

การแปลงแผนภาพยูเอ็มแอลเป็นจาวา

นางสาวปรียาณูช ขลุ่ยศรีตระกูล

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)  
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)  
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2558

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

# Transformation of UML Diagrams to Java

Miss Preyanoot Kluisitrakul



A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2015

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

การแปลงแผนภาพยูเอ็มแอลเป็นจาวา

โดย

นางสาวปรียาณูช ขลุ่ยศรีตระกูล

สาขาวิชา

วิศวกรรมซอฟต์แวร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

รองศาสตราจารย์ ดร. ญาใจ ลิมปิยะภรณ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วน  
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร. สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(ศาสตราจารย์ ดร. บุญเสริม กิจศิริกุล)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร. ญาใจ ลิมปิยะภรณ์)

..... กรรมการภายนอกมหาวิทยาลัย

(อาจารย์ ดร. ภาสกร อภิรักษ์วรพินิต)

ปริยาณูช ขลุ่ยศรีตระกูล : การแปลงแผนภาพยูเอ็มแอลเป็นจาวา (Transformation of UML Diagrams to Java) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. ญาใจ ลิ้มปิยะกรณ, 79 หน้า.

แนวทางการสร้างโค้ดบนพื้นฐานแบบจำลองเชิงวัตถุมีสองประเภท ได้แก่ เชิงโครงสร้าง และเชิงพฤติกรรม แนวทางเชิงโครงสร้างจะสร้างโค้ดจากแบบจำลองเชิงสถิติ เช่น แผนภาพคลาส ในทางตรงกันข้าม โค้ดเป้าหมายที่ครอบคลุมพฤติกรรมของระบบสามารถถูกสร้างโดยการเทียบหรือแมปจากแผนภาพเชิงพฤติกรรมไปยังภาษาเชิงวัตถุ วิทยานิพนธ์ฉบับนี้ได้นำเสนอแนวทางหนึ่งสำหรับการสร้างโค้ดจาวาจากแผนภาพคลาสและแผนภาพลำดับของยูเอ็มแอล โดยใช้การสับเปลี่ยนเมทาเดตาเอกซ์เอ็มแอล หรือเอกซ์เอ็มไอ เป็นตัวแบบตรงกลางระหว่างกระบวนการดำเนินการแปลง ทั้งนี้ การสร้างโค้ดเชิงวัตถุแบบอัตโนมัติจากแผนภาพยูเอ็มแอลจะมีผลในการลดต้นทุน เพิ่มประสิทธิภาพการทำงาน และความถูกต้องของผลิตภัณฑ์



ภาควิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์

ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2558

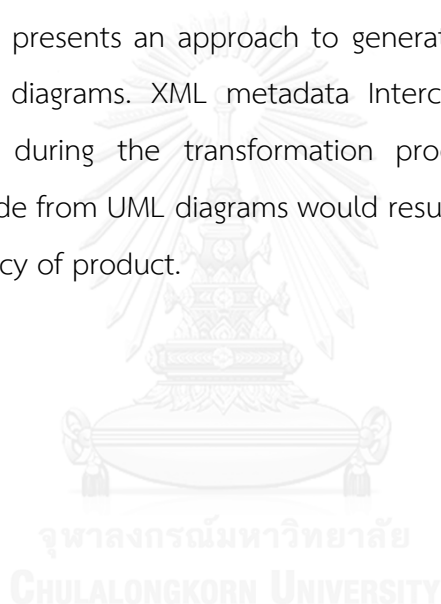
# # 5770938221 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: SEQUENCE DIAGRAM, CLASS DIAGRAM, CODE GENERATION, OBJECT-ORIENTED, METAMODEL.

PREYANOOT KLUISRITRAKUL: Transformation of UML Diagrams to Java.

ADVISOR: ASSOC. PROF. DR. YACHAI LIMPIYAKORN, 79 pp.

There are two approaches of object-oriented model based code generation, namely structural and behavioral. The structural approach generates code from static models, such as class diagrams. In contrast, the target code covering system behaviors could be generated by mapping from the behavior diagrams to object-oriented languages. This paper presents an approach to generating Java code from the UML class and sequence diagrams. XML metadata Interchange (XMI) is used as the intermediate model during the transformation process. Automated generation of object-oriented code from UML diagrams would result in cost reduction, increased efficiency, and accuracy of product.



Department: Computer Engineering      Student's Signature .....

Field of Study: Software Engineering      Advisor's Signature .....

Academic Year: 2015

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์จากรองศาสตราจารย์ ดร.ญาใจ ลิ้มปิยะกรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ได้สละเวลาให้ความรู้ คำปรึกษา ตรวจสอบและแก้ไขข้อผิดพลาดต่างๆ ตลอดจนการกำกับดูแลและติดตามความก้าวหน้า ทำให้งานวิจัยนี้ สำเร็จไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณ ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล และ อาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณบิดา มารดา และญาติพี่น้องที่ให้การสนับสนุน ให้กำลังใจที่ตีเสมอมา และสนับสนุนด้านทุนทรัพย์ในการศึกษา รวมไปถึงทุกท่านที่มีส่วนช่วยเหลือในการทำวิทยานิพนธ์ครั้งนี้ ซึ่งมีได้กล่าวนามในที่นี้ด้วย

ท้ายที่สุด ผู้วิจัยขอขอบคุณเพื่อนๆ พี่ๆ ทุกคน ที่คอยติดตามและให้กำลังใจ รวมถึงท่านอื่นๆ ที่มีได้กล่าววลงนามไว้ ณ ที่นี้ที่มีส่วนทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์บ้างไม่มากก็น้อยสำหรับผู้สนใจจะศึกษารายละเอียดต่อไป

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญรูปภาพ.....	ฎ
บทที่ 1 บทนำ .....	1
1.1. ความเป็นมาและความสำคัญของปัญหา.....	1
1.2. วัตถุประสงค์ของการวิจัย .....	2
1.3. ขอบเขตงานวิจัย .....	2
1.4. ขั้นตอนและวิธีการดำเนินการวิจัย.....	2
1.5. ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6. โครงสร้างของเนื้อหาในวิทยานิพนธ์.....	3
1.7. ผลงานตีพิมพ์จากวิทยานิพนธ์.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	4
2.1. ทฤษฎีที่เกี่ยวข้อง .....	4
2.1.1. แผนภาพลำดับ (Sequence Diagram).....	4
2.1.2. แผนภาพคลาส (Class Diagram).....	5
2.1.3. เอกซ์เอ็มไอ (XMI: XML Metadata Interchange).....	6
2.1.4. Visual Paradigm Tool.....	6
2.2. งานวิจัยที่เกี่ยวข้อง.....	7
2.2.1. A Simple Implementation of UML Sequence Diagram to Java Code Generation through XMI Representation [5] .....	7

2.2.2. Automatic generation of Java code from UML diagrams using UJECTOR [6].....	9
บทที่ 3 แนวคิดและวิธีดำเนินการวิจัย.....	12
3.1. สร้างแผนภาพลำดับ.....	14
3.2. สร้างแผนภาพคลาส.....	19
3.3. วิเคราะห์การสกัดข้อมูลในส่วนของคลาสและสร้างคลาสจากแผนภาพคลาส.....	22
3.4. วิเคราะห์การสกัดข้อมูลในส่วนของแอทริบิวต์และสร้างแอทริบิวต์จากแผนภาพคลาส.....	24
3.5. วิเคราะห์การสกัดข้อมูลในส่วนของเมทอดและสร้างเมทอดจากแผนภาพคลาส.....	25
3.6. วิเคราะห์ความสัมพันธ์ระหว่างเมทอดที่ติดกับการรับส่งข้อมูลจากแผนภาพลำดับ.....	27
3.7. เพิ่มรายละเอียดของการสกัดข้อมูลของเงื่อนไขจากแผนภาพลำดับ.....	30
3.8. สร้างรหัสต้นทาง.....	34
บทที่ 4 การพัฒนาเครื่องมือ.....	35
4.1. ความต้องการเชิงฟังก์ชัน.....	35
4.2. การวิเคราะห์ความต้องการและแผนภาพฟังก์ชันงานของระบบ.....	35
4.3. สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือสนับสนุน.....	37
4.4. การออกแบบหน้าจอ.....	38
4.5. ขั้นตอนการทำงานของเครื่องมือ.....	42
บทที่ 5 การทดสอบและการวิเคราะห์ผล.....	59
5.1. วัตถุประสงค์ของการทดสอบ.....	59
5.2. การทดสอบระบบ.....	59
5.3. สรุปผลการทดสอบ.....	66
บทที่ 6 สรุปผลการวิจัย.....	67
6.1. สรุปผลงานวิจัย.....	67



6.2. ข้อจำกัดของงานวิจัย.....	68
6.3. งานวิจัยในอนาคต.....	68
รายการอ้างอิง.....	69
ภาคผนวก.....	70
ภาคผนวก ก. การติดตั้งซอฟต์แวร์โหนดเจเอส (NodeJs).....	71
ภาคผนวก ข. การติดตั้งส่วนเสริมกัล์พ์ (GulpJs).....	74
ภาคผนวก ค. การติดตั้งอินเทลลิเจไอเดีย (intellij idea).....	76
ประวัติผู้เขียนวิทยานิพนธ์.....	79



## สารบัญตาราง

ตารางที่ 1 รายละเอียดของส่วนประกอบย่อยในโครงสร้างของคลาสที่ใช้ในงานวิจัย .....	23
ตารางที่ 2 รายละเอียดของส่วนประกอบย่อยในโครงสร้างแอทริบิวต์ของโค้ดจาวาที่ใช้ในงานวิจัย..	25
ตารางที่ 3 รายละเอียดของส่วนประกอบย่อยในโครงสร้างเมทอดของโค้ดจาวาที่ใช้ในงานวิจัย .....	26
ตารางที่ 4 รายละเอียดของส่วนประกอบในโครงสร้างการรับส่งข้อมูลของโค้ดจาวาที่ใช้ในงานวิจัย.....	29
ตารางที่ 5 รายละเอียดของส่วนประกอบย่อยในโครงสร้างเงื่อนไขของโค้ดจาวาที่ใช้ในงานวิจัย .....	31
ตารางที่ 6 รายละเอียดของส่วนประกอบย่อยในโครงสร้างวนซ้ำของโค้ดจาวาที่ใช้ในงานวิจัย .....	32
ตารางที่ 7 รายละเอียดของส่วนประกอบย่อยในโครงสร้างสวิตช์เคสของโค้ดจาวาที่ใช้ในงานวิจัย...	33
ตารางที่ 8 ทดสอบการนำเข้าของแผนภาพลำดับในรูปแบบของไฟล์เอกซ์เอ็มไอ .....	60
ตารางที่ 9 ทดสอบการนำเข้าของแผนภาพคลาสในรูปแบบของไฟล์เอกซ์เอ็มไอ .....	60
ตารางที่ 10 ทดสอบเครื่องมือในการออกแบบการใช้งานและกำหนดตำแหน่งของแต่ละส่วนประกอบ.....	61
ตารางที่ 11 ทดสอบเครื่องมือในการสกัดข้อมูล เพื่อให้ได้โครงสร้างของคลาส.....	62
ตารางที่ 12 ทดสอบเครื่องมือในการสกัดข้อมูล เพื่อให้ได้ความสัมพันธ์ของเมทอด .....	63
ตารางที่ 13 ทดสอบเครื่องมือในการสกัดข้อมูล เพื่อให้ได้ความสัมพันธ์เงื่อนไข .....	64
ตารางที่ 14 ทดสอบเครื่องมือในการรวบรวมส่วนประกอบย่อยทั้งหมด เพื่อให้ได้โค้ดจาวา.....	65

## สารบัญรูปภาพ

รูปที่ 1 องค์ประกอบหลักของแผนภาพลำดับ [2] .....	4
รูปที่ 2 องค์ประกอบหลักของแผนภาพคลาส [3] .....	5
รูปที่ 3 คุณลักษณะของเครื่องมือ Visual Paradigm [4].....	7
รูปที่ 4 กระบวนการดำเนินการ [5] .....	8
รูปที่ 5 รูปแบบเอกซ์เอ็มไอจากเครื่องมือ BOUML [5].....	8
รูปที่ 6 โค้ดจาวา [5].....	8
รูปที่ 7 สถาปัตยกรรมของ UJECTOR [6] .....	9
รูปที่ 8 UJECTOR Implementation package diagram [6] .....	10
รูปที่ 9 โครงสร้างหน่วยความจำภายในสำหรับเก็บการสร้างโค้ด [6] .....	11
รูปที่ 10 แผนภาพกิจกรรมแนวคิดและวิธีการวิจัย.....	12
รูปที่ 11 Tailored OMG Sequence diagram metamodel [7].....	13
รูปที่ 12 OMG Class diagram metamodel [7].....	13
รูปที่ 13 ตัวอย่างแผนภาพลำดับที่ได้จากเครื่องมือวิชชัวพาราตามยูเอ็มแอล.....	14
รูปที่ 14 ตัวอย่างไฟล์เอกซ์เอ็มไอในส่วนของ “WebCyberAgent” ที่ได้จากเครื่องมือ .....	15
รูปที่ 15 ตัวอย่างไฟล์เอกซ์เอ็มไอในส่วนของ “loginMCA(user, password)” ที่ได้จากเครื่องมือ ..	15
รูปที่ 16 การดาวน์โหลดโปรแกรมประยุกต์ผ่านคอมมานไลน์ด้วยคำสั่ง “npm install gulp -g” ...	16
รูปที่ 17 การเรียกใช้โปรแกรมประยุกต์ผ่านคอมมานไลน์เพื่อเปลี่ยนรูปแบบไฟล์เป็นเจสันของ แผนภาพลำดับ .....	17
รูปที่ 18 ตัวอย่างรูปแบบไฟล์เจสันของแผนภาพลำดับที่ได้จากโปรแกรมประยุกต์ .....	18
รูปที่ 19 ตัวอย่างแผนภาพคลาสที่ได้จากเครื่องมือวิชชัวพาราตามยูเอ็มแอล .....	19
รูปที่ 20 ตัวอย่างไฟล์เอกซ์เอ็มไอของแผนภาพคลาส.....	20

รูปที่ 21 การเรียกใช้โปรแกรมประยุกต์ผ่านคอมมานไลน์เพื่อเปลี่ยนรูปแบบไฟล์เป็นเจสันของ แผนภาพคลาส .....	21
รูปที่ 22 ตัวอย่างรูปแบบไฟล์เจสันของแผนภาพคลาสที่ได้จากโปรแกรมประยุกต์.....	21
รูปที่ 23 โครงสร้างยูเอ็มแอลโมเดลของแผนภาพคลาสที่ใช้ในงานวิจัย .....	22
รูปที่ 24 ข้อมูลของคลาส.....	23
รูปที่ 25 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากการสร้างคลาส.....	23
รูปที่ 26 ข้อมูลของแอทริบิวต์.....	24
รูปที่ 27 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากการสร้างแอทริบิวต์.....	25
รูปที่ 28 ข้อมูลของเมทอด.....	26
รูปที่ 29 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากการสร้างเมทอด.....	27
รูปที่ 30 ข้อมูลของการรับส่งข้อมูล .....	28
รูปที่ 31 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากการรับส่งข้อมูล .....	29
รูปที่ 32 ข้อมูลของเงื่อนไข.....	30
รูปที่ 33 ข้อมูลของวงซ้ำ.....	31
รูปที่ 34 ข้อมูลของสวิตช์เคส.....	32
รูปที่ 35 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากความสัมพันธ์ของคลาสอย่างมีเงื่อนไข.....	33
รูปที่ 36 ตัวอย่างรูปแบบไฟล์จาวา.....	34
รูปที่ 37 แผนภาพยูสเคสของระบบแปลงแผนภาพยูเอ็มแอลเป็นจาวา.....	36
รูปที่ 38 หน้าจอเครื่องมือของการแปลงแผนภาพยูเอ็มแอลไปเป็นจาวา.....	38
รูปที่ 39 หน้าจอเครื่องมือของผลสรุปของการแปลงโค้ดจาวา.....	40
รูปที่ 40 หน้าจอเครื่องมือของการแสดงผลโค้ดจาวา.....	41
รูปที่ 41 หน้าจอหลังจากการแปลงโค้ดจาวาเสร็จสมบูรณ์.....	42
รูปที่ 42 การสร้างแผนภาพลำดับด้วยเครื่องมือวิชชัวพาราตามยูเอ็มแอล.....	43
รูปที่ 43 การสร้างแผนภาพคลาสด้วยเครื่องมือวิชชัวพาราตามยูเอ็มแอล.....	43

รูปที่ 44 รูปแบบไฟล์เอกซ์เอ็มไอที่ได้จากการนำออกของแผนภาพลำดับด้วยเครื่องมือวิชัวพารา ตามยูเอ็มแอล.....	44
รูปที่ 45 รูปแบบไฟล์เอกซ์เอ็มไอที่ได้จากการนำออกของแผนภาพคลาสด้วยเครื่องมือวิชัวพารา ตามยูเอ็มแอล.....	44
รูปที่ 46 คำสั่งการแปลงรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจสันของแผนภาพลำดับ .....	45
รูปที่ 47 คำสั่งการแปลงรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจสันของแผนภาพคลาส.....	45
รูปที่ 48 รูปแบบไฟล์เจสันที่ได้จากคำสั่งการแปลงรูปแบบไฟล์ของแผนภาพลำดับ.....	46
รูปที่ 49 รูปแบบไฟล์เจสันที่ได้จากคำสั่งการแปลงรูปแบบไฟล์ของแผนภาพคลาส.....	46
รูปที่ 50 คลาสของ “WebCyberAgent” .....	47
รูปที่ 51 คลาสของ “MoileCyberAgent” .....	47
รูปที่ 52 คลาสของ “KeyCodeService” .....	47
รูปที่ 53 คลาสของ “MainMenuService” .....	48
รูปที่ 54 คลาสของ “CustomerService” .....	48
รูปที่ 55 คลาสของ “PolicyStatusService” .....	48
รูปที่ 56 แอทริบิวต์ที่เป็นส่วนหนึ่งของคลาส “MobileCyberAgent” .....	49
รูปที่ 57 แอทริบิวต์ที่เป็นส่วนหนึ่งของคลาส “MainMenuService” .....	49
รูปที่ 58 เมท็อดของ “WebCyberAgent” .....	50
รูปที่ 59 เมท็อดของ “MobileCyberAgent” .....	50
รูปที่ 60 เมท็อดของ “KeyCodeService” .....	50
รูปที่ 61 เมท็อดของ “MainMenuService” .....	51
รูปที่ 62 เมท็อดของ “CustomerService” .....	51
รูปที่ 63 เมท็อดของ “PolicyStatusService” .....	51
รูปที่ 64 ความสัมพันธ์ของคลาส “WebCyberAgent” .....	52
รูปที่ 65 ความสัมพันธ์ของคลาส “MobileCyberAgent” .....	52

รูปที่ 66 ความสัมพันธ์ของคลาส “KeyCodeService” .....	53
รูปที่ 67 ความสัมพันธ์ของคลาส “MainMenuService” .....	53
รูปที่ 68 ความสัมพันธ์ของคลาส “CustomerService” .....	53
รูปที่ 69 ความสัมพันธ์ของคลาส “PolicyStatusService” .....	54
รูปที่ 70 คลาสที่มีเงื่อนไขของออลเทอร์เนทิฟและวนซ้ำ.....	54
รูปที่ 71 คลาสที่มีเงื่อนไขของสวิตช์เคส .....	55
รูปที่ 72 ไฟล์จาวาของ “WebCyberAgent” .....	55
รูปที่ 73 ไฟล์จาวาของ “MobileCyberAgent” .....	56
รูปที่ 74 ไฟล์จาวาของ “KeyCodeService” .....	56
รูปที่ 75 ไฟล์จาวาของ “MainMenuService” .....	57
รูปที่ 76 ไฟล์จาวาของ “CustomerService” .....	57
รูปที่ 77 ไฟล์จาวาของ “PolicyStatusService” .....	58
รูปที่ 78 เข้าสู่การติดตั้งของโนนดเจเอส.....	71
รูปที่ 79 การกำหนดค่าเริ่มต้นและการติดตั้งซอฟต์แวร์โนนดเจเอส.....	72
รูปที่ 80 ติดตั้งโนนดเจเอสเสร็จสมบูรณ์.....	72
รูปที่ 81 การตรวจสอบการติดตั้งซอฟต์แวร์ผ่านตัวพร้อมคำสั่งงาน .....	73
รูปที่ 82 รูปแบบการทำงาน .....	74
รูปที่ 83 กำหนดการตั้งค่าฟังก์ชันของส่วนเสริมที่ใช้ในงานวิจัย .....	75
รูปที่ 84 เข้าสู่การติดตั้งของอินเทลลิเจไอเดีย.....	76
รูปที่ 85 การติดตั้งซอฟต์แวร์อินเทลลิเจไอเดีย.....	77
รูปที่ 86 การติดตั้งซอฟต์แวร์อินเทลลิเจไอเดียเสร็จสมบูรณ์.....	77
รูปที่ 87 หน้าจอการทำงานซอฟต์แวร์ของอินเทลลิเจไอเดีย .....	78

# บทที่ 1

## บทนำ

### 1.1. ความเป็นมาและความสำคัญของปัญหา

การพัฒนาซอฟต์แวร์ จะใช้แผนภาพยูเอ็มแอล (UML Diagrams) เป็นส่วนหนึ่งในการวางโครงสร้างการดำเนินการ แผนภาพนี้จึงมีความสำคัญที่สามารถวิเคราะห์ พัฒนา และอธิบายการออกแบบระบบเชิงวัตถุ (Object Oriented) โดยพิจารณาจากมุมมองที่เกิดขึ้น เพื่อนำไปเป็นแนวทางโครงสร้างของซอฟต์แวร์ ใช้สื่อสารสำหรับการแลกเปลี่ยนและทำความเข้าใจให้ตรงกันระหว่างผู้ร่วมงาน และช่วยสนับสนุนโมเดลต่างๆของซอฟต์แวร์ด้วย การปฏิสัมพันธ์นั้น เป็นกลไกสำหรับการอธิบายถึงพฤติกรรมของระบบ ซึ่งยังไม่สามารถที่จะอธิบายขั้นตอนการทำงานของระบบได้อย่างชัดเจน นอกจากนี้การรับส่งข้อมูลก็เป็นสิ่งสำคัญสำหรับการทำความเข้าใจของสถานการณ์ เพื่อให้บรรลุความเข้าใจร่วมกันของสถานการณ์ที่เกิดขึ้น และการสื่อสารระหว่างขั้นตอนการดำเนินการของกระบวนการที่ถูกต้องจะต้องเป็นไปตามข้อกำหนดระหว่างขั้นตอนการออกแบบรายละเอียดด้วยแผนภาพยูเอ็มแอล ซึ่งจะอธิบายถึงลำดับความเป็นไปได้ทั้งหมดของเหตุการณ์หรือร่องรอยของระบบ และจะได้รับการบันทึกการสื่อสารผ่านทางรูปแบบของแผนภาพลำดับ

แผนภาพลำดับ (Sequence Diagram) เป็นแผนภาพที่มีการใช้อย่างแพร่หลาย โดยจะใช้อธิบายการทำงานของแต่ละยูสเคส เพื่อแสดงถึงขั้นตอนการทำงานและลำดับการสื่อสารระหว่างอ็อบเจกต์ ซึ่งทำให้มองเห็นภาพของระบบได้อย่างชัดเจน พร้อมทั้งแสดงให้เห็นถึงลำดับของเวลา สำหรับผู้พัฒนาระบบ และแผนภาพคลาส (Class Diagram) คือแผนภาพที่ใช้แสดงคลาส และความสัมพันธ์ในแง่ต่างๆ ระหว่างคลาส โดยจะเป็นความสัมพันธ์เชิงสถิต (Static Relationship) และความสัมพันธ์เชิงพลวัต (Dynamic Relationship) ซึ่งสามารถบ่งบอกถึงความสัมพันธ์ระหว่างคลาสได้ชัดเจนอีกด้วย นอกจากนี้ แผนภาพยูเอ็มแอลยังถูกสร้างขึ้นบนพื้นฐานแนวคิดของระบบเชิงวัตถุ ทั้งระดับคลาสและขั้นตอนของการดำเนินการ ดังนั้นจึงเหมาะสำหรับภาษาเชิงวัตถุ ตัวอย่างเช่น ภาษาจาวา และภาษาซีพลัสพลัส สามารถที่จะแปลงแผนภาพยูเอ็มแอลไปเป็นภาษาเชิงวัตถุได้ ซึ่งจะทำให้มีความสอดคล้องและเพิ่มประสิทธิภาพมากขึ้น

งานวิจัยได้เสนอแนวคิดการวิจัยสำหรับการสร้างโค้ดจาวาจากเมตาโมเดลของแผนภาพลำดับยูเอ็มแอล และได้ทำการศึกษาเอกซ์เอ็มไอ (XMI : XML Metadata Interchange) ที่ใช้เป็นสื่อกลางสำหรับการแปลงระหว่างโมเดล รวมถึงเป็นองค์ประกอบของรูปแบบและข้อมูลที่เกี่ยวข้อง

## 1.2. วัตถุประสงค์ของการวิจัย

นำเสนอวิธีการและพัฒนาเครื่องมือสำหรับการสร้างโค้ดจาวาจากแผนภาพลำดับยูเอ็มแอล เพื่อปรับปรุงกระบวนการซอฟต์แวร์

## 1.3. ขอบเขตงานวิจัย

- 1) แหล่งข้อมูลในการสร้างแผนภาพยูเอ็มแอล (Unified Modelling Language) ใช้เครื่องมือ Visual Paradigm สำหรับยูเอ็มแอล เวอร์ชัน 13.1
- 2) การแปลงแผนภาพลำดับและแผนภาพคลาสเป็นวากยสัมพันธ์ของภาษาจาวาเท่านั้น
- 3) ครอบคลุมการสร้างโค้ดที่ประกอบด้วยโครงสร้างเงื่อนไขและวนซ้ำ
- 4) ความสัมพันธ์ของแผนภาพลำดับจะมีลักษณะแบบเจเนอรัลไรเซชัน (Generalization) ที่ไม่เกิน 2 ลำดับ
- 5) ผลลัพธ์รหัสต้นทางที่ได้ถูกพัฒนาขึ้นมาจากจาวาแอปพลิเคชัน

## 1.4. ขั้นตอนและวิธีการดำเนินการวิจัย

- 1) ศึกษาและทำความเข้าใจเกี่ยวกับโครงสร้างของโค้ดจาวา
- 2) ศึกษาแนวทาง ทฤษฎีในการแก้ไขและงานวิจัยที่เกี่ยวข้อง
- 3) ศึกษาการใช้เครื่องมือวิชชัวพาราตามยูเอ็มแอล (Visual Paradigm for UML) เวอร์ชัน 13.1
- 4) วิเคราะห์และออกแบบการแก้ไขปัญหา
- 5) ตั้งสมมติฐานที่เกี่ยวข้องกับงานวิจัย
- 6) พัฒนาระบบ
- 7) ทดสอบและประเมินผลงานวิจัย
- 8) สรุปผลงานวิจัย
- 9) ตีพิมพ์ผลงานวิจัย
- 10) จัดทำเล่มวิทยานิพนธ์



### 1.5. ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีการออกแบบแผนภาพยูเอ็มแอล และการแปลงแผนภาพยูเอ็มแอลเป็นรูปแบบไฟล์ เอกซ์เอ็มไอ พร้อมทั้งได้เครื่องมือสนับสนุนที่สามารถนำมาสร้างเป็นโค้ดจาวา ซึ่งจะช่วยลดทรัพยากร การพัฒนาซอฟต์แวร์

### 1.6. โครงสร้างของเนื้อหาในวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 6 บทด้วยกันคือ บทที่ 1 อธิบายถึงที่มาและความสำคัญของปัญหา รวมถึงขอบเขตและประโยชน์ของงานวิจัย บทที่ 2 อธิบายถึงทฤษฎีที่เกี่ยวข้อง และงานวิจัยที่เกี่ยวข้อง บทที่ 3 อธิบายถึงแนวคิดและวิธีการดำเนินการวิจัยในการแปลงแผนภาพ ยูเอ็มแอลเป็นโค้ดจาวา บทที่ 4 อธิบายถึงวิธีการพัฒนาเครื่องมือสนับสนุนแนวคิดของงานวิจัย บทที่ 5 อธิบายถึงวิธีการทดสอบและวิเคราะห์ผลและในบทสุดท้ายจะสรุปงานวิจัยทั้งหมด รวมถึงงานวิจัยในอนาคต

### 1.7. ผลงานตีพิมพ์จากวิทยานิพนธ์

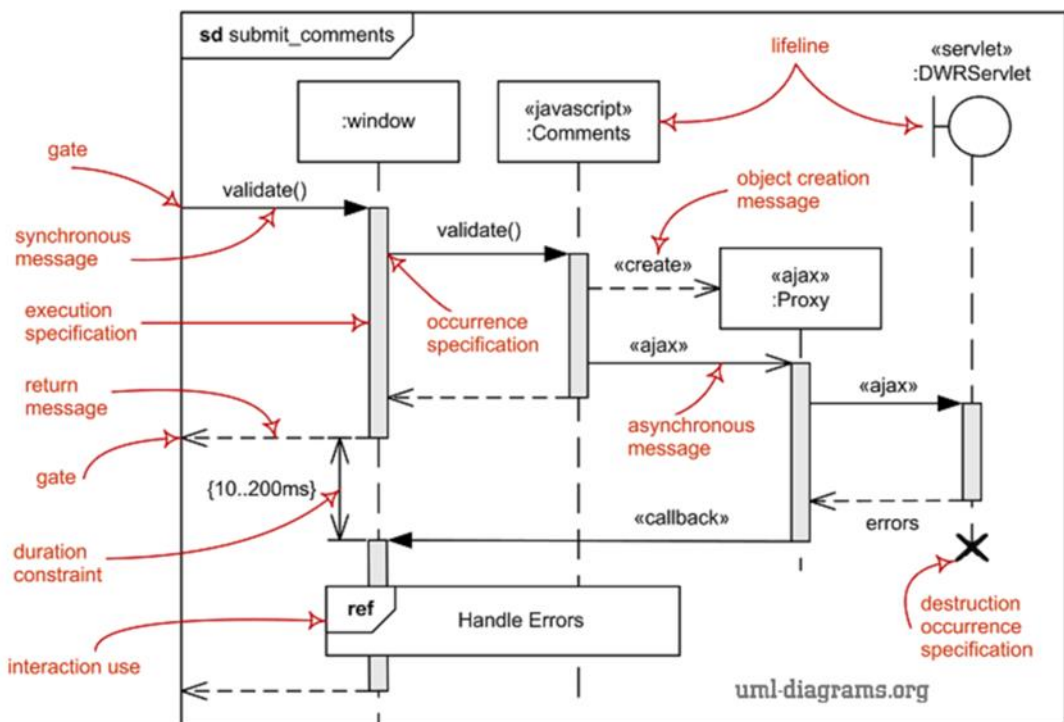
ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์ในรายงานสืบเนื่องจากการประชุมวิชาการระดับนานาชาติเรื่อง “Generation of Java Code from UML Sequence and Class Diagrams”, Preyanoot Kluisitrakul and Yachai Limpiyakorn, Proceeding of 7th International Conference on Information Science and Application (ICISA 2016), February 15-18, 2016, Ho Chi Minh City, Vietnam

## บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

### 2.1. ทฤษฎีที่เกี่ยวข้อง

#### 2.1.1. แผนภาพลำดับ (Sequence Diagram)

แผนภาพลำดับเป็นแผนภาพที่แสดงถึงลำดับเหตุการณ์ ซึ่งจะมุ่งเน้นสำหรับการอธิบายกิจกรรมหรือ ปฏิสัมพันธ์ (Interaction) ระหว่างอ็อบเจกต์ตามลำดับของเวลาที่เกิดขึ้น (Time-Ordering Description) หรือเป็นลำดับก่อนหลังในการส่งข้อความ (message) [1]



รูปที่ 1 องค์ประกอบหลักของแผนภาพลำดับ [2]

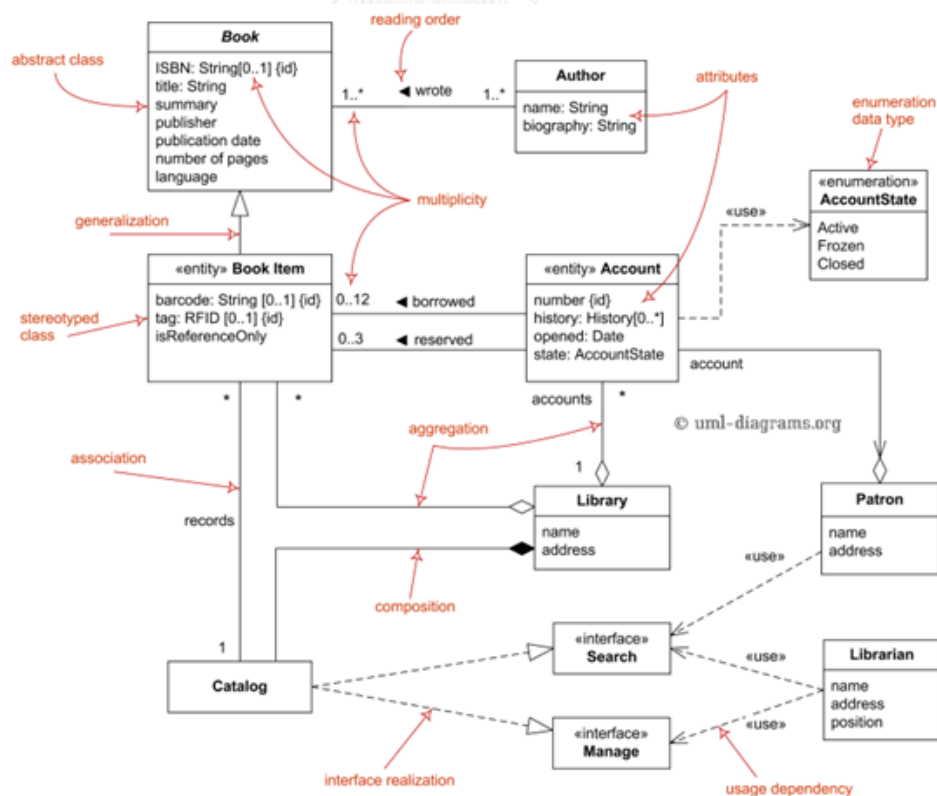
รูปที่ 1 แสดงองค์ประกอบหลักของแผนลำดับภาพ ประกอบด้วย

- Lifeline เป็นองค์ประกอบของชื่อที่แสดงถึงการมีส่วนร่วมในการทำงานร่วมกัน
- Message เป็นการสื่อสารระหว่างอ็อบเจกต์ของการรับส่งข้อมูล
- Occurrence specification เป็นการอธิบายเหตุการณ์ของการทำงานร่วมกันในช่วงจุดเริ่มต้นหรือจุดสิ้นสุดของข้อความ

- Interaction fragment เป็นองค์ประกอบที่แสดงถึงปฏิสัมพันธ์การทำงานร่วมกันของแต่ละอ็อบเจกต์
- Gate เป็นจุดที่เชื่อมต่อของข้อความจากภายนอก ซึ่งจะนำมาเป็นส่วนหนึ่งในการทำงานร่วมกันของอ็อบเจกต์ภายใน
- Combined fragment เป็นการแสดงการทำงานร่วมกันของอ็อบเจกต์ โดยแสดงในรูปแบบของออลเทอร์นาทิฟ (alternatives) ทางเลือก (options) และวนซ้ำ (loop)

### 2.1.2. แผนภาพคลาส (Class Diagram)

แผนภาพคลาสเป็นแบบจำลองโครงสร้าง (Structural Model) คือมุมมองของระบบที่เน้นโครงสร้างของวัตถุ รวมทั้งคลาสของวัตถุ (Class) ความสัมพันธ์ระหว่างคลาส (Relationship) แอทริบิวต์ (Attribute) และโอเปอเรชัน (Operation) ความสัมพันธ์ที่แสดงเป็นความสัมพันธ์เชิงสถิติ คือความสัมพันธ์ระหว่างคลาสต่างๆที่มีอยู่แล้ว ไม่ใช่ความสัมพันธ์ที่เกิดขึ้นเนื่องจากกิจกรรม จึงไม่สามารถเห็นการเปลี่ยนแปลงต่างๆที่เกิดขึ้นได้



รูปที่ 2 องค์ประกอบหลักของแผนภาพคลาส [3]

รูปที่ 2 แสดงองค์ประกอบหลักของแผนภาพคลาส ประกอบด้วย

- ชื่อคลาส เป็นการแสดงชื่อของแต่ละคลาส
- แอททริบิวต์ เป็นคุณสมบัติหรือลักษณะของเอนทิตี โดยการแสดงรายละเอียด
- เมทอด (Method) เป็นวิธีการทำงานของคลาส
- ความสัมพันธ์ เป็นการแสดงถึงความสัมพันธ์ระหว่างคลาสกับคลาส

### 2.1.3. เอกซ์เอ็มไอ (XMI: XML Metadata Interchange)

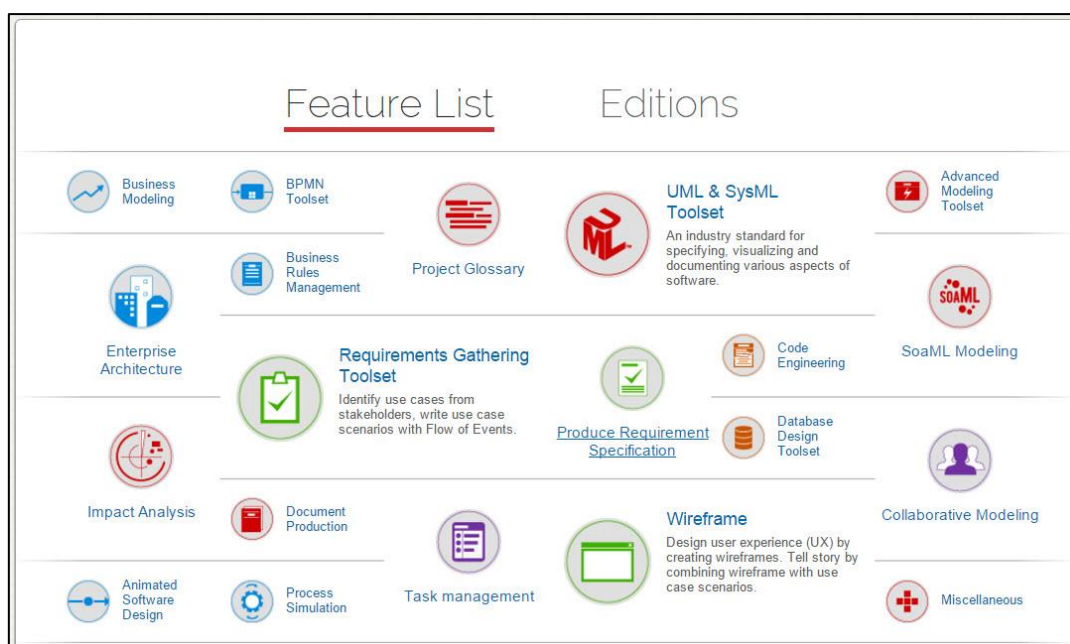
เอกซ์เอ็มไอเป็นรูปแบบในการแสดงรายละเอียดข้อมูลของแผนภาพยูเอ็มแอล ซึ่งใช้สำหรับการเปลี่ยนแปลงข้อมูลระหว่างแบบจำลองแผนภาพยูเอ็มแอล เอกซ์เอ็มไอจึงมีความเกี่ยวข้องกับข้อมูลเมตาเดตาของเอกซ์เอ็มแอล (XML) ซึ่งถูกนิยามไว้ดังนี้

- 1) ใช้สำหรับเป็นข้อกำหนดในการสร้างหรือจัดทำเอกสาร
- 2) เป็นการแสดงถึงรูปแบบขององค์ประกอบของเอกซ์เอ็มแอล ประเภท (Type) แอททริบิวต์ และอีลิเมนต์ (Element)
- 3) โครงร่างของเอกซ์เอ็มแอลจะไว้สำหรับตรวจสอบเอกซ์เอ็มไอ

### 2.1.4. Visual Paradigm Tool

วิชชัวาราดตามยูเอ็มแอล เป็นเครื่องมือที่ช่วยสำหรับพัฒนาการวิเคราะห์และออกแบบระบบของผู้ใช้หลายประเภท ได้แก่ วิศวกรซอฟต์แวร์ (Software Engineers) นักวิเคราะห์ระบบ (System Analysts) นักวิเคราะห์ธุรกิจ (Business Analysts) และนักออกแบบโครงสร้างของระบบ (System Architects) และยังสามารถสนับสนุนการออกแบบ ยกตัวอย่างเช่น

- สนับสนุนการออกแบบระบบซอฟต์แวร์ขนาดใหญ่
- สนับสนุนแผนภาพยูเอ็มแอลทั้งหมด
- สร้างเอกสารในรูปแบบของพีดีเอฟ (PDF) และเอชทีเอ็มแอล (HTML)
- สามารถนำเข้า (import) และส่งออก (export) ได้โดยง่าย
- สนับสนุนรูปแบบที่ใช้ในฟังก์ชันต่างๆ เช่น การเปลี่ยนจากภาษาอังกฤษเป็นภาษาจีนได้

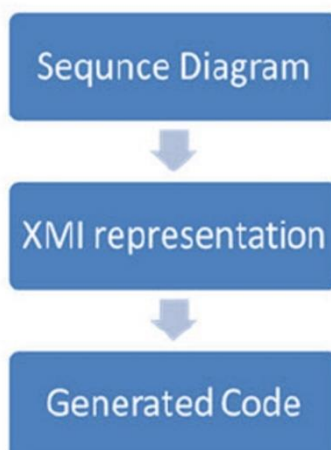


รูปที่ 3 คุณลักษณะของเครื่องมือ Visual Paradigm [4]

## 2.2. งานวิจัยที่เกี่ยวข้อง

### 2.2.1. A Simple Implementation of UML Sequence Diagram to Java Code Generation through XMI Representation [5]

งานวิจัยนี้นำเสนอพื้นฐานการประยุกต์ของโค้ดจาวาจากแผนภาพลำดับด้วยเอกซ์เอ็มไอ ด้วย BOUML ซึ่งเป็นเครื่องมือที่ช่วยสร้างแผนภาพยูเอ็มแอล และยังใช้สำหรับการแปลงจากแผนภาพยูเอ็มแอลไปเป็นรูปแบบเอกซ์เอ็มไอ ข้อจำกัดของงานวิจัยนี้ คือ ไม่ครอบคลุมเงื่อนไข ได้แก่ ออลเทอร์นาทีฟ (if-else) ทางเลือก (switch) และวนซ้ำ (for / do-while / while) รูปที่ 4 แสดงถึงกระบวนการดำเนินการแปลงจากแผนภาพลำดับไปเป็นโค้ดจาวา ซึ่งการแปลงแผนภาพลำดับไปยังเอกซ์เอ็มไอ จะใช้เครื่องมือ BOUML ที่แสดงดังรูปที่ 5 และนำไฟล์เอกซ์เอ็มไอที่ได้มาสกัดไปเป็นโค้ดจาวา แสดงดังรูปที่ 6



รูปที่ 4 กระบวนการดำเนินการ [5]

```

24 <packagedElement xmi:type="uml:Class" name="A" xmi:id="BOUML_0x1f412_4" visibility="public">
25 </packagedElement>
26 <packagedElement xmi:type="uml:Class" name="B" xmi:id="BOUML_0x1f492_4" visibility="public">
27 <ownedOperation xmi:type="uml:Operation" name="sum" xmi:id="BOUML_0x1f412_2" visibility="public" isAbstract="f
28 <ownedParameter name="return" xmi:id="BOUML_return_1" direction="return">
29 <type xmi:type="uml:Class" xmi:idref="BOUML_datatype_0"/>
30 </ownedParameter>
31 <ownedParameter xmi:type="uml:Parameter" name="num1" xmi:id="BOUML_op_param_1" direction="in">
32 <type xmi:type="uml:Class" xmi:idref="BOUML_datatype_1"/>
33 </ownedParameter>
34 <ownedParameter xmi:type="uml:Parameter" name="num2" xmi:id="BOUML_op_param_2" direction="in">
35 <type xmi:type="uml:Class" xmi:idref="BOUML_datatype_1"/>
36 </ownedParameter>
37 </ownedOperation>
38 <ownedOperation xmi:type="uml:Operation" name="minus" xmi:id="BOUML_0x1f492_2" visibility="public" isAbstract=
39 <ownedParameter name="return" xmi:id="BOUML_return_2" direction="return">
40 <type xmi:type="uml:Class" xmi:idref="BOUML_datatype_0"/>
41 </ownedParameter>
42 <ownedParameter xmi:type="uml:Parameter" name="num1" xmi:id="BOUML_op_param_3" direction="in">
43 <type xmi:type="uml:Class" xmi:idref="BOUML_datatype_1"/>
44 </ownedParameter>
45 <ownedParameter xmi:type="uml:Parameter" name="num2" xmi:id="BOUML_op_param_4" direction="in">
46 <type xmi:type="uml:Class" xmi:idref="BOUML_datatype_1"/>
47 </ownedParameter>
48 </ownedOperation>
  
```

รูปที่ 5 รูปแบบเอกซ์เอ็มไอจากเครื่องมือ BOUML [5]

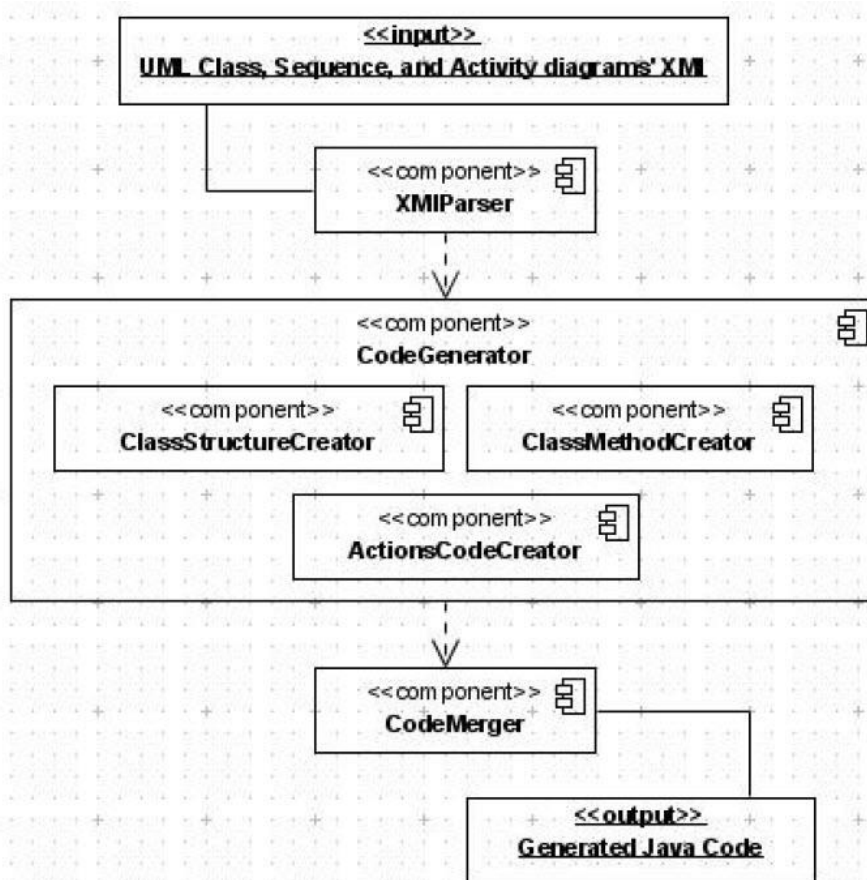
```

public class A
{
    public sum(int num1)
    {
    }
    Sum();
}
  
```

รูปที่ 6 โค้ดจาวา [5]

## 2.2.2. Automatic generation of Java code from UML diagrams using UJECTOR [6]

งานวิจัยนี้นำเสนอการสร้างของโค้ดจาวาแบบอัตโนมัติจากแผนภาพยูเอ็มแอลโดยใช้เครื่องมือที่ชื่อว่า UJECTOR ซึ่งแผนภาพที่ใช้คือแผนภาพคลาส แผนภาพลำดับ และแผนภาพกิจกรรม เพื่อให้ได้โครงสร้างของโค้ดที่มีความสัมพันธ์เชิงพฤติกรรมของระบบ

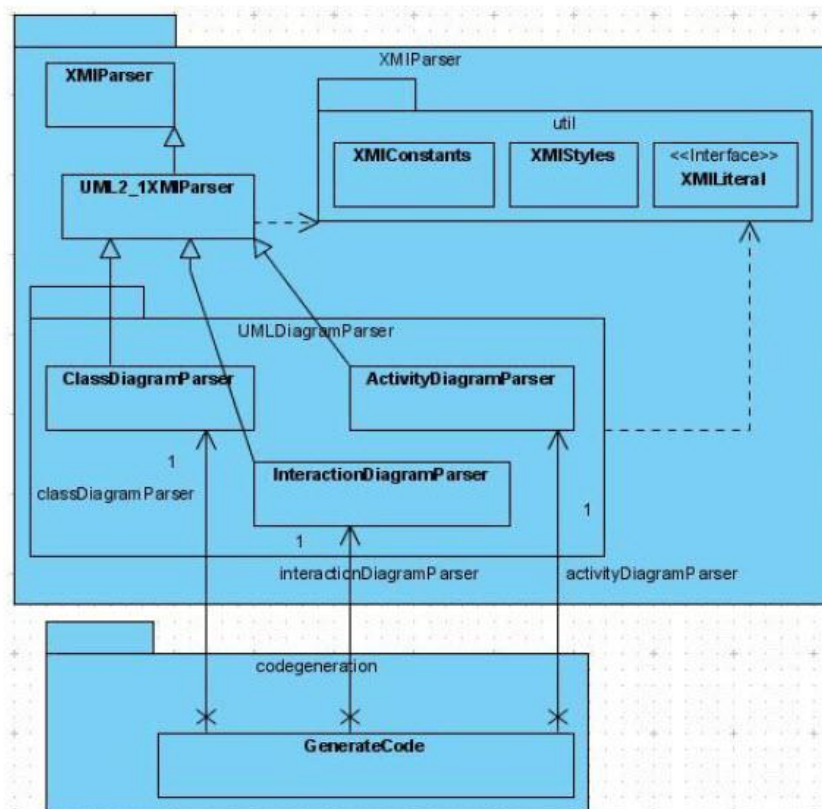


รูปที่ 7 สถาปัตยกรรมของ UJECTOR [6]

รูปที่ 7 แสดงถึงสถาปัตยกรรมทั้งหมดของเครื่องมือ UJECTOR ซึ่งทำให้มีผลต่อการสร้างโค้ด และสถาปัตยกรรมนี้ยังประกอบด้วย 3 ส่วนหลัก ได้แก่

- 1) XMLParser เป็นการเริ่มต้นการดำเนินงาน ซึ่งใช้แผนภาพยูเอ็มแอลแปลงไปยังรูปแบบของเอกซ์เอ็มไอ เพื่อที่จะใช้เอกซ์เอ็มไอเป็นสื่อนำเข้า และสร้างเมตาโมเดลของแผนภาพยูเอ็มแอล

- 2) CodeGenerator เป็นองค์ประกอบหลักของผู้ที่รับผิดชอบในการสร้างโค้ดจาวา ซึ่งประกอบด้วย ดังนี้
- ClassStructureCreator เป็นส่วนประกอบเมตาโมเดลของแผนภาพคลาส ซึ่งผลที่ได้คือโครงสร้างของโค้ดจาวาที่แสดงถึงความสัมพันธ์ระหว่างคลาสของแผนภาพคลาส
  - ActionsCodeCreator เป็นการนำเข้าเมตาโมเดลของแผนภาพกิจกรรม และผลที่ได้คือแผนภาพกิจกรรมและจำนวนบรรทัดของโค้ดจาวา ซึ่งจะนำแผนภาพกิจกรรมมาใช้ในการอ้างอิงแผนภาพลำดับ
  - ClassMethodCreator เป็นส่วนประกอบเมตาโมเดลของแผนภาพลำดับ ซึ่งผลที่ได้คือคลาสเมทอดที่มีการควบคุมการไหลของข้อมูล และออลเทอร์นาทิฟ โดยใช้ แผนภาพกิจกรรมสำหรับการอ้างอิง
- 3) CodeMerger เป็นผู้ที่แยกองค์ประกอบของโค้ดจาวาจากส่วนประกอบของโค้ดจาวาจากส่วนประกอบของผู้ที่สร้างโค้ดและใช้แผนภาพกิจกรรมแทนที่สำหรับการอ้างอิงถึงโค้ด

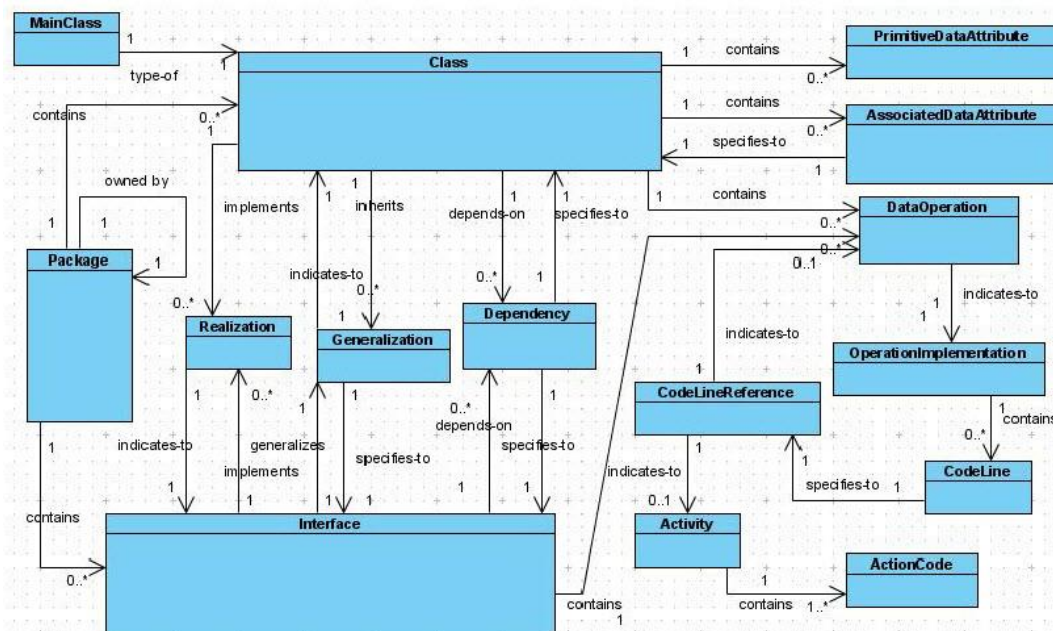


รูปที่ 8 UJECTOR Implementation package diagram [6]



รูปที่ 8 แสดงถึงส่วนประกอบภายในของ XMIParser จะใช้แผนภาพคลาส แผนภาพลำดับ และแผนภาพกิจกรรม ซึ่งแบ่งออกเป็น 3 ส่วน ดังนี้

- ClassDiagramParser เป็นการสกัดข้อมูลของแผนภาพคลาสจากรูปแบบไฟล์เอกซ์เอ็มไอ และไฟล์เมตาโมเดลของแผนภาพคลาส
- InteractionDiagramParser เป็นการสกัดข้อมูลของแผนภาพลำดับจากไฟล์เอกซ์เอ็มไอและไฟล์เมตาโมเดลของแผนภาพลำดับ
- ActivityDiagramParser เป็นการสกัดข้อมูลของแผนภาพกิจกรรมจากไฟล์เอกซ์เอ็มไอและไฟล์เมตาโมเดลของแผนภาพกิจกรรม

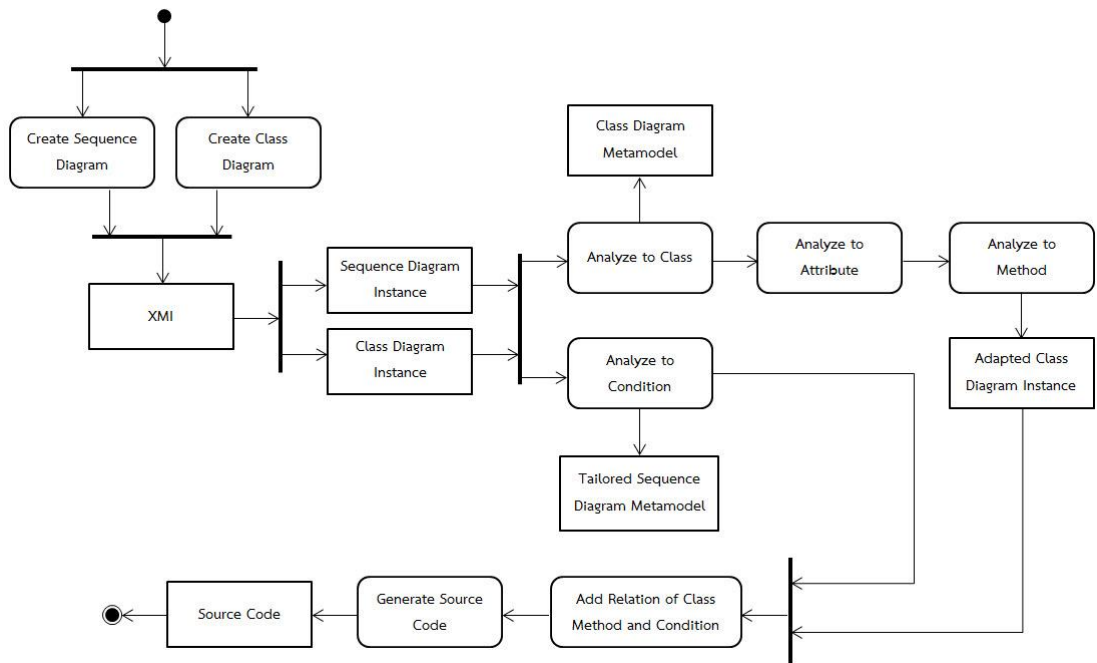


รูปที่ 9 โครงสร้างหน่วยความจำภายในสำหรับการสร้างโค้ด [6]

รูปที่ 9 แสดงโครงสร้างหน่วยความจำภายในสำหรับการสร้างโค้ด ซึ่งจะเป็นการอธิบายรายละเอียดการทำงานภายในของระบบ

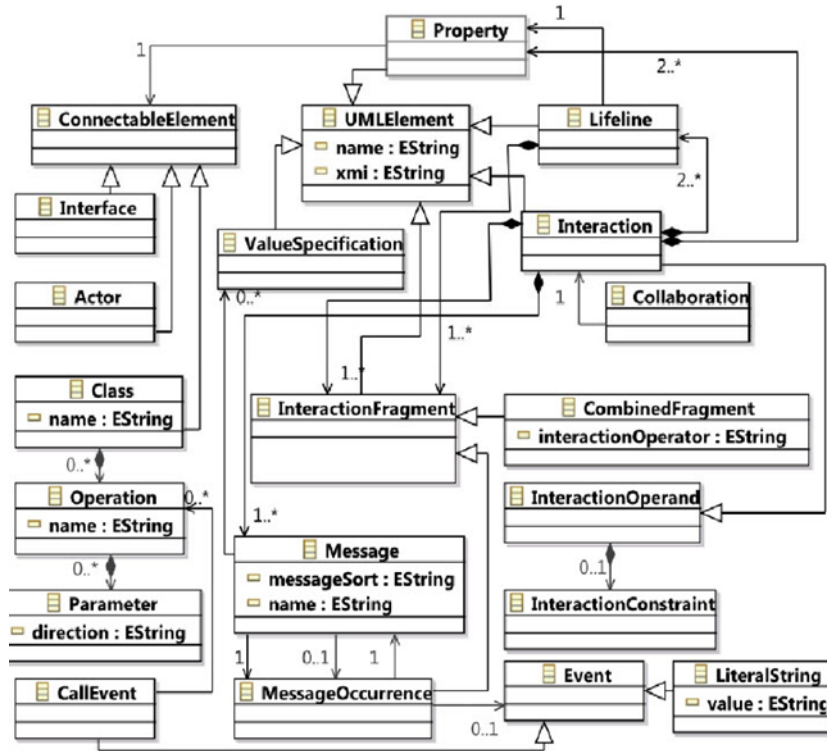
### บทที่ 3 แนวคิดและวิธีดำเนินการวิจัย

ในบทนี้จะกล่าวถึงแนวคิดและวิธีการดำเนินการวิจัย โดยแนวคิดและดำเนินการวิจัยในการสร้างโค้ดจาวาจากแผนภาพลำดับและแผนภาพคลาส ซึ่งจะครอบคลุมในรูปแบบของพื้นฐาน ออลเทอร์นาทิฟ และวนซ้ำ สามารถแบ่งออกเป็นทั้งหมด 8 ขั้นตอนมีดังต่อไปนี้ 1) สร้างแผนภาพลำดับ 2) สร้างแผนภาพคลาส 3) วิเคราะห์การสกัดข้อมูลในส่วนของคลาสและสร้างคลาสจากแผนภาพคลาส 4) วิเคราะห์การสกัดข้อมูลในส่วนของแอทริบิวต์และสร้างแอทริบิวต์จากแผนภาพคลาส 5) วิเคราะห์การสกัดข้อมูลในส่วนของเมทอดและสร้างเมทอดจากแผนภาพคลาส 6) วิเคราะห์ความสัมพันธ์ระหว่างเมทอดกับการรับส่งข้อมูลจากแผนภาพลำดับ 7) เพิ่มรายละเอียดของการสกัดข้อมูลของเงื่อนไขจากแผนภาพลำดับ 8) สร้างรหัสต้นทาง

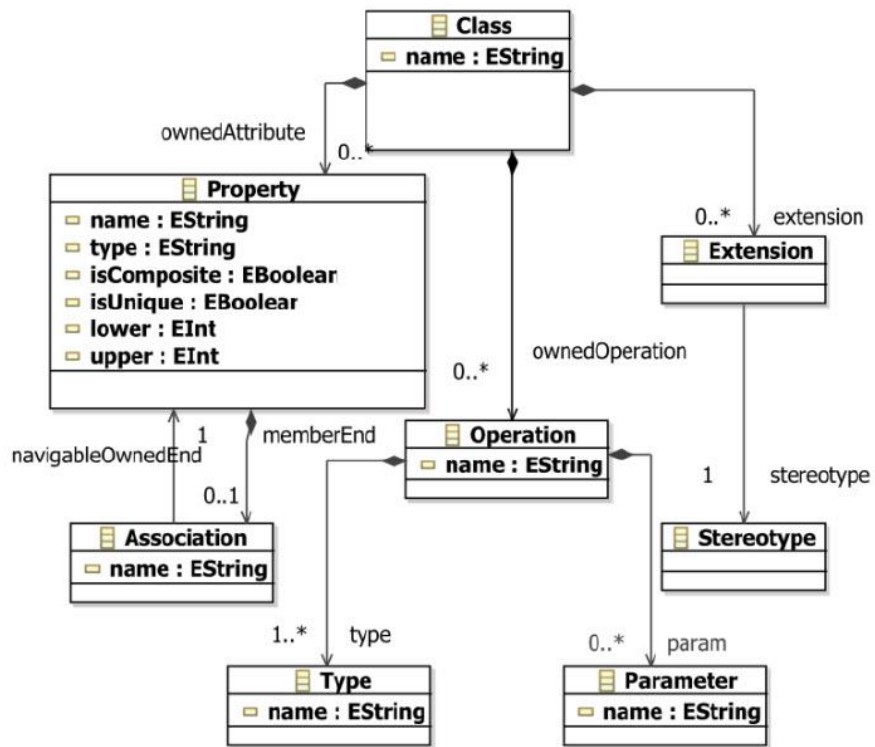


รูปที่ 10 แผนภาพกิจกรรมแนวคิดและวิธีการวิจัย

รูปที่ 10 แสดงถึงวิธีการแลกเปลี่ยนข้อมูลเมตาเดตาของเอกซ์เอ็มแอลที่จะใช้เป็นตัวกลางกระบวนการดำเนินการของงานวิจัยนี้ ซึ่งจะอยู่ในมาตรฐานของ Object Management Group (OMG) สำหรับการแลกเปลี่ยนข้อมูลของโมเดล และแสดงเมตาโมเดลของแผนภาพลำดับและแผนภาพคลาสที่นำไปใช้ในงานวิจัยนี้ ดังรูปที่ 11 และรูปที่ 12 ตามลำดับ



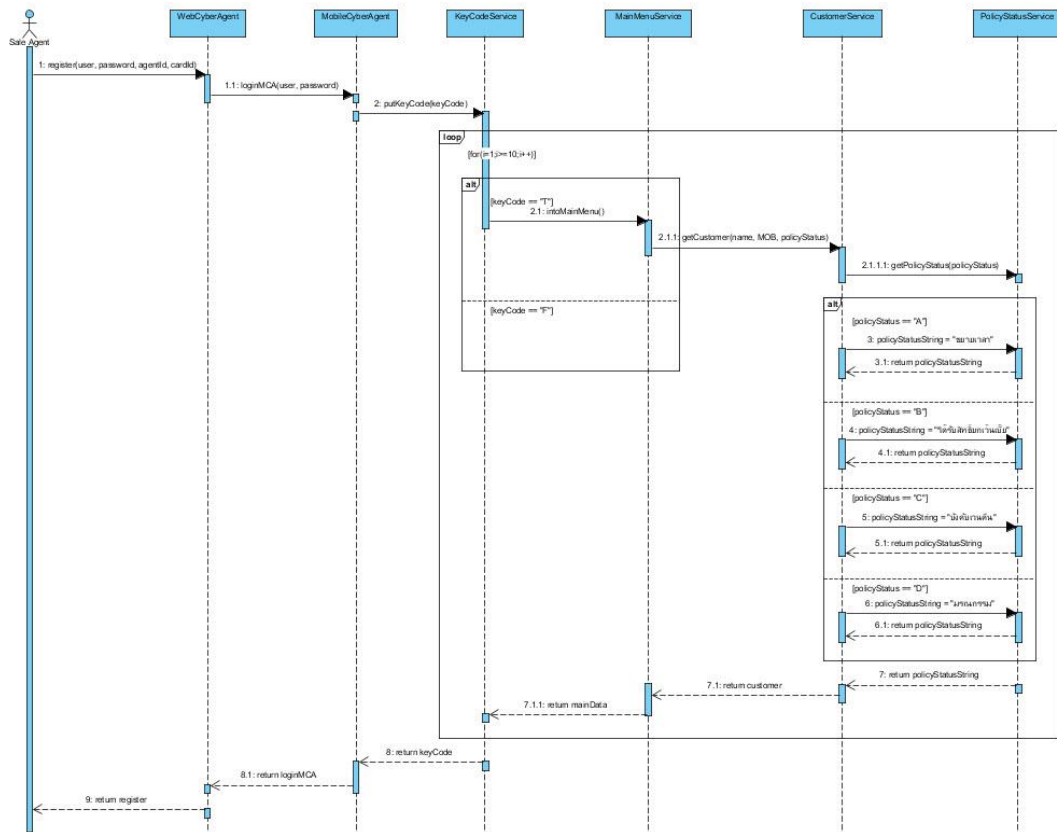
รูปที่ 11 Tailored OMG Sequence diagram metamodel [7]



รูปที่ 12 OMG Class diagram metamodel [7]

### 3.1. สร้างแผนภาพลำดับ

การสร้างโค้ดจาวา ในงานวิจัยนี้ได้นำแผนภาพลำดับมาใช้เป็นพื้นฐานในการออกแบบ เนื่องจากแผนภาพลำดับ จะแสดงถึงลำดับขั้นตอนของเหตุการณ์ แล้วผ่านไปยังตัวแจบของแผนภาพลำดับ ซึ่งในแต่ละลำดับเหตุการณ์ของแผนภาพลำดับ จะประกอบด้วยแอทริบิวต์ต่างๆที่สามารถนำมาสร้างเป็นโค้ดจาวา โดยการสร้างแผนภาพลำดับ จะใช้เครื่องมือที่มีชื่อว่า วิชชัวพาราตาม ซึ่งเครื่องมือนี้สามารถสร้างแบบแผนภาพลำดับ พร้อมทั้งเปลี่ยนแผนภาพลำดับดังกล่าวให้อยู่ในรูปแบบไฟล์เอกซ์เอ็มไอ ซึ่งมีตัวอย่างในการสร้างแผนภาพลำดับดังรูปที่ 13



รูปที่ 13 ตัวอย่างแผนภาพลำดับที่ได้จากเครื่องมือวิชชัวพาราตามยูเอ็มแอล

จากรูปที่ 13 แสดงให้เห็นตัวอย่างในการสร้างแผนภาพลำดับ ยกตัวอย่างเช่น

- Lifeline ที่มีชื่อว่า WebCyberAgent จะทำหน้าที่แสดงเป็นชื่อของคลาส
- Message ที่มีชื่อว่า loginMCA(user, password) จะทำหน้าที่แสดงเป็นเม็ทอด

จากนั้นนำแผนภาพลำดับนี้ มาสร้างเป็นรูปแบบไฟล์เอกซ์เอ็มไอ โดยใช้เครื่องมือวิชชัวพาราตามยูเอ็มแอล ซึ่งจะได้ผลดังรูปที่ 14

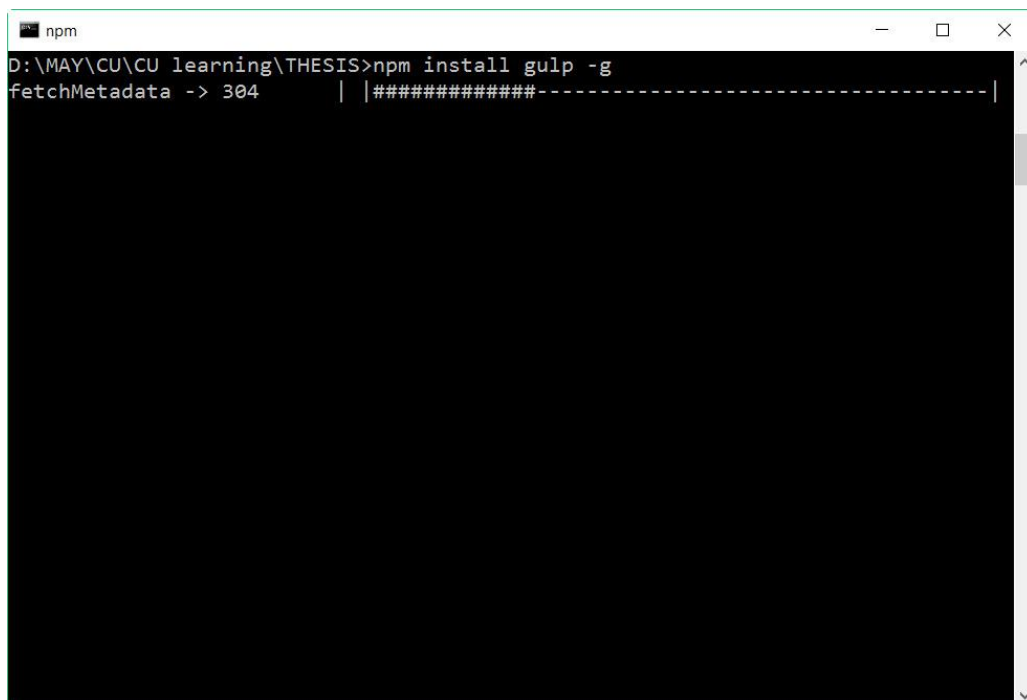
```
<ownedMember xmi:type="uml:Collaboration">
  <xmi:Extension extender="Visual Paradigm">
    <dummyModel value="true"/>
  </xmi:Extension>
  <ownedBehavior name="InsuranceThesis" xmi:id="CBGD7yqGAgACMA8K" xmi:type="uml:Interaction">
    <lifeline name="WebCyberAgent" represents="IhND7yqGAgACMA8o_represents" xmi:id="IhND7yqGAgACMA8o" xmi:type="uml:Lifeline">
      <xmi:Extension extender="Visual Paradigm">
        <active xmi:value="false"/>
        <stopped xmi:value="false"/>
        <multiObject xmi:value="false"/>
        <activation name="Activation" type="false" xmi:id="GqBj7yqGAgACMBNq" xmi:type="activation">
          <xmi:Extension extender="Visual Paradigm">
            <qualityScore value="-1"/>
          </xmi:Extension>
        </activation>
        <activation name="Activation" type="false" xmi:id="b7Pb7yqGAgACMBtv" xmi:type="activation">
          <xmi:Extension extender="Visual Paradigm">
            <qualityScore value="-1"/>
          </xmi:Extension>
        </activation>
        <activation name="Activation" type="false" xmi:id="0FQ77yqGAgACMBuW" xmi:type="activation">
          <xmi:Extension extender="Visual Paradigm">
            <qualityScore value="-1"/>
          </xmi:Extension>
        </activation>
        <lifeline/>
        <qualityScore value="-1"/>
      </xmi:Extension>
      <coveredBy xmi:idref="FRKT7yqGAgACMBfv"/>
      <coveredBy xmi:idref="iFQ77yqGAgACMBub"/>
      <coveredBy xmi:idref="OqBj7yqGAgACMBNW"/>
      <coveredBy xmi:idref="37Pb7yqGAgACMBt1"/>
    </lifeline>
  </ownedBehavior>
</ownedMember>
```

รูปที่ 14 ตัวอย่างไฟล์เอกซ์เอ็มไอในส่วนของ “WebCyberAgent” ที่ได้จากเครื่องมือ  
วิชชัวพาราตามยูเอ็มแอล

```
<message name="loginMCA(user, password)" receiveEvent="FRKT7yqGAgACMBfv" sendEvent="FRKT7yqGAgACMBfu" xmi:id="FRKT7yqGAgACMBfu" xmi:type="uml:Message">
  <signature xmi:id="FRKT7yqGAgACMBfu_type" xmi:type="uml:Action"/>
  <xmi:Extension extender="Visual Paradigm">
    <number xmi:value="1.1"/>
    <asynchronous xmi:value="false"/>
    <fromActivation>
      <activation xmi:value="GqBj7yqGAgACMBNq"/>
    </fromActivation>
    <toActivation>
      <activation xmi:value="xRKT7yqGAgACMBfq"/>
    </toActivation>
    <qualityScore value="-1"/>
  </xmi:Extension>
</message>
```

รูปที่ 15 ตัวอย่างไฟล์เอกซ์เอ็มไอในส่วนของ “loginMCA(user, password)” ที่ได้จากเครื่องมือ  
วิชชัวพารามตามยูเอ็มแอล

จากนั้นนำไฟล์เอกซ์เอ็มไอของแผนภาพลำดับที่ได้มาเปลี่ยนให้อยู่ในรูปแบบของไฟล์เจสัน ซึ่งในงานวิจัยนี้เลือกใช้รูปแบบไฟล์เจสัน เพราะรูปแบบไฟล์เจสันมีโครงสร้างข้อมูลที่สามารถเข้าถึงและเรียกใช้งานได้ง่ายกว่ารูปแบบไฟล์เอกซ์เอ็มไอ โดยการเปลี่ยนรูปแบบไฟล์เจสันนั้น งานวิจัยนี้ได้ใช้โปรแกรมประยุกต์ในการเปลี่ยนรูปแบบไฟล์จากเอกซ์เอ็มไอให้อยู่ในรูปแบบไฟล์เจสันผ่านช่องทางที่มีชื่อว่า “Gulp” ซึ่งวิธีก็ลพ์จะต้องดาวน์โหลดโปรแกรมผ่านคอมมานไลน์ด้วยคำสั่ง “npm install gulp -g” ดังรูปที่ 16 เมื่อโปรแกรมถูกติดตั้งแล้วจะสามารถเรียกใช้หรือเข้าถึงฟังก์ชัน เพื่อใช้ในการเปลี่ยนรูปแบบไฟล์เอกซ์เอ็มไอของแผนภาพลำดับเป็นรูปแบบไฟล์เจสัน ดังรูปที่ 17 และผลลัพธ์ที่ได้จะแสดงอยู่ในรูปแบบไฟล์เจสันของแผนภาพลำดับ ดังรูปที่ 18



```
npm
D:\MAY\CU\CU learning\THESIS>npm install gulp -g
fetchMetadata -> 304 | |#####-----|
```

รูปที่ 16 การดาวน์โหลดโปรแกรมประยุกต์ผ่านคอมมานไลน์ด้วยคำสั่ง “npm install gulp -g”

```
Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\MAY>cd C:\Users\MAY\Desktop\ConvertUMLPROJ\gulp-example

C:\Users\MAY\Desktop\ConvertUMLPROJ\gulp-example>gulp convertToJson
(node:21532) fs: re-evaluating native module sources is not supported. If you are
e using the graceful-fs module, please update it to a more recent version.
[21:32:21] Using gulpfile ~\Desktop\ConvertUMLPROJ\gulp-example\gulpfile.js
[21:32:21] Starting 'convertToJson'...
[21:32:21] Finished 'convertToJson' after 8.65 ms
[21:32:22] gulp-xml2json:  xml\SDinput.xml

C:\Users\MAY\Desktop\ConvertUMLPROJ\gulp-example>
```

รูปที่ 17 การเรียกใช้โปรแกรมประยุกต์ผ่านคอมมานด์ไลน์เพื่อเปลี่ยนรูปแบบไฟล์เป็นเจสันของ

แผนภาพลำดับ

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

```

{
  "$": {
    "xmi:type": "uml:Collaboration"
  },
  "xmi:Extension": [
    {
      "$": {
        "extender": "Visual Paradigm"
      },
      "dummyModel": [
        {
          "$": {
            "value": "true"
          }
        }
      ]
    }
  ],
  "ownedBehavior": [
    {
      "$": {
        "name": "InsuranceThesis",
        "xmi:id": "CBGD7yqGAqACMA8K",
        "xmi:type": "uml:Interaction"
      },
      "lifeline": [
        {
          "$": {
            "name": "WebCyberAgent",
            "represents": "IhND7yqGAqACMA8o_represents",
            "xmi:id": "IhND7yqGAqACMA8o",
            "xmi:type": "uml:Lifeline"
          },
          "xmi:Extension": [
            {
              "$": {
                "extender": "Visual Paradigm"
              },
              "active": [
                {
                  "$": {
                    "xmi:value": "false"
                  }
                }
              ],
              "stopped": [
                {
                  "$": {
                    "xmi:value": "false"
                  }
                }
              ],
              "multiObject": [
                {
                  "$": {
                    "xmi:value": "false"
                  }
                }
              ]
            }
          ]
        }
      ]
    }
  ],
}

```

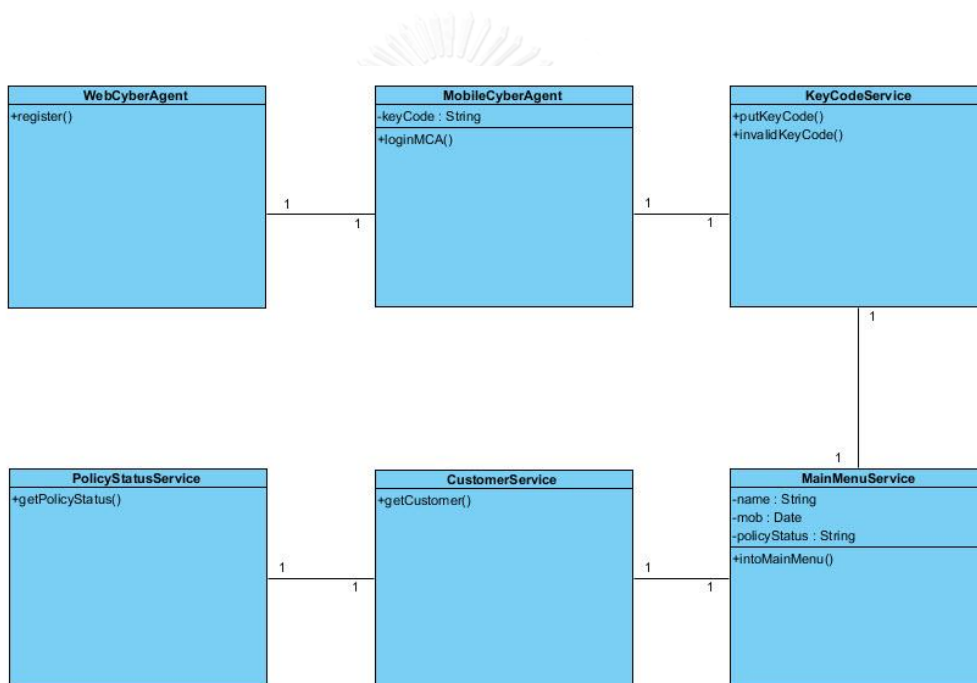
รูปที่ 18 ตัวอย่างรูปแบบไฟล์เจสันของแผนภาพลำดับที่ได้จากโปรแกรมประยุกต์



จากนั้นจะนำรูปแบบไฟล์เจสันที่ได้มาสกัด เพื่อนำไปเป็นส่วนหนึ่งในการสร้างโค้ดจาวา ซึ่งจะกล่าวรายละเอียดในขั้นตอนถัดไป

### 3.2. สร้างแผนภาพคลาส

ขั้นตอนนี้เป็นการสร้างแผนภาพคลาสที่ใช้เป็นส่วนหนึ่งของงานวิจัยนี้ เนื่องจากแผนภาพคลาส จะแสดงถึงคลาส แอทริบิวต์ เมทอดและความสัมพันธ์ระหว่างคลาสที่สามารถนำส่วนประกอบนี้มาสร้างเป็นโค้ดจาวา โดยการสร้างแผนภาพคลาส จะใช้เครื่องมือที่มีชื่อว่า วิชช่วพาราตาม ซึ่งเครื่องมือนี้สามารถสร้างแบบแผนภาพคลาส พร้อมทั้งเปลี่ยนแผนภาพคลาสดังกล่าวให้อยู่ในรูปแบบไฟล์เอกซ์เอ็มแอล ซึ่งมีตัวอย่างในการสร้างแผนภาพคลาสดังรูปที่ 19



รูปที่ 19 ตัวอย่างแผนภาพคลาสที่ได้จากเครื่องมือวิช่วพาราตามยูเอ็มแอล

จากรูปที่ 19 แสดงให้เห็นตัวอย่างในการสร้างแผนภาพคลาส ยกตัวอย่างเช่น

- MobileCyberAgent จะแสดงเป็นชื่อคลาส
- keycode จะแสดงแอทริบิวต์ของคลาส
- loginMCA จะแสดงเมทอดของคลาส

จากนั้นนำแผนภาพคลาสนี้ มาสร้างเป็นรูปแบบไฟล์เอกซ์เอ็มไอ โดยใช้เครื่องมือวิซช่วพารา ตามยูเอ็มแอล ซึ่งจะได้ผลดังรูปที่ 20 และเมื่อได้ไฟล์เอกซ์เอ็มไอของแผนภาพคลาสจะนำมาเปลี่ยนให้อยู่ในรูปแบบของไฟล์เจสัน ซึ่งมีวิธีเดียวกันกับแผนภาพลำดับ โดยการเปลี่ยนรูปแบบไฟล์เอกซ์เอ็มไอของแผนภาพคลาสเป็นรูปแบบไฟล์เจสันจะแสดงดังรูปที่ 21 และผลลัพธ์ที่ได้จะแสดงอยู่ในรูปแบบไฟล์เจสันของแผนภาพคลาส ดังรูปที่ 22

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xmi:XMI xmi:version="2.1" xmlns:uml="http://schema.omg.org/spec/UML/2.0" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1">
3  <xmi:Documentation exporter="Visual Paradigm" exporterVersion="7.0.2">
4  <xmi:Extension extender="Visual Paradigm">
5  <nickname exportedFromDifferentName="false"/>
6  </xmi:Extension>
7  </xmi:Documentation>
8  <xmi:Extension extender="Visual Paradigm">
220 <uml:Model name="CD" xmi:id="KioIraqAUNimHgOC">
7593 <uml:Diagram diagramType="ClassDiagram" documentation="" name="Class Diagram1" toolName="Visual Paradigm" xmi:id="T2EIraqAUNimHgWB">
7594 <xmi:Extension extender="Visual Paradigm">
7595 <properties>
7596 <ashowShapeStereotypeIconName value="true"/>
7597 <ashowAllocatedFrom value="true"/>
7598 <ashowAllocatedTo value="true"/>
7599 <apmAuthor value="MAY"/>
7600 <apmCreateDateTime value="1467621607154"/>
7601 <apmLastModified value="1467657456109"/>
7602 <ateamworkCreateDateTime value="0"/>
7603 <ashowPackageNameStyle value="0"/>
7604 <ashowConnectorName value="0"/>
7605 <apaintConnectorThroughLabel value="1"/>
7606 <ashowActivityStateNodeCaption value="65535"/>
7607 <adocumentation htmlValue="&lt;html&gt;&#13;&#10; &lt;head&gt;&#13;&#10; &lt;style type=&quot;text/css&quot;&gt;&#10;
7608 &lt;table border=&quot;&#13;&#10; &lt;tbody&gt;&#13;&#10; &lt;tr&gt;&#13;&#10; &lt;td&gt;&#13;&#10; &lt;/tr&gt;&#13;&#10; &lt;/tbody&gt;&#13;&#10; &lt;/table&gt;&#13;&#10; &lt;/html&gt;&#13;&#10; &lt;/adocumentation&gt;
7609 <aeditable value="true"/>
7610 <afollowDiagramParentElement value="true"/>
7611 <aautoFitShapesSize value="false"/>
7612 <ashowTaggedValues value="false"/>
7613 <ahideEmptyTaggedValues value="false"/>
7614 <ashowStereotypes value="true"/>
7615 <ashowPMIteration value="false"/>
7616 <ashowPMPhase value="false"/>
7617 <ashowPMDiscipline value="false"/>
7618 <ashowPMVersion value="false"/>

```

รูปที่ 20 ตัวอย่างไฟล์เอกซ์เอ็มไอของแผนภาพคลาส

```

Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\MAY>cd C:\Users\MAY\Desktop\ConvertUMLPROJ\gulp-example

C:\Users\MAY\Desktop\ConvertUMLPROJ\gulp-example>gulp convertToJson
(node:9052) fs: re-evaluating native module sources is not supported. If you are
using the graceful-fs module, please update it to a more recent version.
[21:36:30] Using gulpfile ~\Desktop\ConvertUMLPROJ\gulp-example\gulpfile.js
[21:36:30] Starting 'convertToJson'...
[21:36:30] Finished 'convertToJson' after 8.92 ms
[21:36:31] gulp-xml2json:  xml\CDinput.xml

C:\Users\MAY\Desktop\ConvertUMLPROJ\gulp-example>

```

รูปที่ 21 การเรียกใช้โปรแกรมประยุกต์ผ่านคอมมานด์ไลน์เพื่อเปลี่ยนรูปแบบไฟล์เป็นเจสันของ  
แผนภาพคลาส

```

{
  "xmi:XMI": {
    "$": {
      "xmi:Documentation": [{"
      "xmi:Extension": [{"
      "uml:Model": [{"
      "uml:Diagram": [{"
        "$": {
          "diagramType": "ClassDiagram",
          "documentation": "",
          "name": "Class Diagram1",
          "toolName": "Visual Paradigm",
          "xmi:id": "T2EIraqAUNimHgWB"
        },
        "xmi:Extension": [{"
          "$": {
            "extender": "Visual Paradigm"
          },
          "properties": [{"
            "ashowShapeStereotypeIconName": [{"
              "$": {
                "value": "true"
              }
            }],
            "ashowAllocatedFrom": [{"
              "$": {
                "value": "true"
              }
            }
          ]
        }
      ]
    }
  ]
}

```

รูปที่ 22 ตัวอย่างรูปแบบไฟล์เจสันของแผนภาพคลาสที่ได้จากโปรแกรมประยุกต์

### 3.3. วิเคราะห์การสกัดข้อมูลในส่วนของคลาสและสร้างคลาสจากแผนภาพคลาส

ขั้นตอนนี้มีจุดประสงค์ในการสร้างคลาสด้วยโค้ดจาวา ซึ่งจะเป็นส่วนประกอบหนึ่งที่สำคัญในการสร้างคลาสของอีอบเจกต์ให้ออกมาอยู่ในรูปแบบของโครงสร้างคลาสของโค้ดจาวา โดยงานวิจัยนี้ได้กำหนดใช้ไฟล์เจสันของแผนภาพคลาสในการสกัดข้อมูล เพื่อให้ได้โครงสร้างของข้อมูลที่ชัดเจนให้อีกกับการสร้างโค้ดจาวา พร้อมทั้งจัดข้อมูลที่ไม่เกี่ยวข้องกับการสร้างโค้ดจาวา ซึ่งโครงสร้างของโค้ดจาวามีดังต่อไปนี้ ดังรูปที่ 23

```
{
  "xmi:XMI": {
    "$": {
      "xmi:Documentation": [{"
      "xmi:Extension": [{"
      "uml:Model": [{"
        "$": {
          "xmi:Extension": [{"
          "ownedMember": [
            {
              {
                {
                  {
                    {
                      "$": {
                        "isAbstract": "false",
                        "isActive": "false",
                        "isLeaf": "false",
                        "name": "MobileCyberAgent",
                        "visibility": "public",
                        "xmi:id": "ck0iraqAUNimHgZC",
                        "xmi:type": "uml:Class"
                      },
                    },
                  },
                },
              },
            ],
          },
        },
      },
    },
  },
}
```

รูปที่ 23 โครงสร้างยูเอ็มแอลโมเดลของแผนภาพคลาสที่ใช้ในงานวิจัย

จากรูปที่ 23 แสดงถึงโครงสร้างยูเอ็มแอลโมเดลของแผนภาพคลาสที่เป็นส่วนประกอบในการสร้างคลาส ซึ่งจากรูปที่ 24 จะแสดงถึงโครงสร้างของคลาสที่เป็นรูปแบบไฟล์เจสัน ซึ่งจะมีรายละเอียดดังตารางที่ 1

```

"$": {
  "isAbstract": "false",
  "isActive": "false",
  "isLeaf": "false",
  "name": "MobileCyberAgent",
  "visibility": "public",
  "xmi:id": "ck0iraqAUNimHgZC",
  "xmi:type": "uml:Class"
},

```

รูปที่ 24 ข้อมูลของคลาส

ตารางที่ 1 รายละเอียดของส่วนประกอบย่อยในโครงสร้างของคลาสที่ใช้ในงานวิจัย

แอทริบิวต์	ประเภท
xmi:id	String
xmi:type	String
name	String
visibility	Boolean

เมื่อได้ผลลัพธ์ของการแปลงรูปแบบไฟล์เจสันของแผนภาพคลาสเป็นคลาส จะนำมาสร้างเป็นรูปแบบไฟล์จาวา ซึ่งจะแสดงดังรูปที่ 25

```

public class MobileCyberAgent {
}

```

รูปที่ 25 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากการสร้างคลาส

### 3.4. วิเคราะห์การสกัดข้อมูลในส่วนของแทรชีบิวต์และสร้างแทรชีบิวต์จากแผนภาพคลาส

ขั้นตอนนี้มีจุดประสงค์ เพื่อสร้างแทรชีบิวต์ของคลาส โดยจะนำเอารูปแบบไฟล์เจสันของแผนภาพคลาสในส่วนแทรชีบิวต์ มาสร้างแทรชีบิวต์ของแต่ละคลาส

```
{
  "$": {
    "isAbstract": "false",
    "isActive": "false",
    "isLeaf": "false",
    "name": "MobileCyberAgent",
    "visibility": "public",
    "xmi:id": "ck0iraqAUNimHgZC",
    "xmi:type": "uml:Class"
  },
  "xmi:Extension": [{
    "ownedAttribute": [{
      "$": {
        "aggregation": "none",
        "isDerived": "false",
        "isDerivedUnion": "false",
        "isID": "false",
        "isLeaf": "false",
        "isReadOnly": "false",
        "isStatic": "false",
        "name": "keyCode",
        "type": "String_id",
        "visibility": "private",
        "xmi:id": "6vCiraqAUNimHgZQ",
        "xmi:type": "uml:Property"
      },
    ],
  ],
}
```

รูปที่ 26 ข้อมูลของแทรชีบิวต์

จากรูปที่ 26 แสดงถึงแทรชีบิวต์ของคลาสในไฟล์เจสัน ซึ่งจะแสดงส่วนประกอบย่อยในโครงสร้างแทรชีบิวต์ของโค้ดจาวา ซึ่งจะมีรายละเอียดดังตารางที่ 2

ตารางที่ 2 รายละเอียดของส่วนประกอบย่อยในโครงสร้างแอทริบิวต์ของโค้ดจาวาที่ใช้ในงานวิจัย

แอทริบิวต์	ประเภท
xmi:id	String
type	String
name	String
visibility	Boolean

เมื่อได้ผลลัพธ์ของการแปลงรูปแบบไฟล์เจสันของแผนภาพคลาสเป็นแอทริบิวต์ จะนำมาสร้างเป็นรูปแบบไฟล์จาวา ซึ่งจะแสดงดังรูปที่ 27

```
public class MobileCyberAgent {
    private String keyCode;
}
```

รูปที่ 27 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากการสร้างแอทริบิวต์

### 3.5. วิเคราะห์การสกัดข้อมูลในส่วนของเมทอดและสร้างเมทอดจากแผนภาพคลาส

ขั้นตอนนี้มีจุดประสงค์ เพื่อสร้างเมทอดระหว่างคลาส โดยนำเอารูปแบบไฟล์เจสันในส่วน  
ของเมทอด มาสร้างความสัมพันธ์ของเมทอดกับคลาส

```

{
  "$": {
    "isAbstract": "false",
    "isActive": "false",
    "isLeaf": "false",
    "name": "MobileCyberAgent",
    "visibility": "public",
    "xmi:id": "ck0iraqAUNimHgZC",
    "xmi:type": "uml:Class"
  },
  "xmi:Extension": [{
    "ownedAttribute": [{
    "ownedOperation": [{
      "$": {
        "isAbstract": "false",
        "isLeaf": "false",
        "isOrdered": "false",
        "isQuery": "false",
        "isStatic": "false",
        "isUnique": "true",
        "name": "loginMCA",
        "visibility": "public",
        "xmi:id": "V_KiraqAUNimHgZa",
        "xmi:type": "uml:Operation"
      },
    ],
  ],
  ],
  },

```

รูปที่ 28 ข้อมูลของเมทอด

จากรูปที่ 28 แสดงถึงเมทอดของคลาสในไฟล์เจสัน ซึ่งจะแสดงส่วนประกอบย่อยในโครงสร้างเมทอดของโค้ดจาวา ซึ่งจะมีรายละเอียดดังตารางที่ 3

ตารางที่ 3 รายละเอียดของส่วนประกอบย่อยในโครงสร้างเมทอดของโค้ดจาวาที่ใช้ในงานวิจัย

แอทริบิวต์	ประเภท
xmi:id	String
xmi_type	String
name	String
visibility	Boolean



เมื่อได้ผลลัพธ์ของการแปลงรูปแบบไฟล์เจสันของแผนภาพคลาสเป็นเมทอด จะนำมาสร้างเป็นรูปแบบไฟล์จาวา ซึ่งจะแสดงดังรูปที่ 29

```
public class MobileCyberAgent {  
  
    private String keyCode;  
  
    public String loginMCA() {  
  
    }  
  
}
```

รูปที่ 29 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากการสร้างเมทอด

### 3.6. วิเคราะห์ความสัมพันธ์ระหว่างเมทอดกับการรับส่งข้อมูลจากแผนภาพลำดับ

ขั้นตอนนี้มีจุดประสงค์ เพื่อสร้างการรับส่งข้อมูลของอ็อบเจกต์ และพารามิเตอร์ของเมทอดในระหว่างคลาส โดยจะนำเอารูปแบบไฟล์เจสันของแผนภาพลำดับในส่วนของการรับส่งข้อมูล มาสร้างความสัมพันธ์ระหว่างเมทอดและการรับส่งข้อมูล

```

"message": [
{
{
"$": {
"name": "loginMCA(user, password)",
"receiveEvent": "FRKT7yqGAqACMBfw",
"sendEvent": "FRKT7yqGAqACMBfv",
"xmi:id": "FRKT7yqGAqACMBfU",
"xmi:type": "uml:Message"
},
"signature": [{
"$": {
"xmi:id": "FRKT7yqGAqACMBfU_type",
"xmi:type": "uml:Action"
}
}],
"xmi:Extension": [{
"$": {
"extender": "Visual Paradigm"
},
"number": [{
"$": {
"xmi:value": "1.1"
}
}],
"asynshronous": [{
"$": {
"xmi:value": "false"
}
}],
}],
}
}
}

```

รูปที่ 30 ข้อมูลของการรับส่งข้อมูล

จากรูปที่ 30 แสดงถึงแอทริบิวต์ของการรับส่งข้อมูล (message) ไฟล์เจสันของแผนภาพลำดับ ซึ่งจะแสดงส่วนประกอบย่อยในโครงสร้างของจาวา ซึ่งจะมีรายละเอียดดังตารางที่ 4

ตารางที่ 4 รายละเอียดของส่วนประกอบในโครงสร้างการรับส่งข้อมูลของโค้ดจาวาที่ใช้ในงานวิจัย

แอทริบิวต์	ประเภท
xmi:id	String
xmi:type	String
name	String
covered	Array
receiveEvent	String
sendEvent	String

ผลลัพธ์ที่ได้จากขั้นตอนนี้จะเป็นไฟล์จาวาตามแต่ละคลาส โดยแต่ละคลาสจะประกอบด้วยชื่อของคลาสรวมถึงความสัมพันธ์ระหว่างคลาสกับการรับส่งข้อมูล โดยจะมีแอทริบิวต์ที่บ่งบอกถึงความสัมพันธ์ของแต่ละส่วนของคลาส การส่งข้อมูลจะใช้แอทริบิวต์ที่ชื่อว่า “sendEvent” และการรับข้อมูลจะใช้แอทริบิวต์ที่ชื่อว่า “receiveEvent” ทั้งสองแอทริบิวต์นี้จะแสดงถึงข้อมูลที่ได้ส่งข้อมูลมาจากคลาสใดและคลาสใดเป็นคลาสที่รับข้อมูล

```

public class MobileCyberAgent {
    private String keyCode;

    public String loginMCA(String user, String password) {
        return loginMCA;
    }
}

```

รูปที่ 31 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากการรับส่งข้อมูล

จากรูปที่ 31 แสดงถึงการรับส่งข้อมูลระหว่างคลาส หรือเรียกอีกอย่างว่าเมทอดของคลาส โดยจะนำมาแสดงส่วนประกอบย่อยในโครงสร้างของโค้ดจาวา

### 3.7. เพิ่มรายละเอียดของการสกัดข้อมูลของเงื่อนไขจากแผนภาพลำดับ

ขั้นตอนนี้มีจุดประสงค์ เพื่อสร้างเงื่อนไขของความสัมพันธ์ที่เกิดขึ้นระหว่างคลาส โดยจะนำเอารูปแบบไฟล์เจสันของแผนภาพลำดับในส่วนของเงื่อนไข คือออลเทอร์นาทิฟ และวนซ้ำมาสร้างเป็นโค้ดจาวา

```

"fragment": [
  {
    "$": {
      "interactionOperator": "alt",
      "name": "CombinedFragment2",
      "xmi:id": "LzjT7yqGAqACMBhB",
      "xmi:type": "uml:CombinedFragment"
    },
    "ownedMember": [
      {
        "$": {
          "name": "ifKeyCodeT",
          "xmi:id": "LzjT7yqGAqACMBhF",
          "xmi:type": "uml:InteractionOperand"
        },
        "covered": [
          {
            "$": {
              "xmi:idref": "iIdD7yqGAqACMA9A"
            }
          },
          {
            "$": {
              "xmi:idref": "7.dD7yqGAqACMA9I"
            }
          }
        ]
      }
    ]
  }
],

```

รูปที่ 32 ข้อมูลของเงื่อนไข

จากรูปที่ 32 แสดงถึงเงื่อนไข (if-else) ไฟล์เจสันของแผนภาพลำดับ ซึ่งจะแสดงส่วนประกอบย่อยในโครงสร้างของจาวา ซึ่งจะมีรายละเอียดดังตารางที่ 5

ตารางที่ 5 รายละเอียดของส่วนประกอบย่อยในโครงสร้างเงื่อนไขของโค้ดจาวาที่ใช้ในงานวิจัย

แอทริบิวต์	ประเภท
xmi:id	String
xmi:type	String
name	String
covered	Array
interactionOperator	String

```

"fragment": [
  {
    "$": {
      "interactionOperator": "loop",
      "name": "CombinedFragment",
      "xmi:id": "5c5T7yqGAqACMBgK",
      "xmi:type": "uml:CombinedFragment"
    },
    "ownedMember": [
      {
        "$": {
          "name": "for",
          "xmi:id": "zc5T7yqGAqACMBgO",
          "xmi:type": "uml:InteractionOperand"
        },
        "covered": [
          {
            "$": {
              "xmi:idref": "iIdD7yqGAqACMA9A"
            }
          },
          {
            "$": {
              "xmi:idref": "7.dD7yqGAqACMA9I"
            }
          },
          {
            "$": {
              "xmi:idref": "GYij7yqGAqACMBKW"
            }
          },
          {
            "$": {
              "xmi:idref": "0vij7yqGAqACMBKl"
            }
          }
        ]
      }
    ]
  }
],

```

รูปที่ 33 ข้อมูลของวนซ้ำ

จากรูปที่ 33 แสดงถึงวนซ้ำ (loop) ไฟล์เจสันของแผนภาพลำดับ ซึ่งจะแสดงส่วนประกอบย่อยในโครงสร้างของจาวา ซึ่งจะมีรายละเอียดดังตารางที่ 6

ตารางที่ 6 รายละเอียดของส่วนประกอบย่อยในโครงสร้างวนซ้ำของโค้ดจาวาที่ใช้ในงานวิจัย

แอทริบิวต์	ประเภท
xmi:id	String
xmi:type	String
name	String
covered	Array
interactionOperator	String

```

"fragment": [
  "§":{
    "interactionOperator": "alt",
    "name": "CombinedFragment3",
    "xmi:id": "g6jL7yqGAqACMB1z",
    "xmi:type": "uml:CombinedFragment"
  },
  "ownedMember": [
    {
      "§": {
        "name": "switchPSA",
        "xmi:id": "c6jL7yqGAqACMB13",
        "xmi:type": "uml:InteractionOperand"
      },
      "covered": [
        {
          "§": {
            "xmi:idref": "GYij7yqGAqACMBKW"
          }
        },
        {
          "§": {
            "xmi:idref": "0vij7yqGAqACMBK1"
          }
        }
      ]
    }
  ],
]

```

รูปที่ 34 ข้อมูลของสวิตช์เคส

จากรูปที่ 34 แสดงถึงสวิตช์เคส (switch case) ไฟล์เจสันของแผนภาพลำดับ ซึ่งจะแสดงส่วนประกอบย่อยในโครงสร้างของจาวา ซึ่งจะมีรายละเอียดดังตารางที่ 7

ตารางที่ 7 รายละเอียดของส่วนประกอบย่อยในโครงสร้างสวิตช์เคสของโค้ดจาวาที่ใช้ในงานวิจัย

แอทริบิวต์	ประเภท
xmi:id	String
xmi:type	String
name	String
covered	Array
interactionOperator	String

จากที่กล่าวไปข้างต้นแสดงถึงเงื่อนไขในคลาส โดยจะนำมาแสดงส่วนประกอบย่อยในโครงสร้างของโค้ดจาวาในส่วนของออลเทอร์นาทิฟ

```

if (keyCode == "T") {

} else if (keyCode == "F") {

}

```

รูปที่ 35 ตัวอย่างรูปแบบไฟล์จาวาที่ได้จากความสัมพันธ์ของคลาสอย่างมีเงื่อนไข

จากรูปที่ 35 แสดงถึงเงื่อนไขในคลาส โดยจะนำมาแสดงส่วนประกอบย่อยในโครงสร้างของโค้ดจาวาในส่วนของออลเทอร์นาทิฟ

### 3.8. สร้างรหัสต้นทาง

ขั้นตอนนี้มีจุดประสงค์ เพื่อสร้างรหัสต้นทาง โดยรวบรวมส่วนประกอบย่อยทั้งหมดที่ได้กล่าวข้างต้นนำมาสร้างเป็นโค้ดจาวา ซึ่งผลลัพธ์ที่ได้จะเป็นรูปแบบไฟล์จาวา ดังรูปที่ 36

```
public class KeyCodeService {  
    public String putKeyCode(String keyCode) {  
        if (keyCode == "T") {  
            String keyCode = mainMenuService.intoMainMenu();  
        } else if (keyCode == "F") {  
        }  
        return keyCode;  
    }  
}
```

รูปที่ 36 ตัวอย่างรูปแบบไฟล์จาวา





## บทที่ 4

### การพัฒนาเครื่องมือ

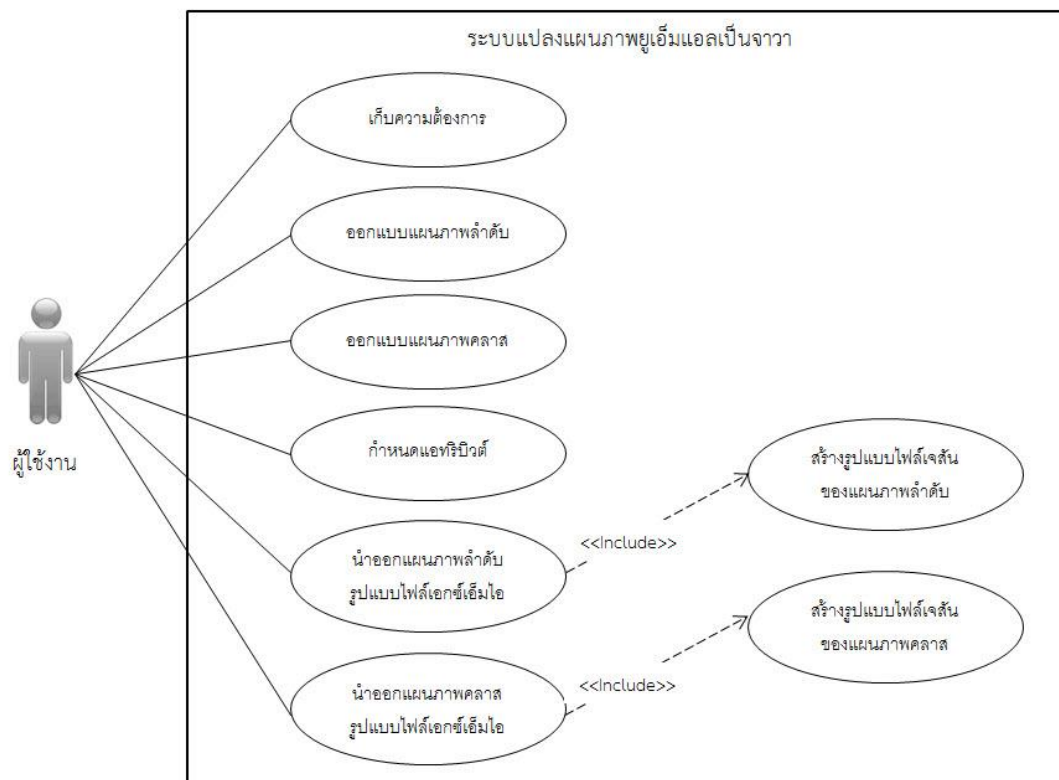
รายละเอียดในบทนี้จะกล่าวถึงวิธีการออกแบบและพัฒนาเครื่องมือต้นแบบสนับสนุนแนวทางในการทดสอบคุณลักษณะของการแปลงแผนภาพยูเอ็มแอลเป็นโค้ดจาวา โดยสร้างจากส่วนประกอบย่อยที่ได้จากการออกแบบแผนภาพลำดับและแผนภาพคลาสตามความต้องการที่ได้นำเสนอในบทที่ 3 โดยกล่าวถึงโครงสร้างของซอฟต์แวร์และสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ ซึ่งมีรายละเอียดดังต่อไปนี้

#### 4.1. ความต้องการเชิงฟังก์ชัน

- 1) ระบบจะสามารถอ่านไฟล์ที่ได้จากการนำออกของการออกแบบแผนภาพลำดับและแผนภาพคลาสที่เป็นรูปแบบไฟล์เอกซ์เอ็มไอได้
- 2) ระบบจะสามารถสร้างความสัมพันธ์ที่เกิดขึ้นระหว่างแผนภาพลำดับและแผนภาพคลาสแล้วนำมาสร้างเป็นคลาส เมทอด และเงื่อนไข ตามลำดับ
- 3) ระบบจะสามารถเชื่อมโยงความสัมพันธ์ระหว่างคลาสกับแอทริบิวต์ของแผนภาพลำดับและแผนภาพคลาสที่นำเสนอได้
- 4) ระบบจะสามารถนำเอาส่วนประกอบย่อยมาแสดงผลในรูปแบบของไฟล์จาวาได้

#### 4.2. การวิเคราะห์ความต้องการและแผนภาพฟังก์ชันงานของระบบ

จากความต้องการเชิงฟังก์ชัน จะสามารถวิเคราะห์เพื่อออกแบบเครื่องมือ ซึ่งหลังจากวิเคราะห์จะได้แบบจำลองเชิงฟังก์ชันที่สามารถแสดงเป็นแผนภาพยูสเคส (Use case diagram) ดังรูปที่ 37



รูปที่ 37 แผนภาพยูสเคสของระบบแปลงแผนภาพยูเอ็มแอลเป็นจาวา

จากแผนภาพยูสเคสเชิงฟังก์ชันแสดงให้เห็นถึงความเกี่ยวข้องของผู้ใช้งานในระบบและการตอบสนองจากผู้ใช้งาน โดยอธิบายรายละเอียดของแต่ละฟังก์ชันงานดังต่อไปนี้

1) ออกแบบแผนภาพลำดับและแผนภาพคลาส

ผู้ใช้งานจะทำการออกแบบแผนภาพลำดับและแผนภาพคลาสสำหรับโครงสร้างในการสร้างโค้ดจาวาในเครื่องมือวิซชั้วพาราตามยูเอ็มแอล โดยจะต้องระบุลำดับเหตุการณ์เพื่อที่จะสร้างเป็นไลฟ์ไลน์ การรับส่งข้อมูลและความสัมพันธ์ระหว่างคลาส รวมถึงการระบุคลาส โดยภายในคลาสจะประกอบด้วยแอทริบิวต์ เมท็อด ออลเทอร์นาทีฟ ทางเลือก และวนซ้ำ โดยอ้างอิงจากลำดับเหตุการณ์ที่ได้กำหนดไว้ เพื่อที่จะสามารถนำมาสร้างเป็นโค้ดจาวาได้

2) นำออกแผนภาพลำดับและแผนภาพคลาสในรูปแบบไฟล์เอกซ์เอ็มไอ

ผู้ใช้ทำการออกแบบแผนภาพลำดับและแผนภาพคลาสเสร็จแล้วจะต้องนำแผนภาพลำดับและแผนภาพคลาสออกมาอยู่ในรูปแบบไฟล์เอกซ์เอ็มไอ เพื่อให้ผู้ใช้ใช้ในการนำเข้าของข้อมูลระบบในส่วนต่อไป

### 3) นำเข้าข้อมูลไฟล์เอกซ์เอ็มไอ

ผู้ใช้นำไฟล์ที่ได้จากการนำออกของแผนภาพลำดับและแผนภาพคลาสที่อยู่ในรูปแบบของไฟล์เอกซ์เอ็มไอ โดยจะต้องนำเอาไฟล์ดังกล่าวไปวางยังตำแหน่งที่สามารถเข้าถึงไฟล์ได้และหลังจากนั้น ผู้ใช้จะต้องใช้คำสั่งในการแปลงรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์ของเจสัน ซึ่งการแปลงรูปแบบไฟล์เป็นรูปแบบไฟล์เจสันจะช่วยให้ระบบสามารถเข้าถึงข้อมูลได้ง่ายและสะดวกมากขึ้น

### 4) กำหนดส่วนประกอบแอทริบิวต์

กำหนดส่วนประกอบแอทริบิวต์ โดยจะกำหนดขึ้นจากรูปแบบไฟล์เจสันที่ได้จากขั้นตอนของการนำเข้าข้อมูลไฟล์เอกซ์เอ็มไอ เพื่อที่จะแบ่งแยกข้อมูลแต่ละประเภทได้อย่างถูกต้อง

## 4.3. สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือสนับสนุน

สภาพแวดล้อมที่ใช้ในการพัฒนาระบบจะอ้างอิงมาจากคอมพิวเตอร์ที่ใช้ในการพัฒนา โดยประกอบด้วยฮาร์ดแวร์ (hardware) และซอฟต์แวร์ (software) ที่ใช้ในการพัฒนาระบบ ซึ่งมีรายละเอียดดังต่อไปนี้

### 1) ระบบฮาร์ดแวร์

เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาระบบควรมีฮาร์ดแวร์ขั้นต่ำดังต่อไปนี้

- หน่วยการประมวล (CPU) ซีพียูอินเทลคอร์ไอ 7 ความเร็ว 2.50 กิกะเฮิร์ตซ์ (Intel(R) Core(TM) i7 CPU 2.50 HZ)
- หน่วยความจำสำรอง (Memory) ความเร็ว 16 กิกะไบต์ (Ram 16 GB)
- จานบันทึกแบบแข็ง (Hardisk) ความจุ 500 กิกะไบต์ (Hardisk 500 GB)

### 2) ซอฟต์แวร์

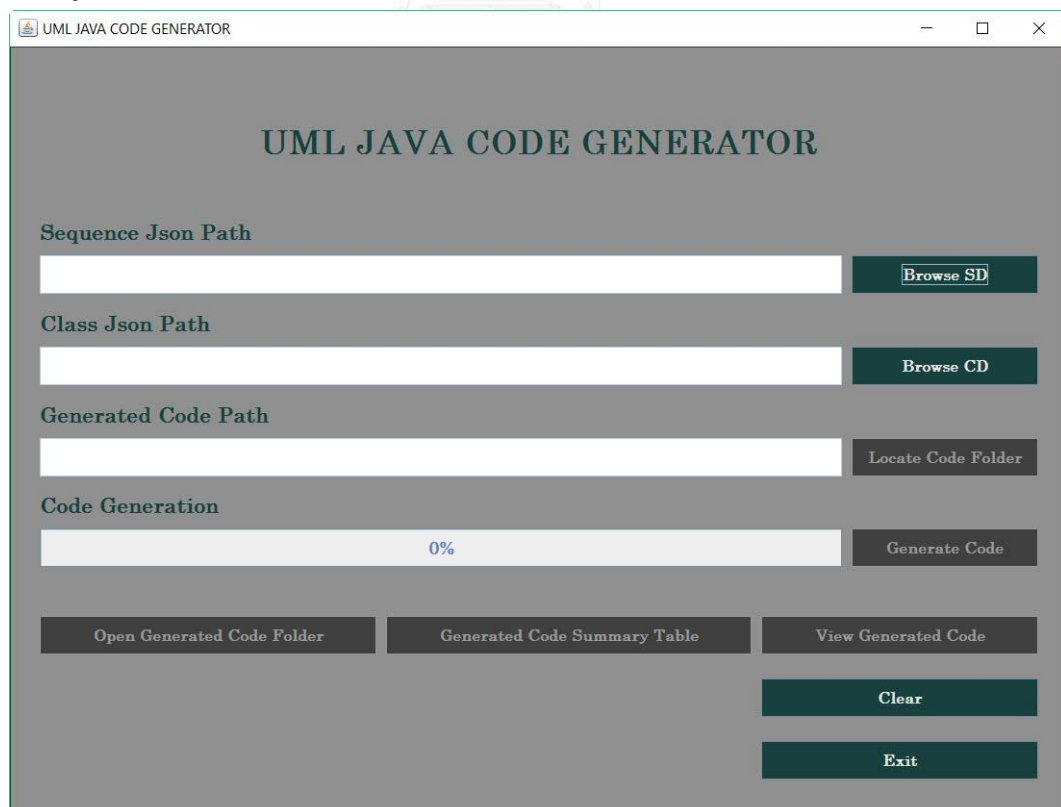
เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาระบบมีซอฟต์แวร์ดังต่อไปนี้

- ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 10 โฮม 64 บิต (Microsoft Windows 10 Home 64x bit)
- เครื่องมือสร้างแบบจำลองวิซิว พาราตาม Visual Paradigm
- โหนดเจเอส (node.js)
- โปรแกรมอินนิจเจ ไอเดีย IntelliJ IDEA เวอร์ชัน 14.1.4

- กูเกิ้ลโครม เว็บเบราว์เซอร์ (Google Chrome web browser) เวอร์ชัน 50.0.2661.102 m
- 3) การติดตั้งซอฟต์แวร์
- ทำการติดตั้งเครื่องมือในการพัฒนาระบบทั้งหมดในเครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาระบบ โดยเริ่มลำดับการติดตั้งตามขั้นตอนดังต่อไปนี้
- 1) ติดตั้งระบบปฏิบัติการไมโครซอฟวินโดวส์ 10 โสสม 64
  - 2) ติดตั้งโปรแกรมวิชชิวพาราดามยูเอ็มแอล
  - 3) ติดตั้งโปรแกรมโนนดเจเอส
  - 4) ติดตั้งโปรแกรมอินเทลลิเจไอเดีย
  - 5) ติดตั้งโปรแกรมกูเกิ้ลโครม เว็บเบราว์เซอร์

#### 4.4. การออกแบบหน้าจอ


จากการออกแบบหน้าจอเครื่องมือของการแปลงแผนภาพยูเอ็มแอลไปเป็นจาวา โดยงานวิจัยนี้จะใช้เครื่องมือที่ชื่อว่า “อินเทลลิเจไอเดีย” สำหรับการออกแบบหน้าจอ เพื่อให้สามารถใช้งานได้ง่าย ดังรูปที่ 38



รูปที่ 38 หน้าจอเครื่องมือของการแปลงแผนภาพยูเอ็มแอลไปเป็นจาวา

จากรูปที่ 38 แสดงหน้าจอของเครื่องมือนี้ โดยจะแสดงรายละเอียดดังนี้

- 1) Sequence Json Path  
แสดงรายละเอียดของเส้นทางในการนำเข้ารูปแบบไฟล์เจสันของแผนภาพลำดับ
- 2) ปุ่ม Browse SD  
เพื่อการค้นดูของเส้นทางที่ต้องการเรียกใช้งานของไฟล์เจสันของแผนภาพลำดับ
- 3) Class Json Path  
แสดงรายละเอียดของเส้นทางในการนำเข้ารูปแบบไฟล์เจสันของแผนภาพคลาส
- 4) ปุ่ม Browse CD  
เพื่อการค้นดูของเส้นทางที่ต้องการเรียกใช้งานของไฟล์เจสันของแผนภาพคลาส
- 5) Generated Code Path  
แสดงรายละเอียดของเส้นทางในการนำออกของรูปแบบไฟล์จาวา
- 6) ปุ่ม Locate Code Path  
เพื่อก่อตั้งเส้นทางของโค้ดในการนำออกของรูปแบบไฟล์จาวา
- 7) Code Generation  
แสดงเปอร์เซ็นต์กระบวนการทำงานของการแปลงโค้ด
- 8) ปุ่ม Generate Code  
เพื่อแปลงรูปแบบไฟล์เจสันให้อยู่ในรูปแบบของไฟล์จาวา
- 9) ปุ่ม Open Generated Code Folder  
เพื่อนำไปยังเส้นทางที่ได้ลงบันทึกของรูปแบบไฟล์จาวา
- 10) ปุ่ม Generated Code Summary Table  
เพื่อนำไปยังผลสรุปของการแปลงโค้ด
- 11) ปุ่ม View Generated Code  
เพื่อแสดงโค้ดจาวาที่ได้จากการแปลงจากระบบออกมาจากหน้าจอ
- 12) ปุ่ม Clear  
เพื่อลบข้อมูลออกจากหน้าจอทั้งหมด
- 13) ปุ่ม Exit  
เพื่อออกจากระบบของการแปลงเป็นโค้ดจาวา



The screenshot shows a window titled "UML JAVA CODE GENERATOR" with a table titled "GENERATED CODE SUMMARY TABLE". The table has two columns: "Name" and "Value".

Name	Value
Class	6
Method	6
Condition and Loop	3

รูปที่ 39 หน้าจอเครื่องมือของผลสรุปของการแปลงโค้ดจาวา

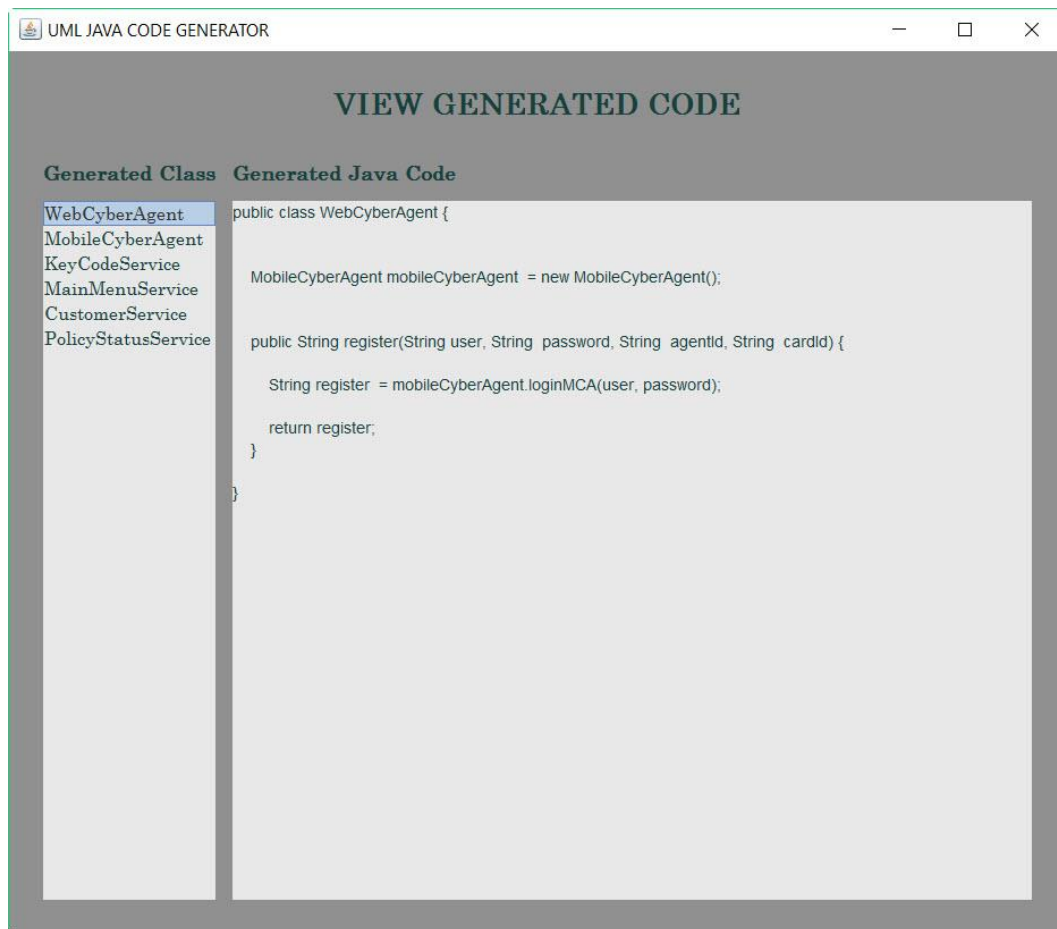
จากรูปที่ 39 แสดงหน้าจอผลสรุปของเครื่องมือนี้ โดยจะแสดงรายละเอียดดังนี้

1) ตาราง Name

แสดงรายละเอียดชื่อของคลาส เมทอด เงื่อนไขและวนซ้ำ

2) ตาราง Value

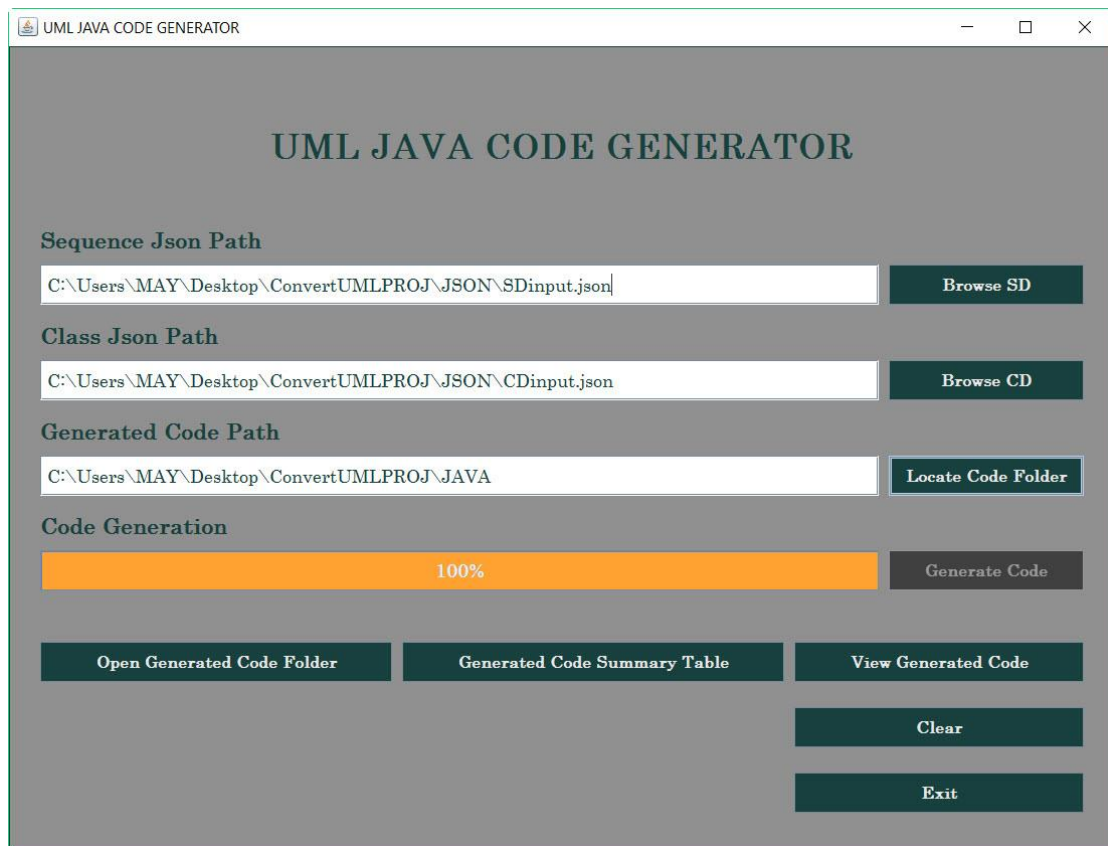
แสดงจำนวนทั้งหมดของคลาส เมทอด เงื่อนไขและวนซ้ำ



รูปที่ 40 หน้าจอเครื่องมือของการแสดงผลโค้ดจาวา

จากรูปที่ 40 แสดงหน้าจอผลสรุปของเครื่องมือนี้ โดยจะแสดงรายละเอียดดังนี้

- 1) Generated Class
  - แสดงชื่อของคลาสทั้งหมดที่มีความสัมพันธ์ในไฟล์เจสัน
- 2) Generated Java Class
  - แสดงโค้ดจาวาที่ได้จากการแปลงจากเครื่องมือ



รูปที่ 41 หน้าจอหลังจากการแปลงโค้ดจาวาเสร็จสมบูรณ์

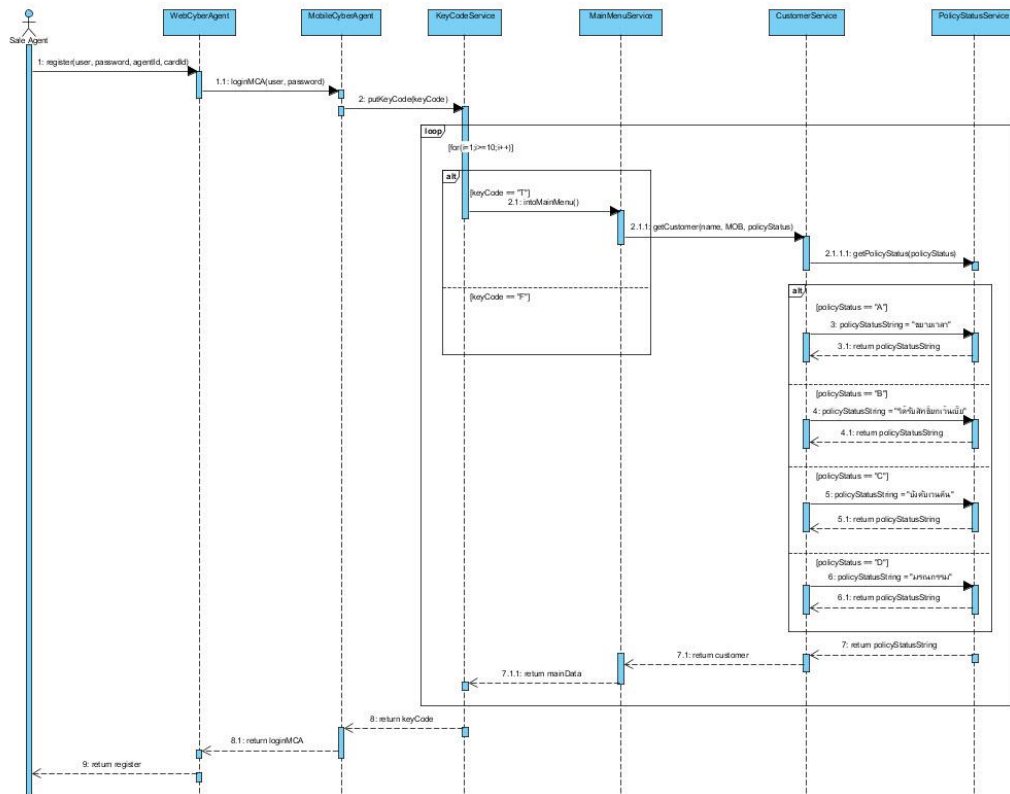
จากรูปที่ 41 แสดงถึงหน้าจอของผลลัพธ์สุดท้ายหลังจากการแปลงโค้ดจาวาจากโปรแกรม

#### 4.5. ขั้นตอนการทำงานของเครื่องมือ

เครื่องมือมีขั้นตอนการทำงานดังต่อไปนี้

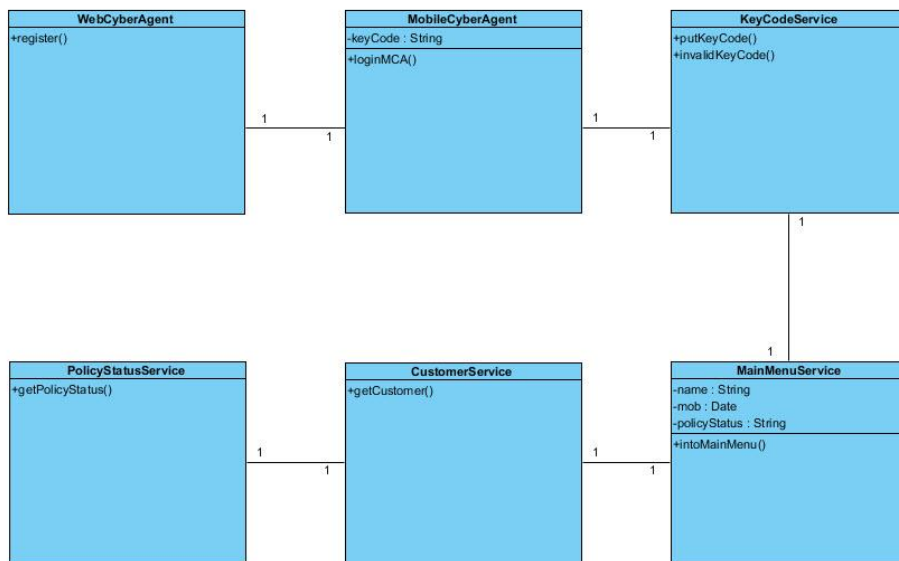
- 1) ผู้ใช้ทำการสร้างแผนภาพลำดับ เพื่อใช้ในการสร้างโค้ดจาวาจากเครื่องมือวิชัวพาราตาม ยูเอ็มแอล ดังรูปที่ 42





รูปที่ 42 การสร้างแผนภาพลำดับด้วยเครื่องมือวิชชัฟพาราตามยูเอ็มแอล

2) ผู้ใช้ทำการสร้างแผนภาพคลาส เพื่อใช้ในการสร้างโค้ดจาวาจากเครื่องมือวิชชัฟพาราตามยูเอ็มแอล ดังรูปที่ 43



รูปที่ 43 การสร้างแผนภาพคลาสด้วยเครื่องมือวิชชัฟพาราตามยูเอ็มแอล

- 3) ผู้ใช้จะต้องทำการนำออกแบบแผนภาพลำดับจากเครื่องมือวิซัวลพาราตามยูเอ็มแอล เพื่อให้ได้เป็นไฟล์เอกซ์เอ็มไอ ดังรูปที่ 44

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xmi:XMI xmi:version="2.1" xmlns:uml="http://schema.omg.org/spec/UML/2.0" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1">
3   <xmi:Documentation exporter="Visual Paradigm" exporterVersion="7.0.2">
4     <xmi:Extension extender="Visual Paradigm">
5       <nickname exportedFromDifferentName="false"/>
6     </xmi:Extension>
7   </xmi:Documentation>
8   <xmi:Extension extender="Visual Paradigm">
220 <uml:Model name="InsuranceThesis" xmi:id="001D7yqGAqACMAz5">
7831 <uml:Diagram diagramType="InteractionDiagram" documentation="" name="InsuranceThesis" toolName="Visual Paradigm" xmi:id="gBGD7yqGAqACMA8G">
7832 <xmi:Extension extender="Visual Paradigm">
7833   <properties>
7834     <ashowShapeStereotypeIconName value="true"/>
7835     <ashowAllocatedFrom value="true"/>
7836     <ashowAllocatedTo value="true"/>
7837     <apmAuthor value="MAY"/>
7838     <apmCreateDateTime value="1464445536257"/>
7839     <apmLastModified value="1467657064764"/>
7840     <ateamworkCreateDateTime value="0"/>
7841     <ashowPackageNameStyle value="0"/>
7842     <ashowConnectorName value="0"/>
7843     <apaintConnectorThroughLabel value="0"/>
7844     <ashowActivityStateNodeCaption value="65535"/>
7845     <adocumentation htmlValue="&lt;html&gt;&#13;&#10; &lt;head&gt;&#13;&#10; &lt;style type=&quot;text/css&quot;&gt;&#10;
7846     <aditable value="true"/>
7847     <afollowDiagramParentElement value="true"/>
7848     <aautoFitShapesSize value="false"/>
7849     <ashowTaggedValues value="false"/>
7850     <ahideEmptyTaggedValues value="false"/>
7851     <ashowStereotypes value="true"/>
7852     <ashowPMIteration value="false"/>
7853     <ashowPMPhase value="false"/>
7854     <ashowPMDiscipline value="false"/>
7855     <ashowPMVersion value="false"/>

```

รูปที่ 44 รูปแบบไฟล์เอกซ์เอ็มไอที่ได้จากการนำออกของแผนภาพลำดับด้วยเครื่องมือวิซัวลพาราตามยูเอ็มแอล

- 4) ผู้ใช้จะต้องทำการนำออกแบบแผนภาพลำดับจากเครื่องมือวิซัวลพาราตามยูเอ็มแอล เพื่อให้ได้เป็นไฟล์เอกซ์เอ็มไอ ดังรูปที่ 45

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xmi:XMI xmi:version="2.1" xmlns:uml="http://schema.omg.org/spec/UML/2.0" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1">
3   <xmi:Documentation exporter="Visual Paradigm" exporterVersion="7.0.2">
4     <xmi:Extension extender="Visual Paradigm">
5       <nickname exportedFromDifferentName="false"/>
6     </xmi:Extension>
7   </xmi:Documentation>
8   <xmi:Extension extender="Visual Paradigm">
220 <uml:Model name="CD" xmi:id="KioIraqAUNimHgOC">
7593 <uml:Diagram diagramType="ClassDiagram" documentation="" name="Class Diagram1" toolName="Visual Paradigm" xmi:id="T2EIraqAUNimHgWB">
7594 <xmi:Extension extender="Visual Paradigm">
7595   <properties>
7596     <ashowShapeStereotypeIconName value="true"/>
7597     <ashowAllocatedFrom value="true"/>
7598     <ashowAllocatedTo value="true"/>
7599     <apmAuthor value="MAY"/>
7600     <apmCreateDateTime value="1467621607154"/>
7601     <apmLastModified value="1467657456109"/>
7602     <ateamworkCreateDateTime value="0"/>
7603     <ashowPackageNameStyle value="0"/>
7604     <ashowConnectorName value="0"/>
7605     <apaintConnectorThroughLabel value="1"/>
7606     <ashowActivityStateNodeCaption value="65535"/>
7607     <adocumentation htmlValue="&lt;html&gt;&#13;&#10; &lt;head&gt;&#13;&#10; &lt;style type=&quot;text/css&quot;&gt;&#10;
7608     <aditable value="true"/>
7609     <afollowDiagramParentElement value="true"/>
7610     <aautoFitShapesSize value="false"/>
7611     <ashowTaggedValues value="false"/>
7612     <ahideEmptyTaggedValues value="false"/>
7613     <ashowStereotypes value="true"/>
7614     <ashowPMIteration value="false"/>
7615     <ashowPMPhase value="false"/>
7616     <ashowPMDiscipline value="false"/>
7617     <ashowPMVersion value="false"/>

```

รูปที่ 45 รูปแบบไฟล์เอกซ์เอ็มไอที่ได้จากการนำออกของแผนภาพคลาสด้วยเครื่องมือวิซัวลพาราตามยูเอ็มแอล

- 5) เมื่อได้ไฟล์เอกซ์เอ็มไอจะนำเข้าไปในเครื่องมือที่สนับสนุนของงานวิจัยนี้ โดยจะนำเข้าผ่านคอมมานด์ไลน์ และจะต้องมีการกำหนดค่าสำหรับการแปลง เช่น ระบุที่อยู่ของไฟล์ ระบุปลายทางที่จะนำไฟล์บันทึก เป็นต้น โดยในงานวิจัยนี้จะกำหนดฟังก์ชันการแปลงจากรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจสัน ดังรูปที่ 46 และ 47 ตามลำดับ

```

gulpfile.js x
1  var gulp = require('gulp');
2  var xml2json = require('gulp-xml2json');
3  var rename = require('gulp-rename');
4
5  gulp.task('convertToJson', function () {
6      gulp.src('src/**/*.SDinput.xml')
7          .pipe(xml2json())
8          .pipe(rename({extname: '.json'}))
9          .pipe(gulp.dest('dist'));
10 });

```

รูปที่ 46 คำสั่งการแปลงรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจสันของแผนภาพลำดับ

```

gulpfile.js x
1  var gulp = require('gulp');
2  var xml2json = require('gulp-xml2json');
3  var rename = require('gulp-rename');
4
5  gulp.task('convertToJson', function () {
6      gulp.src('src/**/*.CDinput.xml')
7          .pipe(xml2json())
8          .pipe(rename({extname: '.json'}))
9          .pipe(gulp.dest('dist'));
10 });

```

รูปที่ 47 คำสั่งการแปลงรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจสันของแผนภาพคลาส

โดยหลังจากใช้คำสั่งจะได้รูปแบบไฟล์เจสัน ซึ่งจะช่วยในการทำงาน ซึ่งจะสามารถเข้าถึงแอทริบิวต์ของวัตถุได้ง่ายมากขึ้น ดังรูปที่ 48 และ 49 ตามลำดับ

```

{
  "xmi:XMI": {
    "$": {
      "xmi:Documentation": [{}
      "xmi:Extension": [{}
      "uml:Model": [{}
      "uml:Diagram": [{}
        "$": {
          "diagramType": "InteractionDiagram",
          "documentation": "",
          "name": "InsuranceThesis",
          "toolName": "Visual Paradigm",
          "xmi:id": "gBGD7yqGAqACMA8G"
        },
        "xmi:Extension": [{}
          "$": {
            "extender": "Visual Paradigm"
          },
          "properties": [{}
            "ashowShapeStereotypeIconName": [{}
              "$": {
                "value": "true"
              }
            ]
          ]
        ]
      ]
    }
  }
}

```

รูปที่ 48 รูปแบบไฟล์เจสันที่ได้จากคำสั่งการแปลงรูปแบบไฟล์ของแผนภาพลำดับ

```

{
  "xmi:XMI": {
    "$": {
      "xmi:Documentation": [{}
      "xmi:Extension": [{}
      "uml:Model": [{}
      "uml:Diagram": [{}
        "$": {
          "diagramType": "ClassDiagram",
          "documentation": "",
          "name": "Class Diagram1",
          "toolName": "Visual Paradigm",
          "xmi:id": "T2EIraqAUNimHgWB"
        },
        "xmi:Extension": [{}
          "$": {
            "extender": "Visual Paradigm"
          },
          "properties": [{}
            "ashowShapeStereotypeIconName": [{}
              "$": {
                "value": "true"
              }
            ]
          ]
        ]
      ]
    }
  }
}

```

รูปที่ 49 รูปแบบไฟล์เจสันที่ได้จากคำสั่งการแปลงรูปแบบไฟล์ของแผนภาพคลาส

- 6) การวิเคราะห์รูปแบบไฟล์เจสันของแผนภาพคลาส เพื่อนำมาสกัดข้อมูลที่เป็นประโยชน์ และมีความสัมพันธ์มาใช้ในการสร้างคลาสของโค้ดจาวา ซึ่งในแต่ละคลาสจะแสดงถึง แอ ทริบิวต์และเมทอด เมื่อได้ข้อมูลที่สกัดมาก็นำมาจัดรูปแบบตามที่ได้กำหนดไว้

```

WebCyberAgent.java x
1 public class WebCyberAgent {
2
3
4
5
6
7
8 }

```

รูปที่ 50 คลาสของ “WebCyberAgent”

```

MobileCyberAgent.java x
1 public class MobileCyberAgent {
2
3
4
5
6
7
8 }

```

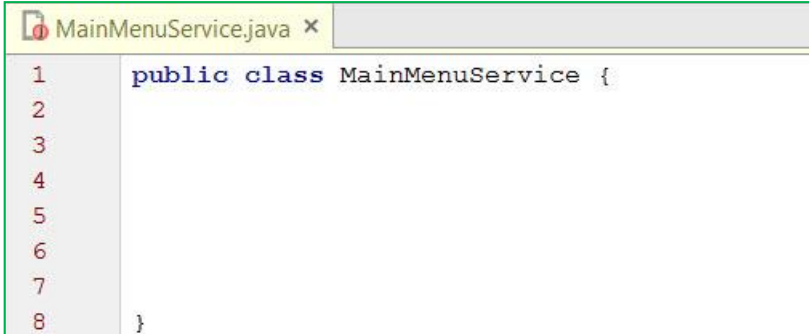
รูปที่ 51 คลาสของ “MoileCyberAgent”

```

KeyCodeService.java x
1 public class KeyCodeService {
2
3
4
5
6
7
8 }

```

รูปที่ 52 คลาสของ “KeyCodeService”



```

1 public class MainMenuService {
2
3
4
5
6
7
8 }

```

รูปที่ 53 คลาสของ “MainMenuService”

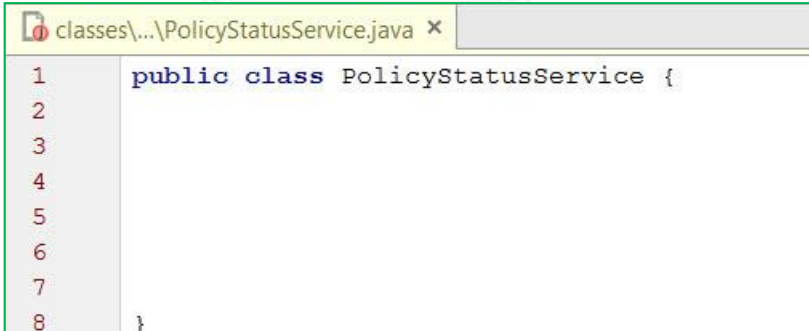


```

1 public class CustomerService {
2
3
4
5
6
7
8 }

```

รูปที่ 54 คลาสของ “CustomerService”



```

1 public class PolicyStatusService {
2
3
4
5
6
7
8 }

```

รูปที่ 55 คลาสของ “PolicyStatusService”

จากรูปที่ 50-55 จะแสดงรูปแบบโครงสร้างของคลาส โดยจะมีความสัมพันธ์กับแผนภาพคลาสที่กล่าวถึงข้างต้น

- 7) การวิเคราะห์รูปแบบไฟล์เจสันของแผนภาพคลาส เพื่อนำมาสกัดข้อมูลที่เป็นประโยชน์และมีความสัมพันธ์มาใช้ในการสร้างแอทริบิวต์ของแต่ละคลาส เมื่อได้ข้อมูลแล้วก็นำมาจัดในตำแหน่งที่ถูกต้องตามหลักโครงสร้างของโค้ดจาวา

```

MobileCyberAgent.java x
1 public class MobileCyberAgent {
2
3     private String keyCode;
4
5
6
7
8 }

```

รูปที่ 56 แอทริบิวต์ที่เป็นส่วนหนึ่งของคลาส “MobileCyberAgent”

```

MainMenuService.java x
1 public class MainMenuService {
2
3     private String name;
4     private Date mob;
5     private String policyStatus;
6
7
8 }

```

รูปที่ 57 แอทริบิวต์ที่เป็นส่วนหนึ่งของคลาส “MainMenuService”

จากรูปที่ 56-57 จะแสดงรูปแบบโครงสร้างของคลาส โดยจะมีความสัมพันธ์กับแผนภาพคลาสที่กล่าวถึงข้างต้น

- 8) การวิเคราะห์รูปแบบไฟล์เจสันของแผนภาพคลาส เพื่อนำมาสกัดข้อมูลที่เป็นประโยชน์และมีความสัมพันธ์มาใช้ในการสร้างเมทอดของแต่ละคลาส เมื่อได้ข้อมูลแล้วก็นำมาจัดรูปแบบตามที่ได้กำหนดไว้

```
WebCyberAgent.java ×
1 public class WebCyberAgent {
2
3
4     public String register() {
5
6     }
7
8 }
```

รูปที่ 58 เมท็อดของ “WebCyberAgent”

```
MobileCyberAgent.java ×
1 public class MobileCyberAgent {
2
3     private String keyCode;
4
5     public String loginMCA() {
6
7     }
8
9 }
```

รูปที่ 59 เมท็อดของ “MobileCyberAgent”

```
KeyCodeService.java ×
1 public class KeyCodeService {
2
3     public String putKeyCode() {
4
5     }
6
7
8 }
```

รูปที่ 60 เมท็อดของ “KeyCodeService”



```

MainMenuService.java x
1 public class MainMenuService {
2
3     private String name;
4     private Date mob;
5     private String policyStatus;
6
7
8     public String intoMainMenu() {
9
10
11     }
12
13 }

```

รูปที่ 61 เมทอดของ “MainMenuService”

```

classes\...\CustomerService.java x
1 public class CustomerService {
2
3     public String getCustomer() {
4
5
6     }
7
8 }

```

รูปที่ 62 เมทอดของ “CustomerService”

```

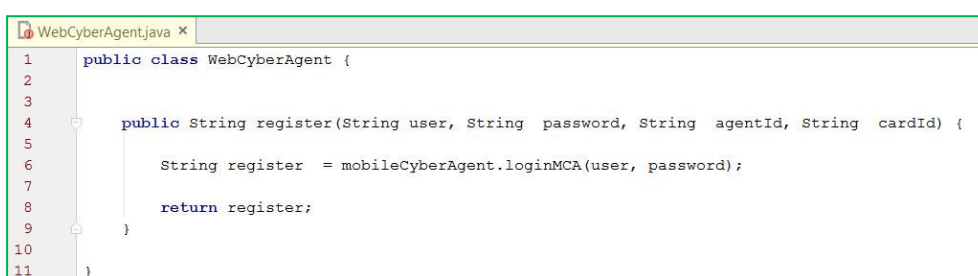
classes\...\PolicyStatusService.java x
1 public class PolicyStatusService {
2
3
4     public String getPolicyStatus() {
5
6     }
7
8 }

```

รูปที่ 63 เมทอดของ “PolicyStatusService”

จากรูปที่ 58-63 จะแสดงรูปแบบโครงสร้างของเมทอดที่มีความสัมพันธ์กับคลาสแต่ละคลาส โดยจะมีความสัมพันธ์กับ แผนภาพคลาสที่กล่าวถึงข้างต้น

- 9) การวิเคราะห์ความสัมพันธ์ระหว่างเมทอดและการรับส่งข้อมูลจากรูปแบบไฟล์เจสันของแผนภาพลำดับและแผนภาพคลาส เพื่อนำมาสกัดข้อมูลที่เป็นประโยชน์และมีความสัมพันธ์มาใช้ในการสร้างพารามิเตอร์ของเมทอด การรับส่งข้อมูลกลับไปยังคลาสที่เรียก และความสัมพันธ์ระหว่างคลาส เมื่อได้ข้อมูลแล้วก็นำมาจัดรูปแบบตามที่กำหนดไว้

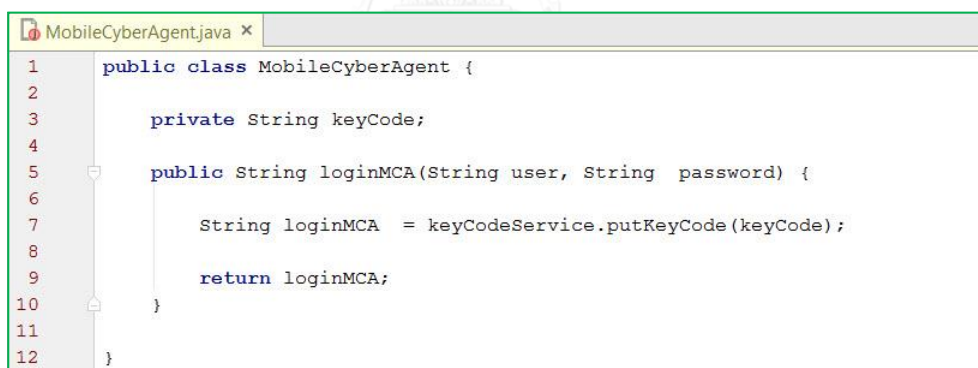


```

WebCyberAgent.java x
1 public class WebCyberAgent {
2
3
4     public String register(String user, String password, String agentId, String cardId) {
5
6         String register = mobileCyberAgent.loginMCA(user, password);
7
8         return register;
9     }
10
11 }

```

รูปที่ 64 ความสัมพันธ์ของคลาส “WebCyberAgent”



```

MobileCyberAgent.java x
1 public class MobileCyberAgent {
2
3     private String keyCode;
4
5     public String loginMCA(String user, String password) {
6
7         String loginMCA = keyCodeService.putKeyCode(keyCode);
8
9         return loginMCA;
10    }
11
12 }

```

รูปที่ 65 ความสัมพันธ์ของคลาส “MobileCyberAgent”

```

KeyCodeService.java x
1  public class KeyCodeService {
2
3
4  public String putKeyCode(String keyCode) {
5
6
7
8      return keyCode;
9  }
10
11 }

```

รูปที่ 66 ความสัมพันธ์ของคลาส “KeyCodeService”

```

MainMenuService.java x
1  public class MainMenuService {
2
3      private String name;
4      private Date mob;
5      private String policyStatus;
6
7  public String intoMainMenu() {
8
9      String mainData = customerService.getCustomer(name, MOB, policyStatus);
10
11     return mainData;
12 }
13
14 }

```

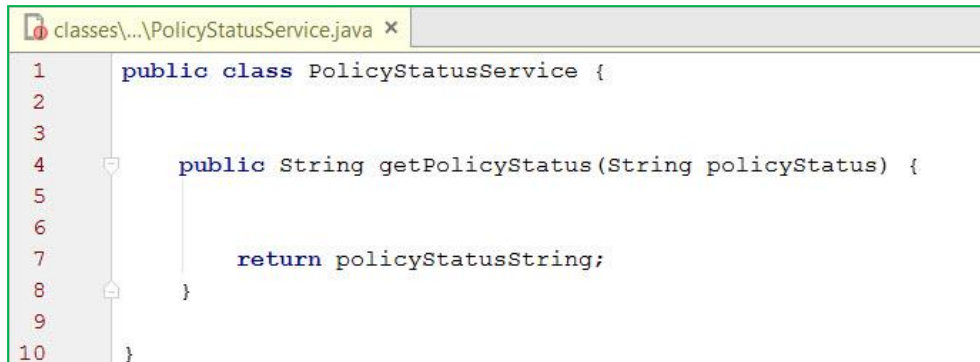
รูปที่ 67 ความสัมพันธ์ของคลาส “MainMenuService”

```

classes\...\CustomerService.java x
1  public class CustomerService {
2
3
4  public String getCustomer(String name, Date MOB, String policyStatus) {
5
6      String customer = policyStatusService.getPolicyStatus(policyStatus);
7
8      return customer;
9  }
10
11 }

```

รูปที่ 68 ความสัมพันธ์ของคลาส “CustomerService”



```

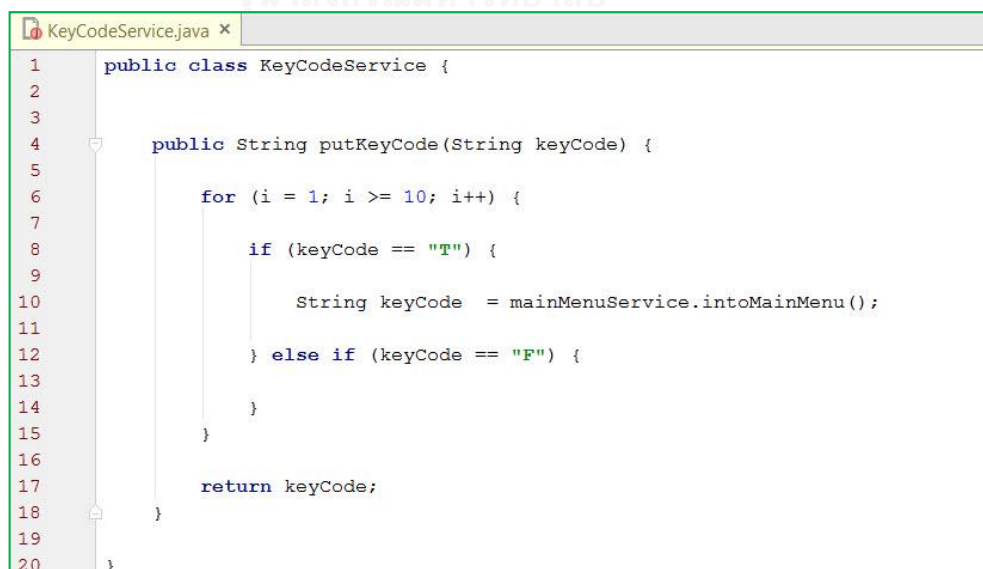
1 public class PolicyStatusService {
2
3
4     public String getPolicyStatus(String policyStatus) {
5
6
7         return policyStatusString;
8     }
9
10 }

```

รูปที่ 69 ความสัมพันธ์ของคลาส “PolicyStatusService”

จากรูปที่ 64-69 จะแสดงรูปแบบโครงสร้างของเมทอดที่มีความสัมพันธ์กับคลาสแต่ละคลาส รวมถึงการรับส่งข้อมูลระหว่างคลาส โดยจะมีความสัมพันธ์กับแผนภาพลำดับและแผนภาพคลาสที่กล่าวถึงข้างต้น

- 10) การวิเคราะห์รูปแบบไฟล์เจสันของแผนภาพลำดับในส่วนของเงื่อนไข เพื่อสกัดข้อมูลที่เป็นประโยชน์ และมีความสัมพันธ์มาใช้ในการสร้างเงื่อนไขที่จะประกอบไปด้วยออลเทอร์เนทิฟ ทางเลือกและวนซ้ำของแต่ละคลาสที่มีความสัมพันธ์กับเงื่อนไขที่ได้กล่าวถึง ซึ่งในส่วนของโปรแกรมของงานวิจัยนี้ได้ทำการสร้างความสัมพันธ์ให้อยู่ในรูปของโครงสร้างของโค้ดจาวา



```

1 public class KeyCodeService {
2
3
4     public String putKeyCode(String keyCode) {
5
6         for (i = 1; i >= 10; i++) {
7
8             if (keyCode == "T") {
9
10                String keyCode = mainMenuService.intoMainMenu();
11            } else if (keyCode == "F") {
12
13            }
14        }
15
16        return keyCode;
17    }
18
19
20 }

```

รูปที่ 70 คลาสที่มีเงื่อนไขของออลเทอร์เนทิฟและวนซ้ำ

```

classes\...\PolicyStatusService.java x
1  public class PolicyStatusService {
2
3  public String getPolicyStatus(String policyStatus) {
4
5      String policyStatusString = "";
6
7      switch (String input) {
8          case "A" :
9              policyStatusString = "ขยายเวลา"
10             break;
11         case "B" :
12             policyStatusString = "ได้รับสิทธิ์ยกเว้นเบี้ย"
13             break;
14         case "C" :
15             policyStatusString = "บังคับเวนคืน"
16             break;
17         case "D" :
18             policyStatusString = "มรดกกรรม"
19             break;
20     }
21
22
23     return policyStatusString;
24 }
25
26 }

```

รูปที่ 71 คลาสที่มีเงื่อนไขของสวิตช์เคส

จากรูปที่ 70 และ 71 จะแสดงรูปแบบโครงสร้างของเงื่อนไข อย่างเช่น ออลเทอร์เนทีฟ ทางเลือก และวนซ้ำ เฉพาะเมทอดที่มีความสัมพันธ์กับเงื่อนไข

- 11) การสร้างโค้ดจาวาจะนำส่วนประกอบย่อยจากที่กล่าวข้างต้นมารวบรวม โดยจะรวมถึงชื่อของคลาส เมทอด และเงื่อนไข เพื่อที่จะนำมาสร้างเป็นโค้ดจาวาที่สมบูรณ์ ซึ่งรูปแบบไฟล์ที่ได้จะเป็นนามสกุลจาวา และจะเพิ่มส่วนประกอบย่อย อาทิเช่น อ็อบเจกต์ที่ไว้สำหรับอ้างอิงถึงเมทอดที่คลาสนั้นเรียกใช้

```

WebCyberAgent.java x
1  public class WebCyberAgent {
2
3
4      MobileCyberAgent mobileCyberAgent = new MobileCyberAgent();
5
6
7  public String register(String user, String password, String agentId, String cardId) {
8
9      String register = mobileCyberAgent.loginMCA(user, password);
10
11     return register;
12 }
13
14 }

```

รูปที่ 72 ไฟล์จาวาของ “WebCyberAgent”

```

MobileCyberAgent.java x
1  public class MobileCyberAgent {
2
3      private String keyCode;
4
5      KeyCodeService keyCodeService = new KeyCodeService();
6
7
8      public String loginMCA(String user, String password) {
9
10         String loginMCA = keyCodeService.putKeyCode(keyCode);
11
12         return loginMCA;
13     }
14
15 }

```

รูปที่ 73 ไฟล์จาวาของ “MobileCyberAgent”

```

KeyCodeService.java x
1  public class KeyCodeService {
2
3
4      MainMenuService mainMenuService = new MainMenuService();
5
6
7      public String putKeyCode(String keyCode) {
8
9          for (i = 1; i >= 10; i++) {
10
11              if (keyCode == "T") {
12                  String keyCode = mainMenuService.intoMainMenu();
13              } else if (keyCode == "F") {
14              }
15          }
16
17          return keyCode;
18      }
19
20
21
22 }
23
24 }

```

รูปที่ 74 ไฟล์จาวาของ “KeyCodeService”

```

MainMenuService.java x
1  public class MainMenuService {
2
3      private String name;
4      private Date mob;
5      private String policyStatus;
6
7      CustomerService customerService = new CustomerService();
8
9
10     public String intoMainMenu() {
11
12         String mainData = customerService.getCustomer(name, MOB, policyStatus);
13
14         return mainData;
15     }
16
17 }

```

รูปที่ 75 ไฟล์จาวาของ “MainMenuService”

```

classes\...\CustomerService.java x
1  public class CustomerService {
2
3
4      PolicyStatusService policyStatusService = new PolicyStatusService();
5
6
7     public String getCustomer(String name, Date MOB, String policyStatus) {
8
9         String customer = policyStatusService.getPolicyStatus(policyStatus);
10
11         return customer;
12     }
13
14 }

```

รูปที่ 76 ไฟล์จาวาของ “CustomerService”

```

classes\...\PolicyStatusService.java x
1  public class PolicyStatusService {
2
3
4  public String getPolicyStatus(String policyStatus) {
5
6      String policyStatusString = "";
7
8      switch (String input) {
9          case "A" :
10             policyStatusString = "ขยายเวลา"
11             break;
12          case "B" :
13             policyStatusString = "ได้รับสิทธิ์ยกเว้นเบี้ย"
14             break;
15          case "C" :
16             policyStatusString = "บังคับเกณฑ์"
17             break;
18          case "D" :
19             policyStatusString = "มรดกกรม"
20             break;
21      }
22
23
24      return policyStatusString;
25  }
26
27  }

```

รูปที่ 77 ไฟล์จาวาของ “PolicyStatusService”

จากรูปที่ 72-77 จะแสดงผลลัพธ์ทั้งหมดที่ได้จากโปรแกรม โดยจะแสดงถึงคลาส เมทีอด และเงื่อนไข ที่มีความสัมพันธ์กัน ซึ่งจะได้จากการอ้างอิงของแผนภาพลำดับและ แผนภาพคลาสที่ได้กล่าวข้างต้น



## บทที่ 5

### การทดสอบและการวิเคราะห์ผล

#### 5.1. วัตถุประสงค์ของการทดสอบ

จุดประสงค์ของการทดสอบ เพื่อสนับสนุนแนวทางในการสร้างโค้ดจาวาที่ได้ออกแบบและพัฒนาเครื่องมือที่สนับสนุนแนวทางในบทที่ 4 โดยเนื้อหาจะประกอบไปด้วยการทดสอบระบบ ตั้งแต่การนำเข้าของแผนภาพลำดับและแผนภาพคลาส รวมถึงผลลัพธ์ส่วนประกอบย่อยของโครงสร้างของโค้ดจาวา ตลอดจนผลลัพธ์ของรูปแบบไฟล์จาวา

#### 5.2. การทดสอบระบบ

การทดสอบระบบที่ใช้ในแนวทางสำหรับการสร้างโค้ดจาวา ได้ออกแบบวิธีทดสอบแบบ (Black Box Testing) ซึ่งจะทดสอบตามส่วนประกอบย่อยของความถี่ และมุ่งเน้นทดสอบฟังก์ชันการทำงาน รวมถึงผลลัพธ์ที่ได้จากการรวมทั้งหมดของส่วนประกอบย่อย ดังต่อไปนี้

##### 1) ทดสอบการนำเข้าของแผนภาพลำดับในรูปแบบของไฟล์เอกซ์เอ็มไอ

การทดสอบการนำเข้าของรูปแบบไฟล์เอกซ์เอ็มไอของแผนภาพลำดับ มีวัตถุประสงค์เพื่อนำเข้าและสกัดข้อมูลที่สำคัญที่จะนำมาใช้ในเครื่องมือ โดยรูปแบบไฟล์เอกซ์เอ็มไอของแผนภาพลำดับจะเป็นข้อมูลนำเข้าที่สำคัญสำหรับเครื่องมือ เพราะเป็นข้อมูลนำเข้าต้นแบบสำหรับการสร้างโค้ดจาวา

##### 2) ทดสอบการนำเข้าของแผนภาพคลาสในรูปแบบไฟล์เอกซ์เอ็มไอ

การทดสอบการนำเข้าของรูปแบบไฟล์เอกซ์เอ็มไอของแผนภาพคลาส มีวัตถุประสงค์เพื่อนำเข้าและสกัดข้อมูลที่สำคัญที่จะนำมาใช้ในเครื่องมือ โดยรูปแบบไฟล์เอกซ์เอ็มไอของแผนภาพคลาสจะเป็นข้อมูลนำเข้าที่สำคัญสำหรับเครื่องมือ เพราะเป็นข้อมูลนำเข้าต้นแบบสำหรับการสร้างโค้ดจาวา

##### 3) ทดสอบเครื่องมือในการออกแบบการใช้งานและกำหนดตำแหน่งของแต่ละส่วนประกอบ

วัตถุประสงค์คือสามารถใช้งานได้ตรงตามข้อกำหนดสำหรับการออกแบบหน้าจอ ว่ามีความถูกต้องหรือไม่และกำหนดตำแหน่งของแต่ละส่วนประกอบย่อย

##### 4) ทดสอบเครื่องมือในการสกัดข้อมูล เพื่อให้ได้โครงสร้างของคลาส

วัตถุประสงค์คือสามารถสร้างส่วนประกอบย่อยโครงสร้างของคลาส เพื่อนำเอาไปแสดงผลในรูปแบบของไฟล์จาวา

5) ทดสอบเครื่องมือในการสกัดข้อมูล เพื่อให้ได้ความสัมพันธ์ของเมท็อด

วัตถุประสงค์คือสามารถสร้างส่วนประกอบย่อยโครงสร้างของเมท็อด รวมทั้งชนิดของข้อมูล เพื่อนำเอาไปแสดงผลได้ถูกต้อง

6) ทดสอบเครื่องมือในการสกัดข้อมูล เพื่อให้ได้ความสัมพันธ์เงื่อนไข

วัตถุประสงค์คือสามารถสร้างส่วนประกอบย่อยโครงสร้างของเงื่อนไขที่ประกอบไปด้วย ออลเทอร์เนทิฟ ทางเลือก และวนซ้ำ เพื่อนำเอาไปแสดงผล รวมถึงการเชื่อมความสัมพันธ์ระหว่างคลาส

7) ทดสอบเครื่องมือในการรวบรวมส่วนประกอบย่อยทั้งหมด เพื่อให้ได้โค้ดจาวา

วัตถุประสงค์คือสามารถรวบรวมส่วนประกอบย่อยทั้งหมดไม่ว่าจะเป็นคลาส เมท็อด และเงื่อนไข เพื่อนำมาแสดงอยู่ในรูปแบบของไฟล์จาวา

ตารางที่ 8 ทดสอบการนำเข้าของแผนภาพลำดับในรูปแบบของไฟล์เอกซ์เอ็มไอ

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC01	นำเข้าข้อมูลทดสอบที่ได้จากการนำออกจากเครื่องมือวิชชัวพาราตามยูเอ็มแอลในรูปแบบไฟล์เอกซ์เอ็มไอของแผนภาพลำดับ	ระบบจะต้องแปลงข้อมูลรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจสันจากแผนภาพลำดับ รวมถึงตำแหน่งไฟล์ผลลัพธ์จะต้องอยู่ในตำแหน่งที่กำหนดไว้	ถูกต้อง

ตารางที่ 9 ทดสอบการนำเข้าของแผนภาพคลาสในรูปแบบของไฟล์เอกซ์เอ็มไอ

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC02	นำเข้าข้อมูลทดสอบที่ได้จากการนำออกจากเครื่องมือวิชชัวพาราตามยูเอ็มแอลในรูปแบบไฟล์เอกซ์เอ็มไอของแผนภาพคลาส	ระบบจะต้องแปลงข้อมูลรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจสันจากแผนภาพคลาส รวมถึงตำแหน่งไฟล์ผลลัพธ์จะต้องอยู่ในตำแหน่งที่กำหนดไว้	ถูกต้อง

ตารางที่ 10 ทดสอบเครื่องมือในการออกแบบการใช้งานและกำหนดตำแหน่งของแต่ละส่วนประกอบ

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC03	การค้นดูข้อมูลของในรูปแบบไฟล์เจสันของแผนภาพลำดับ	ระบบจะสามารถนำเข้าข้อมูลที่อยู่ในรูปแบบไฟล์เจสันของแผนภาพลำดับเท่านั้น	ถูกต้อง
TC04	การค้นดูข้อมูลของในรูปแบบไฟล์เจสันของแผนภาพคลาส	ระบบจะสามารถนำเข้าข้อมูลที่อยู่ในรูปแบบไฟล์เจสันของแผนภาพคลาสเท่านั้น	ถูกต้อง
TC05	การก่อตั้งเส้นทางของโค้ดในการนำออกของรูปแบบไฟล์จาวา	ระบบจะสามารถนำออกไฟล์ของข้อมูลตามตำแหน่งไฟล์ผลลัพธ์ที่กำหนดไว้ได้	ถูกต้อง
TC06	การแปลงเป็นโค้ดในรูปแบบไฟล์จาวา	ระบบจะสามารถนำออกมาเป็นรูปแบบไฟล์จาวาได้	ถูกต้อง
TC07	การทำงานการดำเนินการของการสร้างโค้ดจาวาที่แสดงในรูปแบบของเปอร์เซ็นต์	ระบบจะสามารถแสดงเปอร์เซ็นต์การดำเนินการในการแปลงโค้ดของระบบได้	ถูกต้อง
TC08	การค้นดูข้อมูลหลังจากบันทึกรูปแบบไฟล์จาวา	ระบบจะสามารถเชื่อมโยงไปยังเส้นทางของรูปแบบไฟล์จาวาที่บันทึกไว้	ถูกต้อง
TC09	การแสดงผลลัพธ์หลังจากแปลงเป็นไฟล์จาวา	ระบบจะสามารถแสดงผลลัพธ์ออกมาทางหน้าจอว่ามีจำนวนของคลาส เมทอด เงื่อนไขและวนซ้ำได้	ถูกต้อง
TC10	การแสดงผลโค้ดจาวาทางหน้าจอหลังจากแปลงเป็นรูปแบบไฟล์จาวา	ระบบสามารถแสดงโค้ดจาวาออกมาทางหน้าจอได้	ถูกต้อง
TC11	การลบล้างของหน้าจอ	ระบบสามารถลบล้างข้อมูลที่แสดงผลทางหน้าจอได้	ถูกต้อง
TC12	การออกจากโปรแกรม	ระบบจะสามารถออกจากโปรแกรมได้	ถูกต้อง
TC13	ในส่วนของหน้าจอจะสามารถแสดงผลส่วนของหน้าจอส่วนแรก	ระบบจะสามารถแสดงผลของหน้าจอส่วนแรกตามที่กำหนดได้	ถูกต้อง

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC14	ในส่วนของหน้าจอจะสามารถแสดงผล ส่วนของหน้าจอส่วนกลาง	ระบบจะสามารถแสดงผลของหน้าจอ ส่วนกลางตามที่กำหนดได้	ถูกต้อง
TC15	ในส่วนของหน้าจอจะสามารถแสดงผล ส่วนของหน้าจอส่วนท้าย	ระบบจะสามารถแสดงผลของหน้าจอ ส่วนท้ายตามที่กำหนดได้	ถูกต้อง
TC16	ในส่วนหน้าจอของปุ่มจะสามารถ แสดงผลตรงตามตำแหน่ง	ระบบจะสามารถแสดงผลหน้าจอของ ปุ่มได้ถูกต้องตามที่กำหนด	ถูกต้อง
TC17	ในส่วนการแสดงผลของจำนวนจะ สามารถระบุถึงจำนวนของคลาส เมที่ อด เงื่อนไข และวนซ้ำ	ระบบจะสามารถแสดงผลหน้าจอของ ผลลัพธ์ตามที่กำหนดได้	ถูกต้อง
TC18	ในส่วนการแสดงผลของโค้ดจาวา จะสามารถแสดงโค้ดจาวาทั้งหมด	ระบบจะสามารถแสดงโค้ดจาวา ทั้งหมดที่ได้อ้างอิงจากแผนภาพลำดับ และแผนภาพคลาสได้	ถูกต้อง

ตารางที่ 11 ทดสอบเครื่องมือในการสกัดข้อมูล เพื่อให้ได้โครงสร้างของคลาส

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC19	นำเข้าข้อมูลทดสอบที่ได้จากการนำ ออกของโปรแกรมประยุกต์ในรูปแบบ ไฟล์เจสันของแผนภาพคลาสสำหรับ โครงสร้างคลาส	ระบบจะสามารถอ่านรูปแบบไฟล์เจสัน ของแผนภาพคลาสในส่วน of คลาสได้	ถูกต้อง
TC20	การหาความสัมพันธ์ระหว่างคลาสใน รูปแบบไฟล์เจสันของแผนภาพคลาส	ระบบจะสามารถหาความสัมพันธ์ ระหว่างคลาส แต่ละคลาสได้	ถูกต้อง
TC21	การจัดรูปแบบโครงสร้างคลาสของโค้ด จาวา	ระบบจะสามารถจัดรูปแบบของ โครงสร้างของคลาสได้	ถูกต้อง
TC22	การเพิ่มในส่วน of ชนิดข้อมูลสำหรับ พารามิเตอร์ของคลาส	ระบบจะสามารถเพิ่มในส่วน of ชนิด ข้อมูลของคลาสได้	ถูกต้อง
TC23	การจัดตำแหน่งรูปแบบ of โครงสร้าง ของโค้ดจาวา	ระบบจะสามารถจัดตำแหน่งรูปแบบ ของโค้ดจาวาได้	ถูกต้อง

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC24	การแสดงผลแอทริบิวต์ของแต่ละคลาส	ระบบจะสามารถแสดงผลแอทริบิวต์ของแต่ละคลาสได้	ถูกต้อง

ตารางที่ 12 ทดสอบเครื่องมือในการสกัดข้อมูล เพื่อให้ได้ความสัมพันธ์ของเมท็อด

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC25	นำเข้าข้อมูลทดสอบที่ได้จากการนำออกของโปรแกรมประยุกต์ในรูปแบบไฟล์เจสันของแผนภาพคลาสสำหรับโครงสร้างของเมท็อด	ระบบจะสามารถอ่านรูปแบบไฟล์เจสันของแผนภาพคลาสในส่วนของเมท็อดได้	ถูกต้อง
TC26	นำเข้าข้อมูลทดสอบที่ได้จากการนำออกของโปรแกรมประยุกต์ในรูปแบบไฟล์เจสันของแผนภาพลำดับสำหรับการรับส่งข้อมูลระหว่างคลาส	ระบบจะสามารถอ่านรูปแบบไฟล์เจสันของแผนภาพลำดับในส่วนของการรับส่งข้อมูลได้	ถูกต้อง
TC27	การหาความสัมพันธ์ระหว่างเมท็อดของแผนภาพลำดับและแผนภาพคลาสในรูปแบบไฟล์เจสัน	ระบบจะสามารถหาความสัมพันธ์ระหว่างเมท็อดในคลาสแต่ละคลาสได้	ถูกต้อง
TC28	การจัดรูปแบบโครงสร้างเมท็อดของโค้ดจาวา	ระบบจะสามารถจัดรูปแบบของโครงสร้างของเมท็อดได้	ถูกต้อง
TC29	การเพิ่มในส่วนของชนิดข้อมูลสำหรับพารามิเตอร์ของเมท็อด	ระบบจะสามารถเพิ่มในส่วนของชนิดข้อมูลของเมท็อดได้	ถูกต้อง
TC30	การนำส่วนประกอบย่อยของโค้ดของคลาสและเมท็อดมารวมกันในแต่ละคลาส	ระบบจะสามารถรวบรวมส่วนประกอบย่อยโค้ดของคลาสและเมท็อดได้รวมถึงความสัมพันธ์	ถูกต้อง
TC31	การจัดตำแหน่งในรูปแบบของโครงสร้างโค้ดจาวาของคลาสและเมท็อด	ระบบจะสามารถจัดรูปแบบโครงสร้างของคลาสและเมท็อดได้	ถูกต้อง

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC32	การส่งข้อมูลกลับไปยังคลาสที่เรียกใช้	ระบบจะสามารถส่งข้อมูลกลับไปยังคลาสที่เรียกใช้ได้	ถูกต้อง

ตารางที่ 13 ทดสอบเครื่องมือในการสกัดข้อมูล เพื่อให้ได้ความสัมพันธ์เงื่อนไข

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC33	นำเข้าข้อมูลทดสอบที่ได้จากการนำออกของโปรแกรมประยุกต์ในรูปแบบไฟล์เจสันของแผนภาพลำดับสำหรับความสัมพันธ์ของเงื่อนไข	ระบบจะสามารถอ่านรูปแบบไฟล์เจสันของแผนภาพลำดับในส่วนของแพกเมนต์ได้	ถูกต้อง
TC34	การหาความสัมพันธ์ระหว่างออลเทอร์เนทิฟกับออลเทอร์เนทิฟ	ระบบจะสามารถหาความสัมพันธ์ระหว่างออลเทอร์เนทิฟกับออลเทอร์เนทิฟได้	ถูกต้อง
TC35	การหาความสัมพันธ์ระหว่างการวนซ้ำกับออลเทอร์เนทิฟ	ระบบจะสามารถหาความสัมพันธ์ระหว่างการวนซ้ำกับออลเทอร์เนทิฟได้	ถูกต้อง
TC36	การหาความสัมพันธ์ระหว่างสวิตช์เคสกับเมทอด	ระบบจะสามารถหาความสัมพันธ์เงื่อนไขของเมทอดได้	ถูกต้อง
TC37	การจัดรูปแบบโครงสร้างของออลเทอร์เนทิฟ	ระบบจะสามารถจัดรูปแบบโครงสร้างของออลเทอร์เนทิฟได้	ถูกต้อง
TC38	การจัดรูปแบบโครงสร้างของการวนซ้ำ	ระบบจะสามารถจัดรูปแบบโครงสร้างของการวนซ้ำได้	ถูกต้อง
TC39	การจัดรูปแบบโครงสร้างของสวิตช์เคส	ระบบจะสามารถจัดรูปแบบโครงสร้างของสวิตช์เคสได้	ถูกต้อง
TC40	การนำส่วนประกอบย่อยของการวนซ้ำและออลเทอร์เนทิฟมาสร้างความสัมพันธ์	ระบบจะสามารถนำส่วนประกอบย่อย	ถูกต้อง

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC41	การจัดตำแหน่งในรูปแบบโครงสร้างของออลเทอร์เนทิฟ	ระบบจะสามารถจัดตำแหน่งในรูปแบบโครงสร้างของออลเทอร์เนทิฟได้	ถูกต้อง
TC42	การจัดตำแหน่งในรูปแบบโครงสร้างของวงซ้ำ	ระบบจะสามารถจัดตำแหน่งในรูปแบบโครงสร้างของวงซ้ำได้	ถูกต้อง
TC43	การจัดตำแหน่งในรูปแบบโครงสร้างของสวิตช์เคส	ระบบจะสามารถจัดตำแหน่งในรูปแบบโครงสร้างของสวิตช์เคสได้	ถูกต้อง
TC44	การเพิ่มในส่วนของชนิดข้อมูลสำหรับพารามิเตอร์ของออลเทอร์เนทิฟ	ระบบจะสามารถเพิ่มในส่วนของชนิดข้อมูลของออลเทอร์เนทิฟได้	ถูกต้อง
TC45	การเพิ่มในส่วนของชนิดข้อมูลสำหรับพารามิเตอร์ของวงซ้ำ	ระบบจะสามารถเพิ่มในส่วนของชนิดข้อมูลของวงซ้ำได้	ถูกต้อง
TC46	การเพิ่มในส่วนของชนิดข้อมูลสำหรับพารามิเตอร์ของสวิตช์เคส	ระบบจะสามารถเพิ่มในส่วนของชนิดข้อมูลของสวิตช์เคสได้	ถูกต้อง
TC47	การจัดตำแหน่งในรูปแบบโครงสร้างของออลเทอร์เนทิฟ	ระบบจะสามารถจัดตำแหน่งในรูปแบบโครงสร้างของออลเทอร์เนทิฟได้	ถูกต้อง

ตารางที่ 14 ทดสอบเครื่องมือในการรวบรวมส่วนประกอบย่อยทั้งหมด เพื่อให้ได้โค้ดจาวา

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC48	นำส่วนประกอบย่อยของคลาส เมทอด และเงื่อนไขมารวมกัน โดยอ้างอิงความสัมพันธ์	ระบบจะสามารถนำส่วนประกอบของคลาส เมทอดและเงื่อนไขมารวมกัน โดยอ้างอิงความสัมพันธ์ได้	ถูกต้อง
TC49	การรวบรวมของส่วนประกอบย่อยทั้งหมดจะต้องได้ผลลัพธ์เป็นโครงสร้างของโค้ดจาวา	ระบบจะสามารถรวบรวมส่วนประกอบย่อยทั้งหมด เพื่อจะแสดงผลลัพธ์เป็นโครงสร้างของโค้ดจาวาได้	ถูกต้อง

รหัส	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลลัพธ์
TC50	การนำออกผลลัพธ์สุดท้ายต้องอยู่ในรูปแบบของไฟล์จาวา	ระบบจะสามารถนำออกผลลัพธ์ให้อยู่ในรูปแบบของไฟล์จาวาเท่านั้น	ถูกต้อง
TC51	การจัดตำแหน่งของส่วนประกอบย่อยทั้งหมดให้เป็นโครงสร้างของโค้ดจาวา	ระบบจะสามารถจัดตำแหน่งของส่วนประกอบย่อยทั้งหมดให้เป็นโครงสร้างของโค้ดจาวาที่ตรงตามข้อกำหนดและครบถ้วนได้	ถูกต้อง

### 5.3. สรุปผลการทดสอบ

ผู้วิจัยสามารถสรุปผลการทดสอบตามวัตถุประสงค์ของการทดสอบได้ดังต่อไปนี้

- 1) การนำเข้าของข้อมูลตั้งแต่ต้นในลักษณะของรูปแบบไฟล์เอกซ์เอ็มไอที่แสดงถึงการออกแบบของแผนภาพลำดับและแผนภาพคลาสที่ได้ออกแบบไว้ จะสามารถนำเอามาสกัดเพื่อให้ได้ข้อมูลที่สำคัญสำหรับการสร้างโค้ดจาวาได้อย่างถูกต้องและครบถ้วน
- 2) ความถูกต้องของการสร้างโค้ดจาวา เพื่อนำมาใช้สำหรับการสร้างโค้ดในรูปแบบไฟล์จาวา ได้ถูกต้องและครบถ้วน
- 3) ส่วนประกอบย่อยทั้งหมดที่มีความสัมพันธ์กันที่สร้างขึ้นจากแผนภาพลำดับและแผนภาพคลาสมีความสัมพันธ์ที่ถูกต้องและครบถ้วน
- 4) การแสดงผลของผลลัพธ์ตามความต้องการที่ได้ออกแบบไว้ รวมถึงการกำหนดแอทริบิวต์ที่ใช้สำหรับแสดงผลได้อย่างถูกต้องและครบถ้วน
- 5) การแสดงผลของผลลัพธ์ของโค้ดจาวาจะแสดงได้ตรงตามรูปแบบและตำแหน่งที่กำหนดไว้ ได้ถูกต้องแล้วครบถ้วน



## บทที่ 6

### สรุปผลการวิจัย

ในบทนี้จะกล่าวถึงสรุปผลการวิจัย ข้อจำกัดของงานวิจัย งานวิจัยในอนาคตและผลงานตีพิมพ์จากวิทยานิพนธ์ โดยแต่ละส่วนที่กล่าวมานั้นมีรายละเอียดดังต่อไปนี้

#### 6.1. สรุปผลงานวิจัย

การสร้างโค้ดได้ถูกออกแบบขึ้น เพื่อเป็นช่องทางการพัฒนาจากการรับสิ่งที่ต้องการมาพัฒนาเป็นซอฟต์แวร์ โดยทั่วไปของการพัฒนาซอฟต์แวร์จะมีการใช้ภาษาพัฒนาที่หลากหลายภาษา ดังเช่น ภาษาซี ภาษาซีพลัสพลัส ภาษาวิซิวเบสิก ซึ่งจะขึ้นอยู่กับสิ่งที่ต้องการพัฒนาว่าเหมาะสมกับภาษาแบบไหน เนื่องจากการพัฒนาซอฟต์แวร์ของแต่ละภาษาเป็นไปได้อย่างล่าช้า รวมถึงเพิ่มภาระในขั้นตอนของการออกแบบแผนภาพยูเอ็มแอล งานวิจัยนี้จึงได้นำเสนอแนวทางในการสร้างโค้ดจาวาจากการออกแบบแผนภาพลำดับและแผนภาพคลาส ประกอบไปด้วย คลาส เมทอด ออลเทอร์เนทิฟ ทางเลือก และวนซ้ำ โดยใช้หลักการการในการแปลงรูปแบบไฟล์เอกซ์เอ็มไอให้เป็นโค้ดจาวา ซึ่งจะนำรูปแบบไฟล์เอกซ์เอ็มไอที่ได้จากการแปลงแผนภาพลำดับและแผนภาพคลาสของเครื่องมือวิซิวพาราตามยูเอ็มแอลมาช่วยในการสร้างโค้ดจาวา และยังใช้รูปแบบไฟล์เจสันมาช่วยในการสร้างความสัมพันธ์ที่ได้ออกแบบไว้ในแผนภาพลำดับและแผนภาพคลาส นอกจากนี้งานวิจัยได้พัฒนาเครื่องมือสนับสนุนในการสร้างโค้ดจาวาจากแผนภาพลำดับและแผนภาพคลาส โดยเครื่องมือดังกล่าวได้นำซอฟต์แวร์อินเทลลิเจไอเดียมาช่วยในการสร้างหน้าจอ รวมถึงส่วนของการพัฒนาเครื่องมือของการแปลงโค้ดจาวาจากแผนภาพลำดับและแผนภาพคลาส ซึ่งแนวคิดและเครื่องมือที่สนับสนุนจะสามารถช่วยในการสร้างโค้ดจาวา เพื่อลดขั้นตอนของการพัฒนาซอฟต์แวร์ที่มีคุณลักษณะที่แตกต่าง

การทดสอบและวิเคราะห์ผลที่ได้จากบทที่ 5 หัวข้อ 5.2. การทดสอบระบบ หน้าที่ 59 ได้แสดงให้เห็นถึงผลลัพธ์ของระบบในการสร้างโค้ดจาวาจากแผนภาพลำดับและแผนภาพคลาส รวมถึงส่วนประกอบของการใช้งานของหน้าจอที่ได้ออกแบบได้ครบถ้วนและถูกต้องตามที่กล่าวไว้ในแนวคิดและวิธีการดำเนินการวิจัย ซึ่งนักพัฒนาซอฟต์แวร์หรือนักวิเคราะห์ระบบที่ต้องการสร้างโค้ดจาวาก็สามารถนำแนวคิดและเครื่องมือสนับสนุนนี้มาใช้ในการสร้างโค้ดจาวา เพื่อลดเวลาและขั้นตอนของการพัฒนาซอฟต์แวร์

## 6.2. ข้อจำกัดของงานวิจัย

- 1) ใช้เครื่องมือวิชชัวพาราดามยูเอ็มแอลเวอร์ชัน 13.1 ไว้สำหรับการสร้างและเพิ่มคุณสมบัติในแผนภาพลำดับและแผนภาพคลาสเท่านั้น
- 2) การนำเข้าข้อมูลของระบบจะเกิดจากการนำออกแผนภาพลำดับและแผนภาพคลาสจากเครื่องมือวิชชัวพาราดามยูเอ็มแอล ที่เป็นรูปแบบไฟล์เอกซ์เอ็มไอและจะต้องแปลงรูปแบบไฟล์เอกซ์เอ็มไอให้เป็นรูปแบบไฟล์เจสันก่อนที่จะนำเข้าในระบบเท่านั้น
- 3) ผู้ใช้ต้องกำหนดโครงสร้างของโค้ดจาวา
- 4) ผู้ใช้ต้องกำหนดโครงสร้างของโค้ดจาวาตามที่ได้ออกแบบไว้ในแผนภาพลำดับและแผนภาพคลาส
- 5) ผู้ใช้กำหนดความสัมพันธ์ของแผนภาพลำดับได้ 2 ลำดับขั้นเท่านั้น

## 6.3. งานวิจัยในอนาคต

- 1) นำแนวทางในการสร้างโค้ดจาวามาประยุกต์ใช้กับอุตสาหกรรมซอฟต์แวร์ เพื่อให้ใช้เวลาของการพัฒนาระบบลดลงและส่งมอบผลงานคุณภาพให้กับลูกค้า
- 2) นำแนวทางในการสร้างโค้ดจาวามาสร้างเป็นระบบอัตโนมัติของการตรวจสอบองค์ประกอบของแผนภาพลำดับ เพื่อให้การพัฒนามีความรวดเร็วและมีคุณภาพที่ดี

## รายการอ้างอิง

1. *UML*. Available from: [http://www.sparxsystems.com/downloads/resources/booklets/uml\\_dictionary.pdf](http://www.sparxsystems.com/downloads/resources/booklets/uml_dictionary.pdf).
2. *Sequence Diagram*. Available from: <http://www.uml-diagrams.org/sequence-diagrams.html>.
3. *Class Diagram*. Available from: <http://www.uml-diagrams.org/class-diagrams-overview.html>.
4. *Visual Paradigm*. Available from: <http://www.visual-paradigm.com/>.
5. Vadakkumcheril, T., M. Mythily, and M.L. Valarmathi, *A Simple Implementation of UML Sequence Diagram to Java Code Generation through XMI Representation*. International Journal of Emerging Technology and Advanced Engineering, December 2013. Volume 3(Issue 12).
6. Usman, M. and A. Nadeem, *Automatic generation of Java code from UML diagrams using UJECTOR*. Int. J. Softw. Eng. Appl., 2009. 3(2): p. 21-38.
7. Object Management Group, *Unified Modeling Language™ (OMG UML)*. Superstructure Version 2.4.1, OMG, Inc., Aug. 2011.



ภาคผนวก

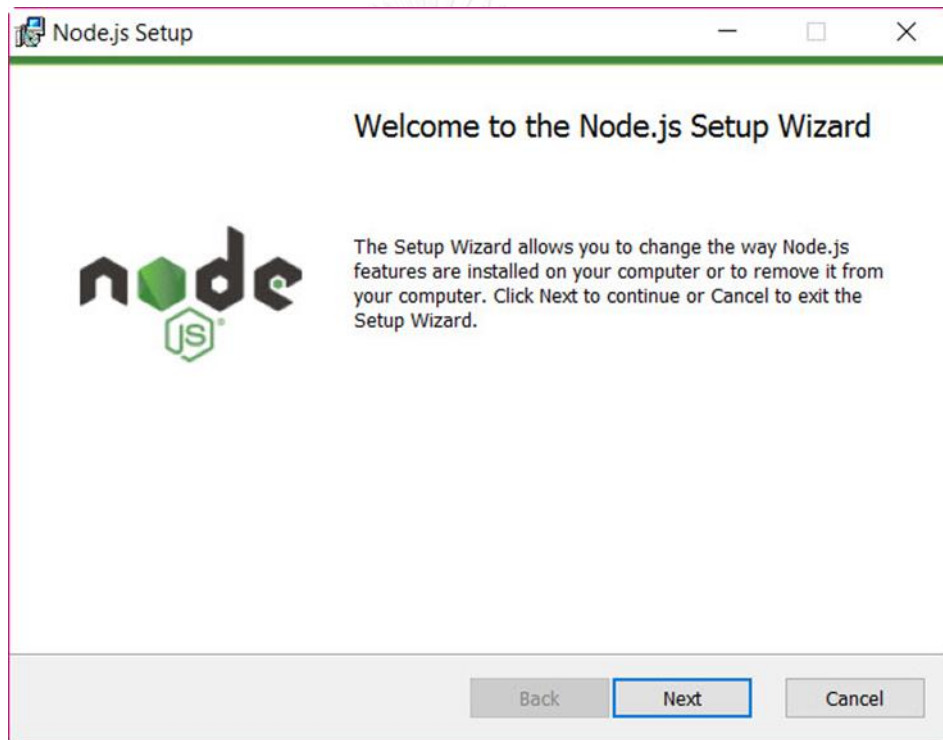
จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

## ภาคผนวก ก.

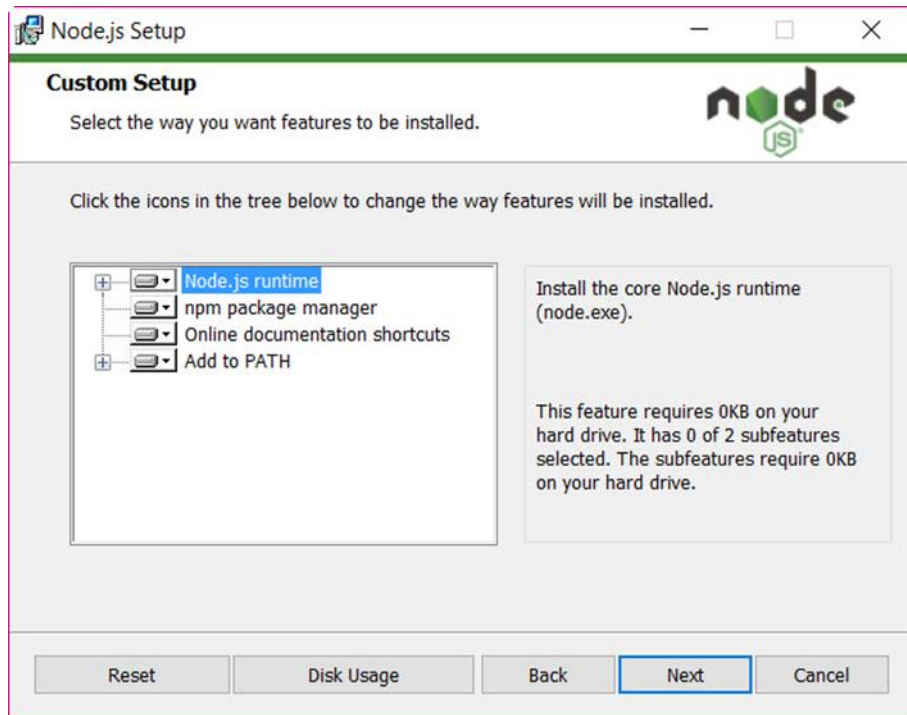
### การติดตั้งซอฟต์แวร์โนดเจเอส (NodeJs)

การนำเข้าข้อมูลของงานวิจัยนี้ ต้องแปลงไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจสัน เพื่อให้ระบบทำงานได้อย่างมีประสิทธิภาพมากขึ้น โดยการติดตั้งซอฟต์แวร์โนดเจเอส มีขั้นตอนดังต่อไปนี้

- 1) ดาวน์โหลดตัวติดตั้งซอฟต์แวร์โนดเจเอสจากลิงค์ <https://nodejs.org/en/>
- 2) เมื่อคลิกติดตั้งจากไฟล์ที่ดาวน์โหลด แล้วกำหนดค่าตามที่ซอฟต์แวร์นั้นได้กำหนดค่าเริ่มต้นจากนั้นให้กดติดตั้ง ดังรูปที่ 78 และ 79 ตามลำดับ

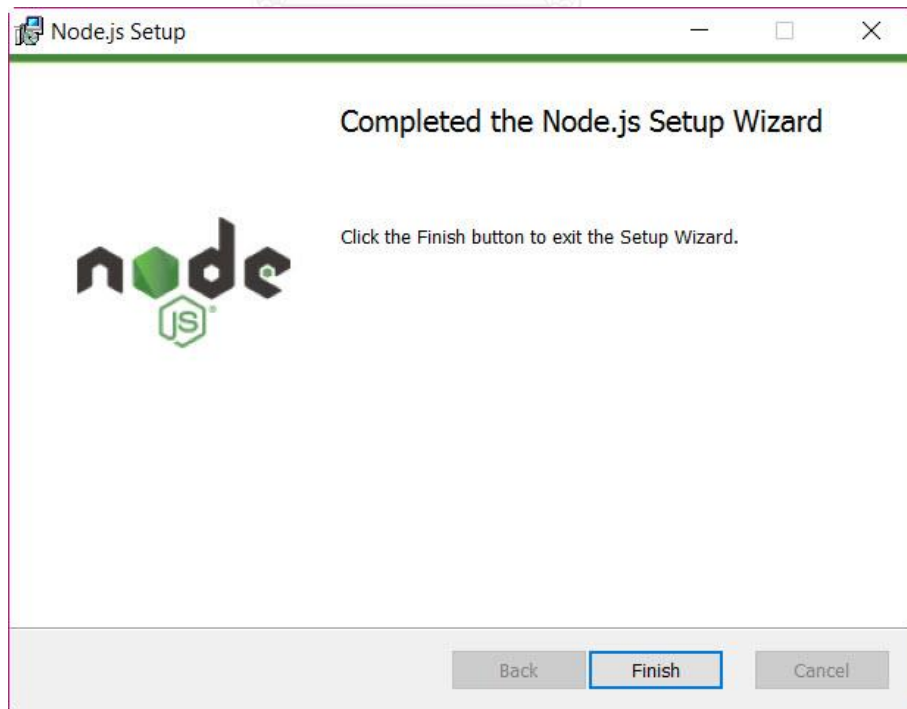


รูปที่ 78 เข้าสู่การติดตั้งของโนดเจเอส



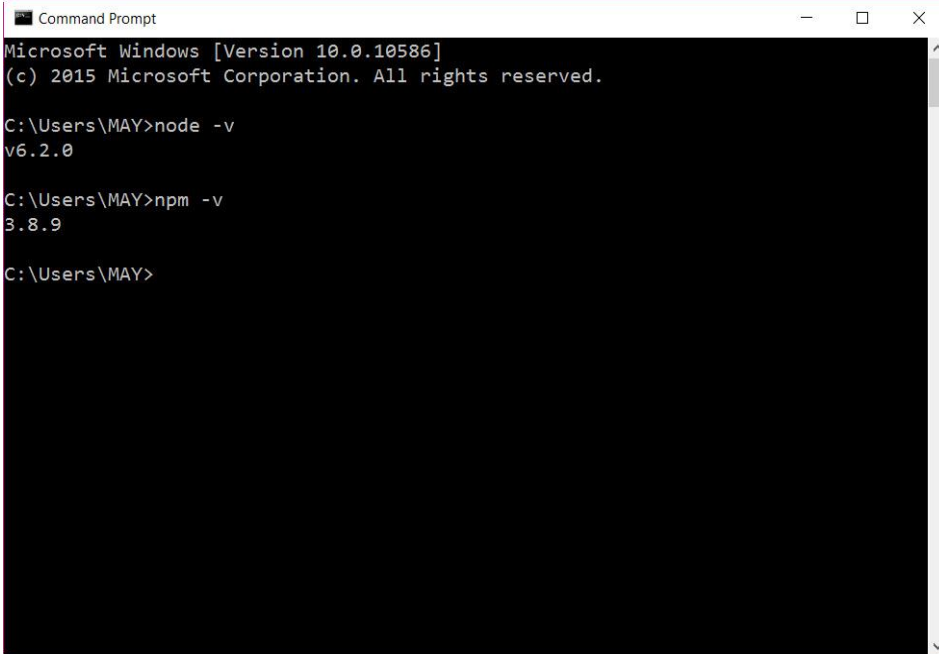
รูปที่ 79 การกำหนดค่าเริ่มต้นและการติดตั้งซอฟต์แวร์โนดเจเอส

- 3) เมื่อติดตั้งซอฟต์แวร์เสร็จสมบูรณ์ จะต้องเริ่มระบบปฏิบัติการของเครื่องคอมพิวเตอร์ใหม่  
ดังรูปที่ 80



รูปที่ 80 ติดตั้งโนดเจเอสเสร็จสมบูรณ์

4) ทดสอบการติดตั้งสมบูรณ์โดยการเปิดตัวพร้อมคำสั่งงานและพิมพ์คำสั่ง ดังรูปที่ 81



```
Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\MAY>node -v
v6.2.0

C:\Users\MAY>npm -v
3.8.9

C:\Users\MAY>
```

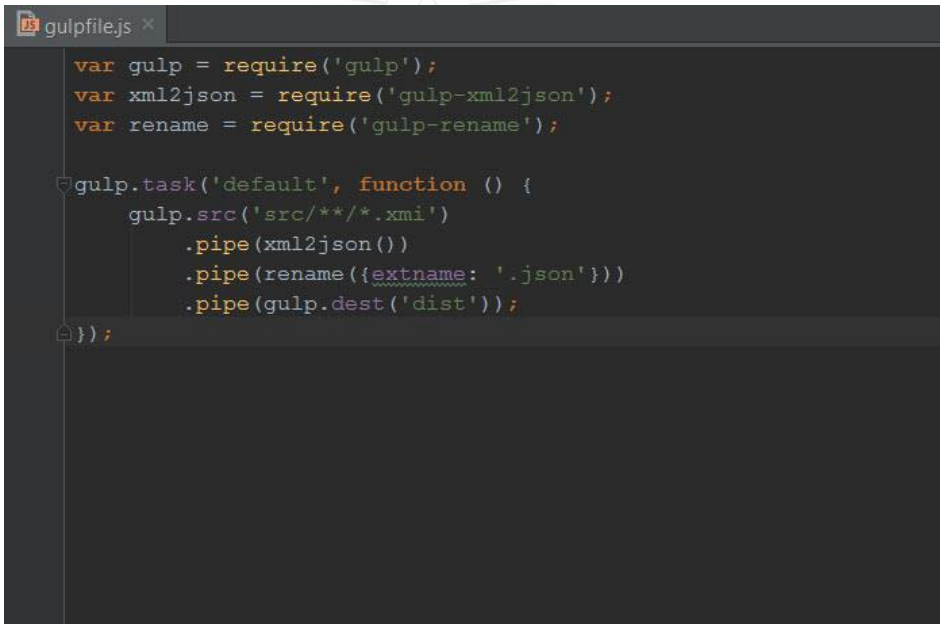
รูปที่ 81 การตรวจสอบการติดตั้งซอฟต์แวร์ผ่านตัวพร้อมคำสั่งงาน

เมื่อรันคำสั่งดังรูปที่ 81 จะแสดงเวอร์ชันของซอฟต์แวร์ที่ติดตั้ง และแสดงถึงการติดตั้งซอฟต์แวร์เสร็จสมบูรณ์

## ภาคผนวก ข. การติดตั้งส่วนเสริมกัลป์ (GulpJs)

จากที่ติดตั้งซอฟต์แวร์โนดเจสเสร็จสมบูรณ์แล้ว จะต้องติดตั้งส่วนเสริมเพื่อช่วยในการ  
สั่งงานผ่านตัวพร้อมคำสั่งงาน เพื่อแปลงรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจสัน นอกจากนี้ส่วน  
เสริมกัลป์ยังสามารถจำลองการแสดงผลที่สร้างขึ้นผ่านตัวพร้อมคำสั่งงาน โดยจะแสดงลำดับการ  
ติดตั้ง เพื่อให้ใช้งานส่วนเสริมกัลป์ได้ดังต่อไปนี้

- 1) สร้างไฟล์ที่ชื่อว่า “gulpfile.js” ซึ่งไว้สำหรับกำหนดการทำงาน ดังรูปที่ 82



```
var gulp = require('gulp');
var xml2json = require('gulp-xml2json');
var rename = require('gulp-rename');

gulp.task('default', function () {
  gulp.src('src/**/*.xml')
    .pipe(xml2json())
    .pipe(rename({extname: '.json'}))
    .pipe(gulp.dest('dist'));
});
```

รูปที่ 82 รูปแบบการทำงาน

- 2) สร้างรูปแบบไฟล์เจสันสำหรับกำหนดการตั้งค่าสำหรับฟังก์ชันที่จะใช้ โดยต้องระบุชื่อไฟล์  
ชื่อว่า “package.json” ซึ่งในงานวิจัยจะกำหนดการตั้งค่าสำหรับฟังก์ชันไว้ดังรูปที่ 83



```
package.json x
{
  "name": "gulp-example",
  "version": "1.0.0",
  "devDependencies": {
    "express": "3.0.3",
    "gulp": "^3.9.1",
    "gulp-xml2json": "^0.1.5",
    "redis": "~0.10.0",
    "request": "~2.40.0",
    "ws.js": "0.0.18",
    "xml2js": "^0.4.3",
    "xml2json": "~0.3.2"
  }
}
```

รูปที่ 83 กำหนดการตั้งค่าฟังก์ชันของส่วนเสริมที่ใช้ในงานวิจัย

หลังจากการติดตั้งส่วนเสริมของกัลป์เจเอสเสร็จสมบูรณ์ ก็จะสามารถใช้ฟังก์ชันในการแปลงรูปแบบไฟล์เอกซ์เอ็มไอเป็นรูปแบบไฟล์เจเอส เพื่อใช้ในงานวิจัยต่อไป

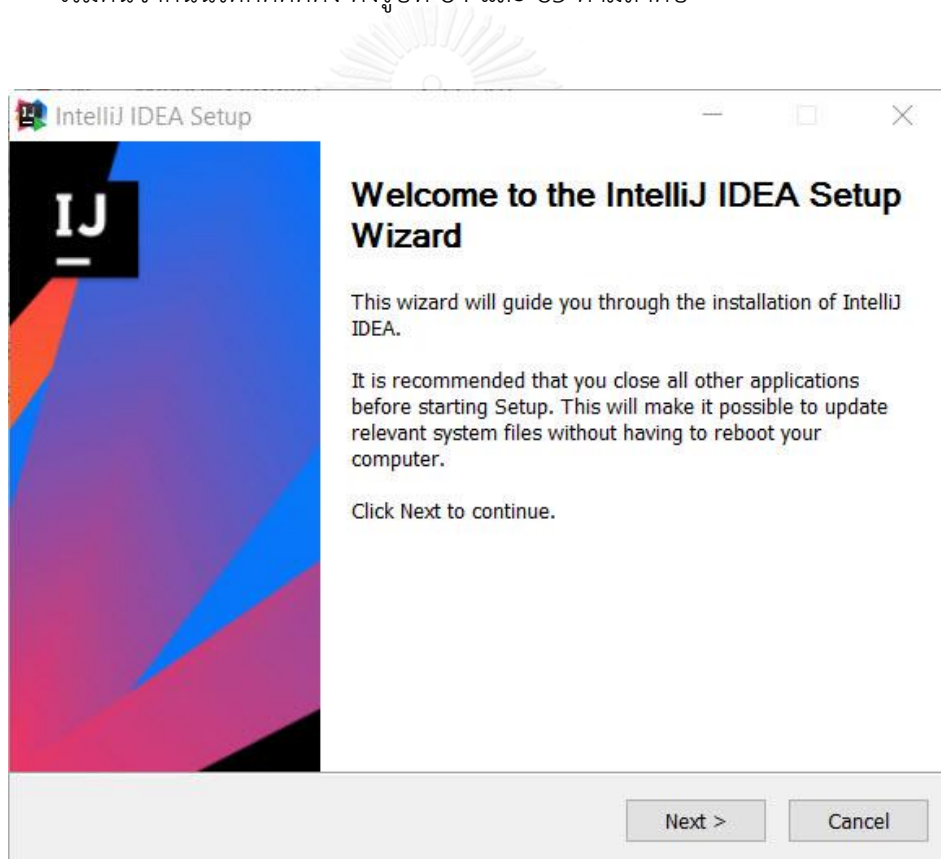
ภาคผนวก ค.  
การติดตั้งอินเทลลิเจไอเดีย (intellij idea)

การสร้างโครงสร้างของโค้ดจาวาของงานวิจัยนี้ จะต้องแปลงรูปแบบไฟล์เจสันเป็นโค้ดจาวา เพื่อให้ได้ง่ายในการพัฒนาระบบ โดยในการติดตั้งซอฟต์แวร์อินเทลลิเจไอเดีย มีขั้นตอนดังต่อไปนี้

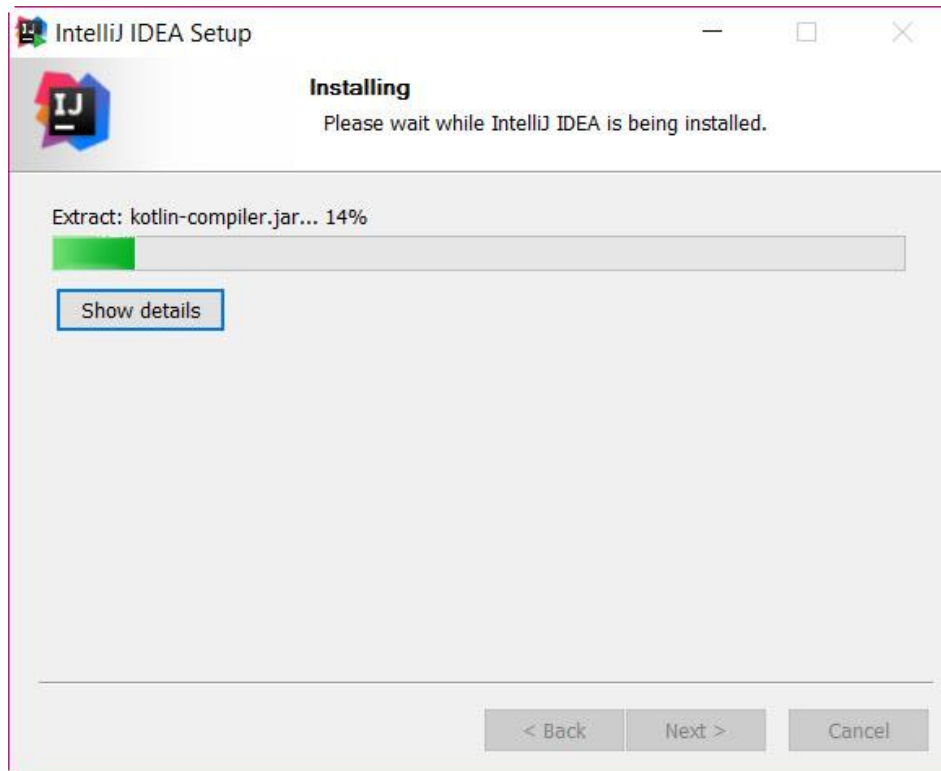
- 1) ดาวน์โหลดตัวติดตั้งซอฟต์แวร์จากลิงค์

<https://www.jetbrains.com/idea/download/#section=windows>

- 2) เมื่อคลิกติดตั้งจากไฟล์ที่ดาวน์โหลด แล้วกำหนดค่าตามที่ซอฟต์แวร์นั้นได้กำหนดค่า เริ่มต้นจากนั้นให้กดติดตั้ง ดังรูปที่ 84 และ 85 ตามลำดับ



รูปที่ 84 เข้าสู่การติดตั้งของอินเทลลิเจไอเดีย



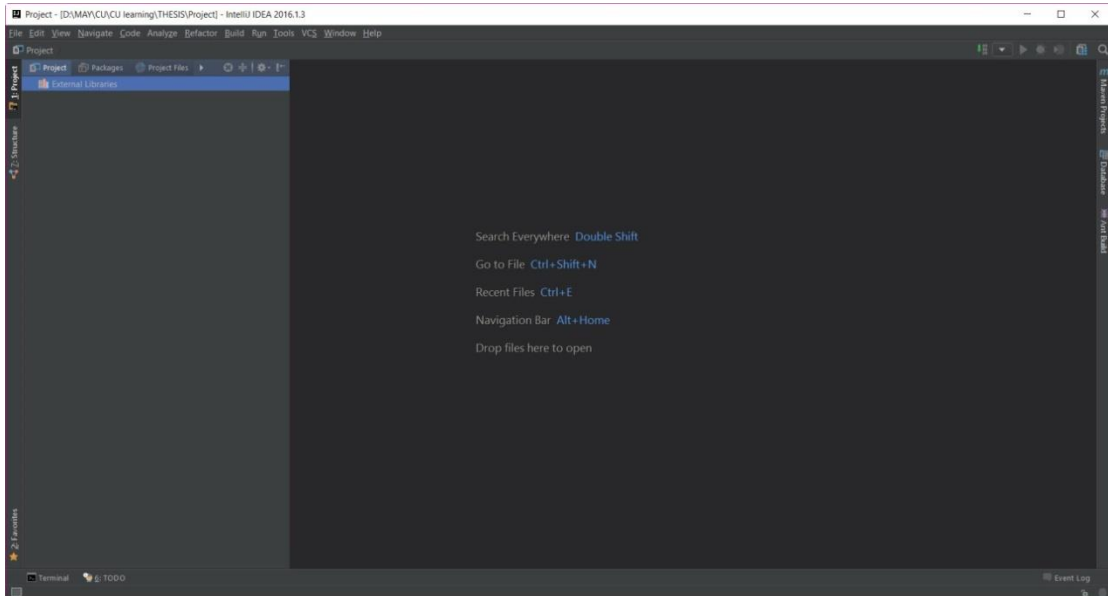
รูปที่ 85 การติดตั้งซอฟต์แวร์อินเทลลิเจไอเดีย

3) เมื่อติดตั้งซอฟต์แวร์เสร็จสมบูรณ์ ดังรูปที่ 86



รูปที่ 86 การติดตั้งซอฟต์แวร์อินเทลลิเจไอเดียเสร็จสมบูรณ์

4) เมื่อทำการติดตั้งเสร็จเรียบร้อยแล้วจะแสดงซอฟต์แวร์ขึ้น ดังรูปที่ 87



รูปที่ 87 หน้าจอการทำงานซอฟต์แวร์ของอินเทลลิเจไอเดีย

หลังจากติดตั้งอินเทลลิเจไอเดียเสร็จสมบูรณ์ ก็จะสามารถพัฒนาระบบ โดยการสร้างคลาส เมทอดและความสัมพันธ์ไปเป็นโค้ดจาวาจากซอฟต์แวร์นี้

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

## ประวัติผู้เขียนวิทยานิพนธ์

นางสาว ปรียาณูช ขลุ่ยศรีตระกูล เกิดเมื่อวันจันทร์ที่ 25 ธันวาคม พ.ศ. 2532 ที่จังหวัดราชบุรี สำเร็จการศึกษาปริญญาตรีหลักสูตรวิทยาศาสตร์บัณฑิต (วท.บ.) สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2554 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2557

