาราง 3.7 ตาราง members

ชื่อตาราง : childrens

รายละเอียด : สำหรับเก็บข้อมูลเด็ก

คีย์หลัก : childrenId

คีย์รอง : -

ตารางชื่อ childrens			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>childrenId</u>	int(4) คีย์หลัก	รหัสข้อมูลเด็ก	0001
childrenName	varchar(100)	ชื่อเด็ก	สุรชัย
childrenLastName	varchar(100)	นามสกุลเด็ก	หวังดี
childrenBirthday	date	วันเกิด	2013-10-24
childrenIDCard	varchar(13)	รหัสบัตรประจำตัวประชาชน	1556984523658

ตาราง 3.8 ตาราง childrens

ชื่อตาราง : address

รายละเอียด : สำหรับเก็บข้อมูลที่อยู่

คีย์หลัก : addressId

คีย์รอง : ownerId, cantonId, districtId, provincId

ตารางชื่อ address			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
addressId	int(5) คีย์หลัก	รหัสข้อมูลที่อยู่	00001
ownerId	int(4) คีย์รอง	รหัสข้อมูลผู้ใช้งาน หรือ รหัสข้อมูลเด็ก	0001
ownerType	enum('members','childents')	ประเภทของที่อยู่ เป็นของ ผู้ใช้งานหรือเด็ก	members
addressDetial	varchar(100)	รายละเอียดข้อมูลที่อยู่	76/3 หมู่ 3
cantonId	int(5) คีย์รอง	รหัสข้อมูลตำบล	05643
districtId	int(5) คีย์รอง	รหัสข้อมูลอำเภอ	00628
provincId	int(5) คีย์รอง	รหัสข้อมูลจังหวัด	00042
street	varchar(30)	ชื่อถนน	ถนนยันตรกิจโกศล
addressType	enum('contactcan','registration')	ประเภทที่อยู่ เป็น ที่อยู่ที่ สามารถติดต่อได้หรือ ข้อมูล ตามทะเบียนบ้าน	registration

ตาราง 3.9 ตาราง address

ชื่อตาราง : province

รายละเอียด : สำหรับเก็บข้อมูลจังหวัด

คีย์หลัก : provinceld

คีย์รอง : -

ตารางชื่อ province			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>provinceld</u>	int(5) คีย์หลัก	รหัสข้อมูลจังหวัด	00001
provinceName	varchar(150)	ชื่อจังหวัด	แพร่

ตาราง 3.10 ตาราง province

ชื่อตาราง : district

รายละเอียด : สำหรับเก็บข้อมูลอำเภอ

คีย์หลัก : districtld

คีย์รอง : provinceld

ตารางชื่อ district			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>districtld</u>	int(5) คีย์หลัก	รหัสข้อมูลอำเภอ	00628
districtName	varchar(150)	ชื่ออำเภอ	สูงเม่น
provinceld	int(5) คีย์รอง	รหัสข้อมูลจังหวัด	00042

ตาราง 3.11 ตาราง district

ชื่อตาราง : canton

รายละเอียด : สำหรับเก็บข้อมูลตำบล

คีย์หลัก : cantonId

คีย์รอง : districtId, provinceId

ตารางชื่อ canton			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>cantonId</u>	int(5) คีย์หลัก	รหัสข้อมูลตำบล	05643
cantonName	varchar(150)	ชื่อตำบล	บ้านโป่ง
districtId	int(5) คีย์รอง	รหัสข้อมูลอำเภอ	00628
provinceId	int(5) คีย์รอง	รหัสข้อมูลจังหวัด	00042

ตาราง 3.12 ตาราง canton

ชื่อตาราง : tel

รายละเอียด : สำหรับเก็บข้อมูลเบอร์โทรศัพท์

คีย์หลัก : telId

คีย์รอง : addressId

ตารางชื่อ tel			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>telId</u>	int(7) คีย์หลัก	รหัสข้อมูลเบอร์โทรศัพท์	000001
addressId	int(5) คีย์รอง	รหัสข้อมูลที่อยู่	00001
tel	varchar(10)	เบอร์โทรศัพท์	0895149852
telNote	varchar(30)	หมายเหตุเบอร์โทร	มือถือ

ตาราง 3.13 ตาราง tel

ชื่อตาราง : zipcodes

รายละเอียด : สำหรับเก็บข้อมูลรหัสไปรษณีย์

คีย์หลัก : zipcodeId

คีย์รอง : provinceId, districtId, cantonId

ตารางชื่อ zipcodes			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>zipcodeId</u>	int(5) คีย์หลัก	รหัสข้อมูลรหัสไปรษณีย์	00548
provinceId	int(5) คีย์รอง	รหัสข้อมูลจังหวัด	00042
districtId	int(5) คีย์รอง	รหัสข้อมูลอำเภอ	00628
cantonId	int(5) คีย์รอง	รหัสข้อมูลตำบล	05643
zipcode	varchar(5)	รหัสไปรษณีย์	54130

ตาราง 3.14 ตาราง zipcodes

ชื่อตาราง : liablearea

รายละเอียด : สำหรับเก็บข้อมูลเขตความรับผิดชอบ

คีย์หลัก : liableAreaId

คีย์รอง : areaId, memberId

ตารางชื่อ liablearea			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>liableAreaId</u>	int(7) คีย์หลัก	รหัสข้อมูลเขตความรับผิดชอบ	000001
areaId	int(5) คีย์รอง	รหัสข้อมูลเขต เช่น รหัสข้อมูลจังหวัด,รหัสข้อมูลอำเภอ,รหัสข้อมูลตำบล	00042
memberId	int(4) คีย์รอง	รหัสข้อมูลผู้ใช้งาน	0001
liableAreaType	enum('province','district','canton')	ประเภทข้อมูลเขตความรับผิดชอบ จะอ้างอิง กับ areaId	district

ตาราง 3.15 ตาราง liablearea

ชื่อตาราง : behaviortype

รายละเอียด : สำหรับเก็บข้อมูลหมวดหมู่พฤติกรรม

คีย์หลัก : behaviorTypeId

คีย์รอง : -

ตารางชื่อ behaviortype			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>behaviorTypeId</u>	int(2) คีย์หลัก	รหัสข้อมูลหมวดหมู่พฤติกรรม	01
behaviorTypeName	varchar(50)	ชื่อหมวดหมู่พฤติกรรม	พฤติกรรมการกิน

ตาราง 3.16 ตาราง behaviortype

ชื่อตาราง : behavior

รายละเอียด : สำหรับเก็บข้อมูลพฤติกรรม

คีย์หลัก : behaviorId

คีย์รอง : behaviorTypeId

ตารางชื่อ behavior			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>behaviorId</u>	int(3) คีย์หลัก	รหัสข้อมูลพฤติกรรม	001
behaviorName	varchar(50)	ชื่อข้อมูลพฤติกรรม	รับประทานนมหวาน
behaviorType	enum('normal','photo')	ประเภทข้อมูลพฤติกรรม เป็น การตรวจฟัน หรือตรวจทั่วไป	normal
behaviorTypeId	int(2) คีย์รอง	รหัสข้อมูลหมวดหมู่พฤติกรรม	01
colorCode	varchar(7)	สีของกราฟที่จะแสดง เก็บเป็น Code สี	#000000

ตาราง 3.17 ตาราง behavior

ชื่อตาราง : distance

รายละเอียด : สำหรับเก็บข้อมูลช่วงระยะเวลาการตรวจ

คีย์หลัก : distanceId

คีย์รอง : -

ตารางชื่อ distance			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>distanceId</u>	int(5) คีย์หลัก	รหัสข้อมูลช่วงระยะเวลาการตรวจ	00001
distanceMonth	int(3)	ระยะเวลาการตรวจ หน่วยเป็น เดือน	3

ตาราง 3.18 ตาราง distance

ชื่อตาราง : diseases

รายละเอียด : สำหรับเก็บข้อมูลโรคประจำตัว

คีย์หลัก : diseasesId

คีย์รอง : childrenId

ตารางชื่อ diseases			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>diseasesId</u>	int(7) คีย์หลัก	ข้อมูลโรคประจำตัว	0000001
childrenId	int(4) คีย์รอง	รหัสข้อมูลเด็ก	0001
diseasesName	varchar(100)	ชื่อโรค	หอบหืด
medicine	varchar(100)	ยาที่กินประจำ	ยาพ่นขยายหลอดลม

ตาราง 3.19 ตาราง diseases

ชื่อตาราง : policings

รายละเอียด : สำหรับเก็บข้อมูลการตรวจ

คีย์หลัก : policingDetialId

คีย์รอง : distanceId, childrenId, memberId

ตารางชื่อ policings			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>policingId</u>	int(9) คีย์หลัก	รหัสข้อมูลการตรวจ	00000001
distanceId	int(5) คีย์รอง	รหัสข้อมูลช่วงระยะเวลาการตรวจ	00001
childrenId	int(4) คีย์รอง	รหัสข้อมูลเด็ก	0001
memberId	int(4) คีย์รอง	รหัสข้อมูลผู้ใช้งาน	0001
policingDate	datetime	วัน/เวลา ที่ตรวจ	2014-1-24

ตาราง 3.20 ตาราง policings

ชื่อตาราง : policingdetial

รายละเอียด : สำหรับเก็บข้อมูลรายละเอียดการตรวจ

คีย์หลัก : policingDetialId

คีย์รอง : policingId, behaviorId

ตารางชื่อ policingdetial			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>policingDetialId</u>	int(11) คีย์หลัก	รหัสข้อมูลรายละเอียดการตรวจ	0000000001
policingId	int(9) คีย์รอง	รหัสข้อมูลการตรวจ	00000001
behaviorId	int(3) คีย์รอง	รหัสข้อมูลพฤติกรรม	001
policingDetialValue	int(2)	ผลรายละเอียดการตรวจ จะอ้างอิงกับ ข้อมูลพฤติกรรม	1

ตาราง 3.21 ตาราง policingdetial

ชื่อตาราง : meetings

รายละเอียด : สำหรับเก็บข้อมูลการนัด

คีย์หลัก : meetingId

คีย์รอง : policingId, childrenId

ตารางชื่อ meetings			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>meetingId</u>	int(11) คีย์หลัก	รหัสข้อมูลการนัด	0000000001
policingId	int(9) คีย์รอง	รหัสข้อมูลการตรวจ	00000001
childrenId	int(4) คีย์รอง	รหัสข้อมูลเด็ก	0001
meetingsDate	datetime	เวลานัด	2014-4-24

ตาราง 3.22 ตาราง meetings

ชื่อตาราง : brokentooth

รายละเอียด : สำหรับเก็บข้อมูลฟันผุ

คีย์หลัก : brokenToothId

คีย์รอง : policingDetailId

ตารางชื่อ brokentooth			
ชื่อ	ชนิด	รายละเอียด	ตัวอย่างข้อมูล
<u>brokenToothId</u>	int(10) คีย์หลัก	รหัสข้อมูลฟันผุ	000000001
policingDetailId	int(11) คีย์รอง	รหัสข้อมูลรายละเอียดการตรวจ	0000000001
brokenToothNumber	int(2)	เลขซี่ฟันที่ผุ	12
brokenToothSide	enum('up','down')	ด้านฟันที่ผุ บน/ล่าง	down

ตาราง 3.23 ตาราง brokentooth