



ระบบอนุมัติสินเชื่อเช่าซื้อรถยนต์และสินเชื่อ SME  
CREDIT APPROVAL PROCESS

นางสาวสิริกาญจน์ กุลวุฒิวิลาส

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีไทย-ญี่ปุ่น

พ.ศ. 2559

ระบบอนุมัติสินเชื่อเช่าซื้อรถยนต์และสินเชื่อ SME

CREDIT APPROVAL PROCESS

นางสาวสิริกาญจน์ กุลวุฒิวิลาศ

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต สาขาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย - ญี่ปุ่น

ปีการศึกษา 2559

คณะกรรมการสอบ

..... ประธานกรรมการสอบ

(ดร.สรมย์พร เจริญพิทย์)

..... กรรมการสอบ

(ผศ.เกษม ทิพย์ธาราจันทร์)

..... กรรมการสอบและอาจารย์ที่ปรึกษา

(อาจารย์นุชนารถ พงษ์พานิช)

..... ประธานที่ปรึกษาสหกิจศึกษาสาขาวิชา IT

(อาจารย์อมรพันธ์ ชมกลีน)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย – ญี่ปุ่น

|                     |                                                                             |
|---------------------|-----------------------------------------------------------------------------|
| ชื่อโครงการ         | ระบบอนุมัติสินเชื่อเช่าซื้อรถยนต์และสินเชื่อ SME<br>CREDIT APPROVAL PROCESS |
| ผู้เขียน            | นางสาวสิริกาญจน์ กุลวุฒิวิลาศ                                               |
| คณะวิชา             | เทคโนโลยีสารสนเทศ สาขาวิชา เทคโนโลยีสารสนเทศ                                |
| อาจารย์ที่ปรึกษา    | อาจารย์นุชนารถ พงษ์พานิช                                                    |
| พนักงานที่ปรึกษา    | คุณอาทิตย์ เลิศยนต์ชีพ                                                      |
| ชื่อบริษัท          | บริษัท อีฟวแลนด์ จำกัด                                                      |
| ประเภทธุรกิจ/สินค้า | บริษัทผู้ให้เช่าปรึกษาทางด้านโซลูชันระบบสารสนเทศ                            |

### บทสรุป

จากการเข้ามาปฏิบัติงานในบริษัท อีฟวแลนด์ จำกัด ได้มีโอกาสมีส่วนร่วมในการทำงานในโปรเจกต์เกี่ยวกับระบบอนุมัติสินเชื่อเช่าซื้อรถยนต์ โดยเทคโนโลยีที่นำมาใช้นับเป็นเทคโนโลยีที่ค่อนข้างใหม่ ถือเป็นประสบการณ์การทำงานที่ดี ระบบนี้เป็นระบบที่มีกระบวนการทำงานรวดเร็ว ช่วยลดเวลาและขั้นตอนการทำงานให้น้อยลงแต่การทำงานมีประสิทธิภาพมากขึ้น

โดยได้ร่วมการพัฒนาระบบเป็นเวลา 4 เดือนตลอดการสหกิจศึกษา และเป็นเพียงส่วนหนึ่งของระบบเท่านั้น โดยในส่วนวิธีการและขั้นตอนการพัฒนาต่างๆนี้ ข้าพเจ้าได้กล่าวถึงเพียงส่วนชิ้นงานที่ข้าพเจ้าได้รับมอบหมาย ซึ่งสามารถนำไปใช้งานได้จริง

TNI

TAHITI - NICHI INSTITUTE OF TECHNOLOGY

## กิตติกรรมประกาศ

การที่ได้มีโอกาสมาสหกิจศึกษาที่ บริษัท อฟวาแลนด์ จำกัด นับเป็นโอกาสที่ดีที่ได้มาศึกษาหาความรู้ หาประสบการณ์ทางด้านไอที จากการทำงานเพื่อเข้าใจหลักการทำงานทางด้านไอที อาทิเช่น การเขียนโปรแกรม เตรียมตัวก่อนที่จะเข้าทำงานจริงในอนาคต การมาสหกิจศึกษาในครั้งนี้สามารถสำเร็จลุล่วงไปด้วยดี เนื่องจากความกรุณาจากบุคคลหลายๆท่านที่คอยดูแล ให้ความรู้ในเรื่องต่างๆ และให้คำปรึกษาในเรื่องต่างๆ รวมไปถึง บริษัท อฟวาแลนด์ จำกัด ที่มอบโอกาสให้เข้ามาศึกษาการทำงานภายในองค์กรและเรียนรู้ชีวิตในการทำงานจริง

ขอขอบพระคุณอาทิตย์ เลิศยนต์ชีพ และ พี่ๆทุกคนที่บริษัท อฟวาแลนด์ จำกัด ที่มีส่วนร่วมในการให้ความรู้และประสบการณ์ทางด้านไอที อันเป็นประโยชน์ต่อการมาสหกิจศึกษาในครั้งนี้ รวมไปถึงอาจารย์นุชนารถ พงษ์พานิช ซึ่งเป็นอาจารย์ที่ปรึกษาที่คอยให้การสนับสนุน ให้คำแนะนำและให้ความช่วยเหลือมาตลอดตั้งแต่ต้นจนจบสหกิจ

นางสาวสิริกาญจน์ กุลวุฒิวิลาส  
ผู้จัดทำ

## สารบัญ

## หน้า

|                                                                    |          |
|--------------------------------------------------------------------|----------|
| บทสรุป                                                             | ก        |
| กิตติกรรมประกาศ                                                    | ข        |
| สารบัญ                                                             | ค        |
| สารบัญรูป                                                          | ฉ        |
| สารบัญตาราง                                                        | ช        |
| <b>บทที่ 1 บทนำ</b>                                                | <b>1</b> |
| 1.1 ชื่อและที่ตั้งของสถานประกอบการ                                 | 1        |
| 1.1.1 ชื่อสถานประกอบการ                                            | 1        |
| 1.1.2 ที่ตั้งสถานประกอบการ                                         | 1        |
| 1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร     | 2        |
| 1.3 จัดองค์กรและการบริหารองค์กร                                    | 4        |
| 1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย                   | 4        |
| 1.5 พนักงานที่ปรึกษา และ ตำแหน่งของพนักงานที่ปรึกษา                | 4        |
| 1.6 ระยะเวลาที่ปฏิบัติงาน                                          | 5        |
| 1.7 ที่มาและความสำคัญของปัญหา                                      | 5        |
| 1.8 วัตถุประสงค์หรือจุดมุ่งหมายของโครงการ                          | 5        |
| 1.9 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย | 5        |
| 1.10 นิยามศัพท์เฉพาะ                                               | 6        |
| <b>บทที่ 2 ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน</b>              | <b>7</b> |
| 2.1 ทฤษฎีที่ใช้ในการปฏิบัติงาน                                     | 7        |
| 2.1.1 OOP: Object-Oriented Programming                             | 7        |
| 2.1.2 Hypertext markup language (HTML)                             | 7        |
| 2.1.3 Java                                                         | 9        |
| 2.1.4 JavaScript                                                   | 11       |

## สารบัญ(ต่อ)

### หน้า

|                                                                                                         |           |
|---------------------------------------------------------------------------------------------------------|-----------|
| 2.1.5 Typescript                                                                                        | 12        |
| 2.1.5 SQL                                                                                               | 13        |
| 2.2 เทคโนโลยีที่ใช้ในการปฏิบัติงาน                                                                      | 14        |
| 2.2.1 Angular2                                                                                          | 14        |
| 2.2.2 JavaScript Object Notation (JSON)                                                                 | 16        |
| 2.2.3 Spring Framework                                                                                  | 17        |
| 2.2.4 Hibernate Framework                                                                               | 18        |
| 2.2.5 โปรแกรม Visual Studio Code                                                                        | 20        |
| 2.2.6 โปรแกรม Rational Application Developer for WebSphere                                              | 20        |
| 2.2.7 NodeJS                                                                                            | 21        |
| 2.2.8 โปรแกรม Hoo WinTail                                                                               | 22        |
| 2.2.9 โปรแกรม Oracle SQL Developer                                                                      | 22        |
| 2.2.10 QMS System (Quality Management System)                                                           | 23        |
| <b>บทที่ 3 แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน</b>                                                | <b>25</b> |
| 3.1 แผนงานการฝึกงาน                                                                                     | 25        |
| 3.2 รายละเอียดที่นักศึกษาปฏิบัติในการฝึกงาน                                                             | 26        |
| 3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงาน                                                            | 26        |
| 3.3.1 ศึกษากระบวนการทำงานและเงื่อนไขต่างๆของชิ้นงานที่ได้รับมอบหมาย                                     | 26        |
| 3.3.3 วางแผน และประมาณเวลาที่ใช้ในการพัฒนา                                                              | 26        |
| 3.3.4 ลงมือพัฒนาโปรแกรมในส่วนหน้าจอตามเอกสารโปรโตไทป์ พร้อมทำการสร้างเส้นทางการเข้าถึงให้กับหน้าจออื่นๆ | 26        |
| 3.3.5 พัฒนาโปรแกรมในส่วนBackEnd การดึงข้อมูลจากฐานข้อมูลและฟังก์ชันต่างๆ                                | 27        |
| 3.3.6 ตรวจสอบและทดสอบระบบที่พัฒนา                                                                       | 27        |
| 3.3.7 อัปเดตไฟล์ขึ้น SVN Server                                                                         | 27        |
| 3.3.8 แก้ไขข้อผิดพลาดจาก Defect                                                                         | 27        |

## สารบัญ(ต่อ)

หน้า

|                                                                                                                       |           |
|-----------------------------------------------------------------------------------------------------------------------|-----------|
| <b>บทที่ 4 สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่างๆ</b>                                                          | <b>28</b> |
| 4.1 ขั้นตอนและผลการดำเนินงาน                                                                                          | 28        |
| 4.1.1 ศึกษากระบวนการทำงานและเงื่อนไขต่างๆของชิ้นงานที่ได้รับมอบหมายอย่างละเอียดจากเอกสาร SRS ตามแต่ละส่วนงาน          | 28        |
| 4.1.2 ศึกษาโครงสร้าง และส่วนฐานข้อมูลที่นำมาใช้ของระบบ                                                                | 28        |
| 4.1.3 ลงมือพัฒนาโปรแกรมในส่วนหน้าจอตามเอกสารโปรโตไทป์ พร้อมทำการสร้างเส้นทางการเข้าถึงให้กับหน้าจออื่นๆ               | 29        |
| 4.1.4 พัฒนาโปรแกรมในส่วน BackEnd การดึงข้อมูลจากฐานข้อมูล และฟังก์ชันต่างๆ                                            | 32        |
| 4.1.5 อัปเดตไฟล์ขึ้น SVN Server                                                                                       | 38        |
| 4.1.6 แก้ไขข้อผิดพลาดจาก Defect                                                                                       | 39        |
| 4.2 ผลการวิเคราะห์ข้อมูล                                                                                              | 40        |
| 4.3 วิเคราะห์และวิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายในการปฏิบัติงานหรือการจัดทำโครงการ | 40        |
| <b>บทที่ 5 บทสรุปและข้อเสนอแนะ</b>                                                                                    | <b>42</b> |
| 5.1 สรุปผลการดำเนินงาน                                                                                                | 42        |
| 5.2 แนวทางการแก้ไขปัญหา                                                                                               | 42        |
| 5.3 ข้อเสนอแนะจากการดำเนินงาน                                                                                         | 42        |
| เอกสารอ้างอิง                                                                                                         | 44        |
| ประวัติผู้จัดทำโครงการ                                                                                                | 46        |

## สารบัญภาพประกอบ

| ภาพ                                                                                | หน้า |
|------------------------------------------------------------------------------------|------|
| ภาพที่ 1.1 โลโก้บริษัท อฟวาแลนด์ จำกัด.....                                        | 1    |
| ภาพที่ 2.2 รูปภาพโลโก้ของ Angular2.....                                            | 15   |
| ภาพที่ 2.3 ภาพสถาปัตยกรรมของ Angular2.....                                         | 16   |
| ภาพที่ 2.4 ตัวอย่าง Code ที่ใช้ Format JSON.....                                   | 17   |
| ภาพที่ 2.5 รูปภาพโลโก้ของ Spring Framework.....                                    | 17   |
| ภาพที่ 2.6 ภาพโครงสร้างของ Spring Framework.....                                   | 18   |
| ภาพที่ 2.7 ภาพโลโก้ของ Hibernate.....                                              | 19   |
| ภาพที่ 2.8 โครงสร้างของ Hibernate.....                                             | 19   |
| ภาพที่ 2.9 ตัวอย่างหน้าต่างภายในโปรแกรม Visual Studio Code.....                    | 20   |
| ภาพที่ 2.10 รูปหน้าต่างโปรแกรมตอนเริ่มเข้าใช้งาน.....                              | 21   |
| ภาพที่ 2.11 รูปตัวอย่างหน้าต่างภายในโปรแกรม.....                                   | 21   |
| ภาพที่ 4.2 ภาพตัวอย่างโครงสร้างตารางจากฐานข้อมูลบางส่วน.....                       | 29   |
| ภาพที่ 4.3 ภาพตัวอย่างจากเอกสารโปรโตไทป์บางส่วน.....                               | 29   |
| ภาพที่ 4.4 ตัวอย่างโค้ด HTML ที่เขียนผ่านโปรแกรม Visual Studio Code.....           | 30   |
| ภาพที่ 4.5 ตัวอย่างโค้ด TypeScript ที่เขียนจากโปรแกรม Visual Studio Code.....      | 30   |
| ภาพที่ 4.6 ตัวอย่างโค้ด TypeScript ที่เขียนจากโปรแกรม Visual Studio Code.....      | 31   |
| ภาพที่ 4.7 ตัวอย่างโค้ด TypeScript ที่เขียนจากโปรแกรม Visual Studio Code.....      | 31   |
| ภาพที่ 4.8 ภาพตัวอย่างการเรียกดูงานผ่านเส้นทางการเข้าถึงที่ได้กำหนดไว้.....        | 32   |
| ภาพที่ 4.9 ภาพหน้าจอที่ยังไม่มีฟังก์ชันการทำงานใดๆที่แสดงผลทางเว็บเบราว์เซอร์..... | 32   |
| ภาพที่ 4.10 ตัวอย่างโค้ดในไฟล์โมเดลส่วน Backend.....                               | 32   |
| ภาพที่ 4.11 ตัวอย่างโค้ดในไฟล์ DAO.....                                            | 33   |
| ภาพที่ 4.12 ตัวอย่างโค้ดในไฟล์ Service ส่วน Backend.....                           | 33   |
| ภาพที่ 4.13 ตัวอย่างโค้ดในไฟล์ Controller.....                                     | 34   |
| ภาพที่ 4.14 ตัวอย่างโค้ดในไฟล์ Service ส่วน FrontEnd.....                          | 34   |
| ภาพที่ 4.15 ตัวอย่างโค้ดในไฟล์โมเดล ส่วน FrontEnd.....                             | 35   |

## สารบัญภาพประกอบ(ต่อ)

ภาพ

หน้า

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| ภาพที่ 4.16 ตัวอย่างโค้ดในไฟล์ Service ส่วน FrontEnd .....                | 35 |
| ภาพที่ 4.17 ตัวอย่างการใส่โมเดลเพื่อบันทึกหรือเรียกแสดงจากฐานข้อมูล.....  | 36 |
| ภาพที่ 4.18 ตัวอย่างการกำหนดมาสเตอร์ .....                                | 36 |
| ภาพที่ 4.19 ตัวอย่างการเรียกใช้มาสเตอร์.....                              | 36 |
| ภาพที่ 4.20 ผลลัพธ์การเรียกใช้มาสเตอร์.....                               | 37 |
| ภาพที่ 4.21 ภาพหน้าจอการทำงานอย่างสมบูรณ์ที่แสดงผลทางเว็บเบราว์เซอร์..... | 37 |
| ภาพที่ 4.22 ภาพโครงสร้างสรุปหลักการทำงาน .....                            | 37 |
| ภาพที่ 4.23 ขั้นตอนการคอมมิตไฟล์ขึ้นไปยัง SVN Server(1).....              | 38 |
| ภาพที่ 4.24 ขั้นตอนการคอมมิตไฟล์ขึ้นไปยัง SVN Server(2).....              | 38 |
| ภาพที่ 4.25 ตัวอย่างดีเฟกต์ที่ได้รับมอบหมาย .....                         | 39 |
| ภาพที่ 4.26 ตัวอย่างรายละเอียดของดีเฟกต์ที่ได้รับมอบหมาย .....            | 40 |

TNI

TAHT - NICHI INSTITUTE OF TECHNOLOGY

## สารบัญตาราง

ตาราง

หน้า

|                                                    |    |
|----------------------------------------------------|----|
| ตารางที่ 3.1 แผนการปฏิบัติงานระยะเวลา 4 เดือน..... | 25 |
|----------------------------------------------------|----|



## บทที่ 1

### บทนำ

#### 1.1 ชื่อและที่ตั้งของสถานประกอบการ



ภาพที่ 1.1 โลโก้บริษัท อฟวาแลนท์ จำกัด

##### 1.1.1 ชื่อสถานประกอบการ

ชื่อภาษาไทย : บริษัท อฟวาแลนท์ จำกัด

ชื่อภาษาอังกฤษ : Avalant Company Limited

##### 1.1.2 ที่ตั้งสถานประกอบการ

ที่อยู่ : 20 ถนนสาทรเหนือ ชั้น 15 อาคารนูปผิจิต แขวงสีลม เขตบางรัก

กรุงเทพฯ 10500 ประเทศไทย

โทรศัพท์ : +66-2-633-9367-9

โทรสาร : +66-2-633-8174

อีเมล : [mkt-avalant@avalant.co.th](mailto:mkt-avalant@avalant.co.th)

เว็บไซต์ : <http://www.avalant.co.th>



ภาพที่ 1.2 แผนที่ตั้งบริษัท อฟวาแลนท์ จำกัด

ที่มา : <http://www.avalant.co.th>

## 1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร

บริษัท อฟวาแลนท์ จำกัด (Avalant Company Limited) เป็นบริษัทผู้ให้คำปรึกษาทางด้านโซลูชันระบบสารสนเทศ (Information Technology) เพื่อตอบสนองการดำเนินงาน ต่างๆภายในองค์กร โดยมีวัตถุประสงค์เพื่อช่วยให้การดำเนินงานต่างๆภายในองค์กรมีประสิทธิภาพและประสิทธิผลที่ดีขึ้นส่งผลสืบเนื่องถึงผลกำไรที่มากขึ้นขององค์กร

บริษัทฯ มีความเชี่ยวชาญในธุรกิจหลากหลายประเภททั้งในด้านธุรกิจประกันภัย ธุรกิจการเงิน ธุรกิจค้าปลีก รวมไปถึงธุรกิจทางด้านโทรคมนาคม โดยการให้บริการนั้นครอบคลุมตั้งแต่การวิเคราะห์และออกแบบกระบวนการทางธุรกิจ การวางแผนเชิงกลยุทธ์ การบริหารโครงการ การวิเคราะห์ความต้องการ การออกแบบและพัฒนาความต้องการ การตรวจสอบและรับประกันคุณภาพ การพัฒนาโครงสร้างพื้นฐานที่สำคัญ (Infrastructure) และการออกแบบโซลูชันที่เหมาะสม (Solution Architect) ให้แก่องค์กร

### วิสัยทัศน์ (Vision)

บริษัทฯ ต้องการเป็นบริษัทชั้นนำในการให้คำปรึกษาทางด้านโซลูชันระบบสารสนเทศ (Information Technology) ในระดับภูมิภาคเอเชีย

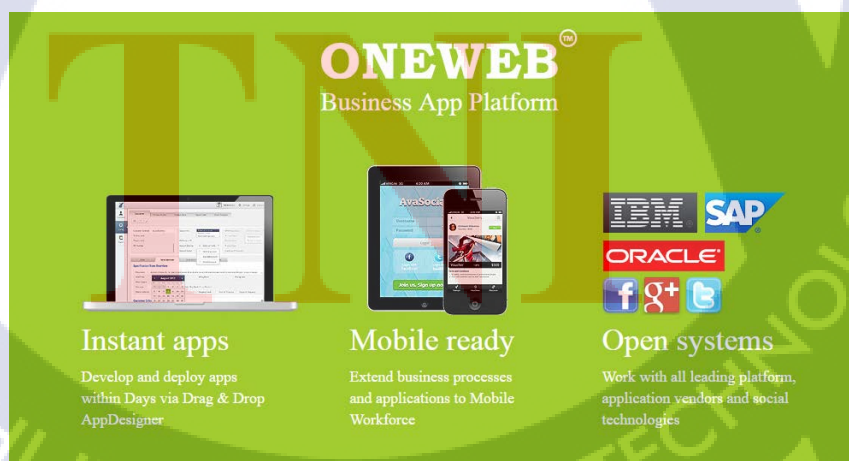
### พันธกิจ (Mission)

เพื่อช่วยเหลือลูกค้าของเราออกแบบโซลูชันทั้งทางด้านธุรกิจและด้านระบบสารสนเทศ (Information Technology) เพื่อให้เกิดผลตอบแทนจากการลงทุนผ่านผลกำไรและผลการดำเนินงานที่ดีขึ้น

### ผลิตภัณฑ์และการบริการ

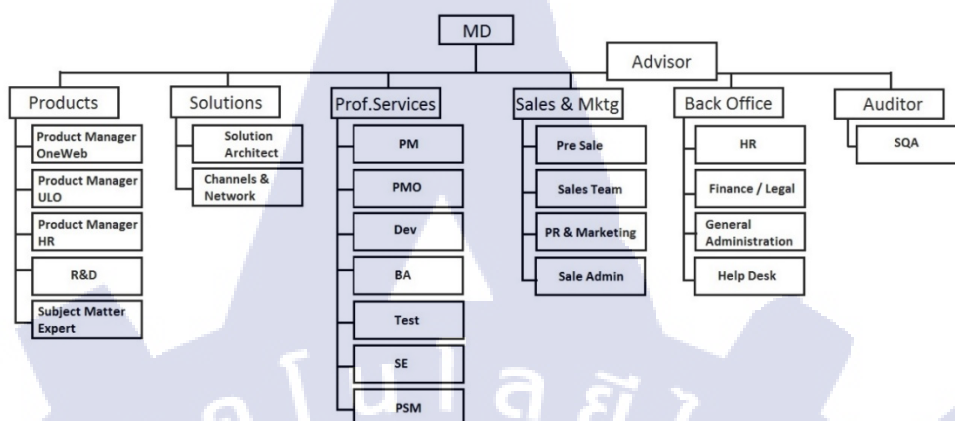
บริษัทมีการวางแผนและจัดการให้ระบบสารสนเทศสอดคล้องกับทางธุรกิจ โดยบริษัทได้ทำงานร่วมกับผู้บริหารในหลายประเภทธุรกิจโดยเฉพาะธุรกิจประกันภัยและธุรกิจการเงินเพื่อส่งมอบกระบวนการดำเนินงานและเทคโนโลยีเพื่อปรับปรุงพัฒนากระบวนการทางธุรกิจและสนับสนุนทิศทางการกลยุทธ์ธุรกิจขององค์กร บริษัทฯ เชื่อมมั่นในแนวทางที่เน้นการปฏิบัติโดยให้ความสำคัญกับความต้องการทางด้านธุรกิจของลูกค้า ก่อนที่จะพัฒนาโซลูชันเพื่อตอบสนองความต้องการของลูกค้า ดังนั้นบริษัทฯ จึงเน้นการทำงานโดยเริ่มที่การสื่อสารกับผู้บริหารและผู้ที่เกี่ยวข้องในองค์กรของลูกค้าเพื่อที่จะกำหนดเป้าหมายในการพัฒนากระบวนการทางธุรกิจ (Business Process Improvement: BPI) ซึ่งวิธีการดังกล่าวจะส่งผลให้สามารถวัดผลตอบแทนจากการลงทุนได้จริง

บริษัท อีฟวแอลท์ จำกัด นั้นเน้นการทำผลิตภัณฑ์ที่ตอบโจทย์ให้แก่ธุรกิจ โดยเน้นแนวความคิดในเรื่องของ “One web” ซึ่งในแนวความคิดนี้จะทำให้ผลิตภัณฑ์ที่จะส่งมอบให้แก่ธุรกิจสามารถนำไปใช้งานในทุก Platform



ภาพที่ 1.3 แสดงผลิตภัณฑ์ของทางบริษัท

### 1.3 จัดองค์กรและการบริหารองค์กร



ภาพที่ 1.4 แสดงผลิตภัณฑ์ของทางบริษัท

บริษัทมีโครงสร้างองค์กรเป็นแบบการแบ่งตามสายงานด้านต่างๆ โดยในแต่ละสายงานจะมีหัวหน้าใหญ่ในแต่ละสายงานนั้น ซึ่งหัวหน้าในแต่ละสายงานจะมีการรายงานผลต่างๆ ให้กับ MD (Managing Director) ได้โดยตรง

### 1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่ง : Internship

แผนกที่เข้ารับการฝึก : Java Developer

หน้าที่งานที่ได้รับมอบหมาย : พัฒนาระบบ Web Application ในส่วน Frontend และ Backend

### 1.5 พนักงานที่ปรึกษา และ ตำแหน่งของพนักงานที่ปรึกษา

พนักงานที่ปรึกษา: นายอาทิตย์ เลิศยนต์ชีพ

ตำแหน่ง : Manager

แผนก: Professional Service

อีเมล : arthit.l@avalant.co.th

เบอร์โทร : 086-345-0055

## 1.6 ระยะเวลาที่ปฏิบัติงาน

เริ่มต้นปฏิบัติงานสหกิจศึกษาตั้งแต่วันที่ 30 พฤษภาคม พ.ศ.2559

สิ้นสุดภาคปฏิบัติงานวันที่ 30 กันยายน พ.ศ. 2559

รวมทั้งสิ้นเป็นระยะเวลา 4 เดือน

## 1.7 ที่มาและความสำคัญของปัญหา

ในปัจจุบันในกระบวนการอนุมัติสินเชื่อภายในธนาคารหลายๆแห่ง การที่จะสามารถอนุมัติสินเชื่อให้ผ่านออกมาได้นั้น จะต้องผ่านกระบวนการอนุมัติหลายขั้นตอน โดยผ่านหลายๆบุคคล จึงทำให้ข้อมูลที่เกิดขึ้นมีความซับซ้อนมีการนำไปใช้ในหลายๆส่วน อีกทั้งยังเป็นปัจจัยสำหรับการอนุมัติสินเชื่อในแต่ละกรณี ซึ่งเป็นผลให้ในการอนุมัติสินเชื่อในแต่ละครั้งนั้นใช้เวลาในการดำเนินการพอสมควร

จากปัญหาด้านความล่าช้า และความซับซ้อนของข้อมูลในกระบวนการอนุมัติสินเชื่อที่เกิดขึ้น เพื่อสนับสนุนกระบวนการดังกล่าว ทางบริษัทได้มีการพัฒนาเว็บแอปพลิเคชัน สำหรับสนับสนุนกระบวนการอนุมัติสินเชื่อให้มีความง่ายขึ้น ซึ่งข้าพเจ้าผู้ซึ่งจัดทำรายงานฉบับนี้ ได้เป็นส่วนหนึ่งในการพัฒนาเว็บแอปพลิเคชันนี้ โดยภายในรายงานฉบับนี้จะอธิบายในส่วนของการดำเนินการที่ข้าพเจ้าซึ่งเป็นผู้จัดทำได้รับมอบหมาย จะอธิบายขั้นตอนรวมถึงโครงสร้างของระบบที่นำมาใช้ในการพัฒนาเว็บแอปพลิเคชันนี้ขึ้นมา

## 1.8 วัตถุประสงค์หรือจุดมุ่งหมายของโครงการ

1. เพื่อแก้ปัญหาด้านความล่าช้าของกระบวนการอนุมัติสินเชื่อ ให้สามารถทำงานได้รวดเร็วขึ้น
2. เพื่อจัดการข้อมูลที่มีความซับซ้อนในกระบวนการอนุมัติสินเชื่อ ให้สามารถจัดการ และเข้าใจได้ง่ายขึ้น

## 1.9 ผลที่คาดว่าจะได้รับการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

1. สามารถเข้าใจในกระบวนการอนุมัติสินเชื่อมากขึ้น และเข้าใจได้อย่างถูกต้อง
2. ได้ฝึกฝนการดำเนินงานอย่างเป็นระบบ และทำงานเป็นทีมได้
3. ฝึกความรับผิดชอบต่องาน และความตรงต่อเวลาในงานที่ได้รับมอบหมายนั้นๆ
4. ได้ฝึกฝนทักษะในด้านการเขียนโปรแกรม และด้านธุรกิจไปพร้อมๆกัน

### 1.10 นิยามศัพท์เฉพาะ

1. Credit Approval Process (CAP) หมายถึง กระบวนการอนุมัติสินเชื่อซึ่งเป็นตัวควบคุมการตรวจสอบ และดำเนินการเป็นกิจกรรมที่แยกจากกิจกรรมการขาย
2. Web Application หมายถึง โปรแกรมประยุกต์ที่เข้าถึงด้วยโปรแกรมค้นดูเว็บผ่านเครือข่ายคอมพิวเตอร์
3. Frontend หมายถึง ส่วนหน้าจอที่แสดงผล สำหรับติดต่อกับผู้ใช้
4. Backend หมายถึง ส่วนที่มีหน้าที่ติดต่อกับฐานข้อมูลและประมวลผลข้อมูล
5. System Requirements specification (SRS) หมายถึง เอกสารข้อกำหนดความต้องการทางซอฟต์แวร์
6. Prototype หมายถึง เอกสารที่สร้างขึ้นเป็นแบบฉบับเพื่อสร้างชิ้นงานให้มีลักษณะเดียวกัน มีรูปทรง ขนาดสัดส่วน โครงสร้างที่แสดงรายละเอียด สามารถใช้งานได้ตามที่ออกแบบไว้ทุกประการ

## บทที่ 2

### ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

ในการปฏิบัติงานสหกิจศึกษาครั้งนี้ เป็นการนำความรู้ทางด้านทฤษฎีและเทคโนโลยีมาใช้ในการปฏิบัติงานทุกส่วนตลอดการปฏิบัติงานสหกิจศึกษา ซึ่งเป็นการนำความรู้ทั้งที่เคยเรียนมาประยุกต์ใช้และเป็นการศึกษาเรียนรู้สิ่งใหม่ ๆ ที่ได้จากการปฏิบัติงาน

#### 2.1 ทฤษฎีที่ใช้ในการปฏิบัติงาน

##### 2.1.1 OOP: Object-Oriented Programming

OOP คือแนวคิดการมองหน่วยต่างๆ ของภาษาโปรแกรม (Entity) ให้เป็นวัตถุ (Object) ดังเช่นที่มนุษย์บุคลธรรมดาตามองสิ่งต่างๆ ในชีวิตประจำวันเป็นวัตถุ

ทุก Object จะประกอบไปด้วย 2 ส่วนคือ

- 1.Attribute (Object Data) – เป็นส่วนที่ใช้ในการเก็บข้อมูลต่างๆของ Object เช่น รถมี attribute คือยี่ห้อรถ ชื่อรุ่น และสี เป็นต้น
- 2.Method (Object behavior) – เป็นสิ่งที่ Object นั้นสามารถทำได้ เช่น รถสามารถเพิ่มความเร็วได้ หรือเปลี่ยนเกียร์ได้ เป็นต้น

##### 2.1.2 Hypertext markup language (HTML)

HTML คือ ภาษาหลักที่ใช้ในการเขียนเว็บเพจ โดยใช้ Tag ในการกำหนดการแสดงผล HTML ย่อมาจากคำว่า Hypertext markup language โดย Hypertext หมายถึง ข้อความที่เชื่อมต่อกันผ่านลิงค์ (Hyperlink) Markup language หมายถึงภาษาที่ใช้ Tag ในการกำหนดการแสดงผลสิ่งต่างๆ ที่แสดงอยู่บนเว็บเพจ ดังนั้น HTML จึงหมายถึง ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่ต่างก็เชื่อมถึงกัน ใน Hyperspace ผ่าน Hyperlink โดยปัจจุบัน HTML ถูกพัฒนามาถึงเวอร์ชันที่ 5 ซึ่งถูกเรียกว่า HTML5



ภาพที่ 2.1 รูปภาพโลโก้ HTML5

### 2.1.2.1 โครงสร้างของภาษา HTML

#### 1. คำสั่งเริ่มต้น

รูปแบบคำสั่ง `<HTML>...</HTML>`

คำสั่ง `<HTML>` เป็นคำสั่งเริ่มต้นในการเขียนเว็บเพจด้วยภาษา HTML และเมื่อสิ้นสุดโปรแกรมจะใช้ `</HTML>`

#### 2. ส่วนหัวของโปรแกรม

รูปแบบคำสั่ง `<HEAD> ...</HEAD>`

คำสั่ง `<HEAD> ...</HEAD>` ภายในจะมีคำสั่งย่อยอยู่หลายคำสั่ง ได้แก่ `<TITLE>` คำสั่ง ปิดท้ายก็คือ `</TITLE>`, `<META` คำสั่งปิดท้ายคือ `/>`, `<LINK` คำสั่งปิดท้ายคือ `/>`

#### 3. กำหนดข้อความในส่วนของ Title bar

รูปแบบคำสั่ง `<TITLE> ...</TITLE>`

คำสั่ง `<TITLE>` เป็นคำสั่งที่ใช้กำหนดข้อความที่จะแสดงในส่วนของ Title Bar และจะปิดท้ายด้วย `</TITLE>`

#### 4. ส่วนของเนื้อหา

รูปแบบคำสั่ง <BODY> ...</BODY>

คำสั่ง <BODY> เป็นส่วนที่มีไว้สำหรับเขียนคำสั่งหรือข้อความต่าง ๆ เพื่อแสดงผลตามที่ เราต้องการไปยัง Browser และจะปิดท้ายโปรแกรมด้วย</BODY>

จากรายละเอียดข้างต้นโครงสร้างของภาษา html มีลักษณะดังนี้

```
<html>
<head>
<title>ข้อความที่ต้องการให้แสดงบนไตเติลบาร์</title>
</head>
<body>
    คำสั่งและข้อความให้แสดงบน Browser
</body>
</html>
```

### 2.1.3 Java

Java หรือ Java programming language คือ ภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นไม้โอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตนด้วย Java ได้

Java เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ ( OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้น คลาส คือ ที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

### ข้อดีของ ภาษา Java

1. ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุจะช่วยให้เราสามารถใช้คำหรือชื่อ ต่าง ๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่ายขึ้น
2. โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็ยังสามารถถูก compile และ run บนเครื่องพีซีธรรมดาได้
3. ภาษาจาวามีการตรวจสอบข้อผิดพลาดทั้งตอน compile time และ runtime ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และช่วยให้ debug โปรแกรมได้ง่าย
4. ภาษาจาวามีความซับซ้อนน้อยกว่าภาษา C++ เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java จะมีจำนวน code น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ทำให้ใช้งานได้ง่ายกว่าและลดความผิดพลาดได้มากขึ้น
5. ภาษาจาวาถูกออกแบบมาให้มีความปลอดภัยสูงตั้งแต่แรก ทำให้โปรแกรมที่เขียนขึ้นด้วยจาวามีความปลอดภัยมากกว่าโปรแกรมที่เขียนขึ้น ด้วยภาษาอื่น เพราะ Java มี security ทั้ง low level และ high level ได้แก่ electronic signature, public and private key management, access control
6. มี IDE, application server, และ library ต่าง ๆ มากมายสำหรับจาวาที่เราสามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย ทำให้เราสามารถลดค่าใช้จ่ายที่ต้องเสียไปกับการซื้อ tool และ s/w ต่าง ๆ

### ข้อเสียของ ภาษา Java

1. ทำงานได้ช้ากว่า native code (โปรแกรมที่ compile ให้อยู่ในรูปของภาษาเครื่อง) หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น อย่างเช่น C หรือ C++ ทั้งนี้ก็เพราะว่าโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาจะถูกแปลงเป็นภาษากลางก่อน แล้วเมื่อโปรแกรมทำงานคำสั่งของภาษากลางนี้จะถูกเปลี่ยนเป็นภาษาเครื่องอีกทีหนึ่ง ทีละคำสั่ง (หรือกลุ่มของคำสั่ง) ณ runtime ทำให้ทำงานช้ากว่า native code ซึ่งอยู่ในรูปของภาษาเครื่องแล้วตั้งแต่ compile โปรแกรมที่ต้องการความเร็วในการทำงานจึงไม่นิยมเขียนด้วยจาวา
2. Tool ที่มีในการใช้พัฒนาโปรแกรมจาวามักไม่ค่อยเก่ง ทำให้หลายอย่างโปรแกรมเมอร์จะต้องเป็นคนทำเอง ทำให้ต้องเสียเวลาทำงานในส่วนที่ tool ทำไม่ได้ ถ้าเราดู tool ของ MS จะใช้

งานได้ง่ายกว่า และพัฒนาได้เร็วกว่า (แต่เราต้องซื้อ tool ของ MS และก็ต้องรันบน platform ของ MS)

#### 2.1.4 JavaScript

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง Java JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียนเตด (Object Oriented Programming) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server)

JavaScript ถูกพัฒนาขึ้นโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปจึงได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี 2538 แล้วตั้งชื่อใหม่ว่า JavaScript ที่สามารถทำให้ การสร้างเว็บเพจ มีลูกเล่น ต่าง ๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือ การกรอกข้อความในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ตาม สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ๆ ออกมาด้วย (ปัจจุบันคือรุ่น 1.5) ดังนั้น ถ้านำโค้ดของเวอร์ชันใหม่ ไปรันบนบราวเซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิด error ได้

### ความสามารถของ JavaScript

1. JavaScript ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายๆได้ โดยไม่ต้องพึ่งภาษาอื่น
2. JavaScript มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่น เมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox ก็สามารถสั่งให้เปิดหน้าต่างใหม่ได้ ทำให้เว็บไซต์ของเรามีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น นี่คือข้อดีของ JavaScript เลยก็ว่าได้ที่ทำให้เว็บไซต์ดังๆทั้งหลายหันมาใช้
3. JavaScript สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้ นั่นคือ สามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้ หรือหน้าแสดงเนื้อหาสามารถซ่อนหรือแสดงเนื้อหาได้แบบง่ายๆนั่นเอง
4. JavaScript สามารถใช้ตรวจสอบข้อมูลได้ สังเกตว่าเมื่อเรากรอกข้อมูลบางเว็บไซต์ เช่น Email เมื่อเรากรอกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่าเรากรอกผิด หรือลืมกรอกอะไรบางอย่าง เป็นต้น
5. JavaScript สามารถใช้ในการตรวจสอบผู้ใช้ได้ เช่น ตรวจสอบว่าผู้ใช้ ใช้ web browser อะไร
6. JavaScript สร้าง Cookies (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

### ข้อดีและข้อเสียของ JavaScript

การทำงานของ JavaScript เกิดขึ้นบนเบราว์เซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้เบราว์เซอร์อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (เรียกว่า server-side script) ดังนั้นจึงต้องใช้บนเซิร์ฟเวอร์ ที่สนับสนุนภาษาเหล่านี้เท่านั้น อย่างไรก็ตาม จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือ ไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา server-side script อยู่

#### 2.1.5 Typescript

Typescript คือ ภาษาโปรแกรมที่ทำงานได้เสมือน Javascript เวอร์ชัน ES6 โดยมีสิ่งที่เพิ่มขึ้นมาคือ สนับสนุน Type System รวมถึงคุณสมบัติอื่นๆที่เพิ่มขึ้น เช่น Enum และความสามารถที่เพิ่มขึ้นของการโปรแกรมเชิงวัตถุ Typescript นั้นเป็น Transpiler ซึ่งนั่นคือ Typescript จะแปลงโปรแกรมที่เขียนขึ้นมาเป็นภาษา Javascript อีกครั้ง

### ข้อดีของการใช้ TypeScript

1. TypeScript ทำให้คุณใช้ JavaScript สมัยใหม่ได้ในปัจจุบัน ความสามารถของ ES2015 และอื่นๆ ได้รวมไว้แล้วใน TypeScript
2. ตัวแปรที่คุณประกาศแล้วใน TypeScript จะเปลี่ยนชนิดข้อมูลไม่ได้อีกต่อไป ข้อผิดพลาดในโปรแกรมของคุณจะน้อยลงเพราะคุณไม่มีโอกาสพลาดในการใส่ข้อมูลผิดชนิดเป็นแน่
3. TypeScript มีการตรวจสอบโค้ดในช่วง compile time ทำให้คุณดักจับข้อผิดพลาดได้แต่ต้นไม่ปล่อยให้ข้อผิดพลาดไปโผล่ในตอนทำงานจริง (runtime)
4. IDE และ Text Editor ที่ดีเยี่ยมสนับสนุนให้คุณใช้งาน TypeScript ได้อย่างสมบูรณ์

### ข้อควรระวังทั่วไปเกี่ยวกับ TypeScript

1. โค้ด JavaScript ธรรมดาก็คือโค้ด TypeScript ด้วยเช่นกัน
2. เราสามารถระบุหรือไม่ระบุชนิดข้อมูลใน TypeScript ก็ได้ แต่เมื่อประกาศตัวแปรแล้วเราจะเปลี่ยนชนิดข้อมูลไม่ได้
3. แม้เราจะใส่ค่าตัวแปรผิดชนิดข้อมูลใน TypeScript แต่ TypeScript ก็ยังใจดีแปลงโค้ดเป็น JavaScript ให้อยู่ดี

### 2.1.5 SQL

SQL ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) หมายถึงเราสามารถใช้คำสั่ง SQL กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อสั่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดขัดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว SQL ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่ง ซึ่งแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้

1. Select query ใช้สำหรับดึงข้อมูลที่ต้องการ
2. Update query ใช้สำหรับแก้ไขข้อมูล
3. Insert query ใช้สำหรับการเพิ่มข้อมูล
4. Delete query ใช้สำหรับลบข้อมูลออกไป

ปัจจุบันมีซอฟต์แวร์ระบบจัดการฐานข้อมูล (DBMS) ที่สนับสนุนการใช้คำสั่ง SQL เช่น Oracle , DB2, MS-SQL, MS-Access นอกจากนี้ภาษา SQL ถูกนำมาใช้เขียนร่วมกับโปรแกรมภาษาต่างๆ เช่น ภาษา C/C++ , Visual Basic และ Java

### ประโยชน์ของภาษา SQL

1. สร้างฐานข้อมูลและ ตาราง
2. สนับสนุนการจัดการฐานข้อมูล ซึ่งประกอบด้วย การเพิ่ม การปรับปรุง และการลบข้อมูล
3. สนับสนุนการเรียกใช้หรือ ค้นหาข้อมูล

### ประเภทของคำสั่งภาษา SQL

1. ภาษานิยามข้อมูล(Data Definition Language : DDL) เป็นคำสั่งที่ใช้ในการสร้างฐานข้อมูล กำหนดโครงสร้างข้อมูลว่ามี Attribute ไດ

ชนิดของข้อมูล รวมทั้งการเปลี่ยนแปลงตาราง และการสร้างดัชนี คำสั่ง :

CREATE,DROP,ALTER

2. ภาษาจัดการข้อมูล (Data Manipulation Language: DML) เป็นคำสั่งที่ใช้ในการเรียกใช้ เพิ่ม ลบ และเปลี่ยนแปลงข้อมูลในตาราง คำสั่ง: SELECT, INSERT, UPDATE, DELETE

3. ภาษาควบคุมข้อมูล (Data Control Language: DCL) เป็นคำสั่งที่ใช้ในการกำหนดสิทธิการ อนุญาต หรือ ยกเลิก การเข้าถึงฐานข้อมูล เพื่อป้องกันความปลอดภัยของฐานข้อมูล คำสั่ง:

GRANT, REVOKE

## 2.2 เทคโนโลยีที่ใช้ในการปฏิบัติงาน

### 2.2.1 Angular2

Angular2 คือ เฟรมเวิร์กการพัฒนาแอปพลิเคชันบนมือถือ เดสก์ท็อป และเว็บ โดยรันบนพื้นฐานภาษา Typescript ซึ่งมีความคล้ายคลึงกับภาษา Javascript และ Angular2 ทำงานได้หลายแพลตฟอร์ม



## ภาพที่ 2.1 รูปภาพโลโก้ของ Angular2

### สถาปัตยกรรมแอปพลิเคชันของ Angular2

**โมดูล** คือ แนวคิดการรวมข้อมูลเป็นกลุ่ม เหมือนการปะป้ายใส่กล่องไว้เพื่อแยกประเภท เพื่อให้ง่ายต่อการเข้าถึง และ ดูแลรักษาได้ง่ายขึ้น หรือสรุปง่ายๆ ก็คือกลุ่มข้อมูลที่มีความเกี่ยวข้องกัน

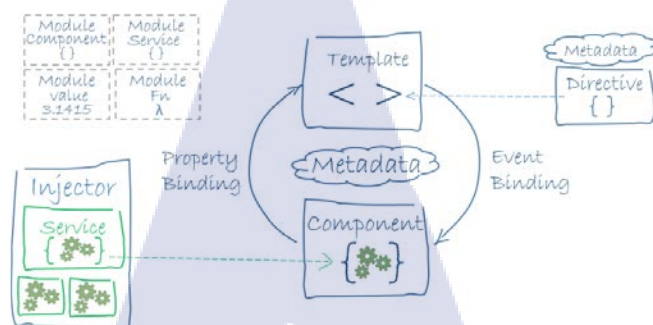
**ไลบรารี** คือ โมดูลที่มีความสามารถพิเศษเพื่อไว้ให้โมดูลอื่นๆ ใช้งาน หรือก็คือเครื่องมือที่ทำให้สามารถสร้างความสามารถต่างๆ แอปพลิเคชันนั่นเอง

**คอมโพเนน** คือ สิ่งที่ควบคุมการแสดงผลทางหน้าจอโดยส่วนนี้จะสัมพันธ์กับส่วน เติม เพลต และ เซอร์วิส

**เท็มเพลต** คือ ตัวที่แสดงผลเองเลยโดยมี คอมโพเนนควบคุม

**เมตาดาต้า** คือ สิ่งที่ใช้บอกอธิบายคลาสว่ามีการตั้งค่าอย่างไรบ้าง และทำอะไรบ้าง

**เซอร์วิส** คือ ส่วนที่เป็นฟังก์ชัน เกิดขึ้นมาเพื่อทำงานเฉพาะทาง เช่น การดึงข้อมูลจากเซิร์ฟเวอร์ เพื่อมาแสดงผลเป็นต้น



ภาพที่ 2.2 ภาพสถาปัตยกรรมของ Angular2

### 2.2.2 JavaScript Object Notation (JSON)

JavaScript Object Notation หรือ JSON เป็นฟอร์แมตสำหรับแลกเปลี่ยนข้อมูลคอมพิวเตอร์ ฟอร์แมต JSON นั้นอยู่ในรูปข้อความธรรมดา (Plain Text) ที่ทั้งมนุษย์และโปรแกรมคอมพิวเตอร์สามารถอ่านเข้าใจได้ ซึ่งปัจจุบันนิยมใช้ใน Web Application

#### โครงสร้างของฟอร์แมต

เจสัน เป็นรูปแบบสายอักขระ (String) ชนิดหนึ่งที่ถูกจัดเก็บในรูปแบบที่สามารถอ่านและเข้าใจได้ง่าย ถ้าจะให้เข้าใจง่าย ๆ ก็คือเจสันเปรียบเสมือนรูปแบบของ อาร์เรย์ (Array) ชนิดหนึ่งที่ใช้รับส่งข้อมูลผ่านอแเจ็กส์เพราะซึ่งปกติแล้วถ้าเราต้องการรับ-ส่งข้อมูลผ่านอแเจ็กส์ต่าง ๆ นั้นจะต้องรับ-ส่งมาในรูปแบบของสายอักขระทั้งก่อน และเมื่อฝั่งอแเจ็กส์ทำการรับค่าที่ทำการส่งค่ากลับมาจากเซิร์ฟเวอร์ ก็จะต้องนำสายอักขระ เหล่านั้นมาตัดตำแหน่งที่ต้องการ เพื่อเอาสายอักขระตัวที่ต้องการมาใช้ แต่สำหรับเจสันแล้ว สามารถรับส่งชุดค่าตัวแปรได้ทั้งฝั่งไคลเอนต์ (Client) และฝั่งเซิร์ฟเวอร์ (Server) โดยทั้ง 2 ฝั่งสามารถทำการเข้ารหัสและถอดรหัสโดยใช้เจสันเอนโค้ด (Json Encode) และ เจสันดีโค้ด (Json Decode) เพื่ออ่านค่าตัวแปรเหล่านั้น และจะเรียกใช้งานมันได้อย่างไร ซึ่งปกติแล้วจะอยู่ในรูปแบบของอาร์เรย์และสำหรับตัวแปรเจสันนั้นไม่จำกัดแค่รับส่งข้อมูลผ่านเว็บเบราว์เซอร์เท่านั้น แต่ยังสามารถนำเจสันไปประยุกต์กับการรับส่งข้อมูลในรูปแบบอื่น ๆ ได้ เช่นการจับเก็บข้อมูลในรูปแบบของ สายอักขระในข้อความหรือการรับส่งผ่านตัวให้บริการเว็บไซต์ (Web Service) ก็สามารถทำได้เช่นเดียวกัน

```
{
  "Students": [
    {
      "Name": "Amit Goenka",
      "Major": "Physics"
    },
    {
      "Name": "Smita Pallod",
      "Major": "Chemistry"
    },
    {
      "Name": "Rajeev Sen",
      "Major": "Mathematics"
    }
  ]
}
```

ภาพที่ 2.3 ตัวอย่าง Code ที่ใช้ Format JSON

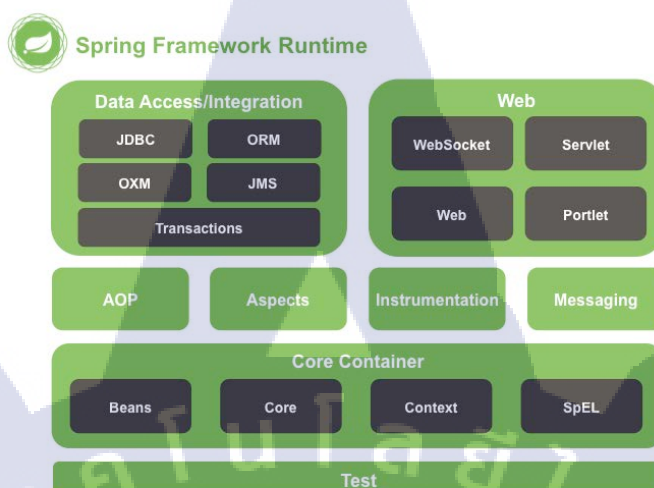
### 2.2.3 Spring Framework

Spring MVC framework คือ framework ในการสร้างเว็บ ที่รองรับแนวคิดแบบ MVC (Model , View ,Controller) นั่นเอง โดยหลักการทำงานของตัวSpring MVC framework จะออกแบบให้การทำงานทุกอย่างขึ้นอยู่กับ Servlet ที่ชื่อว่า DispatcherServlet ยกตัวอย่างเช่น เวลาที่มี Request จาก User Client เข้ามาตัว DispatcherServlet ก็จะต้องทำหน้าที่เป็น Operation รับ Request นั้นๆไว้ก่อน แล้วค่อยส่งต่อไปให้ตัว Controller อื่นๆทำงานต่อไป



ภาพที่ 2.4 รูปภาพโลโก้ของ Spring Framework

## โครงสร้างของ Spring Framework



ภาพที่ 2.5 ภาพโครงสร้างของ Spring Framework

ที่มา : <http://javabeat.net/>

Spring MVC framework มีการออกแบบโครงสร้างการเก็บชิ้นส่วนของเว็บ เช่น พวกหน้าเว็บ (ไฟล์ html, ไฟล์ jsp) ไฟล์ CSS หรือ ไฟล์ Script ต่างๆ ไว้อย่างชัดเจน เช่น หน้า ส่วน Controller ต้องอยู่ภายใน โฟลเดอร์ src เท่านั้น ซึ่งทำให้สะดวกต่อการค้นหา แก้ไข เพิ่มเติม หรือ ลบทิ้ง เพราะไม่ว่าจะเป็น developer คนใดในทีมเป็นคนสร้างไฟล์ ก็ต้องถูกเก็บอยู่ตามโครงสร้างที่ Spring MVC Framework กำหนดไว้เท่านั้น

ตัว Spring MVC framework มีการกำหนด Form การเขียนเว็บในแต่ละส่วน ไม่ว่าจะเป็นส่วน Request, Respond หรือ Controller ต่างๆ ทำให้ไม่ว่าจะเป็น developer คนไหนที่เข้ามาแก้ไขไฟล์ก็สามารถเข้าใจ Code ที่เขียนไว้ได้ง่ายๆ และยังมีตัว Library ที่ไว้ช่วยในการเขียนเพื่อให้ง่ายต่อการทำงานของ developer อีกด้วย

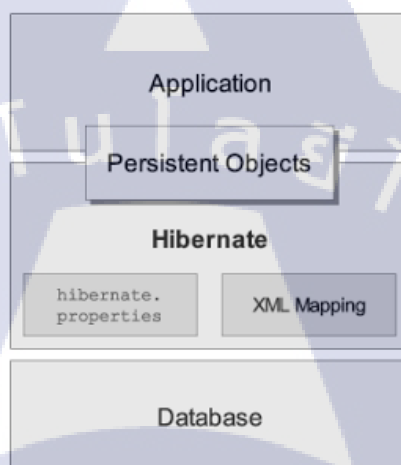
### 2.2.4 Hibernate Framework

Hibernate เป็น Framework Java ใช้ในการจัดการข้อมูลแบบ ORM (Object/Relation Mapping Object กับ ข้อมูลจากฐานข้อมูลแบบอัตโนมัติกลับไปกลับมาได้) เพื่อความสะดวกในการทำงานต่างๆ เช่น การเข้าถึงข้อมูล การเรียกค้นข้อมูล เป็นต้น



ภาพที่ 2.6 ภาพโลโก้ของ Hibernate

### โครงสร้าง Hibernate



ภาพที่ 2.7 โครงสร้างของ Hibernate

จากภาพที่ 2.9 เป็นรูปโครงสร้างระดับบนสุดของ Hibernate ซึ่งทำหน้าที่เป็นตัวกลางในการสื่อสารระหว่าง Application กับ Database

### ประโยชน์ของ Hibernate

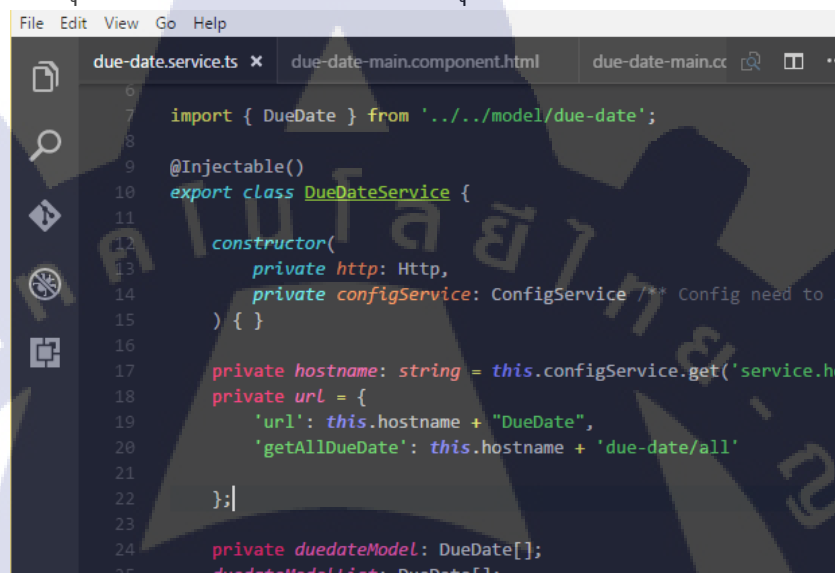
1. มีเวลาในการโฟกัส Business Logic ของโปรแกรมได้อย่างเต็มที่
2. ทำให้การบำรุงรักษาง่ายขึ้นเนื่องจากโค้ดที่เขียนน้อยลง สะอาด และเข้าใจง่าย
3. ไม่ยึดติดกับฐานข้อมูล เราสามารถใช้ฐานข้อมูลได้หลากหลาย
4. มี APIs ที่เรียบง่ายที่ใช้ในการบันทึกและอ่านข้อมูลจากฐานข้อมูลได้โดยตรงผ่าน Java

Objects

5. เข้าถึงฐานข้อมูลได้อย่างมีประสิทธิภาพ ทำให้ลดภาระของ Database Server ซึ่งส่งผลให้การเชื่อมต่อเร็วขึ้น

### 2.2.5 โปรแกรม Visual Studio Code

Visual Studio Code เป็นโปรแกรมที่ถูกพัฒนาขึ้นโดยบริษัท Microsoft โดยโปรแกรม Visual Studio Code นี้เป็นโปรแกรมสำหรับแก้ไขและปรับแต่งโค้ดที่ตัดความสามารถของ Visual Studio รุ่นปกติออกไปเหลือเพียง Editor อย่างเดียวซึ่งสามารถทำงานข้ามแพลตฟอร์ม ทั้งบน วินโดวส์ แมค และลินุกซ์ โดยเป็นโปรแกรมแจกฟรีให้ทุกคนสามารถเข้าไปดาวน์โหลดได้



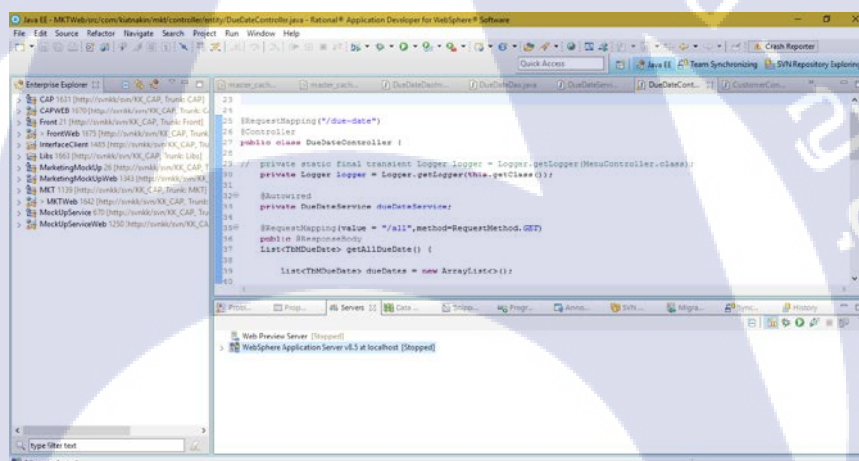
ภาพที่ 2.8 ตัวอย่างหน้าต่างภายในโปรแกรม Visual Studio Code

### 2.2.6 โปรแกรม Rational Application Developer for WebSphere

Rational Application Developer หรือ RAD เป็นโปรแกรมที่พัฒนาขึ้นมาโดยบริษัท IBM โดยมีพื้นฐานการทำงานมาจากโปรแกรม Eclipse ซึ่งสำหรับใช้พัฒนาเว็บไซต์และโปรแกรมที่สามารถเขียนโดยใช้ภาษา Java นอกจากนี้ยังมีเครื่องมือช่วยเหลือเพื่อให้สามารถทำงานได้ไวขึ้น อาทิ ส่วนของการ Synchronize SVN เพื่อให้สามารถอัปเดตไฟล์ที่พัฒนาเข้าเซิร์ฟเวอร์กลางเพื่อให้สามารถคนอื่นๆสามารถนำไปพัฒนาต่อ ตามที่ได้รับมอบหมาย และสิ่งที่เป็นจุดเด่นอีกอย่างคือ WebSphere ซึ่งมาพร้อมกับโปรแกรมนี้ โดยมีไว้สำหรับการใช้เป็นเซิร์ฟเวอร์ทดสอบบนเครื่องผู้พัฒนาได้เลย



ภาพที่ 2.9 รูปหน้าต่างโปรแกรมตอนเริ่มใช้งาน



ภาพที่ 2.10 รูปตัวอย่างหน้าต่างภายในโปรแกรม

## 2.2.7 NodeJS

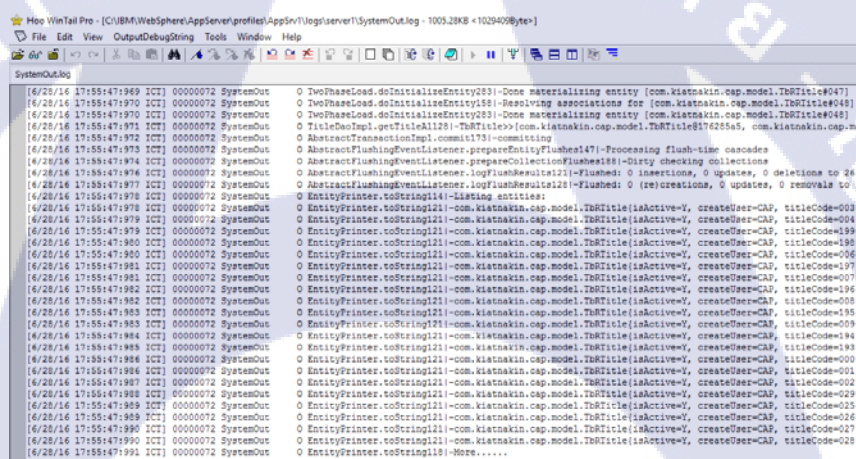
**Node.js** คือ Cross Platform Runtime Environment สำหรับฝั่ง Server และเป็น Open Source ซึ่งเขียนด้วยภาษา JavaScript สรุปรวมๆ NodeJS ก็คือ Platform ตัวหนึ่งที่เขียนด้วย JavaScript สำหรับเป็น Web Server นั้นเอง ซึ่งภายในโปรเจกต์นี้ข้าพเจ้าจะเน้นใช้คือส่วนที่เป็น npm หรือ Node Package Manager ซึ่งก็คือตัวช่วยที่คอยทำหน้าที่จัดการกับ Package ของ Node ให้เรา อย่างเช่น Application/package หรือ module ต่างๆ ที่มีคนเขียนไว้ ซึ่ง npm นั้นสามารถนำมาใช้ได้ อย่างง่ายดาย โดย Package/Module จะถูกติดตั้งไว้ที่โฟลเดอร์ node\_modules ภายในโปรเจกต์ของเรา

```
npm install PACKAGE_NAME
```

ภาพที่ 2.12 ตัวอย่างคำสั่งสำหรับการนำ Package หรือ Module คนอื่นมาใช้

## 2.2.8 โปรแกรม Hoo WinTail

Hoo WinTail เป็น utility สำหรับดูไฟล์แบบ Real-Time เหมือนกับ Unix Tail - Utility สามารถใช้เพื่อดู จุดสิ้นสุดของไฟล์ที่เพิ่มขึ้นเรื่อยๆ เป็นโปรแกรมในสำหรับดูร่องรอยที่โปรแกรมสร้างขึ้นหรือ Server logs ในแบบ Real-Time ขณะเดียวกันที่สร้างขึ้น สามารถดูส่วนสุดท้ายของไฟล์ใหญ่ๆ ได้ทันทีโดยไม่จำเป็นต้อง load ทั้งไฟล์



The screenshot shows the Hoo WinTail Pro application window. The title bar reads 'Hoo WinTail Pro - [C:\IBM\WebSphere\AppServer\profiles\AppSrv1\logs\server1\SystemOut.log - 1005.28KB <1029405Byte>]'. The menu bar includes File, Edit, View, OutputDebugString, Tools, Window, and Help. The main window displays a log file named 'SystemOut.log'. The log content shows a series of messages from 'com.kiatnakin.cap.model.TbRTTitle' and 'com.kiatnakin.cap.model.TbRTTitle' with timestamps and status codes. The messages include 'TwoPhaseLoad.doInitializeEntry2831-Done materializing entity', 'TwoPhaseLoad.doInitializeEntry1581-Resolving associations for', 'TitleDoImpl.getTitleAll1281-TbRTTitle', 'AbstractTransactionImpl.commit1731-committing', 'AbstractFlushingEventListener.prepareEntityFlushes1471-Processing flush-time cascades', 'AbstractFlushingEventListener.prepareCollectionFlushes181-Dirty checking collections', 'AbstractFlushingEventListener.logFlushResults1211-Flushed: 0 insertions, 0 updates, 0 deletions to 24', 'AbstractFlushingEventListener.logFlushResults1211-Flushed: 0 (recreations, 0 updates, 0 removals to', 'EntityPrinter.toString1141-Listing entities:', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=003', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=004', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=199', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=198', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=006', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=197', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=007', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=196', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=008', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=195', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=009', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=194', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=193', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=000', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=001', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=002', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=003', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=005', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=006', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=007', 'EntityPrinter.toString1211-com.kiatnakin.cap.model.TbRTTitle(isActive=W, createUser=CAP, titleCode=008', 'EntityPrinter.toString1211-More.....'

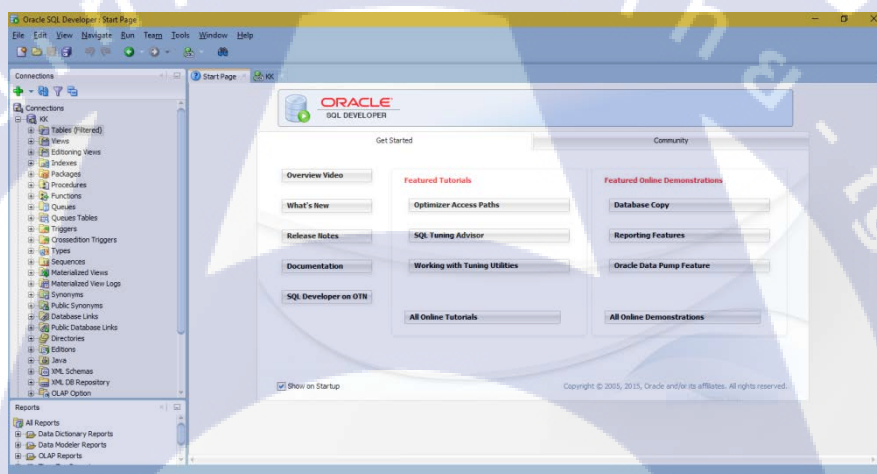
ภาพที่ 2.13 การทำงานภายในโปรแกรม Hoo Wintail

## 2.2.9 โปรแกรม Oracle SQL Developer

โปรแกรม SQL Developer พัฒนาขึ้นโดยบริษัทออราเคิล เป็นเครื่องมือที่ใช้ในการบริหารจัดการระบบฐานข้อมูล ช่วยให้ง่ายต่อการพัฒนา และจัดการระบบฐานข้อมูล โปรแกรม SQL Developer มีเครื่องมือที่ช่วยในการพัฒนา PL SQL ตั้งแต่ ต้นจนจบกระบวนการ ได้อย่างสมบูรณ์แบบ และยังมีเวิร์คชีต (Work sheet) ที่ช่วยแบ่งหน้าสำหรับการป้อนคำสั่งเอสคิวแอล มีคอนโซลสำหรับจัดการฐานข้อมูล มีหน้าต่างสำหรับรายงานข้อผิดพลาด ในการใช้คำสั่งเอสคิวแอล มีการสร้างแบบจำลองข้อมูลที่สมบูรณ์ และสามารถส่งออกฐานข้อมูลเพื่อนำไปใช้งานกับโปรแกรมออราเคิล ที่อยู่ในหลายๆ แพลตฟอร์มอีกด้วย



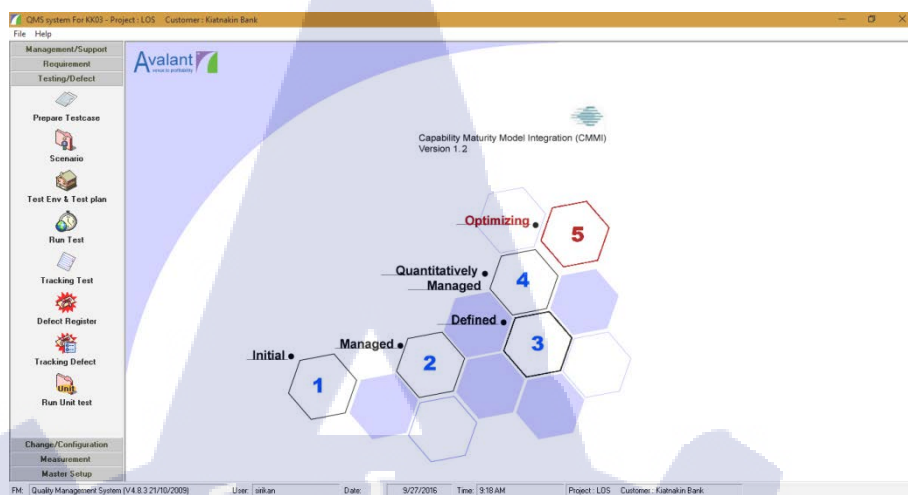
ภาพที่ 2.14 รูปหน้าต่างโปรแกรมตอนเริ่มใช้งานโปรแกรมoracle



ภาพที่ 2.15 ภาพตัวอย่างหน้าต่างภายในโปรแกรม

## 2.2.10 QMS System (Quality Management System)

เป็นเทคโนโลยีขององค์กรที่ใช้ในการช่วยจดบันทึกข้อบกพร่อง (Defect) ของระบบทั้งหมดของในแต่ละโครงการของบริษัทโดยผู้ที่จะจดบันทึกได้จะต้องมีรหัสผู้ใช้งานเป็นของตัวเองเพื่อให้สามารถทราบได้ถึงว่าข้อบกพร่อง (Defect) ในเรื่องนี้ใครเป็นผู้บันทึกนอกจากนี้ไม่ใช่แค่การจดบันทึกเท่านั้นข้อบกพร่อง (Defect) แต่ระบบยังสามารถส่งข้อบกพร่อง (Defect) ให้กับเคเวอลอปเปอร์แต่ละคนที่มีหน้าที่รับผิดชอบในส่วนงานนั้นเพื่อทำการแก้ไขปัญหาแต่ละส่วนได้อีกด้วย



ภาพที่ 2.16 ภาพตัวอย่างหน้าต่างภายในโปรแกรมQMS

### บทที่ 3

## แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

### 3.1 แผนงานการฝึกงาน

| หัวข้องาน                | มิถุนายน | กรกฎาคม | สิงหาคม | กันยายน |
|--------------------------|----------|---------|---------|---------|
| CAP : UI Screen          |          |         |         |         |
| CAP : Front Controller   |          |         |         |         |
| CAP : Data Access Object |          |         |         |         |
| CAP : Service            |          |         |         |         |
| CAP : Entity             |          |         |         |         |
| MKT : UI Screen          |          |         |         |         |
| MKT : Front Controller   |          |         |         |         |
| MKT : Data Access Object |          |         |         |         |
| MKT : Service            |          |         |         |         |
| MKT : Entity             |          |         |         |         |
| Fix Defect From CAP      |          |         |         |         |

 ระยะเวลาวางแผน
  ระยะเวลาปฏิบัติงานจริง

### ตารางที่ 3.1 แผนการปฏิบัติงานระยะเวลา 4 เดือน

#### 3.2 รายละเอียดที่นักศึกษาปฏิบัติในการฝึกงาน

ข้าพเจ้าได้มีโอกาสมาสหกิจศึกษา ณ บริษัท อีฟวแลนด์ จำกัด ในตำแหน่ง Developer และได้ถูกมอบหมายให้พัฒนาซอฟต์แวร์เว็บแอปพลิเคชันเกี่ยวกับการจัดการอนุมัติสินเชื่อของธนาคารแห่งหนึ่ง โดยได้ร่วมพัฒนาเพียงในบางส่วนจากทั้งหมดตามที่ได้รับมอบหมายเป็น Task งานในแต่ละครั้งไป โดยจะดูเงื่อนไขต่างๆ แล้วจึงทำการพัฒนาตามที่ระบุไว้ในเอกสาร Prototype และ เอกสาร SRS (Software Requirement Specification) ซึ่งเป็นเอกสารที่บอกถึงคุณสมบัติและเงื่อนไขต่างๆในส่วนนั้นๆ

#### 3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงาน

##### 3.3.1 ศึกษากระบวนการทำงานและเงื่อนไขต่างๆของชิ้นงานที่ได้รับมอบหมาย

ศึกษากระบวนการทำงานและเงื่อนไขต่างๆของชิ้นงานที่ได้รับมอบหมายเพื่อให้สามารถพัฒนาออกมาได้อย่างถูกต้องและตรงตามที่ระบุไว้ โดยสามารถหาข้อมูลเหล่านี้ได้จากไฟล์ SRS ซึ่งในไฟล์จะมีอธิบายเงื่อนไข Business Rule ไว้อย่างละเอียด

##### 3.3.2 ศึกษาโครงสร้าง และส่วนฐานข้อมูลที่จะนำมาใช้ของระบบ

ศึกษาโครงสร้างระบบหรือไฟล์ที่มีอยู่ก่อน เพื่อให้สามารถทำต่อและพัฒนาได้ และเพื่อให้เข้าถึงข้อมูลได้อย่างถูกต้องและรวดเร็วจึงต้องมีการศึกษาความสัมพันธ์ของแต่ละตาราง ภายในฐานข้อมูล ซึ่งจะนำไปตามที่ผู้ออกแบบกำหนดไว้ตามงานนั้นๆ โดยบางครั้งหากคิดปัญหาไม่เข้าใจจะเข้าไปถามผู้ออกแบบ

##### 3.3.3 วางแผน และประมาณเวลาที่ใช้ในการพัฒนา

ทำการวางแผนการทำงานในแต่ละชิ้นงาน เพื่อที่จะสามารถทำงานที่ได้รับมอบหมายต่อไปได้อย่างไม่ติดปัญหา โดยตัดสินใจจากความซับซ้อน ความยากง่ายของงานชิ้นนั้นๆ ประมาณเวลาที่ใช้ในการพัฒนา และบอกให้กับหัวหน้างานนั้นๆ เพื่อให้สามารถวางแผนในการมอบหมายงานในชิ้นต่อไปได้

##### 3.3.4 ลงมือพัฒนาโปรแกรมในส่วนหน้าจอตตามเอกสารโปรโตไทป์ พร้อมทำการสร้างเส้นทางการเข้าถึงให้กับหน้าจอนั้นๆ

พัฒนาเว็บแอปพลิเคชันตามที่ได้วางแผนไว้ โดยเริ่มลงมือสร้างหน้าจอก่อนโดยทำการเขียนโค้ดส่วน HTML ในส่วนนี้จะยังไม่สามารถทำงานใดๆและยังไม่สามารถเชื่อมต่อกับ

ฐานข้อมูลได้ จะเพียงเข้าผ่านเส้นทางการเข้าถึง หรือ พาท เพื่อไปดูหน้าตาของหน้าจอนั้นๆ ได้เท่านั้น

### 3.3.5 พัฒนาโปรแกรมในส่วนBackEnd การดึงข้อมูลจากฐานข้อมูล และฟังก์ชันต่างๆ

พัฒนาในส่วน FrontEnd และส่วน BackEnd ให้สมบูรณ์ เขียนไฟล์ส่วน BackEnd โดยเริ่มจากการสร้างไฟล์โมเดล ไฟล์ Dao Service และ Controller แล้วจึงมาเขียนไฟล์Serviceในฝั่ง FrontEnd เพื่อเรียกใช้งานข้อมูลจากฐานข้อมูล ไปสู่หน้าจอการทำงาน และหากเกิดปัญหาใดๆขึ้น ก็ จะแจ้งให้กับทางหัวหน้างานเพื่อขอคำปรึกษาปรึกษาหรือหาทางแก้ไขงานให้สามารถดำเนินการต่อไปได้

### 3.3.6 ตรวจสอบและทดสอบระบบที่พัฒนา

เมื่อทำชิ้นงานนั้นๆเสร็จ ก็จะมีการยูนิตเทสก่อนอย่างละเอียด เพื่อหาข้อผิดพลาดหรือบัคในระบบ หากเจอก็จะทำการแก้ไขในจุดนั้นๆ ให้ได้ตามที่มีเงื่อนไขระบุไว้ในเอกสาร SRS

### 3.3.7 อัปเดตไฟล์ขึ้นSVN Server

เมื่อได้ทำการยูนิตเทสและแก้ไขให้ถูกต้องตามเอกสาร SRS เรียบร้อยครบถ้วนและไม่มีข้อผิดพลาดใดๆแล้ว ก็จะมีการ Synchronize ไฟล์ไปยัง SVN Server เพื่ออัปเดตงานในส่วนที่ทำเสร็จเรียบร้อยแล้ว ขึ้นไปเพื่อให้ซอฟต์แวร์ทดสอบสามารถนำไปทดสอบต่อไปได้ตามกระบวนการ

### 3.3.8 แก้ไขข้อผิดพลาดจาก Defect

เมื่องานถูกส่งไปให้ซอฟต์แวร์ทดสอบเรียบร้อยแล้ว ซอฟต์แวร์ทดสอบจะทำการ log Defect แล้วแจ้งมายังฝ่ายคิวเวลลอปเปอร์ให้ทำการแก้ไข โดยสามารถดูจากโปรแกรม QMS ซึ่งจะดูสิ่งที่ถูกต้องได้จากเอกสาร SRS หรือสอบถามโดยตรงกับผู้ที่log Defect นั้นๆขึ้นมา เพราะในบางครั้งก็อาจเกิดข้อผิดพลาดได้เนื่องจากเวอร์ชันของเอกสารSRSที่ดูเป็นคนละเวอร์ชันกัน ซึ่งเกิดจากการเปลี่ยนแปลงความต้องการของลูกค้า

## บทที่ 4

### สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ

#### 4.1 ขั้นตอนและผลการดำเนินงาน

##### 4.1.1 ศึกษากระบวนการทำงานและเงื่อนไขต่างๆของชิ้นงานที่ได้รับมอบหมายอย่างละเอียดจากเอกสาร SRS ตามแต่ละส่วนงาน

6.2.2 DV\_UC002: Claim Case

Table 6-2 DV\_UC002 <Claim Case> description

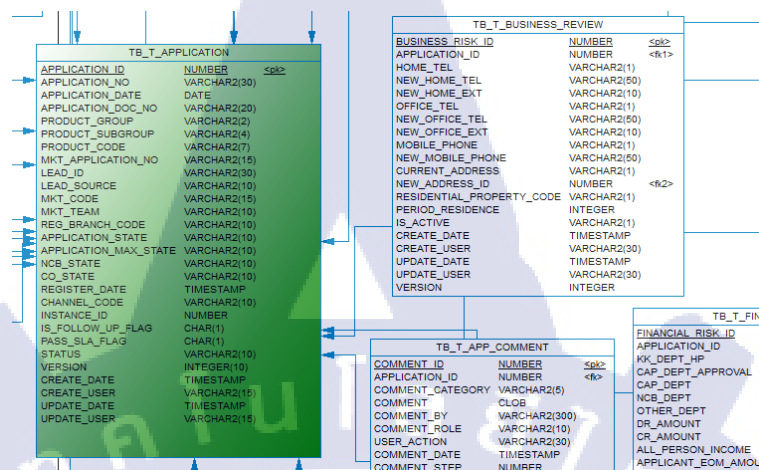
| Category             | Description                                                                                                                                                                                                                                                           |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objectives           | เพื่อกรอกรับงานหรือเปิดงานที่รับแล้วแต่ยังดำเนินการไม่เสร็จ ในกล่องรับงานของตนเอง                                                                                                                                                                                     |
| Actor                | COC1, COC1 Supervisor, COC1 Manager, COC Head Manager                                                                                                                                                                                                                 |
| Pre-Conditions       | 1. งานถูกแจกเข้ากล่องรับงานของ COC1<br>2. เลือกเมนู รายการงานตนเอง<br>3. ระบบแสดงหน้า รายการงานตนเอง                                                                                                                                                                  |
| Primary Scenario     | <u>การรับงานใหม่</u><br>1. กดปุ่มรับงาน  ที่รายการงานใหม่<br>2. ระบบบันทึกเวลารับงานและเปลี่ยนสถานะงาน<br>3. ระบบแสดงหน้าจอการทำงานเห็นแรก (ข้อมูลลูกค้า)                          |
| Business Rules       | N/A                                                                                                                                                                                                                                                                   |
| Alternative Scenario | <u>การเปิดงานที่รับแล้วแต่ยังไม่เสร็จ</u><br>1. กดปุ่ม  ที่รายการที่รับงานแล้ว เพื่อเข้ามาดำเนินการต่อ<br>2. ระบบแสดงหน้าจอการทำงานเห็นแรก (ข้อมูลลูกค้า)                          |
| Exception            | N/A                                                                                                                                                                                                                                                                   |
| Post Condition       | <u>การรับงานใหม่</u><br>1. งานถูกเปลี่ยนสถานะเป็น "Claimed by COC1"<br>2. งานถูกบันทึกวันที่เวลารับงาน และข้อมูลเจ้าหน้าที่ที่กรอกรับงาน<br>3. ไอคอน หน้ารายการงาน ถูกเปลี่ยนเป็น  |
| Use Case Related     | N/A                                                                                                                                                                                                                                                                   |

ภาพที่ 4.1 ภาพตัวอย่างขั้นตอนการทำงานและเงื่อนไขต่างๆจากเอกสาร SRS

##### 4.1.2 ศึกษาโครงสร้าง และส่วนฐานข้อมูลที่น่ามาใช้ของระบบ

ในข้อมูลแต่ละฟิลด์ จะต้องมีการเก็บบันทึกและเรียกใช้ข้อมูลจากฐานข้อมูลทุกๆฟิลด์ ซึ่งจำเป็นต้องศึกษาโครงสร้างและตารางในฐานข้อมูลที่น่ามาใช้งานส่วนนั้นๆ เนื่องจากส่วนมาก

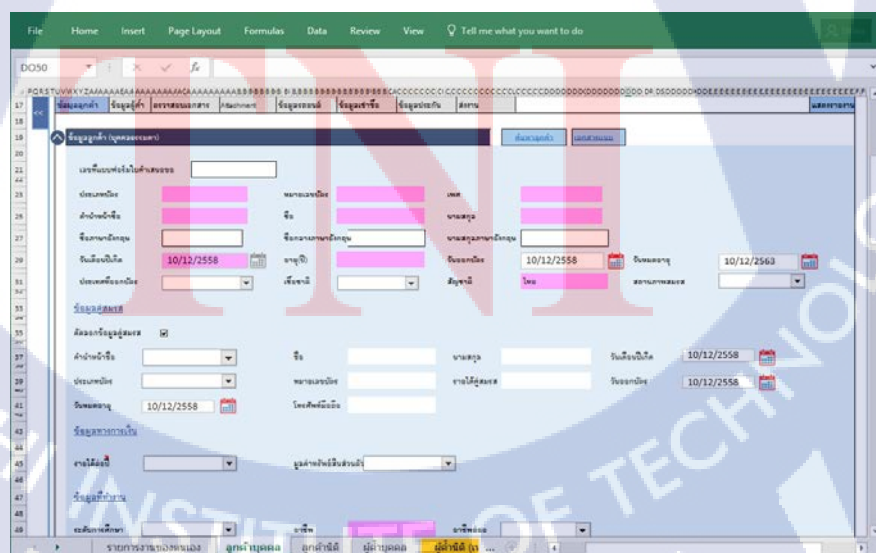
นั่น แต่ละฟิลด์จะมีความซับซ้อนแตกต่างกันไป โดยสามารถดูได้จากไฟล์ความสัมพันธ์ของตารางต่างๆ



ภาพที่ 4.1 ภาพตัวอย่างโครงสร้างตารางจากฐานข้อมูลบางส่วน

#### 4.1.3 ลงมือพัฒนาโปรแกรมในส่วนหน้าจอตามเอกสารโปรโตไทป์ พร้อมทำการสร้างเส้นทางการเข้าถึงให้กับหน้านั้นๆ

ในแต่ละโมดูลจะมีไฟล์เอกสาร Prototype ซึ่งแสดงหน้าตาของหน้าจอการทำงานว่ามีอะไรในหน้านั้นๆบ้าง มีไว้เพื่อให้ developer สะดวกต่อการพัฒนา ในไฟล์นี้จะไม่บอกรายละเอียดเงื่อนไขหรือกระบวนการทำงานใดๆ เพียงแค่สามารถดูข้อมูลในไฟล์แล้วสร้างหน้าจอโดยใช้โค้ดภาษา HTML ได้ เริ่มต้นโดยการสตาร์ทเซิร์ฟเวอร์และ node module แล้วเริ่มทำได้เลย



ภาพที่ 4.2 ภาพตัวอย่างจากเอกสารโปรโตไทป์บางส่วน

```

<div class="card-header">
  ข้อมูลลูกค้า (บุคคลธรรมดา)
  <div class="button-group">
    <button class="btn btn-default" type="button" [disabled]="readonly">แสดงรายงาน</button>
    <button class="btn btn-default" type="button" [disabled]="readonly" (click)="searchCustomer">ค้นหา</button>
    <button class="btn btn-default" type="button">เอกสารแนบ</button>
  </div>
</div>
<!--row1-->
<row>
  <control type="text" name="applicationDocNo" label="เลขที่แบบฟอร์มใบคำเสนอขอ" [(model)]= "applicationDocNo" [cssClass]="cssClass" maxlength="20" [readonly]="readonly" </row>

  <!--row2-->
  <row>
    <control type="dropdown" name="cardType" label="ประเภทบัตร" [(model)]= "cloneCustomerModel" [disabled]="true" [readonly]="true" [master]="CARD_TYPE"></control>
    <control type="text" name="cardId" label="หมายเลขบัตร" [cssClass]="cssClass" [(model)]= "cardId" </control>
  </row>
</row>

```

ภาพที่ 4.3 ตัวอย่างโค้ด HTML ที่เขียนผ่านโปรแกรม Visual Studio Code

เมื่อสร้างหน้าจอการทำงาน จะต้องมีการสร้างเส้นทางการเข้าถึง หรือ พาท ไว้ด้วย สร้างได้โดยใช้ภาษา TypeScript กำหนดให้ไฟล์นั้นๆ เข้าถึงหน้าจอการทำงานทางเว็บเบราว์เซอร์ทางชื่อพาทใดๆ เพื่อให้สามารถเข้าไปดูหน้าจอการทำงานผ่านพาทที่ตั้งเอาไว้ได้ โดยในแต่ละฟอร์ม จะต้องมีการสร้างไฟล์ HTML และไฟล์ TypeScript ทั้งสองชนิด โดยไฟล์ชนิด HTML จะเป็นตัวแสดงหน้าจอการทำงาน และส่วนไฟล์ TypeScript จะเป็นไฟล์ที่เชื่อมต่อกับส่วนอื่นๆ ในระบบ และสำหรับใส่เงื่อนไข ฟังก์ชันต่างๆ ในการทำงาน

```

@Component({
  selector: 'due-date-main',
  template: require('./due-date-main.component.html'),
  directives: [
    HTML_DIRECTIVES,
    FORM_CONTROL_DIRECTIVES,
    MODAL_DIRECTIVES
  ]
})

export class DueDateMainComponent extends SubFormBase {
  // @Input('duedate')
  duedateModelList: DueDate[];
  private duedateTd: number;
  //private duedateModel: DueDate;

  constructor(
    private duedateService: DueDateService
  ) {
    super();
  }
}

```

ภาพที่ 4.4 ตัวอย่างโค้ด TypeScript ที่เขียนจากโปรแกรม Visual Studio Code

ในหนึ่งเส้นทาง การเข้าถึง หรือ พาทนั้น จะมีหลายๆฟอร์มรวมอยู่ด้วยกัน เรียกว่า Subform หรือ ฟอร์มย่อย ซึ่งเมื่อทำการกำหนดเส้นทางจะต้องมีการรวมให้มาอยู่ด้วยกันในฟอร์ม วิธีการใส่ คือนำชื่อคลาสมาใส่ไว้รวมกันในฟอร์มอีกฟอร์มหนึ่ง เพื่อให้สามารถเรียกใช้ได้ในชื่อคลาสเพียง คลาสเดียว ดังภาพด้านล่างเป็นการรวมฟอร์มย่อย GeneralParameter ทั้งสามฟอร์มย่อยใน directives ชื่อที่นำไปใช้คือชื่อคลาส MasterGeneralParameterComponent

```
import { Component } from '@angular/core';
import * as Subform from '../subform';

@Component ({
  selector: 'master-general-parameter',
  template: `<general-parameter-search (selectedSearch)="showGeneralParameter">
    <general-parameter-detail *ngIf="displayGeneralParameter">
    <general-parameter-management *ngIf="displayGeneralParameter">
  directives: [
    Subform.GeneralParameterSearchComponent,
    Subform.GeneralParameterDetail,
    Subform.GeneralParameterManagementComponent
  ]
})

export class MasterGeneralParameterComponent {
```

ภาพที่ 4.5 ตัวอย่างโค้ด TypeScript ที่เขียนจากโปรแกรม Visual Studio Code

ภาพด้านล่าง เป็นการนำฟอร์มที่รวมฟอร์มย่อยอยู่ข้างในแล้วไปใช้ในการสร้างเส้นทาง การเข้าถึง หรือ พาท วิธีการคือ ใส่ชื่อพาทและชื่อคลาสในไฟล์ที่มีไว้สำหรับการกำหนดพาทการเข้าถึง ต่างๆ ใช้เวลาที่จะเรียกดูข้อมูลทางหน้าจอบริเวณเบราว์เซอร์ จะเห็นได้ว่าชื่อคลาสที่ระบุไว้ข้างต้นนั้น ได้นำมาใช้อยู่บรรทัดล่างของชื่อพาท เป็นการบอกว่าหากเรียกใช้ผ่านชื่อพาทนี้ระบบจะไปเรียกใช้ ข้อมูลจากคลาสนี้โดยตรง

```
} from '../../content/master-setup';

import * as Forms from '../../content/forms';

export const masterRoutes: RouterConfig = [

  {
    path: 'phone-verification-list',
    component: MasterPhoneVerificationComponent
  },
];
```

ภาพที่ 4.6 ตัวอย่างโค้ด TypeScript ที่เขียนจากโปรแกรม Visual Studio Code

localhost:3000/#/master/phone-verification-list

ภาพที่ 4.7 ภาพตัวอย่างการเรียกดูงานผ่านเส้นทางที่เข้าถึงที่กำหนดไว้

เมื่อได้มีการสร้างหน้าฟอร์ม และกำหนดเส้นทางเข้าถึงโดยใช้ภาษา TypeScript และได้มีการไปใส่ข้อมูลตั้งค่าในฐานข้อมูลไว้เรียบร้อยแล้ว หากไม่มีข้อผิดพลาดหรือเออเร่อใด เว็บเบราว์เซอร์จะแสดงผลออกมาตามที่ได้สร้างไว้

ข้อมูลที่อยู่

ที่อยู่ทำงาน (ชื่อผู้เช่าชื่อหรือผู้ค้า)

ที่อยู่เดียวกับ

หมายเลขทะเบียนการค้า  คำนวณหน้า  ชื่อบริษัทสถานประกอบการ

เลขที่  หมู่  หมู่บ้าน

อาคาร  ชั้น  ซอย

ภาพที่ 4.8 ภาพหน้าจอที่ยังไม่มีฟังก์ชันการทำงานใดๆที่แสดงผลทางเว็บเบราว์เซอร์

#### 4.1.4 พัฒนาโปรแกรมในส่วน Backend การดึงข้อมูลจากฐานข้อมูล และฟังก์ชันต่างๆ

เริ่มต้นจากสร้างไฟล์โมเดลที่ได้มาจากตารางในฐานข้อมูลลงในโปรแกรมRAD เพื่อให้เป็นตัวกลางในการรับหรือส่งค่าต่างๆ

```
@Table(name = "TB_M_DUE_DATE")
@XmlRootElement
@NamedQuery(name = "TbMDueDate.findAll", query = "SELECT t FROM TbMDu
public class TbMDueDate implements Serializable {
    private static final long serialVersionUID = 1L;
    // @Max(value=?) @Min(value=?)//if you know range of your decima
    @Id
    @Basic(optional = false)
    @Column(name = "DUE_DATE_ID", nullable = false, precision = 0, sc
    @Column(name = "CUSTOMER_TYPE")
    private Character customerType;
    @Column(name = "OCCUPATION_GROUP", length = 10)
    private String occupationGroup;
    @Column(name = "PAYMNET_TYPE", length = 10)
    private String paymnetType;
```

ภาพที่ 4.9 ตัวอย่างโค้ดในไฟล์โมเดลส่วน Backend

สร้างไฟล์ DAO (Data Access Objects) ซึ่งเป็นตัวเข้าถึงฐานข้อมูลกับส่วนอื่นๆ จากภาพตัวอย่างโค้ดในไฟล์DAO เป็นตัวอย่างของการเขียนฟังก์ชันเพื่อเรียกใช้ข้อมูลในโมเดลที่มาจากฐานข้อมูลทั้งหมด

```
@Repository("listBoxDao")
public class ListBoxDaoImpl extends AbstractDao implements ListBoxDao {

    private Logger logger = Logger.getLogger(this.getClass());

    @Override
    @Transactional
    public TbMListBox getListBoxByListBoxId(long listBoxId) {
        Criteria criteria = getSession().createCriteria(TbMListBox.class);
        criteria.add(Restrictions.eq("listBox", listBoxId));

        TbMListBox itemList = (TbMListBox) criteria.uniqueResult();

        logger.debug("listBox>>>" + itemList);
        return itemList;
    }
}
```

ภาพที่ 4.10 ตัวอย่างโค้ดในไฟล์ DAO

สร้างไฟล์ Service ซึ่งเป็นตัวเรียกใช้ข้อมูลจากไฟล์DAO เพื่อส่งต่อไปให้ไฟล์Controller ต่อไปได้

```
@Service("ListBoxService")
public class ListBoxServiceImpl implements ListBoxService{

    @Autowired
    ListBoxDao listBoxDao;

    @Override
    public TbMListBox getListBoxBylistBoxId(long listBoxId) {
        return listBoxDao.getListBoxByListBoxId(listBoxId);
    }

    @Override
    public List<TbMListBox> getAllListBox() {
        return listBoxDao.getAllListBox();
    }
}
```

ภาพที่ 4.11 ตัวอย่างโค้ดในไฟล์ Service ส่วน Backend

สร้างไฟล์ Controller เพื่อเป็นตัวกลางในการรับ-ส่งข้อมูลจาก Service ในส่วน Backend และ Service ในส่วน FrontEnd สามารถทดสอบเออเร่อได้โดยการลองเข้าผ่านพาทที่กำหนดไว้

ได้ หากโปรแกรมไม่มีการฟ้องเออเร่อใดๆ แต่ยังไม่สามารถเข้าไปได้ให้ดูlog เออเร่อ ที่โปรแกรม Wintail

```
@RestController
@RequestMapping("/listbox")
public class ListBoxController {

    private static final transient Logger logger = Logger
        .getLogger(AddressController.class);

    @Autowired
    private ListBoxService listBoxService;

    @RequestMapping(method = RequestMethod.GET,value = "/by/companycode/{code}")
    public @ResponseBody TbMListBox getCompanyTypeByCompanyCode(@PathVariable String code )
    {
        return listBoxService.getCompanyTypeByCompanyCode (code);
    }
}
```

ภาพที่ 4.12 ตัวอย่างโค้ดในไฟล์ Controller

เมื่อเรียกดูข้อมูลจากพาทที่กำหนดไว้ในไฟล์ Controller สำเร็จแล้ว จำเป็นต้องสร้างไฟล์ Service ของทาง FrontEnd ด้วยเช่นกัน เพื่อให้ServiceของFrontEnd ไปรับข้อมูลที่Controller ของทาง BackEnd ส่งมา โดยฟังก์ชันที่เรียกใช้จะต้องตรงกันทั้ง BackEnd และ FrontEnd

```
private hostname: string = this.configService.get('service.hostname');
private url = {
    'url': this.hostname + "DueDate",
    'getAllDueDate': this.hostname + 'due-date/all'
};

private dueDateModel: DueDate[];
private dueDateModelList: DueDate[];

public getAllDueDate(): Observable<DueDate[]> {
    let url = this.url.getAllDueDate;

    return this.http.get(url)
        .map((res: Response) => {
            // Mapping response data
            let body = res.json();
            this.dueDateModelList = body;
            return body || {};
        })
        .catch(this.handleError);
}
```

ภาพที่ 4.13 ตัวอย่างโค้ดในไฟล์ Service ส่วน FrontEnd

สร้างโมเดลของส่วนหน้าบ้าน เพื่อเป็นตัวรับข้อมูลแต่ละฟิลด์ในแต่ละตารางของฐานข้อมูล

```
export class DueDate {
    public dueDateId: number;
    public customerType: string;
    public occupationGroup: string;
    public paymnetType: string;
    public bookDateFrom: number;
    public bookDateTo: number;
    public endPlusDateMin: number;
    public endPlusDateMax: number;
    public endDueDateFrom: number;
    public endDueDateTo: number;
    public endDueDateNextMonth: number;
    public beginPlusDateMin: number;
    public beginPlusDateMax: number;
    public beginDueDateFrom: number;
    public beginDueDateTo: number;
    public beginDueDateNextMonth: number;
    public isActive: string;
    public createDate: Date;
    public createUser: string;
    public updateDate : Date;
    public updateUser: string;
    public version: number;
}
```

ภาพที่ 4.14 ตัวอย่างโค้ดในไฟล์โมเดล ส่วน FrontEnd

เรียกใช้เซอร์วิสส่วนหน้าบ้าน โดยเขียนฟังก์ชันในส่วนไฟล์TypeScript ของฟอร์ม ตัวอย่างเช่นในภาพด้านล่าง เป็นการเรียกใช้ฟังก์ชัน getAllDueDate() จากไฟล์เซอร์วิสหน้าบ้าน

```
constructor(
    private dueDateService: DueDateService
) {
    super();
}

ngOnInit() {
    this.dueDateService.getAllDueDate().
        subscribe((duedateModelList: DueDate[]) => {
            console.log('duedateModelList = ', duedateModelList);
            this.duedateModelList = duedateModelList;
        })
}
```

ภาพที่ 4.15 ตัวอย่างโค้ดในไฟล์ Service ส่วน FrontEnd

และเมื่อได้มีการเรียกใช้ข้อมูลจากเซิร์ฟเวอร์หน้าบ้านแล้ว วิธีการเอาข้อมูลมาใช้ก็น่าชื่อที่ได้กำหนดไว้ไปใส่ ยกตัวอย่างเช่นในภาพด้านล่าง ชื่อที่กำหนดไว้คือ cloneCustomerModel และชื่อฟิลด์ที่กำหนดไว้คือ name วิธีนำมาใส่ จะต้องเขียนเป็นรูปแบบนี้

`[(model)]= "cloneCustomerModel.name"` เมื่อเขียนโค้ดแบบนี้แล้ว จะทำให้ฟิลด์เปล่าๆที่ได้สร้างไว้แค่เป็นไฟล์HTMLในตอนต้น ก็จะสามารถเรียกแสดงข้อมูลที่อยู่ในฐานข้อมูลได้อย่างสมบูรณ์

```
<control type="text" name="name" label="ชื่อ" [cssClass]="cssClass" [(model)]= "cloneCustomerModel.name"></control>
```

ภาพที่ 4.16 ตัวอย่างการใส่โมเดลเพื่อบันทึกหรือเรียกแสดงจากฐานข้อมูล

ในส่วนที่เป็นฟิลด์ครอบดาวลิส ทำให้มีข้อมูลแสดงขึ้นมาได้โดยใส่คำสั่งQuery และกำหนดชื่อที่ใช้ในการเรียกใช้หรือ มาสเตอร์ ไว้ข้างหน้า ดังภาพด้านล่าง ซึ่งก่อนหน้านี้ได้มีการเขียนไฟล์เพื่อกำหนดไว้แล้ว ว่าให้เขียนเป็นรูปแบบนี้ เพื่อง่ายต่อการกำหนดและเรียกใช้ชื่อมาสเตอร์

```

22 #####
23 # Master Cache SQL      (c) Kiatnakin Bank #
24 #####
25
26 ### Beware of using reserved words as a part of query, always put double quotes around the word
27 BRANCH=select BRANCH_CODE AS "value", BRANCH_NAME AS "desc", IS_ACTIVE AS active from TB_R_BRANCH
28 BRAND=select BRAND_CODE AS "value", BRAND_NAME AS "desc", IS_ACTIVE AS active from TB_R_BRAND
29 BUSINESS_TYPE_BOT=select BUSINESS_TYPE_BOT_CODE AS "value", BUSINESS_TYPE_BOT_NAME AS "desc", IS_ACTIV
30 BUSINESS_TYPE=select BUSINESS_TYPE_CODE AS "value", BUSINESS_TYPE_NAME AS "desc", IS_ACTIVE AS active
31 CALCULATE_TYPE=select CALCULATE_CODE AS "value", CALCULATE_DESC AS "desc", IS_ACTIVE AS active from TB
32 CAMPAIGN=select CAMPAIGN_CODE AS "value", CAMPAIGN_DESC AS "desc", IS_ACTIVE AS active from TB_R_CAMPA
33 COLORS=select COLOR_CODE AS "value", DESCRIPTION AS "desc", IS_ACTIVE AS active from TB_R_CAR_COLOR
34 COMPANY=select COMPANY_CODE AS "value", COMPANY_NAME AS "desc", IS_ACTIVE AS active from TB_M_COMPANY
35 COUNTRY=select COUNTRY_CODE AS "value", COUNTRY_NAME AS "desc", IS_ACTIVE AS active from TB_R_COUNTRY

```

ภาพที่ 4.17 ตัวอย่างการกำหนดมาสเตอร์

วิธีการใส่มาสเตอร์ในฟิลด์ครอบดาวลิส ทำได้โดยใส่ชื่อมาสเตอร์ที่กำหนดไว้อยู่ใน `[master]= " " name " "` ดังตัวอย่างในภาพด้านล่าง

```
<control type="dropdown" name="raceCode" label="เชื้อชาติ" [cssClass]="cssClass" [master]= " 'NATIONALITY' "></control>
```

ภาพที่ 4.18 ตัวอย่างการเรียกใช้มาสเตอร์

วันเดือนปีเกิด: 06/08/2535      อายุ: 24  
 ประเทศที่ออกบัตร:   
 เชื้อชาติ: Jin  
 ข้อมูลคู่สมรส  
 คัดลอกข้อมูลคู่สมรส: ☐  
 คำนำหน้าชื่อ:   
 ชื่อ:   
 ประเภทบัตร:   
 หมายเลขบัตร:

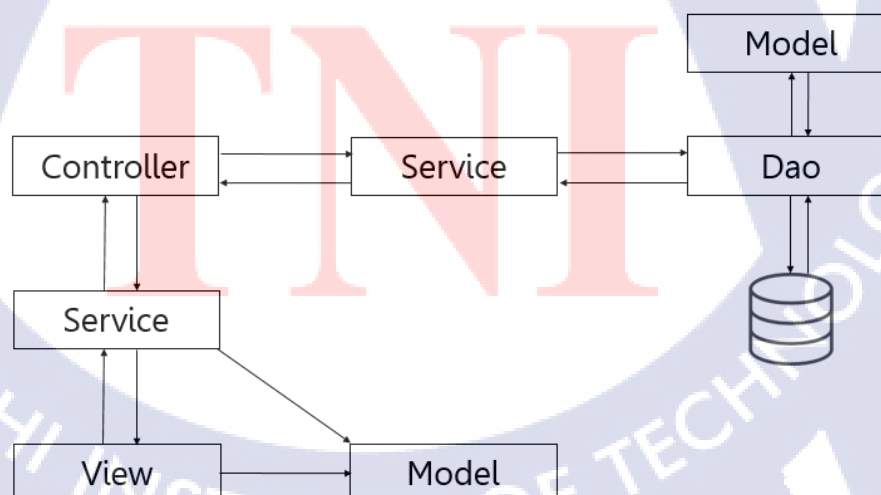
หมู่เกาะฟอล์กแลนด์  
 เฟรนช์เกียนา  
 กานา  
 กัวเตมาลา  
 จิน  
 คอสตาริกา  
 ไชปรัส  
 สาธารณรัฐเคิร์ด

ภาพที่ 4.19 ผลลัพธ์การเรียกใช้มาสเตอร์

ข้อมูลลูกค้า (บุคคลธรรมดา)  
 เลขที่แบบฟอร์มใบคำเสนอขอ: 22  
 ประเภทบัตร:   
 หมายเลขบัตร: 1234567891011  
 เพศ: ชาย  
 คำนำหน้าชื่อ: นาย  
 ชื่อ: กุญชรินทร์  
 นามสกุล: ชีวจีนนิ่ง  
 ชื่อภาษาอังกฤษ: 22  
 ชื่อกลางภาษาอังกฤษ: 22  
 นามสกุลภาษาอังกฤษ: 22  
 วันเดือนปีเกิด: 06/07/2524  
 อายุ: 35  
 วันออกบัตร: 05/09/2559  
 วันหมดอายุ: 29/08/2559  
 ประเทศที่ออกบัตร: ปากีสถาน  
 เชื้อชาติ: เฟรนช์เกียนา  
 สัญชาติ: ไทย  
 สถานภาพสมรส: โสด

ภาพที่ 4.20 ภาพหน้าจอการทำงานอย่างสมบูรณ์ที่แสดงผลทางเว็บเบราว์เซอร์

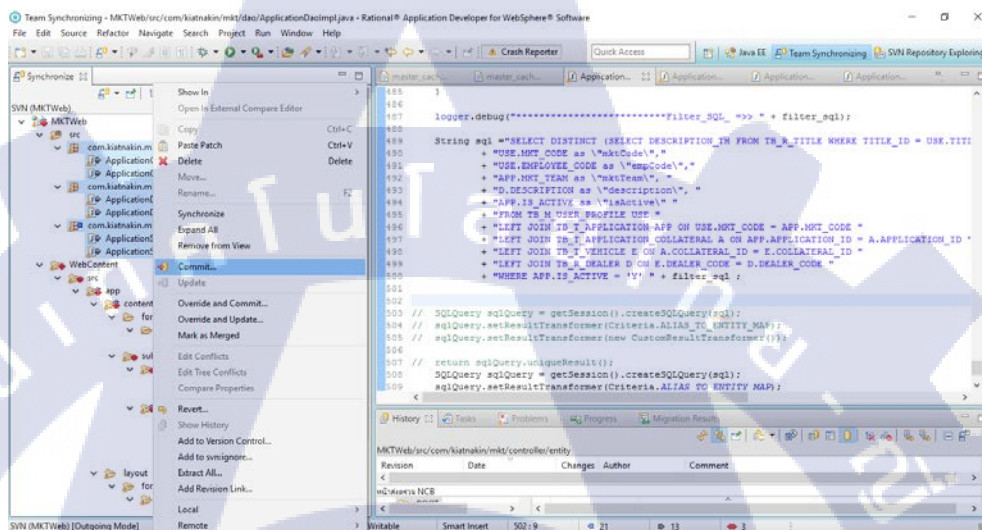
กระบวนการทำงานข้างต้น สามารถสรุปเป็นแผนภาพให้ง่ายต่อการทำความเข้าใจได้ดังนี้



ภาพที่ 4.21 ภาพโครงสร้างสรุปหลักการทำงาน

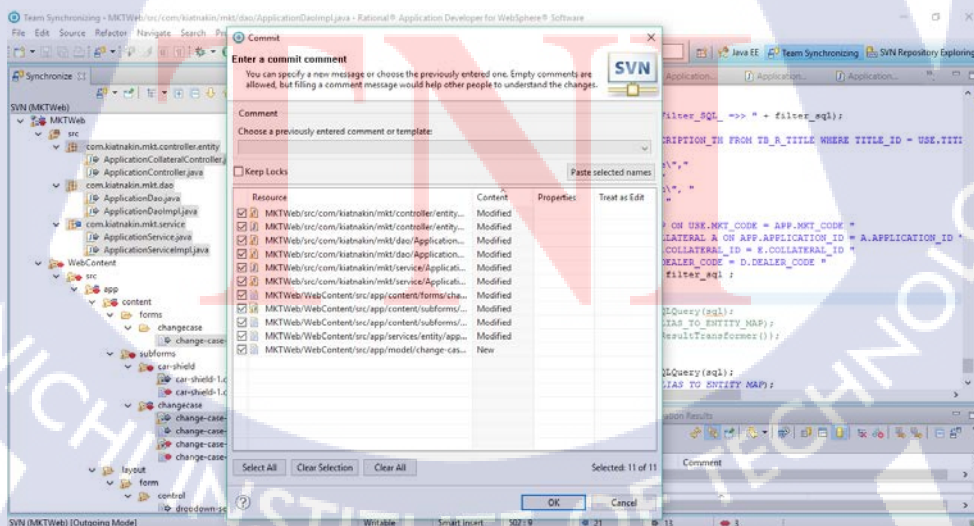
#### 4.1.5 อัปเดตไฟล์ขึ้น SVN Server

เมื่อได้ทำการยูนิตเทสต์ด้วยตัวเองให้แน่ใจแล้วว่าไม่มีข้อผิดพลาดใด ก็ทำการคอมมิทไฟล์ขึ้นไปยัง SVN Server โดยไปที่ Team Synchronizing แล้วเลือกไฟล์ที่ต้องการคอมมิท คลิกขวาที่ไฟล์ แล้วเลือก commit



ภาพที่ 4.22 ขั้นตอนการคอมมิทไฟล์ขึ้นไปยัง SVN Server(1)

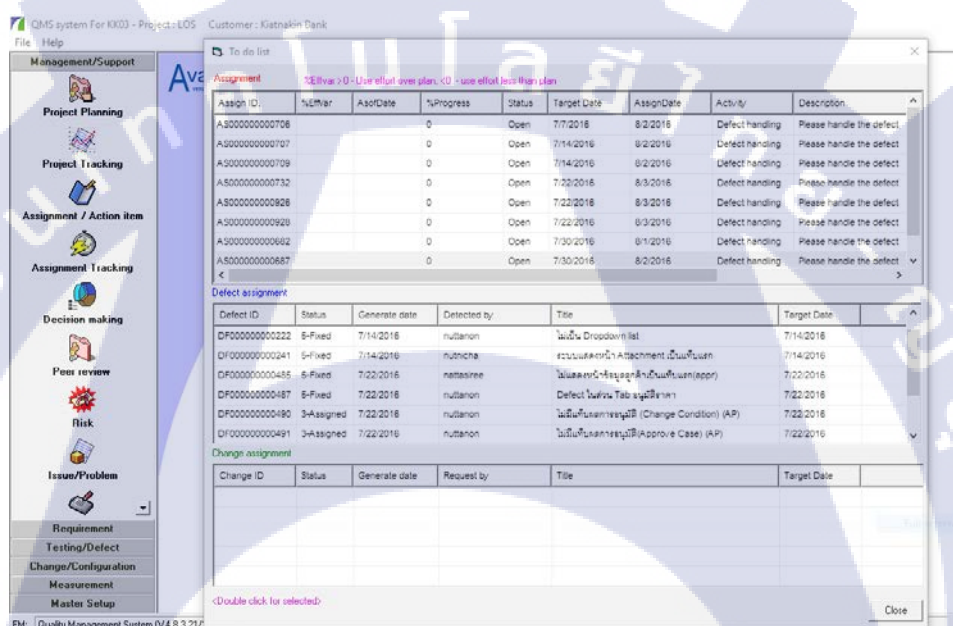
โปรแกรมจะลิสรายชื่อไฟล์ที่เราเลือกไว้ หากถูกต้องตามที่ต้องการ กดปุ่ม OK จะเสร็จสิ้นการคอมมิท



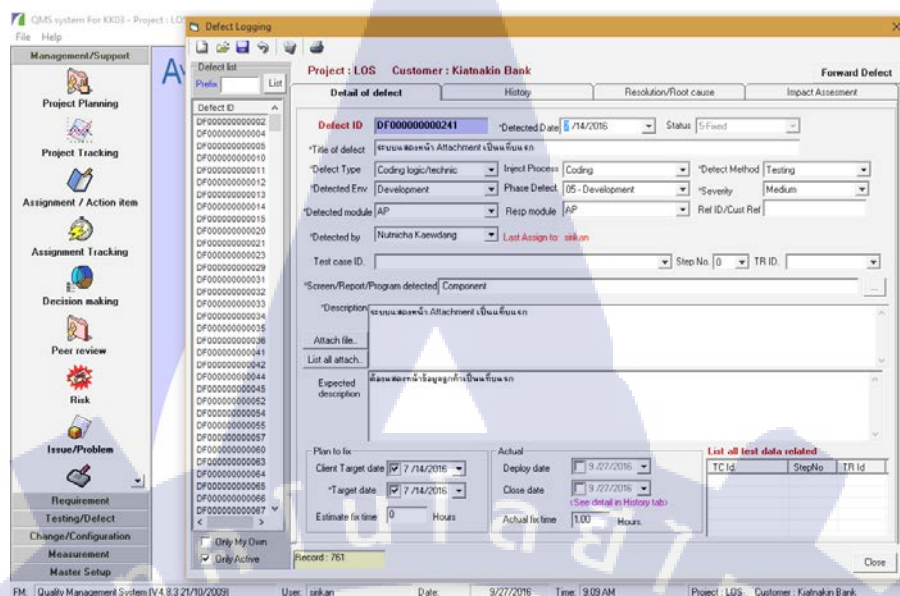
ภาพที่ 4.23 ขั้นตอนการคอมมิทไฟล์ขึ้นไปยัง SVN Server(2)

#### 4.1.6 แก้ไขข้อผิดพลาดจาก Defect

เมื่อได้ทำการคอมมิทไฟล์ไปแล้ว จะมีการdeployไปยังเครื่องของซอฟต์แวร์ทดสอบระบบได้ และหากเจอข้อผิดพลาดที่ได้ก็จะลิสออกมา และlogเป็นดิเฟค ในโปรแกรม QMS เดวेलอปเปอร์มีหน้าที่เข้าไปดูว่ามีดิเฟคตัวใดบ้างที่ถูกมอบหมายมาให้ตนเอง ซึ่งผู้ที่มอบหมายจะเป็นหัวหน้างาน จากนั้นจึงทำการแก้ไขให้สมบูรณ์ไม่ผิดพลาดใดๆและคอมมิทไฟล์ขึ้น SVN Server เพื่อรอการdeployอีกครั้ง และจะต้องเข้าไปเปลี่ยนสถานะในโปรแกรมQMSให้เป็น สถานะ Fixed เพื่อแจ้งให้ซอฟต์แวร์ทดสอบระบบรับทราบและทำการทดสอบระบบต่อไป



ภาพที่ 4.24 ตัวอย่างดิเฟคที่ได้รับมอบหมาย



ภาพที่ 4.25 ตัวอย่างรายละเอียดของดีเฟกต์ที่ได้รับมอบหมาย

## 4.2 ผลการวิเคราะห์ข้อมูล

ในการพัฒนาระบบงานนี้ขึ้นมา หากได้ศึกษาโครงสร้างของฐานข้อมูลตารางต่างๆก่อนลงมือทำ จะเป็นประโยชน์อย่างยิ่ง เนื่องจากจะทำให้ประหยัดเวลาในการทำงานลงไปได้มากพอสมควร รวมถึงเงื่อนไขเชิงธุรกิจต่างๆที่ค่อนข้างมีความซับซ้อนสูง ต้องมีความเข้าใจทุกเงื่อนไขในชิ้นงานที่ได้รับมอบหมาย หากทำงานไปโดยที่ขาดความเข้าใจ ชิ้นงานที่ออกมาก็จะเกิดความผิดพลาด และเกิดเป็นดีเฟกต์ และเมื่อผู้ที่ได้รับมอบหมายให้แก้ไขดีเฟกต์ชิ้นนั้นๆมาแก้ไขต่อ อาจทำให้ไม่เข้าใจในจุดประสงค์ หรือโค้ดที่ผู้พัฒนาคนแรกทำไว้ ทำให้ต้องเสียเวลามากขึ้นในการตามหาตัวผู้พัฒนาคนแรก

## 4.3 วิเคราะห์และวิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายในการปฏิบัติงานหรือการจัดทำโครงการ

โครงการชิ้นนี้มีจุดประสงค์เพื่อพัฒนา Web Application โดยใช้ความรู้ทางภาษา Java, HTML, Javascript, Typescript, SQL เป็นหลัก เพื่อตอบสนองความต้องการของผู้ใช้งานและมีประสิทธิภาพในการทำงานให้มากที่สุด โดยมีขั้นตอนการทำงานที่ง่ายต่อการใช้งาน รวมไปถึงรูปแบบการแสดงผลที่เข้าใจง่าย

ทั้งนี้ในการทำงานจริงก็สามารถทำงานได้เป็นที่น่าพอใจ ในส่วนที่ทาล่วงไปแล้วมีผลที่น่าพอใจและสามารถใช้งานได้จริงและไม่เกิดปัญหาที่เป็นข้อบกพร่องแต่อย่างใด สามารถ

บรรลุเป้าหมายของRequirement ได้สำเร็จเป็นส่วนใหญ่ แต่เนื่องจากเป็นงานที่ยังต้องมีการ  
ปรับแก้ Requirement อยู่เรื่อยๆ จึงทำให้ต้องคอยแก้ไขเปลี่ยนไปตามเอกสารSRSเวอร์ชันใหม่  
หรือตามที่ถูกคำเปลี่ยนแปลงความต้องการ



## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### 5.1 สรุปผลการดำเนินงาน

ผลการปฏิบัติงาน อยู่ในระดับที่คาดหวังไว้ เนื่องจากการในการทำงานจริงอาจจะไม่ตรงตามแผนไปบ้างเพราะต้องมีการปรับเปลี่ยนเอกสาร SRSตามความต้องการของลูกค้า หรือตามความเหมาะสมให้สอดคล้องกับกระบวนการทำงาน และปรับแก้ฐานข้อมูลที่ได้ออกแบบไว้ เพื่อให้ง่ายต่อการพัฒนา ดังนั้นจึงปรึกษาและสรุปร่วมกันกับพนักงานที่ปรึกษาให้พัฒนาในส่วนที่ไม่ได้รับผลกระทบโครงสร้างข้อมูลจากการเปลี่ยนแปลงดังกล่าว เพื่อให้ทางบริษัทปรับการออกแบบฐานข้อมูลที่ได้รับผลกระทบและแก้ไขระบบการพัฒนาระบบงานต่อไป

#### 5.2 แนวทางการแก้ไขปัญหา

1. ถ้าแสดงออกความคิดเห็น ต่อการทำงานของระบบ และความเข้าใจในการทำงานของแต่ละกระบวนการงาน
2. ศึกษาหลักการงานและเงื่อนไขต่างๆของงานส่วนนั้นๆที่ได้รับมอบหมาย เพื่อให้เข้าใจ และสามารถพัฒนาได้อย่างไม่มีข้อผิดพลาด
3. ถ้าที่จะซักถามข้อสงสัยในความพอดี และไม่จมกับปัญหาเป็นเวลานานๆ เพื่อไม่ให้ติดปัญหาเป็นเวลานานจะส่งผลกระทบต่อการทำงานของคนอื่น
4. ตรวจสอบการทำงานของตนเองและผู้ร่วมงานคนอื่น ทั้งก่อนและหลังการคอมมิตไฟล์งานลง SVN Server ทุกครั้งเพื่อลดปัญหา ในการทำงาน ซึ่งอาจเป็นการคอมมิตทับซ้อนกัน จะทำให้งานของคนที่คอมมิตก่อนหายไปเนื่องจากการคอมมิตอีกหลังจากนั้น

#### 5.3 ข้อเสนอแนะจากการดำเนินงาน

1. เนื่องจากในส่วนงานที่ทำนั้น ก่อนข้างซับซ้อนและมี กระบวนการที่ค่อนข้างเยอะและเนื่องด้วยไม่มีประสบการณ์ในด้านนี้มาก่อน จึงใช้เวลาไปค่อนข้างมากในการศึกษาและเข้าใจระบบในการทำงาน เพราะไม่มีเอกสารประกอบการศึกษา ส่วนใหญ่จะเป็นการสอนกันโดยปากต่อปาก หรือให้ทำความเข้าใจเองเป็นส่วนมาก

2. นักศึกษาควรกล้าที่จะแสดงออกถึงความพร้อมและชวนขายที่จะเรียนรู้ด้วยตัวเองและทำส่วนที่ทำได้ก่อนเมื่อมีโอกาสถามในข้อสงสัยถึงถามในยามที่จำเป็นเพื่อไม่รบกวนพนักงานคนอื่นมากเกินไป ควรมีความรับผิดชอบต่องานปรับตัวสู่สถานการณ์ทำงานจริงเพื่อประโยชน์ของตนและเพื่อนร่วมงาน อีกทั้งลดความกดดันต่อตนเอง เพราะการทำงานนั้น เหมือนการทดสอบการแก้ปัญหาของตนเองทุกวันและผลลัพธ์ของการทดสอบนั้นจะมีผลดีหรือไม่ก็อยู่ที่ผู้ปฏิบัติว่าเรียนรู้จากความสำเร็จและความผิดพลาดของตนเองได้แค่ไหน คำนึงกับเวลาที่ใช้ไปกับการปฏิบัติหรือไม่ ต่อให้เป็นความผิดพลาดหากได้เรียนรู้ก็คุ้มค่างกับเวลาที่จะเสียไปแต่ได้รับความรู้หรือวิธีแก้ปัญหาใหม่ๆ



TNI

## เอกสารอ้างอิง

1. Coding Basic [Online], Available: <http://www.codingbasic.com/html.html> [2016, August 31].
2. Mindphp [Online], Available: <http://www.mindphp.com/คู่มือ/73-คืออะไร/2185-java-คืออะไร.html> [2016, August 31].
3. 2013, Srisunthorn's Blog [Online], Available: <https://srisunthorn.wordpress.com/2013/03/13/1056/> [2016, August 31].
4. Mindphp [Online], Available: <http://www.mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html> [2016, August 31].
5. Nuttavut Thongjor, 2016, Babel Coder [Online], Available: <https://www.babelcoder.com/blog/posts/typescript-data-types#TypeScript-คืออะไร> [2016, August 31].
6. Nuttavut Thongjor, 2016, Babel Coder [Online], Available: <https://www.babelcoder.com/blog/posts/angular2-introduction-to-angular2-and-architecture-overview> [2016, August 31].
7. 2016, Wikipedia [Online], Available: <https://th.wikipedia.org/wiki/%E0%B9%80%E0%B8%88%E0%B8%8B%E0%B8%B1%E0%B8%99> [2016, August 31].
8. Nopphanan Mayoe, 2016, Programming Hunter [Online], Available: <http://programminghunter.blogspot.com/2015/07/what-is-hibernate-framework.html> [2016, August 31].
9. Mk, 2015, Blognone [Online], Available: <https://www.blognone.com/node/68057> [2016, August 31].
10. IBM [Online], Available: <http://www-03.ibm.com/software/products/en/application> [2016, August 31].
11. Chai Phonbopit, 2015, Devahoy [Online], Available: <http://devahoy.com/posts/getting-started-with-nodejs/> [2016, August 31].
12. Mindphp [Online], Available: <http://www.mindphp.com/คู่มือ/73-คืออะไร/2088-sql-คืออะไร.html> [2016, August 31].

13.     เกร็ดความรู้ [Online], Available: <http://www.xn--12cg1cxchd0a2gzc1c5d5a.net/css/>  
[2016, September 27]
14.     [Online], Available: <http://www.fusionidea.biz/spring-mvc-framework-part-1-ชินดีต้อนรับสู่-spring-mvc-framework/>



## ประวัติผู้จัดทำโครงการ

ชื่อ – สกุล

นางสาวสิริกาญจน์

กุลวุฒิวิลาส



วัน เดือน ปีเกิด

1 กันยายน พ.ศ.2537

ประวัติการศึกษา

ระดับประถมศึกษา

โรงเรียนประจักษ์ศิลปาคม

ระดับมัธยมศึกษา

โรงเรียนสตรีรัตนบุรี

ระดับอุดมศึกษา

คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีสารสนเทศ พ.ศ. 2556

สถาบันเทคโนโลยีไทย – ญี่ปุ่น

ทุนการศึกษา

- ไม่มี -

ผลงานที่ได้รับการตีพิมพ์

- ไม่มี -

TNI

TAI - NICHII INSTITUTE OF TECHNOLOGY