

การพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองของต้นแบบแอนดรอยด์แอปพลิเคชันจากแผนภาพนำทาง  
หน้าต่าง



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

MODEL DRIVEN DEVELOPMENT OF ANDROID APPLICATION PROTOTYPE FROM WINDOWS  
NAVIGATION DIAGRAM

Mr. Thanaseth Channonthawat



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	การพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองของต้นแบบแอนดรอยด์แอปพลิเคชันจากแผนภาพนำทางหน้าต่าง
โดย	นายธนเสกฐ์ ชาญนนทวัฒน์
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร. ญาใจ ลี้มปิยะกรณ์

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์  
(รองศาสตราจารย์ ดร. สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. สุกรี สินธุภิญโญ)  
.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก  
(รองศาสตราจารย์ ดร. ญาใจ ลี้มปิยะกรณ์)  
.....กรรมการภายนอกมหาวิทยาลัย  
(อาจารย์ ดร. ภาสกร อภิรักษ์วรพินิต)

ธนเสฏฐ์ ชาญนนทวัฒน์ : การพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองของต้นแบบแอนดรอยด์แอปพลิเคชันจากแผนภาพนำทางหน้าต่าง (MODEL DRIVEN DEVELOPMENT OF ANDROID APPLICATION PROTOTYPE FROM WINDOWS NAVIGATION DIAGRAM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. ญาใจ ลีมีปิยะภรณ์, 95 หน้า.

ในช่วงหลายปีที่ผ่านมา การเติบโตของตลาดสมาร์ทโฟนได้นำไปสู่การพัฒนาแอปพลิเคชันมือถือที่เพิ่มมากขึ้น การพัฒนาแอปพลิเคชันมือถือด้วยแนวทางแบบรวดเร็วจะตอบสนองต่อการเติบโตของตลาดได้ งานวิจัยนี้ได้นำเสนอแนวทางการพัฒนาเชิงขับเคลื่อนด้วยแบบจำลอง สำหรับการสร้างรหัสต้นทางจียูไอเบื้องต้นของแอนดรอยด์แอปพลิเคชันจากแผนภาพนำทางหน้าต่าง แผนภาพนำเข้าจะถูกแปลงเป็นเอกซ์เอ็มแอล เพื่อใช้เป็นเมตาดาตาสำหรับการแปลงแบบจำลอง ผลลัพธ์สุดท้ายที่ได้คือ ไฟล์เอกซ์เอ็มแอลและไฟล์จาวาสำหรับการจัดหน้าของแต่ละส่วนต่อประสานผู้ใช้ โดยแต่ละไฟล์เอกซ์เอ็มแอลจะประกอบด้วยข้อมูลเกี่ยวกับยูไออิลิเมนต์ และไฟล์จาวาจะบรรจุส่วนการดำเนินการ ระเบียบวิธีการที่นำเสนอจะสนับสนุนเพิ่มการนำกลับมาใช้ใหม่อันเนื่องมาจากการใช้แม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้ นอกจากนี้ ความเป็นอัตโนมัติจากการใช้วิธีการแปลงแบบจำลองจะทำให้มั่นใจในความถูกต้องของส่วนต่อประสานที่ถูกสร้างจากการออกแบบด้วยแผนภาพนำทางหน้าต่าง

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

ภาควิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์

ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2558

# # 5770995021 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: MODEL DRIVEN DEVELOPMENT / MOBILE APPLICATION / PROTOTYPE /  
METADATA / WINDOWS NAVIGATION DIAGRAM

THANASETH CHANNONTHAWAT: MODEL DRIVEN DEVELOPMENT OF ANDROID  
APPLICATION PROTOTYPE FROM WINDOWS NAVIGATION DIAGRAM. ADVISOR:  
ASSOC. PROF. YACHAI LIMPIYAKORN, Ph.D., 95 pp.

In recent years, the growth of smartphone market has led to the increasing development of mobile application. The rapid approach of mobile application development would respond to the market growth. This research presents a model driven development approach for creating preliminary GUI source code of Android application from windows navigation diagrams. The input diagram is converted to XML used as the metadata for model transformation. The final results of XML and Java files will be obtained for each UI layout where each XML file contains UI element information, and the Java file contains the actions. The proposed methodology would promote the reusability due to the use of UI layout template. The automation with model transformation would also ensure the integrity of interfaces generated from the design with windows navigation diagrams.

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

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

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

Academic Year: 2015

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์จากรองศาสตราจารย์ ดร.ญาใจ ลิ้มปิยะกรณ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ได้สละเวลาให้ความรู้ คำปรึกษา ตรวจสอบและแก้ไขข้อผิดพลาดต่าง ๆ ตลอดจนการกำกับดูแลและคอยติดตามความก้าวหน้า ทำให้การวิจัยนี้สำเร็จไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ และ อาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณบิดา มารดา และญาติพี่น้องที่ให้การสนับสนุนและเป็นกำลังใจที่ดีให้เสมอมาและสนับสนุนด้านทุนทรัพย์ในการศึกษารวมไปถึงทุกท่านที่มีส่วนช่วยเหลือในการทำวิทยานิพนธ์ครั้งนี้ ซึ่งมีได้กล่าวนามในที่นี้

ท้ายที่สุด ผู้วิจัยขอขอบพระคุณเพื่อน ๆ ทุกคน ที่คอยติดตามและให้กำลังใจ รวมถึงท่านอื่น ๆ ที่มีได้กล่าวลงนามไว้ ณ ที่นี้ที่มีส่วนทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดีผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์บ้างไม่มากก็น้อยสำหรับผู้ที่สนใจจะศึกษารายละเอียดต่อไป

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	1
สารบัญรูป.....	1
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	1
1.3 ขอบเขตของงานวิจัย.....	1
1.4 ขั้นตอนและวิธีการดำเนินงานวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 เจสัน (JavaScript-Object Notation, JSON).....	4
2.2 เอกซ์เอ็มแอล (eXtensible Markup Language, XML).....	7
2.3 ดอม (The Document Object Model, DOM).....	7
2.4 เอ็มดีเอ (Model-Driven Architecture, MDA).....	8
2.4.1 CIM (Computation Independent Model).....	9
2.4.2 PIM (Platform Independent Model).....	9
2.4.3 PSM (Platform Specific Model).....	9
2.4.4 Code (โค้ด).....	10
2.5 ยูเอ็มแอล (Unified Modeling Language, UML).....	10

2.6) งานวิจัยที่เกี่ยวข้อง .....	11
2.6.1) GUI Code Generation for Android Applications Using a MDA Approach [10].....	11
2.6.2) Generating Android Graphical User Interfaces using an MDA Approach [11].....	12
บทที่ 3 แนวคิดวิธีวิจัย.....	14
3.1) Create Windows Navigation Diagram.....	30
3.2) Export Windows Navigation Diagram.....	39
3.3) Generate Conceptual Model.....	41
3.4) Generate Navigation Model.....	42
3.5) Generate Presentation Model.....	44
3.6.1) การสร้างแบบเค้าร่าง.....	46
3.6.2) การเตรียมสร้างส่วนควบคุม.....	48
3.7) Generate Source Code.....	49
บทที่ 4 การพัฒนาเครื่องมือ.....	52
4.1) ความต้องการเชิงฟังก์ชัน.....	52
4.2) การวิเคราะห์ความต้องการและแผนภาพฟังก์ชันงานของระบบ.....	52
4.3) สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือสนับสนุน.....	55
4.3.1) ฮาร์ดแวร์.....	55
4.3.2) ซอฟต์แวร์.....	55
4.3.3) การติดตั้งซอฟต์แวร์.....	56
4.4) ขั้นตอนการทำงานของเครื่องมือ.....	57
บทที่ 5 การทดสอบและการวิเคราะห์ผล.....	71
5.1) วัตถุประสงค์ของการทดสอบ.....	71



5.2) การทดสอบระบบ .....	71
5.2.1) การทดสอบการนำข้อมูลเข้าเครื่องมือสนับสนุนด้วยรูปแบบไฟล์เอ็กซ์เอ็มไอ .....	71
5.2.2) การทดสอบเครื่องมือการแปลงข้อมูลจากรูปแบบเอ็กซ์เอ็มแอลไปรูปแบบเจสัน .....	71
5.2.3) การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองทางความคิด .....	71
5.2.4) การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำทาง .....	72
5.2.5) การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำเสนอ .....	72
5.2.6) การทดสอบเครื่องมือการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติ .....	72
5.3) สรุปผลการทดลอง .....	86
บทที่ 6 สรุปผลการวิจัย .....	87
6.1) สรุปผลการวิจัย .....	87
6.2) ข้อจำกัดของงานวิจัย .....	87
6.3) งานวิจัยในอนาคต .....	88
รายการอ้างอิง .....	89
ภาคผนวก การสร้างแผนภาพนำทางหน้าต่าง .....	90
ประวัติผู้เขียนวิทยานิพนธ์ .....	95

## สารบัญตาราง

ตารางที่ 1 อธิบายรูปแบบโครงสร้างของตัวอย่างเจสันแบบวัตถุ.....	5
ตารางที่ 2 อธิบายรูปแบบโครงสร้างของตัวอย่างเจสันแบบแอ็รเรย์.....	6
ตารางที่ 3 ตารางองค์ประกอบคุณลักษณะพื้นฐานของ DOM.....	7
ตารางที่ 4 คำอธิบายสัญลักษณ์บนเครื่องมือ Android Studio.....	16
ตารางที่ 5 คำอธิบายสัญลักษณ์ในแถบเครื่องมือของสตาร์ยูเอ็มแอลที่ใช้ในงานวิจัย.....	28
ตารางที่ 6 คำอธิบายองค์ประกอบคุณสมบัติของ Instance Object.....	31
ตารางที่ 7 คำอธิบายองค์ประกอบคุณสมบัติของ Link.....	32
ตารางที่ 8 องค์ประกอบคุณสมบัติแผนภาพนำทางหน้าต่าง ชื่อ Login .....	34
ตารางที่ 9 องค์ประกอบคุณสมบัติ แผนภาพนำทางหน้าต่างชื่อ Register .....	35
ตารางที่ 10 องค์ประกอบคุณสมบัติแผนภาพนำทางหน้าต่าง ชื่อ MainActivity.....	35
ตารางที่ 11 องค์ประกอบคุณสมบัติของแผนภาพนำทางหน้าต่าง ชื่อ ClickButton ฟังก์ชัน ซ้ายมือ .....	36
ตารางที่ 12 องค์ประกอบคุณสมบัติของแผนภาพนำทางหน้าต่างชื่อ ClickButton ฟังก์ชันขวามือ..	37
ตารางที่ 13 คำอธิบายค่าเจสันที่สำคัญ.....	40
ตารางที่ 14 การทดสอบการนำข้อมูลเข้าเครื่องมือสนับสนุนด้วยไฟล์รูปแบบเอ็กซ์เอ็มไอ.....	72
ตารางที่ 15 การทดสอบเครื่องมือการแปลงข้อมูลจากรูปแบบเอ็กซ์เอ็มแอลไปรูปแบบเจสัน.....	73
ตารางที่ 16 การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองทางความคิด.....	73
ตารางที่ 17 การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำทาง.....	73
ตารางที่ 18 การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำเสนอ.....	74
ตารางที่ 19 การทดสอบเครื่องมือการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติ.....	81

## สารบัญรูป

รูปที่ 1 ตัวอย่างโครงสร้างค่าของเจสัน [2].....	5
รูปที่ 2 โครงสร้างประเภทต่าง ๆ ของ UML Diagrams [9].....	11
รูปที่ 3 ภาพรวมการทำงาน [10].....	12
รูปที่ 4 ภาพรวมวิธีการเพื่อสร้างส่วนต่อประสาน [11] .....	13
รูปที่ 5 ขั้นตอนการแปลงโมเดล [11] .....	13
รูปที่ 6 ภาพรวมแนวคิดงานวิจัย.....	14
รูปที่ 7 รายละเอียดเครื่องมือสนับสนุน Shorthand .....	29
รูปที่ 8 ตัวอย่างแผนภาพนำทางหน้าต่าง.....	31
รูปที่ 9 ภาพรวมโปรแกรมสตาร์ยูเอ็มแอล .....	31
รูปที่ 10 ตัวอย่างโครงสร้างเอกซ์เอ็มแอลของแผนภาพนำทางหน้าต่าง .....	39
รูปที่ 11 รูปแบบโครงสร้างเจสันที่แปลงมาจากไฟล์เอกซ์เอ็มไอ.....	40
รูปที่ 12 รหัสเทียมของขั้นตอนวิธี Export Windows Navigation Diagram .....	41
รูปที่ 13 รหัสเทียมของขั้นตอนวิธี Generate Conceptual Model .....	42
รูปที่ 14 รหัสเทียมของขั้นตอนวิธี Generate Navigation Model.....	44
รูปที่ 15 รหัสเทียมของขั้นตอนวิธี Generate Presentation Model.....	46
รูปที่ 16 ตัวอย่างแม่แบบการจัดหน้าส่วนต่อประสาน.....	47
รูปที่ 17 ตัวอย่างไฟล์เอกซ์เอ็มแอลรับผิดชอบในส่วนต่อประสานผู้ใช้ .....	47
รูปที่ 18 ตัวอย่างไฟล์จาวารับผิดชอบในส่วนพัฒนาโปรแกรม .....	48
รูปที่ 19 รหัสเทียมของขั้นตอนวิธี Parse.....	49
รูปที่ 20 รหัสเทียมของขั้นตอนวิธี Generate Source Code .....	50
รูปที่ 21 แผนภาพยูสเคสของระบบพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองต้นแบบแอนดรอยด์แอป พลิกเคชันจากแผนภาพนำทางหน้าต่าง .....	53

รูปที่ 22	ลักษณะการออกแบบแผนภาพนำทางหน้าต่างด้วยเครื่องมือสตาร์ยูเอ็มแอล .....	57
รูปที่ 23	ระบบเครื่องมือสนับสนุน .....	58
รูปที่ 24	รูปแบบการทำงานแปลงรูปแบบข้อมูลเอกซ์เอ็มแอลไปเจสัน .....	59
รูปที่ 25	เนื้อหาข้อมูลเอกซ์เอ็มแอลจากไฟล์เอกซ์เอ็มไอ .....	59
รูปที่ 26	เนื้อหาในรูปแบบข้อมูลเจสัน .....	60
รูปที่ 27	รูปแบบการสกัดข้อมูลของแบบจำลองทางความคิด .....	60
รูปที่ 28	ภาพรวมการทำงานของแบบจำลองนำทาง .....	61
รูปที่ 29	รูปแบบการหารายละเอียดที่มีส่วนเกี่ยวข้องกับการเชื่อมโยงระหว่างแผนภาพนำทาง หน้าต่าง .....	62
รูปที่ 30	รูปแบบการหารายละเอียดที่ไม่มีส่วนเกี่ยวข้องกับการเชื่อมโยงระหว่างแผนภาพนำทาง หน้าต่าง .....	62
รูปที่ 31	รูปแบบการสกัดข้อมูลแผนภาพนำทางหน้าต่างที่มีส่วนเกี่ยวข้องกับการเชื่อมโยงระหว่าง แผนภาพนำทางหน้าต่าง .....	63
รูปที่ 32	รูปแบบการสกัดข้อมูลเฉพาะโดยเน้นที่แผนภาพนำทางหน้าต่าง .....	63
รูปที่ 33	รูปแบบการสกัดข้อมูลเฉพาะโดยเน้นที่แบบจำลองนำทางหน้าต่างปลายทาง .....	64
รูปที่ 34	ภาพรวมรูปแบบการสกัดรายละเอียดที่ไม่มีส่วนเกี่ยวข้องกับการนำทาง .....	64
รูปที่ 35	การเลือกไฟล์โปรเจคแอนดรอยด์แอปพลิเคชัน .....	65
รูปที่ 36	ภาพรวมเครื่องมือสนับสนุนหลังจากกดปุ่มเจเนอเรท .....	65
รูปที่ 37	รูปแบบวิธีการค้นหาคลาสในโปรเจคแอนดรอยด์แอปพลิเคชันที่เกี่ยวข้อง .....	66
รูปที่ 38	รูปแบบค้นหาส่วนต่อประสานผู้ใช้ที่ถูกแสดงผลและไม่แสดงผลออกทางแอนดรอยด์ แอปพลิเคชัน .....	66
รูปที่ 39	รูปแบบค้นหาส่วนต่อประสานผู้ใช้ที่แสดงผลในการจำลองแอนดรอยด์แอปพลิเคชัน .....	67
รูปที่ 40	รูปแบบค้นหาส่วนต่อประสานผู้ใช้ที่ไม่แสดงผลในการจำลองแอนดรอยด์แอปพลิเคชัน ..	67
รูปที่ 41	รูปแบบการทวนสอบค่าข้อมูล .....	67
รูปที่ 42	เนื้อหาภายในไฟล์เอกซ์เอ็มแอลส่วนควบคุม .....	68

รูปที่ 43 รูปแบบเนื้อหาคลาสตัวช่วยส่วนควบคุมการนำเสนอ .....	69
รูปที่ 44 รูปแบบการนำคลาสตัวช่วยส่วนควบคุมการนำเสนอไปประยุกต์ใช้.....	69
รูปที่ 45 การเลือกไฟล์ส่วนควบคุมเพื่อนำมาวิเคราะห์.....	69
รูปที่ 46 รูปแบบการอ่านและเขียนไฟล์เอกซ์เอ็มแอล.....	70
รูปที่ 47 ภาพรวมเครื่องมือสตาร์ยูเอ็มแอลที่เลือกแม่แบบประเภท Class Diagram .....	90
รูปที่ 48 ทดสอบการวาดแผนภาพนำทางหน้าต่าง .....	91
รูปที่ 49 ทดสอบการเปลี่ยนชื่อแผนภาพ <<Button>> btnLogin .....	91
รูปที่ 50 ภาพการซ้อนแผนภาพกับแผนภาพ .....	92
รูปที่ 51 ภาพหลังจากทำการซ้อนแผนภาพ.....	92
รูปที่ 52 ภาพเครื่องมือ Shorthand.....	93
รูปที่ 53 ภาพผลลัพธ์หลังจากใช้เครื่องมือ Shorthand ช่วยการสร้างความสัมพันธ์การเชื่อมโยง ระหว่างแผนภาพนำทางหน้าต่าง .....	93
รูปที่ 54 ภาพรวมการกำหนดรายละเอียดแผนภาพนำทางหน้าต่าง.....	94
รูปที่ 55 รายละเอียดภาพรวมเมื่อทำการกำหนดรายละเอียดเสร็จเรียบร้อยแล้ว .....	94

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบัน มีระบบปฏิบัติการบนโทรศัพท์มือถือที่หลากหลาย เช่น Windows phone ของไมโครซอฟต์, iOS ของแอปเปิ้ล, แอนดรอยด์ (Android) ของกูเกิ้ล และอื่น ๆ เนื่องจากโทรศัพท์มือถือมีส่วนสำคัญกับการใช้ชีวิตประจำวันมากขึ้น จึงเกิดกลยุทธ์แข่งขันกันเพื่อที่จะผลิตซอฟต์แวร์แอปพลิเคชันสำหรับตอบโจทย์ผู้ใช้งานรวมถึงการออกแบบส่วนต่อประสานให้ตอบโจทย์ผู้ใช้งาน

ส่วนต่อประสานเป็นระบบส่วนงานที่สามารถติดต่อกับสิ่งที่อยู่ภายนอกระบบ และมักเป็นสิ่งที่แรกที่ผู้ออกแบบคำนึงถึงเพื่อสร้างความประทับใจให้แก่ผู้ใช้ ในขั้นตอนการออกแบบมักจะใช้เวลาและเกิดความล่าช้าบ่อยครั้ง เนื่องด้วยความล่าช้านี้อาจจะส่งผลถึงนักพัฒนาโปรแกรมด้วย นักออกแบบจึงได้สร้างต้นแบบส่วนต่อประสานผู้ใช้ (UI Prototype) ในรูปแบบแผนภาพจาก user scenario และ user experience เพื่อใช้สื่อสารความต้องการระบบด้านส่วนต่อประสานระหว่างนักออกแบบและนักพัฒนา

งานวิจัยนี้ได้นำเสนอวิธีการสร้างต้นแบบส่วนต่อประสานบนแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติจากแผนภาพนำทางหน้าต่าง (Windows Navigation Diagram— WND) แนวทางที่นำเสนอจะช่วยอำนวยความสะดวกในการออกแบบส่วนต่อประสานเพื่อให้ความเข้าใจกับลูกค้าได้เร็วขึ้น รวมถึงช่วยลดข้อบกพร่องที่เกิดจากความไม่สอดคล้องกัน (Inconsistency) และสามารถนำสิ่งที่ออกแบบกลับมาใช้ใหม่อีกรอบได้ (Reusability)

### 1.2 วัตถุประสงค์ของงานวิจัย

นำเสนอวิธีการและพัฒนาเครื่องมือสำหรับสร้างต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติด้วยหลักการสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลองจากแผนภาพนำทางหน้าต่าง เพื่อปรับปรุงกระบวนการซอฟต์แวร์

### 1.3 ขอบเขตของงานวิจัย

- 1) ใช้เครื่องมือสตาร์ยูเอ็มแอล (StarUML) [1] เวอร์ชัน 2.6.0 สำหรับสร้างแผนภาพนำทางหน้าต่าง
- 2) ความสัมพันธ์ในแผนภาพนำทางหน้าต่าง มีความลึกและกว้างไม่เกิน 2 ลำดับ

#### 1.4 ขั้นตอนและวิธีการดำเนินงานวิจัย

- 1) ศึกษาและทำความเข้าใจทฤษฎีและงานวิจัยที่เกี่ยวข้อง
- 2) ศึกษาการใช้เครื่องมือที่ใช้สำหรับการพัฒนาโปรแกรม
- 3) วิเคราะห์และกำหนดระเบียบวิธีวิจัย
- 4) ออกแบบ ตั้งสมมติฐาน ที่เกี่ยวข้องกับงานวิจัย
- 5) พัฒนาระบบ
- 6) ทดสอบและประเมินผลงานวิจัย
- 7) สรุปผลงานวิจัย
- 8) ตีพิมพ์ผลงานทางวิชาการ
- 9) จัดทำวิทยานิพนธ์

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีการและเครื่องมือสนับสนุนแนวทางการพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองต้นแบบ แอนดรอยด์แอปพลิเคชันแบบอัตโนมัติจากแผนภาพนำทางหน้าต่าง ซึ่งรองรับการนำกลับมาใช้ซ้ำได้

#### 1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

เนื้อหาภายในวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 6 บท โดยเริ่มต้นจากบทที่ 1 บทนำจะกล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของงานวิจัย ขอบเขตของงานวิจัย ขั้นตอนและวิธีการดำเนินงานวิจัย และประโยชน์ที่คาดว่าจะได้รับ บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้องที่นำมาใช้งานวิจัยนี้ บทที่ 3 การออกแบบระบบ ขั้นตอนและวิธีการในการออกแบบระบบ บทที่ 4 การพัฒนาระบบ เครื่องมือที่ใช้ในการวิจัยและวิธีการพัฒนาระบบ บทที่ 5 การทดสอบระบบ ทดสอบการทำงานของระบบ และบทที่ 6 สรุปผลการวิจัย ข้อจำกัด และแนวทางสำหรับการทำวิจัยต่อไปในอนาคต

#### 1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “Model Driven Development of Android Application Prototypes from Windows Navigation Diagrams” โดย ธนเสฏฐ์ ชาญนนทวัฒน์ และ ญาใจ ลิ้มปิยะกรณ์ ในหนังสือรวมบทความการ

ประชุมวิชาการนานาชาติ 2016 INTERNATIONAL CONFERENCE ON SOFTWARE NETWORKING (ICSN) ณ เกาะเชจู สาธารณรัฐเกาหลี วันที่ 25 พฤษภาคม 2559 หน้า 41 - 44





## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1) เจสัน (JavaScript-Object Notation, JSON)

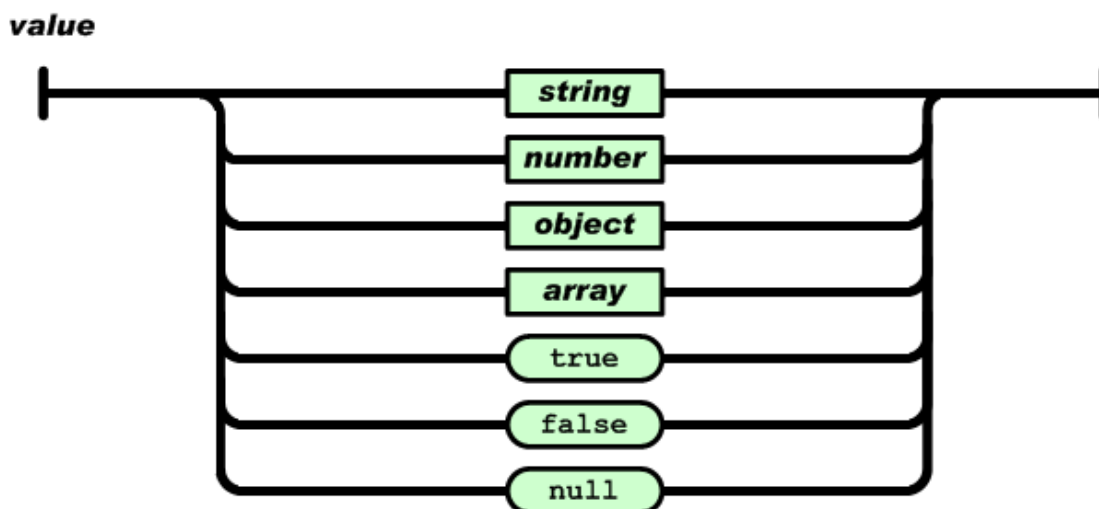
เจสัน (JSON) [2] มีรูปแบบโครงสร้างของข้อมูลที่มีการแลกเปลี่ยนข้อมูลแบบเบา (Lightweight) ที่ง่ายต่อการทำความเข้าใจสำหรับมนุษย์ และง่ายต่อการพัฒนา โดยเจสันเป็นการทำงานชุดย่อย (Subset) ของภาษาเขียนโปรแกรมจาวาสคริป (JavaScript Programming Language) ตามมาตรฐาน ECMA-262 ที่ตีพิมพ์ครั้งที่ 3 เดือนธันวาคม ปี ค.ศ. 1999

เจสันมีรูปแบบโครงสร้างข้อความที่เป็นภาษาอิสระที่ใช้เป็นข้อตกลงกับการเขียนโปรแกรมตระกูลภาษา C ได้แก่ C++, C#, Java, JavaScript, Perl, Python และอื่น ๆ อีกมากมาย คุณสมบัติเหล่านี้ทำให้เจสันเป็นภาษาแลกเปลี่ยนข้อมูลที่มีความสมบูรณ์

เจสันสามารถสร้างได้จาก 2 โครงสร้าง คือ

- 1) คอลเลกชันของชื่อ (Name) และค่า (Value) ในภาษาต่างๆ คอลเลกชันเปรียบเสมือนกับวัตถุ (Object) เรคคอร์ด (Record) สตรัค (Struct) พจนานุกรม (Dictionary) แฮชเทเบิล (Hash Table) รายการที่มีคีย์ (Keyed List) หรือแอสซิเอทีฟแอเรย์ (Associative Array) เป็นต้น
- 2) รายการลำดับของค่า ในภาษาส่วนใหญ่จะเหมือนกับแอสซิเอทีฟแอเรย์ (Array) เวกเตอร์ (Vector) รายการ (List) หรือลำดับ (Sequence) เป็นต้น

รูปแบบเจสันเป็นการจัดรูปแบบของแอสซิเอทีฟแอเรย์ (Array) หรือวัตถุ (Object) ซึ่งแอสซิเอทีฟแอเรย์หรือวัตถุอาจจะมีบรรทัดเลข สตริง แอสซิเอทีฟแอเรย์และวัตถุอื่น ๆ ค่าตัวอักษรที่ไม่มีค่า (Null) ค่าจริง (True) และค่าเท็จ (False) ค่าเจสันเหล่านี้สามารถจำลองขึ้น และอธิบายเป็นส่วนย่อย ดังรูปที่ 1



รูปที่ 1 ตัวอย่างโครงสร้างค่าของเจสัน [2]

ในเจสันจะใช้สัญลักษณ์ ([ ]) แทนแอร์เรย์ และจะใช้สัญลักษณ์ปีกกา ( ) แทน แสทและรวมถึงแอร์เรย์ที่เชื่อมโยงกัน

ตัวอย่างการจัดรูปแบบเจสันแบบวัตถุ มีรูปแบบเป็นดังนี้

```
{
  "name": "value",
  "firstname": "Thanaset",
  "lastname": "Channonthawat"
}
```

โดยตารางที่ 1 อธิบายรูปแบบโครงสร้างของตัวอย่างเจสันแบบวัตถุ

ตารางที่ 1 อธิบายรูปแบบโครงสร้างของตัวอย่างเจสันแบบวัตถุ

ชื่อ	คำอธิบาย
Name	คอลเลคชันของวัตถุ
Value	ค่าคอลเลคชันของวัตถุ
Firstname	คอลเลคชันของวัตถุที่มีชื่อว่า firstname
Lastname	คอลเลคชันของวัตถุที่มีชื่อว่า lastname

Thanaseth	ค่าคอลเลชันของ firstname ที่มีค่าเท่ากับ Thanaseth
Channonthawat	ค่าคอลเลชันของ lastname ที่มีค่าเท่ากับ Channonthawat

จากตารางที่ 1 เป็นการอธิบายการยกตัวอย่างรูปแบบโครงสร้างเจสันแบบวัตถุ โดยที่ได้ อธิบายองค์ประกอบภายในอย่างเช่น fisrtname เรียกว่า คอลเลชันวัตถุที่มีชื่อว่า firstname และ ค่าของคอลเลชันวัตถุที่มีชื่อว่า firstname มีค่าเท่ากับ Thanaseth

ตัวอย่างการจัดรูปแบบเจสันแบบแอ็รเรย์ มีรูปแบบเป็นดังนี้

```
{
  "product":
  [
    "iphone", "ipad", "macbook"
  ]
}
```

โดยที่ ตารางที่ 2 อธิบายรูปแบบโครงสร้างของตัวอย่างเจสันแบบแอ็รเรย์

ตารางที่ 2 อธิบายรูปแบบโครงสร้างของตัวอย่างเจสันแบบแอ็รเรย์

ชื่อ	คำอธิบาย
Product	คอลเลชันของแอ็รเรย์ที่มีชื่อว่า product
["iphone", "ipad", "macbook"]	ค่าคอลเลชันของ product ที่มีค่าเท่ากับ ["iphone", "ipad", "macbook"]

จากตารางที่ 2 เป็นการอธิบายการยกตัวอย่างรูปแบบโครงสร้างเจสันแบบแอ็รเรย์ โดยที่ได้ อธิบายองค์ประกอบภายในอย่างเช่น product เรียกว่า คอลเลชันแอ็รเรย์ที่มีชื่อว่า product และ ค่าของคอลเลชันแอ็รเรย์ที่มีชื่อว่า product มีค่าเท่ากับ ["iphone", "ipad", "macbook"]

## 2.2) เอกซ์เอ็มแอล (eXtensible Markup Language, XML)

XML [3] เป็น meta-language ที่ใช้ในการนิยามให้เข้ากับภาษาอื่น ๆ เป็นการอธิบายในระดับของข้อมูลวัตถุประสงค์สำหรับรองรับการใช้งานทั่วไป ถูกพัฒนาขึ้นโดย W3C โดยมีจุดประสงค์เพื่อเป็นสื่อการติดต่อในระบบที่มีความแตกต่างกัน เป็นการจัดรูปแบบที่สามารถอธิบายถึงรายละเอียดของโครงสร้างและรูปแบบของข้อมูล ให้มีการพัฒนาที่มีประสิทธิภาพและมีศักยภาพในส่วนของโครงสร้างข้อมูลจากหลากหลายรูปแบบมานำเสนอบนเครื่องเดสก์ทอปด้วย ข้อดีของเอกซ์เอ็มแอล คือ ทำให้การจัดการข้อมูลหรือเรียกใช้ข้อมูลจากหลากหลายแหล่งข้อมูลเข้าสู่รูปแบบมาตรฐานเดียวกัน

## 2.3) ดอม (The Document Object Model, DOM)

DOM [4] ถูกพัฒนาขึ้นโดยองค์กร W3C โดยใช้หลักการส่วนหนึ่งของโปรแกรมการจัดหน้า ส่วนต่อประสานผู้ใช้ที่สามารถอนุญาตให้เข้าถึงเอกสารเพื่อทำการปรับปรุงโครงสร้าง รูปแบบ และเนื้อหาของเอกสารเอกซ์เอ็มแอล XML Parser [5] เป็นหลักการพื้นฐานที่ช่วยในการพัฒนาโปรแกรมบนไฟล์เอกซ์เอ็มแอล

วิธีการของหลักการ DOM Parser จะทำงานได้เมื่อเราทราบรายละเอียดโครงสร้างของเอกสารไฟล์เอกซ์เอ็มแอลและต้องการวิเคราะห์ข้อมูลเพื่อปรับเปลี่ยนค่าข้อมูล เนื้อหา และโครงสร้างในเอกสารไฟล์เอกซ์เอ็มแอล คุณลักษณะพื้นฐานของ DOM ที่นิยมใช้จะประกอบดังนี้

ตารางที่ 3 ตารางองค์ประกอบคุณลักษณะพื้นฐานของ DOM

คุณลักษณะ	คำอธิบาย
Node	หน่วยข้อมูลของ DOM
Element	วัตถุองค์ประกอบใน DOM
Attr	ค่าที่แสดงให้เห็นถึงองค์ประกอบของแอทริบิวต์
Text	เนื้อหาในส่วน Element หรือ Attr
Document	แสดงถึงผลลัพธ์ของเอกสารไฟล์เอกซ์เอ็มแอล

จากตารางที่ 3 เป็นการอธิบายองค์ประกอบคุณลักษณะพื้นฐานเบื้องต้นระหว่างการทำงานของ DOM และ DOM Parser

## 2.4) เอ็มดีเอ (Model-Driven Architecture, MDA)

สถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง (MDA) [6] เป็นโมเดลและวิธีการพัฒนาซอฟต์แวร์ที่ถูกพัฒนาขึ้นและนำเสนอโดย The Object Management Group (OMG) [7] ขั้นตอนกระบวนการพัฒนาซอฟต์แวร์ของสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง ประกอบด้วย การวิเคราะห์ การออกแบบ การเขียนโปรแกรม และการทดสอบ การออกแบบและการพัฒนาด้วยสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลองมักใช้โครงสร้างโมเดลพื้นฐานและมาตรฐานของ UML

โมเดลสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลองสามารถจำแนกเป็น 4 ระดับและจะใช้หลักการขั้นตอนพัฒนาจากบนลงล่าง (Top down development) การเริ่มต้นจะเริ่มจากการสร้างโมเดลแบบจำลองลงไปจนถึงขั้นตอนการเขียนโปรแกรมและการทดสอบ การพัฒนาเริ่มจากการวิเคราะห์ จากนั้นเป็นการทำงานกับโมเดลที่เป็นองค์ประกอบของธุรกิจหรือเรียกว่า โดเมนโมเดล (Domain model) โดยในขั้นตอนของการทำงานจะมีการตรวจสอบอย่างต่อเนื่องเพื่อยืนยันความถูกต้องของโดเมนธุรกิจ (Business domain) หลังจากที่ได้โดเมนโมเดลที่ถูกต้องแล้วจะทำการเปลี่ยนโดเมนโมเดลที่ได้ เป็นโมเดลที่จำเพาะเจาะจงกับเทคโนโลยีแบบใดแบบหนึ่งหรือเรียกว่า เทคโนโลยีโมเดล (Technology model) และเมื่อได้เทคโนโลยีโมเดลแล้ว ต่อมาจะทำการออกแบบและพัฒนาซอฟต์แวร์ในส่วนของเทคโนโลยีโมเดล ในขั้นตอนนี้จะเป็นการจัดการกับฟังก์ชันของการทำงานและการจัดการกับส่วนติดต่อกับผู้ใช้ หลังจากทำการออกแบบและพัฒนาส่วนเทคโนโลยีโมเดลเสร็จแล้ว ขั้นตอนต่อไปจะใช้เครื่องมือเปลี่ยนเทคโนโลยีโมเดลไปเป็นโค้ดซึ่งจะมีเครื่องมือทำการแปลงให้อัตโนมัติ เห็นได้ว่า กระบวนการพัฒนาด้วยสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลองมีความสะดวกและง่ายกว่าการเขียนโปรแกรมเองด้วยมือ รวมทั้งลดระยะเวลาของกระบวนการพัฒนาซอฟต์แวร์และทำให้คุณภาพซอฟต์แวร์ดียิ่งขึ้น

กระบวนการพัฒนาซอฟต์แวร์ด้วย MDA ถูกขับเคลื่อนโดยกิจกรรมของการสร้างแบบจำลองโมเดล เพื่อทำการสร้างซอฟต์แวร์ในรูปแบบเชิงธุรกิจ แนวคิดของสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลองจะมองไปยังโครงสร้างในระดับธุรกิจและนำเอากระบวนการและองค์ประกอบต่าง ๆ ของธุรกิจนั้นออกมาในรูปแบบเชิงนามธรรม (Abstract) จากนั้นนำมาพัฒนาในรูปแบบสัญลักษณ์โมเดลโดยปกติแล้ว ธรรมชาติของแต่ละธุรกิจจะมีกระบวนการและองค์ประกอบคล้าย ๆ กันหรือเหมือนกัน ซึ่งสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลองจะสร้างโมเดลจากองค์ประกอบพื้นฐานเหล่านั้นและทำให้ครอบคลุมถึงกระบวนการและองค์ประกอบต่าง ๆ ที่มีอยู่ในแต่ละธุรกิจ เมื่อต้องการสร้างซอฟต์แวร์ให้ธุรกิจก็นำเอาองค์ประกอบของโมเดลเหล่านั้นมาพัฒนาและปรับปรุงให้สามารถนำมาใช้กับธุรกิจนั้น ๆ ที่จะพัฒนา แล้วทำการสร้างซอฟต์แวร์จากโมเดลเหล่านั้น ทำให้การพัฒนาซอฟต์แวร์

เป็นไปอย่างรวดเร็วและมีความถูกต้องเมื่อเทียบกับการพัฒนาแบบเดิมที่ต้องเก็บความต้องการของลูกค้าแล้วนำมาพัฒนาโมเดลใหม่ทุกครั้ง

วัฏจักรการพัฒนาของสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง ประกอบด้วยแบบจำลอง 4 ระดับ ดังนี้

#### 2.4.1) CIM (Computation Independent Model)

เป็นระดับมุมมองของระบบบนสุดที่ไม่แสดงรายละเอียดของโครงสร้างระบบ กล่าวคือ CIM เป็นโมเดลที่เป็นอิสระของทุกระบบคอมพิวเตอร์ (any computerization) สามารถอธิบายผลิตภัณฑ์ที่ไม่เกี่ยวข้องกันกับทุกระบบคอมพิวเตอร์ ให้ความสำคัญไปที่ความต้องการของระบบ รวมถึงกับสภาพแวดล้อมในการที่จะดำเนินการโดยไม่ต้องไปในรายละเอียดของโครงสร้างรวมถึงการพัฒนา ในด้านวิศวกรรมซอฟต์แวร์ CIM เป็นที่รู้จักกันในรูปแบบโดเมนซึ่งส่วนนี้จะเกี่ยวข้องกับผู้เชี่ยวชาญโดเมน

#### 2.4.2) PIM (Platform Independent Model)

แสดงถึงองค์ประกอบของธุรกิจในลักษณะนามธรรม (Abstract) ที่ยังไม่ยึดติดกับเทคโนโลยีใด ๆ จะไม่ทราบรายละเอียดเฉพาะของคอมพิวเตอร์แพลตฟอร์ม PIM จะสร้างโมเดลจากองค์ประกอบต่าง ๆ ที่มีอยู่ในธุรกิจ โมเดลที่ได้จะรองรับความต้องการต่าง ๆ ของธุรกิจ ซึ่งในขั้นตอนนี้เป็นขั้นตอนที่มีความสำคัญมาก โมเดลที่ได้จากขั้นตอนนี้สามารถนำไปปรับใช้หรือนำไปประยุกต์ใช้ได้กับอีกหลายธุรกิจได้หากออกแบบมาได้ดี เนื่องจากธรรมชาติของแต่ละธุรกิจจะมีบางส่วนที่คล้ายคลึงกันหรือเหมือนกันสามารถนำมาใช้ร่วมกันได้ โมเดลที่ได้จะไม่ขึ้นกับเทคโนโลยีใด ๆ เช่น เทคโนโลยีของฐานข้อมูล (MSSQL, MySQL, Oracle) ภาษาโปรแกรมที่ใช้ (ASP.net, JSP, Servlet) และรวมถึงโปรแกรมประยุกต์ย่อยต่าง ๆ ที่นำมาใช้ในการพัฒนา PIM ประกอบไปด้วยรูปแบบและส่วนต่าง ๆ ของ UML เช่น Use Case, Class Diagram, Sequence Diagram และอื่น ๆ เป็นต้น

#### 2.4.3) PSM (Platform Specific Model)

ในขั้นตอน PSM โดเมนโมเดลจะถูกเปลี่ยนรูปเป็นเทคโนโลยีโมเดล โดยจะเลือกเทคโนโลยีที่ใช้ก่อน เช่น .Net ของบริษัทไมโครซอฟต์ หรือ Java ที่เป็น open source โดย PSM หนึ่ง ๆ จะเป็นต้นแบบที่สามารถเปลี่ยนไปใช้ได้กับเทคโนโลยีอื่น ๆ ได้อีกหลายรูปแบบ ซึ่งนักออกแบบและนักพัฒนาต้องมีความรู้พื้นฐาน เพราะแต่ละเทคโนโลยีจะมีรูปแบบและความสามารถที่แตกต่างกันออกไป เมื่อถึงขั้นตอนนี้ นักออกแบบและนักพัฒนาในขั้นตอนการแปลงนี้จะใช้เครื่องมือในการช่วย สำหรับการเปลี่ยนรูปจาก PIM ไปเป็น PSM ซึ่งจะได้โมเดล

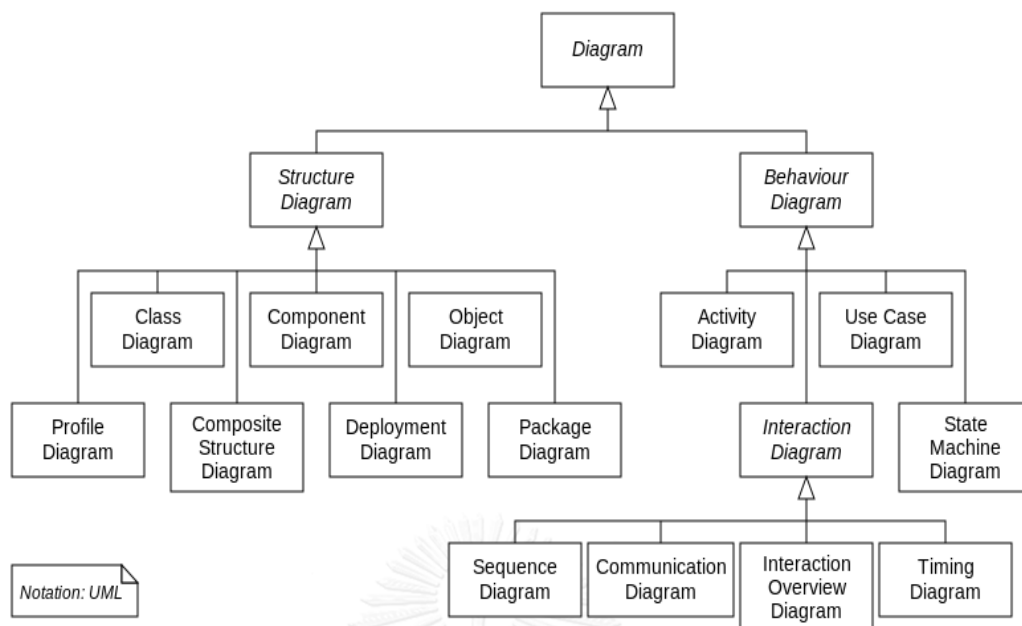
และส่วนประกอบต่าง ๆ จากเทคโนโลยีโมเดล ยังมีแบบโครงสร้างซอฟต์แวร์รูปแบบต่าง ๆ ให้เลือกใช้ ซึ่งการเลือกแบบโครงสร้างซอฟต์แวร์ต้องเลือกให้เหมาะสมกับขนาดของโครงการ ซึ่งโครงการที่มีขนาดไม่ใหญ่ควรเลือกแบบโครงสร้างซอฟต์แวร์ที่เล็กไม่ซับซ้อนทำให้การออกแบบและพัฒนาาง่ายและรวดเร็วมากกว่าการเลือกแบบโครงสร้างซอฟต์แวร์ที่มีขนาดใหญ่และซับซ้อน

#### 2.4.4) Code (โค้ด)

คือ การเขียนโค้ดโปรแกรม เป็นขั้นตอนสุดท้ายของสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง โดย PSM จะถูกแปลงเป็นโค้ดด้วยเครื่องมือ โครงสร้างโค้ดของโปรแกรมที่ได้จะประกอบไปด้วย Class, Properties และ Method or Function การทำงานและไฟล์คุณสมบัติต่างๆ ที่จำเป็นสำหรับการติดตั้งซอฟต์แวร์ นักพัฒนาโปรแกรมสามารถเขียนคำสั่งลงไปตามส่วนต่าง ๆ ของโค้ดที่ได้ ซึ่งจะเขียนเพิ่มเติมเฉพาะส่วนที่เรียกว่า ส่วนอิสระ (Free Block) เท่านั้น โค้ดส่วนที่ไม่สามารถเขียนได้เรียกว่าส่วนป้องกัน (Guarded Block) เนื่องจากเป็นโค้ดจากการสร้างของเครื่องมือ โดยโค้ดส่วนนี้มาจากขั้นตอนการออกแบบ PSM และ PIM เท่านั้น การสร้างโค้ดของเครื่องมือส่วนใหญ่เมื่อสร้างเสร็จแล้วจะสามารถทำงานเบื้องต้นได้เลย ทำให้โปรแกรมเมอร์สามารถทำการทดสอบ พัฒนาและแก้ไขส่วนต่าง ๆ ของโค้ดให้ตรงกับความต้องการได้

#### 2.5) ยูเอ็มแอล (Unified Modeling Language, UML)

UML [8] คือ มาตรฐานของโมเดลที่ใช้หลักการและทักษะการออกแบบของ OOP (Object Oriented Programming) รูปแบบของภาษา UML จะมีสัญกรณ์ (Notation) ที่นำไปใช้ในโมเดลต่าง ๆ UML จะมีข้อกำหนดและกฎระเบียบต่าง ๆ ในการนำไปใช้ โดยกฎระเบียบต่าง ๆ จะมีความหมายต่อการพัฒนาโปรแกรม ดังนั้น การใช้ UML จะต้องทราบความหมายของสัญกรณ์ต่าง ๆ เช่น generalize, association, dependency class และ package สิ่งเหล่านี้มีความจำเป็นอย่างยิ่งต่อการตีความของการออกแบบระบบ ก่อนนำไปพัฒนาเป็นระบบงานจริงปัจจุบัน มีเครื่องมือมากมายที่สามารถแปลง UML Model เป็นโค้ดและภาษาต่าง ๆ ยกตัวอย่าง เช่น ภาษา Java, Power builder และ VB เป็นต้น



รูปที่ 2 โครงสร้างประเภทต่าง ๆ ของ UML Diagrams [9]

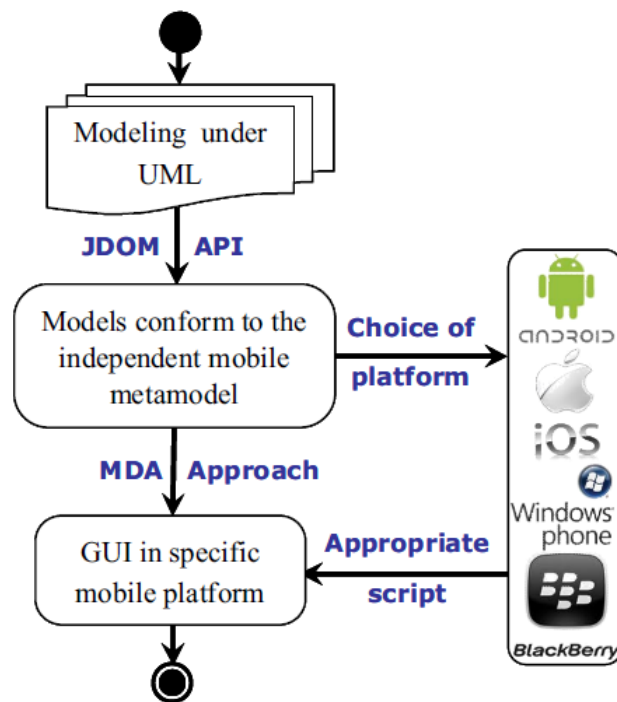
## 2.6) งานวิจัยที่เกี่ยวข้อง

### 2.6.1) GUI Code Generation for Android Applications Using a MDA Approach [10]

งานวิจัยนี้ ได้นำเสนอการพัฒนาโปรแกรมประยุกต์สำหรับแพลตฟอร์มโทรศัพท์มือถือ เป็นสำหรับหลายแพลตฟอร์ม เนื่องจากความหลากหลายของระบบปฏิบัติการของมือถือและความแตกต่างของอุปกรณ์ที่จำเป็นต้องได้รับการสนับสนุน จึงได้ใช้แนวทางสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง เข้ามาแก้ปัญหาจัดการให้เป็นแบบอัตโนมัติ เพื่อสร้างส่วนติดต่อผู้ใช้แบบกราฟิก (Graphical User Interface— GUI) และงานวิจัยได้นำเสนอกระบวนการทำงาน ซึ่งแบ่งการทำงานออกเป็น 3 วิธีการดังต่อไปนี้

- 1) การวิเคราะห์และการสร้างแบบจำลอง GUI ภายใต้อูML
- 2) การเปลี่ยนแผนภาพให้อยู่ในรูปแบบของ XMI โดรนใช้ JDOM API F
- 3) การสร้าง GUI ด้วยวิธีการ MDA



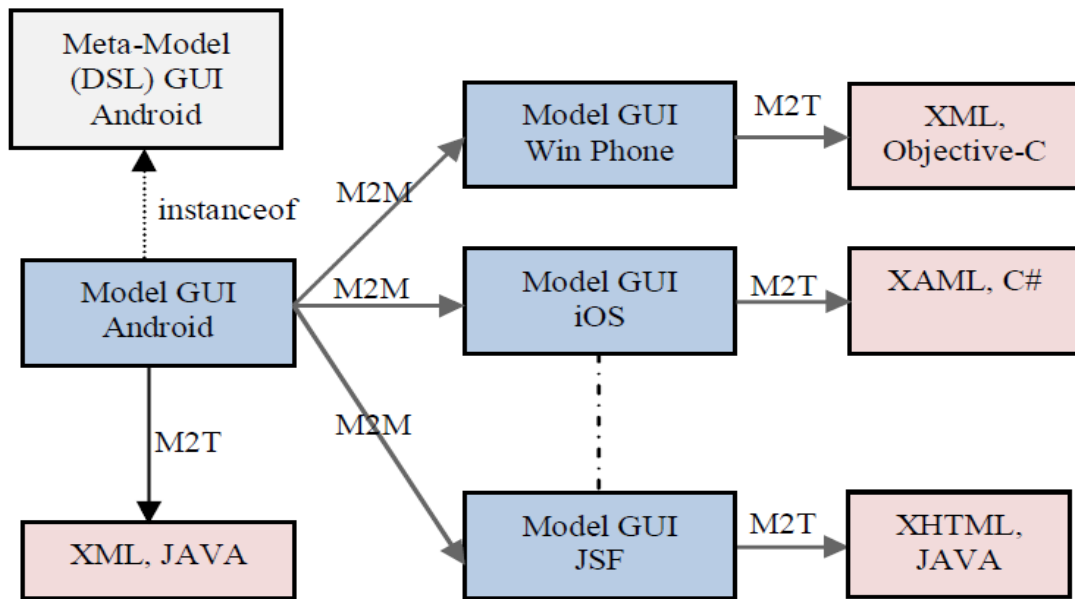


รูปที่ 3 ภาพรวมการทำงาน [10]

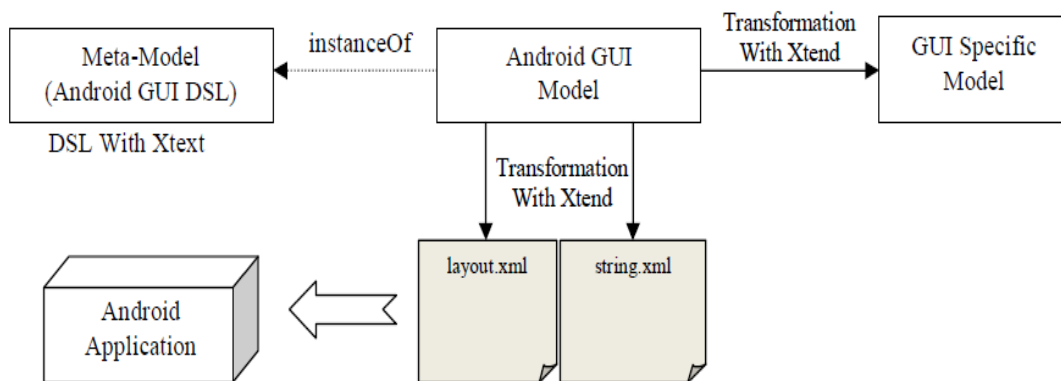
งานวิจัยฉบับนี้สิ่งที่น่าสนใจมาประยุกต์คือ ขั้นตอนที่ 1) จากแผนภาพ Object Diagram ได้เปลี่ยนเป็น WND และใช้ส่วนต่อขยายของเครื่องมือสตาร์ยูเอ็มแอลช่วยในการส่งออกไฟล์เอกซ์เอ็มไอ จากนั้นนำไฟล์เอกซ์เอ็มไอที่ได้ นำเข้าสู่ระบบ

### 2.6.2) Generating Android Graphical User Interfaces using an MDA Approach [11]

งานวิจัยฉบับนี้ได้นำเสนอวิธีการของสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลอง ซึ่งใช้วิธีการนี้เพื่อให้วิธีการแก้ปัญหาและเพิ่มประสิทธิภาพสำหรับการพัฒนาโปรแกรมประยุกต์ข้ามแพลตฟอร์ม และได้นำเสนอ แนวทางในการออกแบบส่วนติดต่อผู้ใช้สำหรับการใช้งานโทรศัพท์มือถือที่นำไปใช้กับแพลตฟอร์มแอนดรอยด์ วิธีการนี้สำหรับแพลตฟอร์มโทรศัพท์มือถือและส่วนประกอบบนเว็บ และการสร้างภาษาสำหรับ Graphical Interface ด้วยเทคโนโลยีภาษาโดเมนเฉพาะ Neutral DSL (Domain Specific Language)



รูปที่ 4 ภาพรวมวิธีการเพื่อสร้างส่วนต่อประสาน [11]

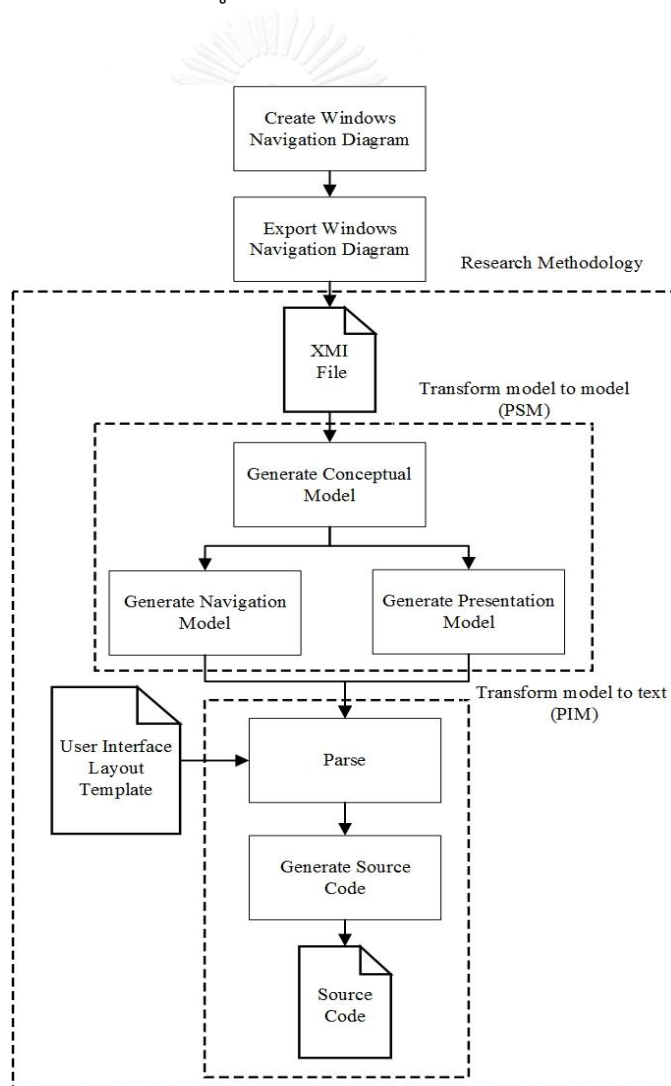


รูปที่ 5 ขั้นตอนการแปลงโมเดล [11]

จากงานวิจัยฉบับนี้สิ่งที่นำมาประยุกต์คือ ผลลัพธ์ของขั้นตอนการทำ Transformation with Xtend ประยุกต์เป็นไฟล์เอกซ์เอ็มแอลที่เก็บเนื้อหารายละเอียดเกี่ยวกับโมเดลการนำทาง

### บทที่ 3 แนวคิดวิธีวิจัย

งานวิจัยนี้ได้นำเสนอแนวทางการพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองอัตโนมัติเพื่อสร้างต้นแบบส่วนต่อประสานผู้ใช้ของแอนดรอยด์แอปพลิเคชันด้วยหลักการสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลองจากแผนภาพนำทางหน้าต่างโดยใช้ XMI [12] (XML Metadata Interchange) ซึ่งอยู่ในรูปแบบของเอกซ์เอ็มแอล ซึ่งเป็นรูปแบบมาตรฐานรองรับเพื่อสนับสนุนการแลกเปลี่ยนข้อมูลของแอปพลิเคชันต่าง ๆ โดยลักษณะของรูปแบบดังกล่าวจะสามารถอธิบายได้ถึงส่วนประกอบคุณสมบัติ และสามารถเชื่อมโยงถึงกันในรูปแบบของเอกซ์เอ็มไอ ภาพรวมแนวคิดงานวิจัยแสดงดังรูปที่ 6

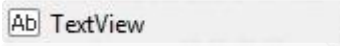
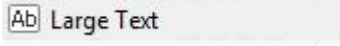
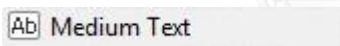
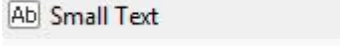

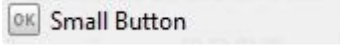


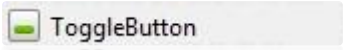
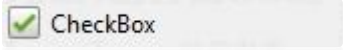

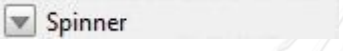
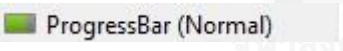
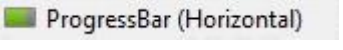
รูปที่ 6 ภาพรวมแนวคิดงานวิจัย

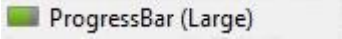


โดยขั้นตอนภาพรวมการสร้างแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้ของแอนดรอยด์แอปพลิเคชันด้วยหลักการสถาปัตยกรรมการขับเคลื่อนด้วยแบบจำลองจากแผนภาพนำทางหน้าต่างอธิบายขั้นตอนเริ่มต้นจาก 1) การสร้างแบบจำลองจากแผนภาพนำทางหน้าต่างด้วยเครื่องมือสตาร์ยูเอ็มแอล (Create Windows Navigation Diagram) เป็นการออกแบบองค์ประกอบแต่ละหน้าจอภายใต้หลักการแบบจำลองแผนภาพนำทางหน้าต่างและการเชื่อมโยงแต่ละหน้าจอเข้าด้วยกัน 2) การส่งออกแบบจำลองแผนภาพนำทางหน้าต่างด้วยเครื่องมือสตาร์ยูเอ็มแอล (Export Windows Navigation Diagram) ที่ถูกติดตั้งด้วยเครื่องมือส่วนต่อขยายที่รองรับการส่งออกไฟล์ชนิดเอกซ์เอ็มไอหลังจากที่ดำเนินการส่งออกแบบจำลองแผนภาพนำทางหน้าต่างสำเร็จ ผลลัพธ์ที่ได้จะเป็นไฟล์ตระกูลรูปแบบของเอกซ์เอ็มไอที่ภายในประกอบด้วยข้อมูลเอกซ์เอ็มแอล จะประกอบไปด้วยรายละเอียดข้อมูลองค์ประกอบต่าง ๆ ที่เกี่ยวกับการออกแบบองค์ประกอบจำลองจากแผนภาพนำทางหน้าต่างบนสตาร์ยูเอ็มแอล ซึ่งองค์ประกอบทั้งหมดจะถูกแสดงอยู่ในรูปแบบโครงสร้างของรูปแบบเอกซ์เอ็มแอล ตัวอย่างขององค์ประกอบของแต่ละหน้าจอจะประกอบด้วย ปุ่ม (Button), การเชื่อมโยง (Link), ข้อความ (Text) เป็นต้น หลังจากนั้นจะนำไฟล์เอกซ์เอ็มไอไปอ่านไฟล์ข้อมูลเพื่อทำการอ่านข้อมูลแล้วทำการเปลี่ยนรูปแบบข้อมูลให้เป็นรูปแบบของเจสัน ซึ่งจะมีรูปแบบของเจสันหลายแบบอย่างเช่น วัตถุเจสันหรือแอ็รเรย์เจสัน เมื่อทำการแปลงข้อมูลเอกซ์เอ็มไอไปรูปแบบเจสันสำเร็จ จะนำเนื้อหาข้อมูลไปสู่ขั้นตอนถัดไปคือ 3) การสร้างโมเดลจำลองทางความคิด (Generate Conceptual Model) ในขั้นตอนการสร้างแบบโมเดลจำลองทางความคิดจะมีกระบวนการทำงานวิเคราะห์ข้อมูลเพื่อการสกัดคัดเลือกข้อมูลเบื้องต้น โดยทำกระบวนการสกัดข้อมูลและเตรียมข้อมูลเฉพาะส่วนที่จำเป็นและสำคัญไว้ และทำการสร้างเป็นแบบโมเดลจำลองทางความคิดเพื่อนำไปใช้ประโยชน์และแนวทางในลำดับขั้นตอนถัดไป 4) การสร้างโมเดลจำลองนำทาง (Generate Navigation Model) เป็นการวิเคราะห์โมเดลจำลองทางความคิดเพื่อค้นหาและสกัดข้อมูลที่เป็นเส้นทางระหว่างหน้าจอเมื่อสกัดข้อมูลเสร็จแล้วนำผลลัพธ์ที่ได้มาสร้างโมเดลจำลองนำทาง โดยรายละเอียดและองค์ประกอบของโมเดลจำลองนำทางจะสอดคล้องกับแผนภาพนำทางหน้าต่าง 5) การสร้างโมเดลจำลองนำเสนอ (Generate Presentation Model) เป็นการวิเคราะห์โมเดลจำลองทางความคิดเข้ากับโมเดลจำลองนำทางเพื่อสกัดและสร้างข้อมูลเพื่อนำมาสร้างโมเดลจำลองนำเสนอ โดยที่โมเดลจำลองนำเสนอเป็นโมเดลที่นำเสนอว่าภายในแผนภาพนำทางหน้าต่างประกอบด้วยหน้าจออะไรบ้าง 6) พาร์ส (Parse) เป็นตัวที่รวบรวมข้อมูลจากโมเดลจำลองนำทางและโมเดลจำลองนำเสนอเพื่อทำการเตรียมเชื่อมโยงข้อมูลเข้ากับรูปแบบโครงสร้างไฟล์เอกซ์เอ็มแอลที่ถูกเตรียมไว้บนโปรเจกต์แอนดรอยด์แอปพลิเคชัน จากนั้นจะส่งผลลัพธ์ที่ได้ไป 7) การสร้างไฟล์รหัสต้นทาง (Generate Source Code) เป็นขั้นตอนการปรับแต่ง ถ่ายโอนข้อมูลเนื้อหาข้อมูลไปยังองค์ประกอบภายในไฟล์เอกซ์เอ็มแอล

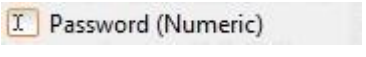
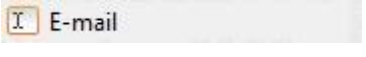

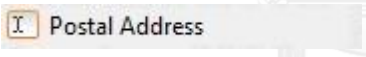
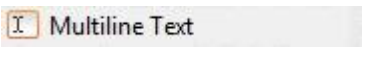


ซอฟต์แวร์ที่ใช้พัฒนาต้นแบบแอนดรอยด์แอปพลิเคชันและแผนภาพนำทางหน้าต่างในงานวิจัยนี้ คือ แอนดรอยด์สตูดิโอ (Android Studio) [13] และสตาร์ยูเอเอ็มแอลสามารถดูรูปเครื่องมือนี้ได้ ดังแสดงในตารางที่ 4 และตารางที่ 5

ตารางที่ 4 คำอธิบายสัญลักษณ์บนเครื่องมือ Android Studio



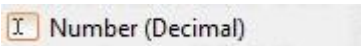
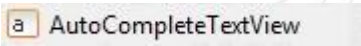
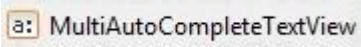
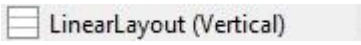
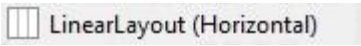
#	Android Studio Shapes	คำอธิบาย
Android Form Widgets		
1	TextView 	เป็น (วิจิต) Widget ใช้สำหรับการจัดการข้อความสายอักขระเช่น การแสดงข้อความต่าง ๆ ทำงานเหมือนกับ เลเบิล
2	Large Text (TextView) 	เป็น Widget อยู่ในกลุ่ม TextView ไว้แสดงผลข้อความที่มีการกำหนดคุณลักษณะเป็นแบบ textAppearanceLarge มีความสามารถคือ ไว้ใช้แสดงผลข้อความในขนาดใหญ่
3	Medium Text (TextView) 	เป็น Widget อยู่ในกลุ่ม TextView ไว้แสดงผลข้อความที่มีการกำหนดคุณลักษณะเป็นแบบ textAppearanceMedium มีความสามารถไว้ใช้สำหรับแสดงผลข้อความในขนาดปานกลาง
4	Small Text (TextView) 	เป็น Widget อยู่ในกลุ่ม TextView ไว้แสดงผลข้อความที่มีการกำหนดคุณลักษณะเป็นแบบ textAppearanceSmall มีความสามารถคือไว้ใช้แสดงผลข้อความในขนาดเล็ก
5	Button 	เป็น Widget ที่ไว้สำหรับการสร้างปุ่ม ต่าง ๆ เช่นปุ่ม Submit และสามารถพัฒนาให้เป็นที่ไว้สำหรับสร้างเหตุการณ์ต่าง ๆ ขึ้นมาได้
6	Small Button (Button) 	เป็น Widget ที่ทำงานเหมือนกับ Button เพียงแต่มีการกำหนดคุณลักษณะเป็นแบบ buttonStyleSmall คือเป็นปุ่มที่มีขนาดเล็ก






7	ToggleButton 	เป็น Widget ที่เป็นปุ่มที่ใช้ไว้สลับการทำงานไปมาระหว่าง เหตุการณ์ 2 เหตุการณ์ ซึ่งในแต่ละเหตุการณ์สามารถเขียนคำสั่งเพื่อควบคุมการกระทำได้ โดยมี 2 สถานะคือ On และ Off
8	CheckBox 	เป็น Widget ที่ไว้สำหรับเลือกรายการว่าจะเลือกหรือไม่เลือก โดย Checkbox สามารถกำหนดค่าเป็น true หรือ false ก็ได้ตามลำดับเหตุการณ์
9	RadioButton 	เป็น Widget ที่ไว้สำหรับกำหนดการเลือกรายการ ซึ่งสามารถมีได้หลายรายการ โดยค่าที่ได้คือ true หรือ false
10	Spinner 	เป็น Widget ที่ไว้สำหรับแสดงรายการใน Spinner หรือ List menu หรือ Drop Down list มีหน้าที่ไว้เลือกรายการ โดยเมื่อคลิกที่ Spinner ก็จะมีรายการลงมาแสดงข้างล่างและสามารถเลือกได้เพียงรายการเดียวเท่านั้น
11	ProgressBar 	เป็น Widget ที่ไว้สำหรับการสร้าง ProgressBar ซึ่งในการพัฒนาแอปพลิเคชันหลาย ๆ ตัว ProgressBar ถือได้ว่าเป็น Widget หลักที่แทบทุกแอปพลิเคชันจำเป็นต้องใช้ เพราะในระหว่างการโหลดข้อมูลหรือการโหลดของแอปพลิเคชันนั้นจะนำ Progress Bar มาใช้งานก็เป็นการแจ้งให้ผู้ใช้หรือผู้ใช้งานทราบว่าในขณะนี้โปรแกรมกำลังทำงานอยู่ และให้ทำการรอนกว่าจะทำงานเสร็จ และหลังจากเสร็จแล้วก็จะแสดงผลข้อมูลหรือรายละเอียดอื่น ๆ อีกครั้ง
12	ProgressBar (Horizontal) 	เป็น Widget ที่ไว้สำหรับการสร้างปุ่ม ProgressBar แบบประเภท แนวนอน (Horizontal) โดยแสดงสถานะการทำงาน รวมถึงสามารถแสดงเป็นตัวเลขเปอร์เซ็นต์กำลังทำงาน โดยข้อมูลเหล่านี้จะทำคำนวณจากการทำงานจริงในขณะนั้น ๆ ซึ่งขึ้นอยู่กับ







		ว่า Progress ในขณะนั้นเกี่ยวกับอะไร และจะนำมาคำนวณเพื่อแสดงผลความคืบหน้าของ Process
13	ProgressBar (Large) 	เป็น Widget ใช้สำหรับสร้าง ProgressBar ที่มีขนาดใหญ่ โดยมีการกำหนดคุณลักษณะเป็นแบบ progressBarStyleLarge
14	ProgressBar (Small) 	เป็น Widget ที่ไว้ใช้สำหรับกำหนดให้ ProgressBar ที่มีขนาดเล็ก โดยมีการกำหนดคุณลักษณะเป็นแบบ progressBarStyleSmall
15	SeekBar 	เป็น Widget สำหรับกำหนดค่าของ Seekbar ซึ่งมีค่าจาก 0 ถึง 100
16	RadioGroup 	เป็น Widget สำหรับควบคุมการทำงานของ RadioButton โดยมีการกำหนดคุณสมบัติของ RadioButton ให้เลือกอย่างใดอย่างหนึ่ง
17	RatingBar 	เป็น Widget ที่เป็นปุ่มสำหรับการเลือกให้คะแนน
18	Switch 	เป็น Widget ที่สามารถกำหนดค่า 2 ค่าระหว่าง On หรือ Off เท่านั้น
Android Text Fields Widgets		
19	Plain Text (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับการรับค่า นำเข้าประภทสายอักขระทั่ว ๆ ไป
20	Personal Name (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับการรับค่า Personal Name โดยมีการกำหนดคุณลักษณะเป็นในส่วนของ EditText ให้เป็นแบบ textPersonName
21	Password (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับการรับค่า Password หรือรหัสผ่าน โดยมีการกำหนดคุณลักษณะในส่วนของ EditText ให้เป็นแบบ

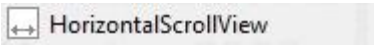


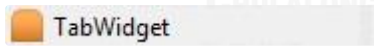


		text Password
22	Password (Numeric) (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับการรับค่า Password ที่เป็นตัวเลข Number เท่านั้น และใน Keyboard ก็จะมีเฉพาะตัวเลข โดยมีการกำหนดคุณลักษณะในส่วนของ EditText ให้เป็นแบบ numberPassword
23	E-mail (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับการรับค่าอีเมล โดยมีการกำหนดคุณลักษณะในส่วนของ EditText ให้เป็นแบบ textEmailAddress
24	Phone (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับการรับค่าหมายเลขโทรศัพท์ โดยมีการกำหนดคุณลักษณะ ในส่วนของ EditText ให้เป็นแบบ phone
25	PostalAddress (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับการรับค่ารหัสไปรษณีย์ โดยมีการกำหนดคุณลักษณะ อยู่ในส่วนของ EditText ให้เป็นแบบ textPostalAddress
26	Multiline Text (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับในการรับค่าข้อความหลายบรรทัด โดยมีการกำหนดคุณลักษณะ ในส่วนของ EditText ให้เป็นแบบ textMultiLine
27	Time (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับในการรับค่าเวลา โดยมีการกำหนดคุณลักษณะ ในส่วนของ EditText ให้เป็นแบบ time
28	Date (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับในการรับค่าวันที่ โดยมีการกำหนดคุณลักษณะ ในส่วนของ EditText ให้เป็นแบบ date















29	Number (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับในการรับตัวเลขโดยมีการกำหนดคุณลักษณะในส่วน of EditText ให้เป็นแบบ number
30	Number (Signed) (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้สำหรับในการรับค่าตัวเลขประเภทติดลบ โดยมีการกำหนดคุณลักษณะ ในส่วน of EditText ให้เป็นแบบ numberSigned
31	Number (Decimal) (EditText) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ EditText ใช้ในการรับค่าตัวเลขประเภททศนิยม โดยกำหนดคุณลักษณะในส่วน of EditText ให้เป็นแบบ numberDecimal
32	AutoCompleterTextView 	เป็น Widget ที่ให้การสนับสนุนในการกรอกหรือนำเข้าข้อมูลของผู้ใช้นั้นให้สะดวกยิ่งขึ้น โดยเมื่อผู้ใช้งานกรอกข้อมูลหรือข้อความลงไปในช่วงว่างที่ไว้ให้ใส่ข้อความ ก็จะแสดงรายการที่เกี่ยวข้องขึ้นเพื่อให้ผู้ใช้ได้เลือกรายการนั้น ๆ ได้
33	MultiAutoCompleteTextView 	เป็น Widget ที่ทำงานเหมือนกับ MultiAutoCompleteTextView แต่มีความสามารถเพิ่มที่ภายใน 1 ช่องข้อมูลจะสามารถใส่ข้อมูลได้หลายรายการ
Android Layouts Widgets		
34	LinearLayout (Vertical) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ Layout โดย LinearLayout แบบแนวตั้ง (Vertical) จะใช้ในการจัดรูปแบบของ Layout ของ Widgets ต่าง ๆ ในลักษณะรูปแบบของเส้นตรง โดยจะเรียงกันในแนวตั้ง
35	LinearLayout (Horizontal) 	เป็น Widget ที่อยู่ในกลุ่มประเภทของ Layout โดย LinearLayout แบบ Horizontal จะใช้ในการจัดรูปแบบของ Layout ของ Widgets ต่าง ๆ ในรูปแบบของเส้นตรง โดยจะเรียงกันในแนวนอน






36	RelativeLayout  RelativeLayout	เป็น Widget ที่อยู่ในกลุ่มประเภทของ Layout โดย RelativeLayout จะใช้ในการจัดรูปแบบของ Layout ของ Widgets ต่าง ๆ จากการอ้างอิงใน Widget อื่น ๆ ที่เกี่ยวข้องกัน ตัวอย่าง เช่น android:-layout_below="@+id/button1" คืออยู่ในตำแหน่งภายใต้การอ้างอิงจาก Widgets ที่มีการกำหนดให้ id="button1" ซึ่งปกติแล้วถ้าเราจัด Layout ผ่าน Graphical Layout จะสร้างคุณลักษณะต่าง ๆ ให้อัตโนมัติอยู่แล้ว
37	AbsoluteLayout  AbsoluteLayout (Deprecated)	เป็น Widget ในประเภทกลุ่มของ Layout โดยตัวของ AbsoluteLayout จะใช้ในการจัดรูปแบบ Layout ของ Widgets ต่าง ๆ ในตำแหน่งแกน X และแกน Y ตามที่กำหนด สำหรับ AbsoluteLayout ใน Android V4.x ถูกยกเลิกใช้ โดยให้ใช้ RelativeLayout แทน
38	FrameLayout  FrameLayout	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Layout โดยตัว FrameLayout จะใช้การจัดรูปแบบ Layout ของ Widgets ต่าง ๆ ในรูปแบบของ เฟรม (Frame) คือ Widget จะซ้อนทับกันแบ่งเป็น Layout
39	Fragment  Fragment	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Layout โดยตัว Fragment จะใช้ในกรณีที่ต้องการขยายวัตถุที่ถูกเลือกหน้าจอการแสดงผลของ Widget ที่อยู่ในแต่ละ Fragment
40	TableLayout  TableLayout	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Layout โดยตัว TableLayout จะใช้การจัดรูปแบบ Layout ของ Widgets ต่าง ๆ ในรูปแบบ Rows และ Column โดยถ้าใช้เฉพาะถ้าใช้ <TableLayout> จะจัดเพียงเฉพาะเป็น Rows ส่วนถ้าจะแบ่ง Column ด้วยให้ใช้ <TableRow>





41	TableRow 	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Layout โดยตัว TableRow จะเป็นหนึ่งในคุณลักษณะของ <TableLayout> โดยทำหน้าที่แบ่ง Column ต่าง ๆ ของ Widget ที่อยู่ภายใต้ <TableRow>
42	Space 	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Layout โดย Space จะใช้ไว้ใช้ในการจัดรูปแบบพื้นที่ว่าง หรือตำแหน่งที่ที่ต้องการปล่อยให้ว่าง
43	GridLayout 	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Layout โดยตัว GridLayout จะใช้ในการจัดรูปแบบ Layout ของ Widgets ต่าง ๆ ในรูปแบบของตารางที่แบ่งเป็น Rows และ Column
Android Composite Widgets		
44	ListView 	เป็น Widget ที่ใช้สำหรับการนำเสนอข้อมูลหรือรายการต่าง ๆ ที่อยู่ในรูปแบบของ Array , ArrayList หรือจากในระบบฐานข้อมูล (Database) ของ SQLite ขึ้นมาแสดงในหน้า Android Form
45	GridView 	เป็น Widget ที่ใช้สำหรับการนำเสนอข้อมูลหรือรายการต่าง ๆ ที่อยู่ในรูปแบบของ Array , ArrayList หรือจาก Database ของ SQLite มานำเสนอในหน้า Android Activity Form โดย GridView จะสามารถทำการแบ่ง คอลัมน์ (Column) ได้ได้หลาย Column หรือสามารถกำหนดเงื่อนไขการแสดงผลอื่น ๆ เช่น แบบ Custom Layout โดยเรียก Layout จากที่อื่น เช่นการแสดงรูปภาพแบบแบ่ง Column หรือทำรายการอื่น ๆ
46	ScrollView 	เป็น Widget ที่ใช้สำหรับการนำเสนอข้อมูลที่เป็น ScrollView หรือ Scrollbar ที่อยู่ในตำแหน่งของ Vertical โดยถ้าใช้ ScrollView ครอบ Widgets อื่น ๆ และข้อมูลใน Form มียาวหรือเยอะลงไปก็จะ

		สามารถใช้ ScrollView เพื่อเลื่อนดูข้อมูลอื่น ๆ ถัดไปได้
47	HorizontalScrollView 	เป็น Widget ที่ใช้สำหรับการนำเสนอข้อมูลที่เป็นการแสดงผลในแนว Horizontal ของ ScrollView หรือ Scrollbar ในตำแหน่งของแนวนอน ส่วนถ้าจะใช้ใน Vertical ให้ใช้ ScrollView
48	SearchView 	เป็น Widget ที่ใช้สำหรับในการสร้างมุมมองของการค้นหาข้อมูลในรูปแบบต่าง ๆ ตามความต้องการ โดยจะมีรูปแบบขยาย สำหรับค้นหาข้อมูลเป็นค่าเริ่มต้น สำหรับ SearchView สามารถนำมาประยุกต์ใช้งานใช้กันกับการใช้กับ Widget หลายตัว เช่นการค้นหาจาก Database ของ SQLite หรือจาก Object Array ต่าง ๆ
49	Sliding Drawer 	เป็น Widget ที่ใช้สำหรับการสร้าง Sliding ที่สามารถลากขึ้นลง หรือลากไปนำหน้าข้างขวา โดยภายใน Sliding จะประกอบด้วย Widgets หลาย ๆ ตัวที่จะใส่คำสั่งหรือ Event ต่าง ๆ ลงไปได้
50	TabWidget 	เป็น Widget ที่ใช้สำหรับการสร้างและควบคุมการแสดงผลของ TabHost
51	TagHost 	เป็น Widget ที่ใช้สำหรับการสร้างข้อมูลภายในแต่ละ layout นั้นให้แบ่งการแสดงผลเป็น Tab โดยใช้ TabHost และ TabWidget ในการจัดการ Widgets และแบ่งขอบเขต ซึ่งแต่ละ Tab จะมีการแสดงผลที่มีความสามารถอิสระต่อกัน โดยจะใช้ FrameLayout มาจัดการการแสดงผลของ Tab อีกที
52	WebView 	เป็น Widget ที่ใช้สำหรับการแสดงผลข้อมูลของเว็บไซต์ที่อยู่ในรูปแบบ URL ต่าง ๆ โดยจะใช้มุมมองของ Web Browser ที่อยู่ใน OS รุ่นนั้น ๆ


Android Images & Media Widgets		
53	ImageView 	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Images & Media โดยที่ ImageView ใช้สำหรับการแสดงรูปภาพ Image และ Picture ต่าง ๆ
54	ImageButton 	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Images & Media โดยที่ ImageButton การทำงานใช้สำหรับการเปลี่ยนรูปปุ่ม Button ให้เป็น Image หรือ Picture
55	Gallery 	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Images & Media โดยที่ Gallery ใช้สำหรับการแสดงรูปภาพเป็นแบบ Gallery คือแสดงใน Horizontal หรือ Vertical และสามารถทำการเลื่อนรูปภาพซ้ายขวาหรือ ขึ้นลงได้
56	MediaController 	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Images & Media โดยที่ MediaController ทำหน้าที่เป็นศูนย์ควบคุมที่มีลักษณะเป็นปุ่ม การทำงานของ Media Player ตัวอย่างเช่น “เล่นหรือหยุด”(Play/Pause), "Rewind", "Fast Forward" เช่นควบคุมการทำงานของ VideoView
57	VideoView 	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Images & Media โดยที่ VideoView ใช้สำหรับแสดงหรือเล่นไฟล์ VDO ต่าง ๆ ตามที่ Format ของ Android รองรับ เช่น .mp4 ,.3gp และ Format อื่น ๆ และในการใช้ VideoView ในบางลักษณะการทำงาน บางครั้งสามารถใช้งานร่วมกับ MediaController เพื่อควบคุมปุ่ม Play/Pause หรืออื่นๆ
Android Time & Date Widgets		
58	TimePicker/TimePickerDialog	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Date &

	 TimePicker	Time โดยที่ TimePicker และ TimePickerDialog ใช้สำหรับการ Picker ข้อมูลในรูปแบบเวลา ชั่วโมง หรือนาที (Hour and Minute)
59	DatePicker/DatePickerDialog  DatePicker	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Date & Time โดยที่ DatePicker และ DatePickerDialog ใช้สำหรับ Picker ข้อมูลในรูปแบบวันที่ วัน เดือน ปี (Year, Month, Day)
60	CalendarView  CalendarView	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Date & Time โดยที่ CalendarView ใช้สำหรับการแสดง Calendar ในรูปแบบต่าง ๆ ตัวอย่างเช่น ในรูปแบบของ Table นอกจากนี้ยังประยุกต์ใช้ได้หลากหลายมาก ทั้ง DatePicker หรือการใช้งานในด้านอื่น ๆ
61	Chronometer  Chronometer	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Date & Time โดยที่ Chronometer ใช้สำหรับ จับเวลา หรือ นับเวลาลอยหลัง
62	AnalogClock  AnalogClock	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Date & Time โดยที่ AnalogClock ใช้สำหรับแสดงเวลาแบบแอนะล็อก (Analog)
63	DigitalClock  DigitalClock	เป็น Widget ที่อยู่ในประเภทกลุ่มของ Date & Time โดยที่ DigitalClock ใช้สำหรับแสดงเวลาแบบดิจิทัล (Digital)
Android Transitions Widgets		
64	ImageSwitcher  ImageSwitcher	เป็น Widget ที่ใช้สำหรับแสดงรูปภาพหรือควบคุมการปรับเปลี่ยนรูปภาพในรูปแบบต่าง ๆ โดยเฉพาะใน ImageSwitcher สามารถใส่ (ผลบังคับ) Effect ให้กับรูปภาพได้ ตัวอย่างเช่น ในกรณีการทำงานลักษณะที่สามารถเปลี่ยนรูปภาพสามารถทำเป็น (เพิ่มสัญญาณทีละน้อย) Fade In หรือ (ลดสัญญาณ

		ที่ละน้อย) Fade Out ได้ ซึ่ง ImageSwitcher สามารถทำงานร่วมกับ Widgets Gallery เพื่อแสดงรูปภาพก่อนแล้วค่อยนำภาพที่ได้จากการเลือกมาแสดงใน ImageSwitcher
65	AdapterViewFlipper  AdapterViewFlipper	เป็น Widget ที่ใช้สำหรับการสร้าง Adapter ให้กับ ViewFlipper
66	StackView  StackView	เป็น Widget ที่ใช้สำหรับแสดง Widgets แบบรูปแบบของ (เรียงทับซ้อน) Stack ตัวอย่างเช่น การแสดงรูปภาพแบบ Stack หรือรูปภาพซ้อนเรียงกันแบบมุมเฉียง
67	TextSwitcher  TextSwitcher	เป็น Widget ที่ใช้สำหรับการกำหนดรูปแบบการเคลื่อนไหวหรือการเปลี่ยนแปลงข้อความต่าง ๆ ที่อยู่ใน TextView ลักษณะของ TextView จะอยู่ในประเภทของ Form ตัวอย่างเช่น การสลับเปลี่ยนข้อความแบบมี Effect หรือกำหนดคุณสมบัติระหว่างการเปลี่ยนแปลง
68	ViewAnimator  ViewAnimator	เป็น Widget ที่ใช้สำหรับกำหนดทิศทางการเปลี่ยนแปลงมุมมอง (View) ของ Widgets ต่าง ๆ โดยสามารถควบคุม มุมมอง Widgets ได้หลาย Widgets โดยไม่จำเป็นจะต้องเป็น Widgets ชนิดเดียวกันและความสามารถอื่น ๆ ตัวอย่างเช่นการกำหนด Effect เป็นภาพเคลื่อนไหวในทิศทางต่างๆ
69	ViewFlipper  ViewFlipper	เป็น Widget ที่ใช้สำหรับ กำหนดทิศทางเคลื่อนไหวของการทำให้เคลื่อนไหว (Animation) ระหว่าง View ของ Widgets จำนวน 2 ตัว หรือมากกว่า 2 ตัว โดยที่ ViewFlipper สามารถกำหนดรูปแบบที่จะมาแทนที่กับ Widgets หรือ Layout เดิมแบบ Flipper และสามารถกำหนดพวก AutoPlay, Play

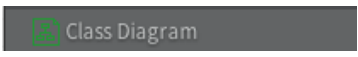
		หรือ Stop
70	ViewSwitcher  ViewSwitcher	เป็น Widget ที่ใช้สำหรับแสดงมุมมองของ Widgets ระหว่าง 2 ตัว โดยจะทำงานสลับกันไปมาระหว่าง มุมมอง 2 มุมมอง หรือ Widgets 2 Widgets ที่อยู่ ภายใต้ ViewSwitcher นั้น ๆ ส่วนถ้ากรณีมากกว่า 2 ตัวให้ใช้ ViewAnimator หรือ ViewFlipper เพราะ ใน Widgets 3-4 ตัวนี้การทำงานจะคล้าย ๆ กัน แต่ ถ้าได้ใช้งานจริง ๆ จะเข้าใจเลยว่า จะมีความสามารถ บางตัว ที่มีไม่เหมือนกันและวัตถุประสงค์ก็ต่างกัน ด้วยเช่นเดียวกัน
Android Advanced Widgets		
71	RequestFocus  requestFocus	เป็น Widget ที่ใช้สำหรับการ Focus ไปยังตำแหน่ง ของ Widgets ต่าง ๆ ที่อยู่ในคุณลักษณะของ Form ตามที่ต้องการ ตัวอย่างเช่น ถ้าเป็น EditText เมื่อใช้ requestFocus ตัว Cursor ก็จะเลื่อนไปตำแหน่งนั้น ๆ
72	ViewStub  ViewStub	เป็น Widget ที่ใช้สำหรับรับผิดชอบการแสดงผล Layout ที่อยู่ในรูปแบบของ XML Layout ที่อยู่ใน ไฟล์อื่น ๆ ในขณะที่กำลังทำงานบน Runtime โดย การทำงานของ ViewStub จะใช้ประเภทการแสดงผล แบบ Layout อื่นด้วยคำสั่ง inflate() และยกเลิกการ แสดงด้วยคำสั่ง setVisibility(int)
73	GestureOverlayView  GestureOverlayView	เป็น Widget ที่ใช้สำหรับ ในการออกแบบเหตุการณ์ ที่จะสามารถเกิดเหตุการณ์ขึ้นบนหน้าจอในรูปแบบ ต่าง ๆ ตัวอย่างเช่นการ Slide ไปยังหน้าจอ หรือการ วาดสัญลักษณ์บนหน้าจอ โดยเราสามารถเขียน เงื่อนไขว่าถ้ามีการกระทำแบบนั้น ๆ จะให้เกิด เหตุการณ์อะไรและทำอะไร ซึ่งปกติแล้วใน Android จะมี Library ที่ เกี่ยว วก ับ Gesture อยู่ เช่น



		GestureDetector ซึ่งกรณีนี้ถ้าใช้ตัวนี้เราจะต้องทำการตรวจสอบเหตุการณ์ต่าง ๆ ขึ้นมาเองได้ด้วยตัวอย่างเช่น เหตุการณ์แบบ onDown, onFling, onLongPress, onScroll, onShowPress และ onSingleTapUp แต่ถ้าอยากจะกำหนดให้ง่ายกว่านั้นเราสามารถให้ GestureOverlayView โดย Widgets ตัวนี้จะอ่านรูปแบบจากรูปแบบ Gesture ที่มีอยู่แล้วว่าเหตุการณ์ที่เกิดขึ้นนั้นมีรูปแบบที่คล้ายคลึงกับเหตุการณ์ที่เราได้ออกแบบไว้หรือไม่ และเราจะต้องทำการออกแบบ Gesture Template ไว้ ซึ่งใน Android ก็มีฟีเจอร์นี้ไว้อยู่แล้วเช่นเดียวกัน
74	NumberPicker 	เป็น Widget ใช้สำหรับการสร้าง Number Picker หรือกล่องสำหรับเลือกตัวเลขลงใน Form ต่าง ๆ ซึ่งในกล่องของตัวเลขสามารถกำหนดค่า Max หรือ Min ต่าง ๆ

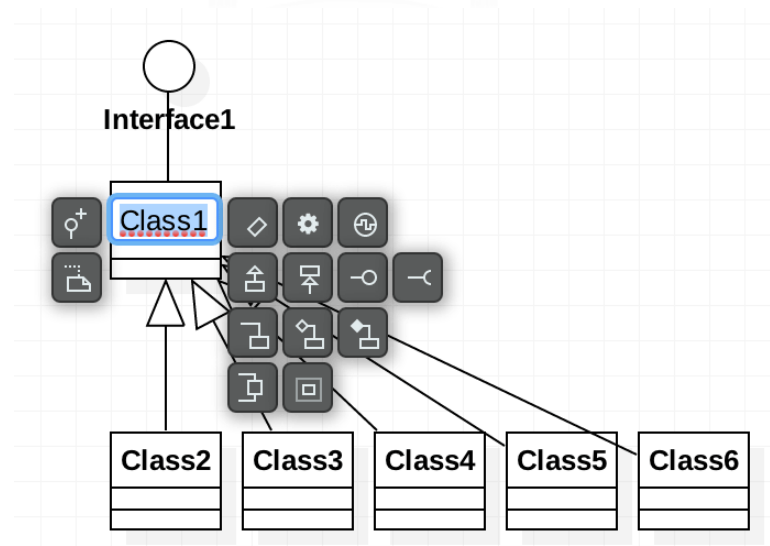
จากตารางที่ 4 เป็นการนำเสนอการจัดกลุ่มสัญลักษณ์เครื่องมือ Android Studio V 2.5 พร้อมคำอธิบายโดยสามารถแบ่งออกได้เป็น 8 กลุ่มหลัก ๆ ได้แก่ Android Form Widgets, Android Text Fields Widgets, Android Layouts Widgets, Android Composite Widgets, Android Images & Media Widgets, Android Time & Date Widgets, Android Transitions Widgets และ Android Advanced Widgets เป็นต้น

ตารางที่ 5 คำอธิบายสัญลักษณ์ในแถบเครื่องมือของสตาร์ยูเอ็มแอลที่ใช้ในงานวิจัย

#	StarUML Shapes	คำอธิบาย
1	Class Diagram 	Class Diagram เป็นหนึ่งในแม่แบบที่ให้เลือกใช้ในเครื่องมือสตาร์ยูเอ็มแอลและอยู่ในมาตรฐานของ UML 2.X
2	Instances	Instances เป็นหนึ่งในคุณลักษณะภายในของ Class Diagram

	Instances	
3	Object Object	Object เป็นการแสดง Shape ที่มีให้ใช้ใน Instances สามารถกำหนดคุณสมบัติของ Object ได้ตามตารางที่ 6
4	Link Link	Link เป็นการแสดง Shape ที่มีให้ใช้ใน Instances เป็นการแสดงความเชื่อมโยงระหว่างวัตถุสองวัตถุเข้าด้วยกัน สามารถกำหนดคุณสมบัติของ Link ได้ตามตารางที่ 7

ตารางที่ 5 อธิบายสัญลักษณ์เบื้องต้นที่ถูกใช้ในงานวิจัยนี้ โดยเครื่องมือสตาร์ยูเอ็มแอลถูกพัฒนาอยู่ภายใต้มาตรฐานของ UML 2.X และสามารถสนับสนุนแผนภาพทั้งหมด 11 แผนภาพ เช่น Class Diagram, Object Diagram, Use Case Diagram, Component Diagram, Deployment Diagram, Composite Diagram, Structure Diagram, Sequence Diagram, Communication Diagram, Activity Diagram and Profile Diagram เครื่องมือสตาร์ยูเอ็มแอลสามารถสร้างองค์ประกอบและการเชื่อมต่อด้วยการสนับสนุนแบบ Shorthands เพื่อสะดวกในการแก้ไขเพื่อสร้างองค์ประกอบและความสัมพันธ์ระหว่าง Class, Sub-Class, Supporting Interfaces เป็นต้น



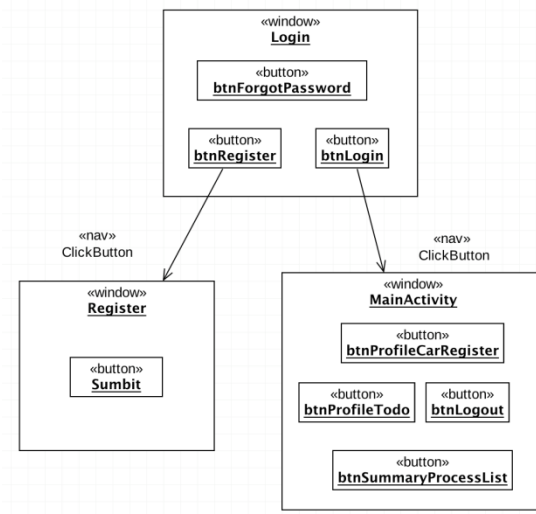
รูปที่ 7 รายละเอียดเครื่องมือสนับสนุน Shorthand

จากรูปที่ 7 เป็นการนำเสนอชุดเครื่องมือพัฒนาที่ช่วยสนับสนุนการสร้างแผนภาพอย่างรวดเร็ว

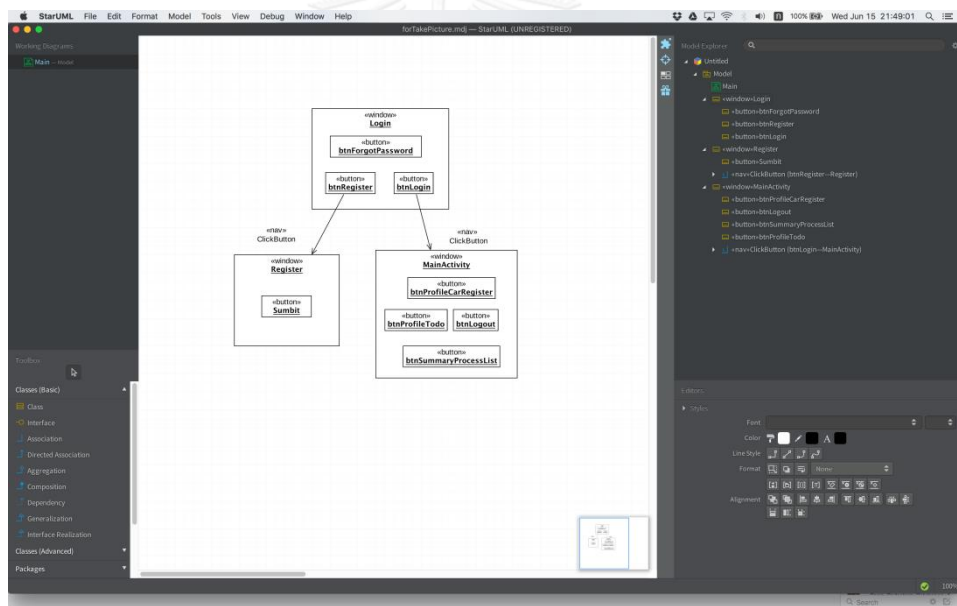
โดยจะอธิบายรายละเอียดขั้นตอนพัฒนาเชิงลึกจะประกอบไปด้วยขั้นตอนต่าง ๆ ดังต่อไปนี้

### 3.1) Create Windows Navigation Diagram

เริ่มต้นจากการออกแบบแผนภาพนำทางหน้าต่างด้วยเครื่องมือสตาร์ยูเอ็มแอลเพื่อใช้เป็นแผนภาพการออกแบบนำเข้าของระบบ ซึ่งเลือกใช้เครื่องมือสตาร์ยูเอ็มแอล เป็นเครื่องมือสนับสนุนการออกแบบและเลือกใช้แม่แบบประเภท Class Diagram ทำให้ผู้ใช้สามารถออกแบบรูปภาพได้ตามรูปที่ 8 และรูปที่ 9 นอกจากนี้ภายในซอฟต์แวร์สตาร์ยูเอ็มแอลยังสามารถติดตั้งส่วนต่อขยายเพิ่มเติมเพื่อสร้างความสะดวกและเพิ่มความหลากหลายให้กับผู้ใช้ อย่างเช่น ภาษา C# สามารถติดตั้งวิธีการ C# code generation and reverse engineering, ภาษา JavaScript สามารถติดตั้งวิธีการ JavaScript code generation, ภาษา C++ สามารถติดตั้งวิธีการ C++ code generation and reverse engineering, ภาษา Ruby สามารถติดตั้งวิธีการ Ruby code generation, การทำ Relationship View สามารถติดตั้งวิธีการ Show relationships of a selected element, การทำ Markdown Documentation สามารถติดตั้งวิธีการ Support Mark down syntax for documentation and preview, ภาษา Java สามารถติดตั้งวิธีการ Java code generation and reverse engineering, ภาษา PHP สามารถติดตั้งวิธีการ PHP code generation, การทำ Teamwork-Server for StarUML สามารถติดตั้งวิธีการ Extension to provide Git functionality and enable collaborative work, ภาษา Merge Generalizations สามารถติดตั้งวิธีการ Quickly merge generalization into a rectilinear form, ภาษา Python สามารถติดตั้งวิธีการ Python code generation, การทำ Generate Getters and Setters สามารถติดตั้งวิธีการ Generate getters and setters for selected attributes or selected classes, ภาษา XMI สามารถติดตั้งวิธีการ XMI Import and Export (XMI 2.1 and UML 2.0), ภาษา DDL (Data Definition Language) สามารถติดตั้งวิธีการ Generate DDL (Data Definition Language) from ERD, การทำ PanesManager สามารถติดตั้งวิธีการ Add two buttons in the status bar to manage to sidebar and the navigator, Hide and Show the panes on the Click



รูปที่ 8 ตัวอย่างแผนภาพนำทางหน้าต่าง



รูปที่ 9 ภาพรวมโปรแกรมสตาร์ยูเอ็มแอล

จากรูปที่ 8 และ 9 เป็นรูปภาพรวมซอฟต์แวร์ที่ถูกออกแบบส่วนประกอบหน้าจอส่วนประกอบของการเชื่อมโยงภายใต้เงื่อนไขแผนภาพหน้าต่างนำทางที่ถูกนำเข้าสู่ระบบ

ตารางที่ 6 คำอธิบายองค์ประกอบคุณสมบัติของ Instance Object

#	คุณสมบัติ	คำอธิบาย
1	Name	เป็นคุณสมบัติของการกำหนดการตั้งชื่อให้กับ Object

		นั้น ๆ
2	Stereotype	เป็นคุณสมบัติของการกำหนดเพิ่มสัญลักษณ์ในภาษา UML ให้กับ Object นั้น ๆ
3	Visibility	เป็นคุณสมบัติของการกำหนดลักษณะการมองเห็นให้กับ Object นั้น ๆ โดยมี 4 คุณลักษณะให้เลือกคือ 1) public 2) protected 3) private และ 4) package
4	Classifier	เป็นคุณสมบัติของการกำหนดชื่อให้ Object นั้น ๆ
5	Value	เป็นคุณสมบัติของการกำหนดค่าให้กับชื่อ Object นั้น ๆ

ตารางที่ 7 คำอธิบายองค์ประกอบคุณสมบัติของ Link

#	คุณสมบัติ	คำอธิบาย
1	Name	เป็นคุณสมบัติของการกำหนดชื่อให้กับ Link นั้น ๆ
2	Stereotype	เป็นคุณสมบัติของการกำหนดเพิ่มสัญลักษณ์ในภาษา UML ให้กับ Link นั้น ๆ
3	Visibility	เป็นคุณสมบัติของการกำหนดลักษณะการมองเห็นให้กับ Link นั้น ๆ
4	Association	เป็นคุณสมบัติในการกำหนดความสัมพันธ์ที่มีการติดต่อสื่อสารกันระหว่างฝั่งรับและส่งข้อความ
5	end1.name	เป็นคุณสมบัติในการตั้งชื่อของจุดฝั่งส่งข้อความ
6	end1.reference	เป็นคุณสมบัติในการอ้างอิงถึง Object ฝั่งส่งข้อความ
7	end1.stereotype	เป็นคุณสมบัติของการกำหนดเพิ่มสัญลักษณ์ในภาษา UML ให้กับฝั่งส่งข้อความ
8	end1.visibility	เป็นคุณสมบัติของการกำหนดลักษณะการมองเห็น

		ข้อความจากฝั่งส่งข้อความ โดยมี 4 คุณลักษณะให้เลือกคือ 1) public 2) protected 3) private และ 4) package
9	end1.navigable	เป็นคุณสมบัติของการกำหนดลักษณะการนำทางของข้อความจากฝั่งส่งข้อความ
10	end1.aggregation	ประกอบไปด้วย 1) none 2) shared 3) composite
11	end1.multiplicity	มี 4 ประเภทดังนี้ 1) 0..1 2) 1 3) 0..* 4) 1..* 5) *
12	end1.defaultValue	เป็นการกำหนดค่าพื้นฐานตรงฝั่งส่งข้อความ
13	end1.isReadOnly	เป็นคุณสมบัติของการกำหนดลักษณะฝั่งส่งข้อความว่าสามารถอ่านได้อย่างเดียว
14	end1.isOrdered	เป็นคุณสมบัติของการกำหนดลักษณะฝั่งส่งข้อความว่าสามารถในการเรียงลำดับข้อความ
15	end1.isUnique	เป็นคุณสมบัติของการกำหนดลักษณะฝั่งส่งข้อความว่าสามารถในการเป็นเอกลักษณ์ของข้อความ
16	end1.isDerived	เป็นคุณสมบัติของการกำหนดลักษณะการนำทางของข้อความจากฝั่งส่งข้อความ
17	end1.isID	เป็นคุณสมบัติของการกำหนดลักษณะการนำทางของข้อความจากฝั่งส่งข้อความ
18	end2.name	เป็นคุณสมบัติในการตั้งชื่อของจุดฝั่งรับข้อความ
19	end2.reference	เป็นคุณสมบัติในการอ้างอิงถึง Object ฝั่งรับข้อความ
20	end2.stereotype	เป็นคุณสมบัติของการกำหนดเพิ่มสัญลักษณ์ในภาษา UML ให้กับฝั่งรับข้อความ
21	end2.visibility	เป็นคุณสมบัติของการกำหนดลักษณะการมองเห็นข้อความจากฝั่งรับข้อความ โดยมี 4 คุณลักษณะให้เลือกคือ 1) public 2) protected 3) private และ

		4) package
22	end2.navigable	เป็นคุณสมบัติของการกำหนดลักษณะการนำทางของข้อความจากฝั่งรับข้อความ
23	end2.aggregation	ประกอบไปด้วย 1) none 2) shared 3) composite
24	end2.multiplicity	มี 4 ประเภทคือ 1) 0..1 2) 1 3) 0..* 4) 1..* 5) *
25	end2.defaultValue	เป็นการกำหนดค่าพื้นฐานตรงฝั่งรับข้อความ
26	end2.isReadOnly	เป็นคุณสมบัติของการกำหนดลักษณะฝั่งรับข้อความที่สามารถอ่านได้อย่างเดียว
27	end2.isOrdered	เป็นคุณสมบัติของการกำหนดลักษณะฝั่งรับข้อความที่สามารถเรียงลำดับข้อความ
28	end2.isUnique	เป็นคุณสมบัติของการกำหนดลักษณะฝั่งรับข้อความที่สามารถเป็นเอกลักษณ์ของข้อความ
29	end2.isDerived	เป็นคุณสมบัติของการกำหนดลักษณะการนำทางของข้อความจากฝั่งรับข้อความ
30	end2.isID	เป็นคุณสมบัติของการกำหนดลักษณะการนำทางของข้อความจากฝั่งรับข้อความ

จากรูปที่ 8 แผนภาพนำทางหน้าต่าง สามารถอธิบายรายละเอียดแบบสรุปได้ตามตารางที่ 8 ตารางที่ 9 ตารางที่ 10 เป็นดังต่อไปนี้

ตารางที่ 8 องค์ประกอบคุณสมบัติแผนภาพนำทางหน้าต่าง ชื่อ Login

#	คุณสมบัติ	ค่าคงที่
1	Name	Login
2	Stereotype	Window

3	Visibility	Public
4	Classifier	-
5	Value	-

จากตารางที่ 8 เป็นการอธิบายถึงรูปที่ 8 แผนภาพนำทางหน้าต่าง โดยเลือกอธิบายองค์ประกอบตามตารางที่ 6 ในส่วนของค่า Instance Object ที่มีชื่อว่า Login

ตารางที่ 9 องค์ประกอบคุณสมบัติ แผนภาพนำทางหน้าต่างชื่อ Register

#	คุณสมบัติ	ค่าคงที่
1	Name	Register
2	Stereotype	Window
3	Visibility	Public
4	Classifier	-
5	Value	-

จากตารางที่ 9 เป็นการอธิบายถึงรูปที่ 8 แผนภาพนำทางหน้าต่าง โดยเลือกอธิบายองค์ประกอบตามตารางที่ 6 ในส่วนของค่า Instance Object ที่มีชื่อว่า Register

ตารางที่ 10 องค์ประกอบคุณสมบัติแผนภาพนำทางหน้าต่าง ชื่อ MainActivity

#	คุณสมบัติ	ค่าคงที่
1	Name	MainActivity
2	Stereotype	Window
3	Visibility	Public
4	Classifier	-
5	Value	-



ตารางที่ 10 อธิบายถึงรูปที่ 8 แผนภาพนำทางหน้าต่าง โดยเลือกอธิบายองค์ประกอบตาม ตารางที่ 6 ในส่วนของค่า Instance Object ที่มีชื่อว่า MainActivity

ตารางที่ 11 องค์ประกอบคุณสมบัติของแผนภาพนำทางหน้าต่าง ชื่อ ClickButton ฝั่งด้านซ้ายมือ

#	คุณสมบัติ	ค่าคงที่
1	Name	ClickButton
2	Stereotype	Nav
3	Visibility	Public
4	Association	-
5	end1.name	-
6	end1.reference	Button
7	end1.stereotype	-
8	end1.visibility	Public
9	end1.navigable	No
10	end1.aggregation	None
11	end1.multiplicity	-
12	end1.defaultValue	-
13	end1.isReadOnly	No
14	end1.isOrdered	No
15	end1.isUnique	No
16	end1.isDerived	No
17	end1.isID	No
18	end2.name	-
19	end2.reference	Setting
20	end2.stereotype	-

21	end2.visibility	Public
22	end2.navigable	Yes
23	end2.aggregation	None
24	end2.multiplicity	-
25	end2.defaultValue	-
26	end2.isReadOnly	No
27	end2.isOrdered	No
28	end2.isUnique	No
29	end2.isDerived	No
30	end2.isID	No

ตารางที่ 11 อธิบายถึงรูปที่ 9 แผนภาพนำทางหน้าต่างโดยเลือกอธิบายองค์ประกอบตามตารางที่ 7 ในส่วนของค่า Instance Link ที่มีชื่อว่า ClickButton ฝั่งด้านซ้ายมือ

ตารางที่ 12 องค์ประกอบคุณสมบัติของแผนภาพนำทางหน้าต่างชื่อ ClickButton ฝั่งด้านขวามือ

#	คุณสมบัติ	ค่าคงที่
1	Name	ClickButton
2	Stereotype	Nav
3	Visibility	Public
4	Association	-
5	end1.name	-

6	end1.reference	Button
7	end1.stereotype	-
8	end1.visibility	Public
9	end1.navigable	No
10	end1.aggregation	None
11	end1.multiplicity	-
12	end1.defaultValue	-
13	end1.isReadOnly	No
14	end1.isOrdered	No
15	end1.isUnique	No
16	end1.isDerived	No
17	end1.isID	No
18	end2.name	-
19	end2.reference	MainActivity
20	end2.stereotype	-
21	end2.visibility	Public
22	end2.navigable	Yes
23	end2.aggregation	None
24	end2.multiplicity	-
25	end2.defaultValue	-
26	end2.isReadOnly	No
27	end2.isOrdered	No
28	end2.isUnique	No

29	end2.isDerived	No
30	end2.isID	No

ตารางที่ 12 อธิบายถึงรูปที่ 9 แผนภาพนำทางหน้าต่างโดยเลือกอธิบายองค์ประกอบตามตารางที่ 7 ในส่วนของค่า Instance Link ที่มีชื่อว่า ClickButton ฝั่งด้านขวามือ

### 3.2) Export Windows Navigation Diagram

ขั้นตอนการ ดำเนินการส่งออก (Export) ในขั้นตอนดำเนินการส่งออกของแผนภาพนำทางหน้าต่างจะใช้ส่วนขยายที่ถูกติดตั้งในซอฟต์แวร์สตาร์ยูเอ็มแอลเป็นตัวสนับสนุนการดำเนินการส่งออก โดยแผนภาพนำทางหน้าต่างจะถูกดำเนินการส่งออกเป็นไฟล์ตระกูลเอกซ์เอ็มไอ หลังจากนั้นจะทำการเปลี่ยนรูปแบบโครงสร้างของเอกซ์เอ็มแอลให้อยู่ในรูปแบบของเจสันด้วยโปรแกรมภาษาจาวาใช้เทคนิคที่มีชื่อว่า XML2JSON โดยงานวิจัยนี้เลือกใช้รูปแบบเจสันเข้ามาช่วยวิเคราะห์เพราะว่าเจสันมีรูปแบบโครงสร้างที่ไม่ซับซ้อน สะดวกต่อการนำไปใช้และพัฒนาในขั้นตอนถัดไป

```
<?xml version="1.0" encoding="UTF-8"?>
<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">
<xmi:Documentation exporter="StarUML" exporterVersion="2.0"/>
<uml:Model xmi:id="AAAAAFVegtQ6G1hvl=" xmi:type="uml:Model" name="RootModel">
<packagedElement xmi:id="AAAAAAFqBWK6M3Z8Y=" name="Model" visibility="public" xmi:type="uml:Model">
<packagedElement xmi:id="AAAAAFS9MqVtLp1BZM=" name="Login" visibility="public" xmi:type="uml:InstanceSpecification">
<ownedMember xmi:id="AAAAAFS9Mf0bbqXsRc=" name="btnLogin" visibility="public" xmi:type="uml:InstanceSpecification">
<xmi:Extension extender="StarUML">
<stereotype value="button"/>
</xmi:Extension>
</ownedMember>
<ownedMember xmi:id="AAAAAFS9N8Ng05rdgE=" name="btnRegister" visibility="public" xmi:type="uml:InstanceSpecification">
<xmi:Extension extender="StarUML">
<stereotype value="button"/>
</xmi:Extension>
</ownedMember>
<ownedMember xmi:id="AAAAAFTE4HQBxmvmc=" name="btnForgotPassword" visibility="public" xmi:type="uml:InstanceSpecification">
<xmi:Extension extender="StarUML">
<stereotype value="button"/>
</xmi:Extension>
</ownedMember>
<xmi:Extension extender="StarUML">
<stereotype value="window"/>
</xmi:Extension>
</packagedElement>
<packagedElement xmi:id="AAAAAFS9NAegbr7HrQ=" name="MainActivity" visibility="public" xmi:type="uml:InstanceSpecification">
<ownedMember xmi:id="AAAAAFS9NAexrsYz6g=" name="ClickButton" visibility="public" xmi:type="uml:InstanceSpecification">
<xmi:Extension extender="StarUML">
<stereotype value="nav"/>
<linkEnd1 value="AAAAAFS9Mf0bbqXsRc="/>
<linkEnd2 value="AAAAAFS9NAegbr7HrQ="/>
</xmi:Extension>
</ownedMember>
<ownedMember xmi:id="AAAAAFDp4Xu9wnZ0=" name="btnLogout" visibility="public" xmi:type="uml:InstanceSpecification">
<xmi:Extension extender="StarUML">
<stereotype value="button"/>
</xmi:Extension>
</ownedMember>
<ownedMember xmi:id="AAAAAFTE4Mxur4QtrU=" name="btnNote" visibility="public" xmi:type="uml:InstanceSpecification">
<xmi:Extension extender="StarUML">
<stereotype value="button"/>
</xmi:Extension>
</ownedMember>
```

รูปที่ 10 ตัวอย่างโครงสร้างเอกซ์เอ็มแอลของแผนภาพนำทางหน้าต่าง

```

[!xmlns:xml:"http://schema.org/spec/XM/2.1",!xmlns:uml:"http://schema.org/spec/XM/2.0",!xmi:documentation:{"!exporter:"StarUML",!exporterVersion:"2.0"},!uml:Model:
{"!xmi:id:"AAAAAFV8yGfKGM9Q=",!name:"rootModel",!packageElement:{"!xmi:id:"AAAAAF+qBmK3Z8F=",!name:"Model",!visibility:"public",!xmi:type:"uml:Model",!packageElement:
[{"!xmi:id:"AAAAAF59vkt19Zk=",!name:"Login",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!ownedMember:
[{"!xmi:id:"AAAAAF59Mlt8b6ksRc=",!name:"btnLogin",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"button"}},
{"!xmi:id:"AAAAAF59Mhg05rdg=",!name:"btnRegister",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"button"}},
{"!xmi:id:"AAAAAF59Hq5kwmcc=",!name:"btnForgotPassword",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"button"}},
{"!xmi:id:"AAAAAF59NAghr7Hrq=",!name:"MainActivity",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!ownedMember:
[{"!xmi:id:"AAAAAF59AexrsYz6g=",!name:"ClickButton",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"nav"},!linkEnd:
{"!value:"AAAAAF59M8b6ksRc="},!linkEnd2:{"!value:"AAAAAF59Mhg05rdg="}},{"!xmi:id:"AAAAAF59KdJubwZ8=",!name:"btnLogout",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:
{"!extender:"StarUML",!stereotype:{"!value:"button"}},{"!xmi:id:"AAAAAF59MouR40rde=",!name:"btnNote",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:
{"!value:"button"}},{"!xmi:id:"AAAAAF59MfA5Xq74=",!name:"btnLocation",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"button"}},
{"!xmi:id:"AAAAAF59Nhb059SM=",!name:"btnSetting",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"button"}},
{"!xmi:id:"AAAAAF59Hq5kwmcc=",!name:"window",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"window"}},
{"!xmi:id:"AAAAAF59PtlNu24eZU=",!name:"btnRegister",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"button"}},
{"!xmi:id:"AAAAAF59d1wNHLJM=",!name:"ClickButton",!visibility:"public",!xmi:type:"uml:InstanceSpecification",!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"nav"},!linkEnd:
{"!value:"AAAAAF59Hq5rdg="},!linkEnd2:{"!value:"AAAAAF59001WlApT6E="}}}],!xmi:extension:{"!extender:"StarUML",!stereotype:{"!value:"window"}}}]]]]]]]
    
```

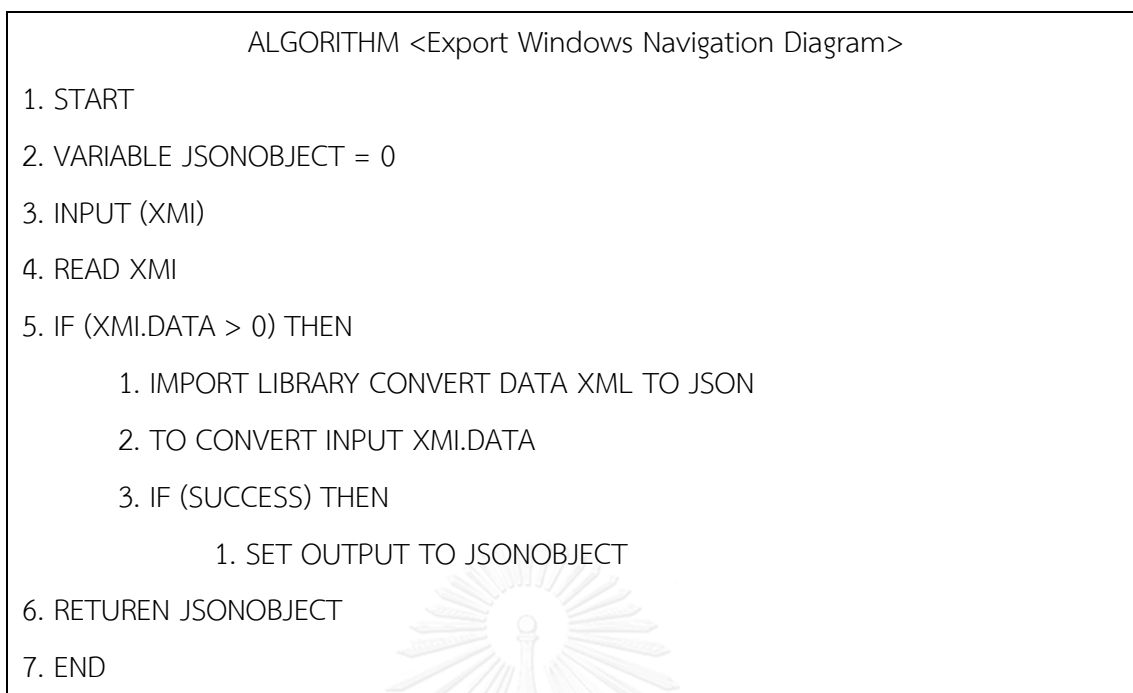
รูปที่ 11 รูปแบบโครงสร้างเจสันที่แปลงมาจากไฟล์เอกซ์เอ็มแอล

จากรูปที่ 10 และรูปที่ 11 แสดงให้เห็นถึงโครงสร้างโดยรวมของแผนภาพนำทางหน้าต่างที่ถูกส่งออกและใช้เทคนิค XML2JSON ตามลำดับ โดยจะเลือกเฉพาะค่าเจสันที่สำคัญมาอธิบายตามตารางที่ 13

ตารางที่ 13 คำอธิบายค่าเจสันที่สำคัญ

#	คอลเลคชัน	คำอธิบาย
1	@xmi:id	เป็นคอลเลคชันที่รับผิดชอบในส่วนการเก็บค่าไอดีของ Object ขึ้นหนึ่ง ๆ
2	@name	เป็นคอลเลคชันที่รับผิดชอบในส่วนการเก็บชื่อของ Object ขึ้นหนึ่ง ๆ
3	packageElement	เป็นคอลเลคชันที่รับผิดชอบในส่วนการเก็บรายละเอียดโมเดลทั้งหมดของแม่แบบขึ้นหนึ่ง ๆ
4	ownedMember	เป็น คอล เลค ชัน ที่ รั บ ผิด ขอบ ใน ส่วน การ เก็บ ส่วนประกอบทั้งหมดใน Object ที่สนใจ
5	xmi :Extension	เป็น คอล เลค ชัน ที่ รั บ ผิด ขอบ ใน ส่วน การ เก็บ ค่า ส่วนประกอบเพิ่มเติมใน Object ที่สนใจ
6	Stereotype	เป็น คอล เลค ชัน ที่ รั บ ผิด ขอบ ใน ส่วน การ เก็บ ส่วนประกอบสัญลักษณ์ในภาษา UML ใน Object ที่สนใจ
7	@value	เป็นคอลเลคชันที่รับผิดชอบในส่วนการเก็บค่าคงที่ใน Object ที่สนใจ

จากตารางที่ 13 เป็นการอธิบายคอลเลคชันที่สำคัญในงานวิจัยนี้

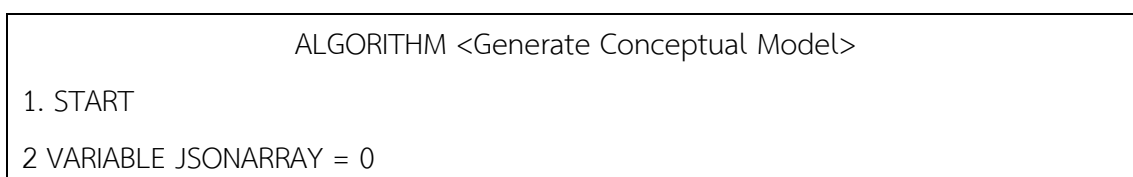


รูปที่ 12 รหัสเทียมของขั้นตอนวิธี Export Windows Navigation Diagram

รูปที่ 12 แสดงรหัสเทียมอธิบายขั้นตอนวิธี (algorithm) Export Windows Navigation Diagram เริ่มต้นจากการอ่านไฟล์เอกซ์เอ็มไอที่ถูกนำเข้าระบบ ถ้าสามารถอ่านค่าได้จะเริ่มกระบวนการแปลงรูปแบบข้อมูลจากเอกซ์เอ็มแอลไปยังเจสัน หลังจากทำการแปลงเสร็จแล้วจะทำการคืนค่าออกมา ค่าที่ถูกคืนออกมาจะอยู่ในรูปแบบตัวแปรของวัตถุเจสัน

### 3.3) Generate Conceptual Model

ในขั้นตอนกระบวนการวิเคราะห์ออกแบบและสร้างโมเดลจำลองทางความคิด (Conceptual model) จะนำหลักการโปรแกรมเชิงวัตถุ (Object-Oriented Programming, OPP) เข้ามาช่วยวิเคราะห์ประมวลผล ซึ่งผลลัพธ์ที่ได้นั้นถูกการสกัดเลือกเฉพาะเนื้อหาที่สำคัญและนำเสนอรายละเอียดองค์ประกอบโมเดลแบบจำลองทางความคิด โดยงานวิจัยนี้จะทำการสร้างโมเดลจำลองความคิดขึ้นจากค่าเจสัน ที่ถูกส่งออกจากขั้นตอนที่ 3.2) ซึ่งค่าเจสันนั้นจะถูกวิเคราะห์เลือกใช้หรือสกัดเฉพาะค่าที่จำเป็น เพื่อนำไปใช้ประโยชน์ในขั้นตอนการแปลงโมเดลในขั้นตอนถัดไป คือ โมเดลจำลองการนำทาง (Navigation model) และโมเดลจำลองการนำเสนอ (Presentation model)



```

3. INPUT (JSON)
4. IF (JSON.SIZE > 0) THEN
5. READ JSON
6. FOR I= 0 to JSON.SIZE
    1. READ JSON SIZE
    2. FIND SPECIFIC SCREEN DATA
    3. IF (FOUND SPECIFIC SCREEN DATA) THEN
        1. SET SPECIFIC SCREEN DATA TO JSONARRAY
    4. NEXT
7. END LOOP
8. RETURN JSONARRAY
9. END

```

รูปที่ 13 รหัสเทียมของขั้นตอนวิธี Generate Conceptual Model

รูปที่ 13 แสดงรหัสเทียมอธิบายขั้นตอนวิธี Generate Conceptual Model โดยขั้นตอนภายในจะเริ่มจากการรับค่าเจสันเข้ามาจากขั้นตอนก่อนหน้าหลังจากนั้นทำการประมวลผล ทำการสกัดข้อมูลส่วนที่ไม่จำเป็นออกและทำการส่งค่ากลับเป็นค่าเจสันแบบแอร์เรย์

### 3.4) Generate Navigation Model

ในขั้นตอนกระบวนการนี้เป็นการวิเคราะห์โมเดลจำลองทางความคิดเพื่อมาออกแบบและสร้างโมเดลจำลองการนำทาง ซึ่งกระบวนการภายในโมเดลการจำลองนำทาง จะเน้นไปที่เส้นทางกิจกรรมจะเป็นไปตามเหตุการณ์ต่าง ๆ ในแอปพลิเคชัน จะเป็นไปตามที่ได้ออกแบบในแผนภาพนำทางหน้าต่างโดยมีการกำหนด Stereotype ในแผนภาพเพื่อนำเข้ามาช่วยวิเคราะห์สำหรับการค้นหาความสัมพันธ์ของเส้นทางอีกด้วย

#### ALGORITHM <Generate Navigation Model>

```

1. START
2. VARIABLE JSONARRAYA = 0, JSONARRAYB = 0
3. INPUT (CONCEPTUALMODEL)
4. CHECK CONCEPTUAL NOT NULL

```

```
5. IF (CONCEPTUALMODEL) > 0 THEN
    1. FIND LINK FROM CONCEPTUALMODEL
    2. READ CONCEPTUAL DATA
    3. LOOP FIND LINK
        1. IF (FOUND LINK) > 0 THEN
            1. CREATE JSONOBJECT
            2. SET HEAD AND END LINKID AND LINDNAME TO JSONOBJECT
            3. SET JSONOBJECT TO JSONARRAYA
            4. NEXT
        4. END FIND LINK LOOP
    5. END IF
6. IF (JSONARRAYA > 0) THEN
    1. FIND RELATED HEAD AND END LINK BY ID AND NAME
    2. READ JSONARRAYA
    3. LOOP JSONARRAYA
        1. LOOP CONCEPTUALMODEL
            1. CREATE JSONOBJECT
            2. FIND START LINK
            3. IF (FOUND RELATED START LINK MATH WITH SOME DATA
                IN CONCEPTUAL MODEL) THEN
                1. SET DATA RELATED BOTH OF START LINK AND
                CONCEPTUAL MODEL TO JSONOBJECT
                2. IF (FOUND RELATED END LINK MATH WITH SOME
                DATA IN CONCEPTUAL MODEL) THEN
                    1. SET DATA RELATED BOTH OF END LINK
                    AND CONCEPTUAL MODEL TO JSONOBJECT
                    2. END IF FOUND RELATED END LINK MATH
                    WITH SOME DATA IN CONCEPTUAL MODEL
                    3. END IF FOUND RELATED START LINK MATH WITH
                    SOME DATA IN CONCEPTUAL MODEL
                4. SET JSONOBJECT TO JSONARRAYB
```



```

5. NEXT
2. END LOOP CONCEPTUALMODEL
3. NEXT
2. END LOOP JSONARRAYA
3. END IF
4. RETURN JSONARRAYB
5. END

```

รูปที่ 14 รหัสเทียมของขั้นตอนวิธี Generate Navigation Model

รูปที่ 14 แสดงรหัสเทียมอธิบายขั้นตอนวิธี Generate Navigation Model เริ่มต้นจากการค้นหาความสัมพันธ์ของ Link จากโมเดลจำลองทางความคิด เพื่อที่หาความสัมพันธ์ระหว่างหน้าจอต้นทางกับหน้าจอปลายทาง และทำการสกัดข้อมูลออกมา หลังจากทำการสกัดข้อมูลเสร็จแล้ว ขั้นตอนต่อไปจะทำการหาความสัมพันธ์ของหน้าจอต้นทางกับหน้าจอปลายทางจากค่าข้อมูลที่ถูกลูกัดกับโมเดลจำลองทางความคิด เพื่อทำการสกัดข้อมูลรายละเอียดองค์ประกอบที่จำเป็น เพื่อนำข้อมูลไปใช้ในขั้นตอนถัดไป

### 3.5) Generate Presentation Model

ในขั้นตอนกระบวนการนี้เป็นการวิเคราะห์โมเดลจำลองทางความคิดเข้ากับโมเดลจำลองนำทางเพื่อมาออกแบบและสร้างโมเดลจำลองนำเสนอ โดยที่โมเดลจำลองนำเสนอจะรับผิดชอบในส่วนการนำเสนอข้อมูลออกทางส่วนต่อประสานผู้ใช้ที่จะถูกนำเสนอออกผ่านทางหน้าจอส่วนต่อประสานมือถือแอนดรอยด์แอปพลิเคชัน

```

ALGORITHM <Generate Presentation Model>
1. START
2. INPUT (NAVIGATIONMODEL, CONCEPTUALMODEL)
3. INPUT (PATHFILE ANDROID PROJECT)
4. READ NAVIGATIONMODEL NOT NULL
5. READ CONCEPTUALMODEL NOT NULL
6. VARIABLE FILEANDRIDPROJECTARRAY, PROJECTANDROIDNAMESTRING,
FULLNAMESTRING, HEADERNAMESTRING, FULLNAMEJSONARRAY
7. TRY ACCESS FOLDER

```

1. ACCESS FOLDER BY PATHFILE
2. FIND CLASS RELATED IN ANDROID PROJECT BY PATHFILE ANDROID PROJECT
3. IF (FOUND CLASS RELATED IN ANDROID PROJECT) THEN
  1. IF (CLASS FILE PATH OF ANDROID PROJECT >0) THEN
    1. GET ALL NAME OF FILE
    2. SET ALL NAME OF FILE TO FILEANDRIDPROJECTARRAY
    3. END IF CLASS FILE PATH OF ANDROID PROJECT
  2. FILTER OUT HELPER CLASS IN FILEANDRIDPROJECTARRAY
8. NEW STRING HEADER COMPONENT OF CLASS NAME BY PATHFILE ANDROID PROJECT
9. SET NEW STRING HEADER TO HEADERNAMESTRING
10. LOOP FILEANDRIDPROJECTARRAY
  1. GET NAME FROM FILEANDRIDPROJECTARRAY
  2. NEW STRING MADE BY APPENDING NAME TO THE HEADERNAMESTRING
  3. SET NEW STRING TO FULLNAMESTRING
  4. CREATE JSONOBJECT
  5. SET FULLNAMESTRING TO JSONOBJECT
  6. SET ID AND TO JSONOBJECT
  7. SET FILENAMEANDROIDPROJECTARRAYATINDEX TO JSONOBJECT
  8. ADD JSONOBJECT TO FULLNAMEARRAY
  9. NEXT
11. END LOOP FILEANDRIDPROJECTARRAY
12. LOOP FILEANDRIDPROJECTARRAY
  1. GET ID AT INDEX FROM FILEANDRIDPROJECTARRAY
  2. LOOP CONCEPTUALMODEL
    1. FIND MATCH WITH ID
    2. IF (ID MATCH)
      1. CREATE JSONOBJECT
      2. ADD DETAIL TO JSONOBJECT
      3. SET JSONOBJECT TO FULLNAMEJSONARRAY
    3. NEXT

```

3. NEXT
4. END LOOP CONCEPTUALMODEL
13. END LOOP FILEANDRIDPROJECTARRAY
14. RETURN FULLNAMEJSONARRAY
15. END

```

รูปที่ 15 รหัสเทียมของขั้นตอนวิธี Generate Presentation Model

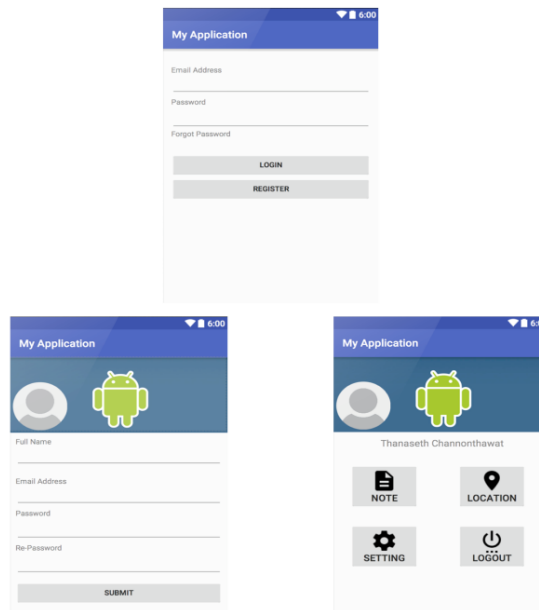
รูปที่ 15 แสดงรหัสเทียมอธิบายขั้นตอนวิธี Generate Presentation Model เริ่มต้นจากการรับค่าผลลัพธ์ที่ได้จากขั้นตอนที่ 3.3) กับ 3.4) และการรับค่าจากโปรเจกแอนดรอย์แอปพลิเคชัน หลังจากนั้นจะทำการเข้าถึงไฟล์ในโปรเจกแอนดรอย์แอปพลิเคชัน เพื่อทำการหาคลาสการทำงานที่เกี่ยวข้องกับการแสดงผลทั้งหมดโดยจะไม่สนใจคลาสตัวช่วย (Class Helper) หลังจากนั้นจะทำการสร้างชื่อภายในของคลาสให้ครบทุกแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้ ในขั้นตอนสุดท้ายจะทำการอ้างอิงข้อมูลจาก Navigation model เข้ากับผลลัพธ์ก่อนหน้าคือแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้เพื่อเปรียบเทียบข้อมูลและกำหนดชื่อให้กับแต่ละคลาส

### 3.6) Parse

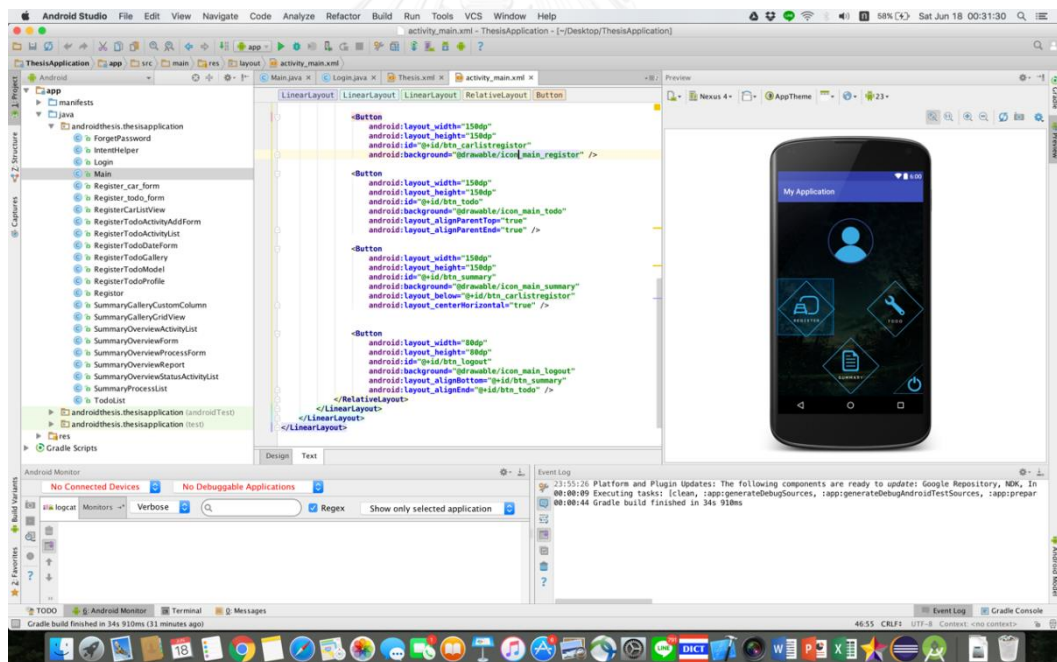
ในขั้นตอนของกระบวนการ Parse กระบวนการนี้เป็นขั้นตอนกระบวนการสร้างส่วนติดต่อแม่แบบส่วนต่อประสานผู้ใช้ในโปรเจกแอนดรอย์แอปพลิเคชันและทำการเตรียมข้อมูลเพื่อที่ถ่ายโอนข้อมูลไปยังไฟล์เอกซ์เอ็มแอลที่อยู่บนโปรเจกแอนดรอย์แอปพลิเคชัน โดยจะแบ่งขั้นตอนกระบวนการทำงานออกเป็น 2 ส่วนการทำงานด้วยกันคือ 3.6.1) การสร้างแม่แบบเค้าร่าง 3.6.2) การเตรียมสร้างส่วนควบคุม โดย 2 ส่วนการทำงานนี้จะถูกอธิบายและนำเสนอต่อไปนี้

#### 3.6.1) การสร้างแบบเค้าร่าง

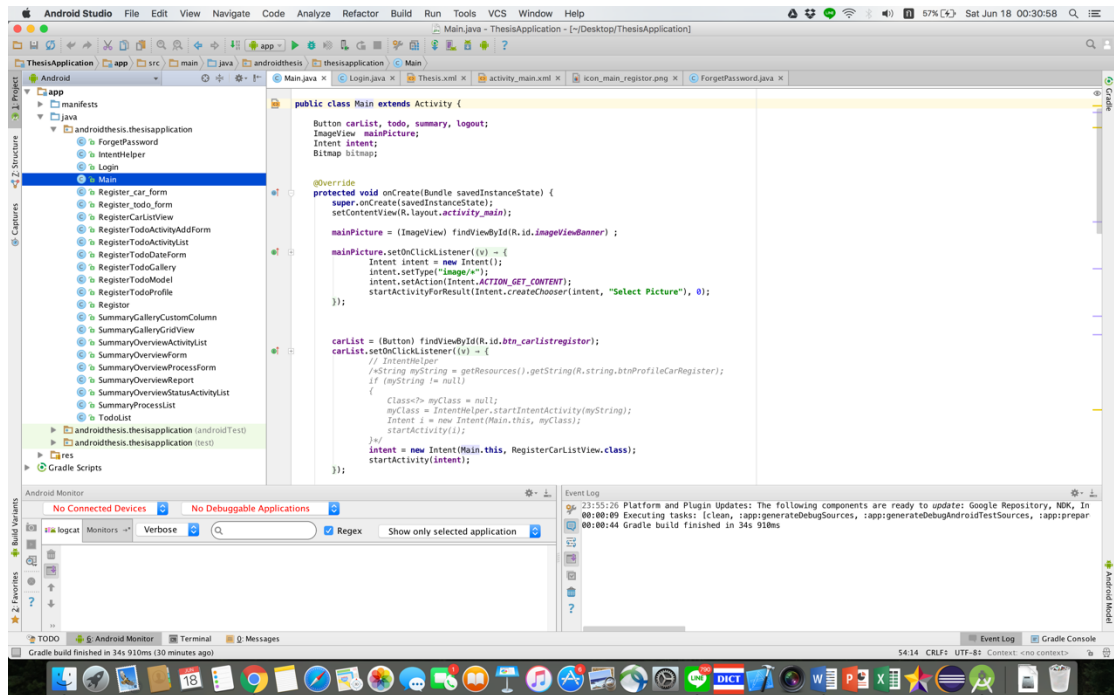
ขั้นตอนนี้เป็นการสร้างแม่แบบเค้าร่างส่วนต่อประสานที่ประกอบด้วยองค์ประกอบโครงสร้างหน้าจอ การจัดวางส่วนประกอบต่าง ๆ ที่แสดงอยู่บนส่วนต่อประสานผู้ใช้บนแอนดรอย์แอปพลิเคชันและรวมถึงกิจกรรมการทำงานเบื้องต้น โดยผู้ใช้สามารถออกแบบต้นแบบส่วนต่อประสานแอนดรอย์ได้เองตามความต้องการ หลังจากสร้างแม่แบบเค้าร่างส่วนต่อประสานผู้ใช้ ผลลัพธ์ในแต่ละส่วนต่อประสานผู้ใช้ประกอบด้วยไฟล์เอกซ์เอ็มแอลเกี่ยวกับการแสดงผลและไฟล์จาวาเกี่ยวกับกิจกรรมการทำงานแบบเบื้องต้น



รูปที่ 16 ตัวอย่างแม่แบบการจัดหน้าส่วนต่อประสาน



รูปที่ 17 ตัวอย่างไฟล์เอกซ์เอ็มแอลรับผิดชอบในส่วนต่อประสานผู้ใช้



รูปที่ 18 ตัวอย่างไฟล์จาวารับผิดชอบในส่วนพัฒนาโปรแกรม

### 3.6.2) การเตรียมสร้างส่วนควบคุม

หลักการทํางานของส่วนการเตรียมการสร้างส่วนควบคุมจะเป็นการวิเคราะห์ระหว่างผลลัพธ์ของ Navigation model เข้ากับ Presentation Model เพื่อที่ทำการเตรียมการสร้างชุดข้อมูลและจะนำไปแปลงข้อมูลในขั้นตอนถัดไปคือ Generate Source Code และเป็นการตรวจทานข้อมูลที่ถูกสร้างในขั้นตอนที่ 3.3) และ 3.4) โดยจะอธิบายด้วยภาษาซูดโค้ดดังรูปที่ 19

#### ALGORITHM <Parse>

1. START
2. VARIABLE JSONARRAY
3. INPUT (NAVIGATIONMODEL, PRESENTATIONMODEL)
4. COMPARE NAVIGATIONMODEL WITH PRESENTATIONMODEL
5. LOOP NAVIGATIONMODEL
  1. LOOP PRESENTATIONMODEL
    1. IF (MATCH) THEN

<ol style="list-style-type: none"> <li>1. CREATE JSONOBEJCT</li> <li>2. ADD MATCH TO JSONOBJECT</li> <li>3. ADD JSONOBJECT TO JSONARRAY</li> <li>4. ENDIF</li> <li>5. NEXT</li> <li>6. END LOOP PRESENTATIONMODEL</li> </ol> <ol style="list-style-type: none"> <li>2. NEXT</li> <li>3. END LOOP NAVIGATIONMODEL</li> </ol> <ol style="list-style-type: none"> <li>6. RETUREN JSONARRAY</li> <li>7. END</li> </ol>
--

รูปที่ 19 รหัสเทียมของขั้นตอนวิธี Parse

รูปที่ 19 แสดงรหัสเทียมอธิบายขั้นตอนวิธีการเตรียมการสร้างความควบคุม โดยวิธีการเป็นการเปรียบเทียบข้อมูลระหว่าง Navigation model กับ Presentation model และเป็นเพื่อเป็นการตรวจทานชุดข้อมูลที่ถูกสร้างขึ้นมา

### 3.7) Generate Source Code

ในขั้นตอนนี้เป็นการสร้างไฟล์รหัสต้นทางจากผลลัพธ์ที่ได้จากขั้นตอนก่อนหน้าคือ ขั้นตอนการเตรียมสร้างความควบคุม หลังจากนั้นจะนำข้อมูลชุดนั้นมาประมวลหาความสัมพันธ์ระหว่างข้อมูลในไฟล์เอกซ์เอ็มแอลกับชุดข้อมูลที่ถูกเตรียมต่อมาจะทำการถ่ายโอนข้อมูลไปยังไฟล์เอกซ์เอ็มแอลด้วยหลักการของ DOM Parser ที่ถูกพัฒนาขึ้นโดยภาษาจาวา หลังจากนั้นใช้ซอฟต์แวร์ Android Studio V 2.2 ทำการรันโปรเจคแอนดรอยด์แอปพลิเคชันเพื่อทำการทดสอบไฟล์เอกซ์เอ็มแอลที่ถูกการถ่ายโอนข้อมูลและเพื่อจำลองการแสดงผลต้นแบบออกทางส่วนจำลองแอนดรอยด์แอปพลิเคชัน

<p style="text-align: center;">ALGORITHM &lt;Generate Source Code&gt;</p> <ol style="list-style-type: none"> <li>1. START</li> <li>2. INPUT (PARSE, PATHFILE)</li> <li>3. VARIABLE PREPARSE</li> <li>4. TRY ACCESS FILE FOLDER <ol style="list-style-type: none"> <li>1. ACCESS XML FILE IN ANDROID PROJECT</li> </ol> </li> </ol>
--

```
2. GET DATA FROM XML FILE
3. PREPARSE = PARSE
4. LOOP DATA FROM XML FILE
    1. GET TAG
    2. GET ATTRIBUTE FROM TAG
    3. LOOP PREPARSE
        1. GET PREPARSE AT INDEX
        2. GET NAME
        3. IF (NAME == ATTRIBUTE) THEN
            1. SET CONTENT TEXT TO TAG
            2. REMOVE OBJECT IN PREPARSE
        4. ELSE
            1. SET NULL TO TAG
        5. NEXT
    4. END LOOP PREPARSE
5. NEXT
6. END DATA FROM XML FILE
5. CHECK PREPARSE
6. IF (PREPARSE >0) THEN
    1. LOOP PREPARSE
        1. CREATE NEW NODE TO XML FILE
        2. SET HEADER TEXT
        3. SET ATTRIBUTE AND VALUE
        4. SET CONTENT TEXT
    2. NEXT
3. END LOOP PREPARSE
7. SAVE TRANSFORM XML FILE
8. END
```

รูปที่ 20 แสดงรหัสเทียมอธิบายขั้นตอนวิธี Generate Source Code ในตอนแรกทำการเข้าถึงข้อมูลที่เป็นไฟล์เอกซ์เอ็มแอลที่อยู่ในโปรเจกแอนดรอยด์แอปพลิเคชัน โดยไฟล์เอกซ์เอ็มแอลเป็นไฟล์ที่สร้างขึ้นมาสำหรับงานวิจัยนี้โดยเฉพาะเพราะว่าไฟล์เอกซ์เอ็มแอลนี้จะสัมพันธ์ถึงการแสดงผลออกทางส่วนต่อประสานผู้ใช้กับไฟล์จาวาที่รับผิดชอบในส่วนพัฒนาแอนดรอยด์แอปพลิเคชันนั้น ๆ หลังจากที่สามารอ่านไฟล์เอกซ์เอ็มแอลได้แล้ว จะทำการหาค่าที่มีความสัมพันธ์ระหว่างแท็กเอกซ์เอ็มแอล (Tag XML) กับค่าเจสันที่ได้มาจากการเตรียมข้อมูล เมื่อพบว่ามีข้อมูลที่สัมพันธ์กันหรือไม่สัมพันธ์กัน จะใช้หลักการของ DOM Parser ในการถ่ายโอนข้อมูล โดยข้อมูลที่ถูถ่ายโอนจะเกี่ยวข้องและสัมพันธ์กับผลลัพธ์ที่ถูกส่งออกไปยังส่วนจำลองแสดงผลแอนดรอยด์แอปพลิเคชัน ถ้าข้อมูลเกิดสัมพันธ์ก็ใช้ชุดข้อมูลนั้นในการถ่ายโอน และถ้าเกิดไม่สัมพันธ์จะให้ค่าว่างไว้ และต่อมาจะใช้ Android Studio เพื่อทำการทดสอบผลลัพธ์ของการถ่ายโอนข้อมูล





## บทที่ 4

### การพัฒนาเครื่องมือ

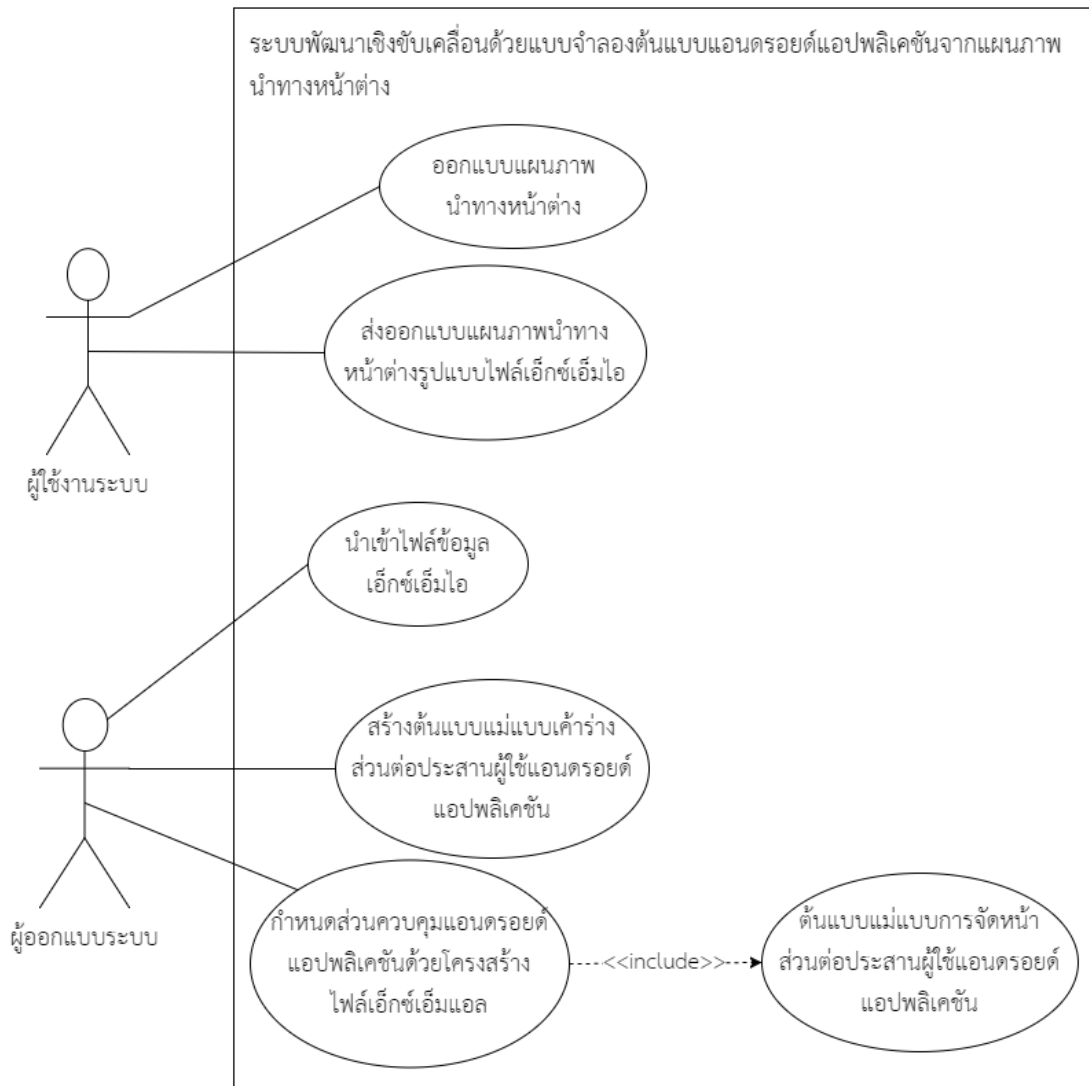
เนื้อหารายละเอียดในบทที่จะกล่าวถึงนี้ จะกล่าวถึงวิธีการออกแบบและพัฒนาเครื่องมือต้นแบบเพื่อสนับสนุนแนวทางอีกแนวทางหนึ่งในการทดสอบคุณลักษณะของแอนดรอยด์แอปพลิเคชันจากความต้องการโดยสร้างจากส่วนประกอบรายละเอียดที่ได้จากการออกแบบแผนภาพนำทางหน้าต่างที่ได้นำเสนอในบทที่ 3 โดยจะกล่าวถึงโครงสร้างของซอฟต์แวร์และสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ ซึ่งรายละเอียดจะมีดังต่อไปนี้

#### 4.1) ความต้องการเชิงฟังก์ชัน

- 1.) ระบบจะสามารถอ่านไฟล์ที่ถูกส่งออกจากการออกแบบแผนภาพนำทางหน้าต่างที่มีรูปแบบเป็นไฟล์เอกซ์เอ็มไอได้
- 2.) ระบบจะสามารถสร้างความสัมพันธ์ที่มีเหตุขึ้นระหว่างแผนภาพนำทางหน้าต่างแล้วสามารถนำมาสร้างการเชื่อมต่อระหว่างแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้ได้
- 3.) ระบบจะอนุญาตและสามารถให้ผู้ใช้สามารถเพิ่มเติมส่วนควบคุมและแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้ขึ้นเองได้
- 4.) ระบบจะอนุญาตและสามารถแก้ไขหรือเพิ่มเติมรายละเอียดแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้เองได้แต่ต้องอยู่ภายใต้เงื่อนไขของการออกแบบ
- 5.) ระบบสามารถแสดงผลบนหน้าจอส่วนต่อประสานได้

#### 4.2) การวิเคราะห์ความต้องการและแผนภาพฟังก์ชันงานของระบบ

จากหัวข้อก่อนหน้านี้ เราจะสามารถนำมาวิเคราะห์เพื่อออกแบบเครื่องมือสนับสนุนการทำงาน ซึ่งหลังจากวิเคราะห์แล้วผลลัพธ์ที่ได้จากแบบจำลองเชิงฟังก์ชันสามารถนำมาเขียนเป็นแผนภาพยูสเคสได้ดังรูปที่ 21



รูปที่ 21 แผนภาพยูสเคสของระบบพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองต้นแบบแอนดรอยด์แอปพลิเคชันจากแผนภาพนำทางหน้าต่าง

จากแผนภาพที่ 21 แผนภาพยูสเคสของระบบพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองต้นแบบแอนดรอยด์แอปพลิเคชันจากแผนภาพนำทางหน้าต่างเป็นการแสดงให้เห็นถึงความเกี่ยวข้องของผู้ใช้งานระบบกับผู้ออกแบบระบบ โดยจะอธิบายรายละเอียดแต่ละฟังก์ชันดังต่อไปนี้

### 1.) ออกแบบแผนภาพนำทางหน้าต่าง

ในขั้นตอนแรกผู้ใช้งานต้องทำการออกแบบแผนภาพนำทางหน้าต่างสำหรับการสร้างโครงสร้างแอนดรอยด์แอปพลิเคชันด้วยเครื่องมือสตาร์ยูเอ็มแอล โดยผู้ใช้งานต้องทำการระบุแผนภาพนำทางหน้าต่าง1 แผนภาพนำทางหน้าต่างชั้น ส่วนต่อประสานบนผู้ใช้แอนดรอยด์แอปพลิเคชัน 1 หน้าจอ แผนภาพนำทางหน้าต่างจำเป็นต้องถูกกำหนดด้วยสตอรี่ไอโทปให้ครบทุกแผนภาพนำ

ทางหน้าต่างเพื่อเป็นตัวสัญลักษณ์กำหนดว่าเป็นส่วนหน้าต่างส่วนต่อประสาน โดยภายในแบบจำลอง แผนภาพนำทางหน้าต่างจะประกอบไปด้วยส่วนประกอบย่อยที่จะแสดงถึงรายละเอียดองค์ประกอบที่สามารถรองรับการเชื่อมโยงระหว่างแบบจำลองแผนภาพนำทางหน้าต่าง เพื่อการกำหนดให้ถูกแสดงอยู่บนหน้าต่างส่วนต่อประสานผู้ใช้ ต่อมาผู้ใช้ระบบจำเป็นต้องทำการกำหนดสแตทริโอไทป์ให้ทุก ๆ ค่าของส่วนประกอบย่อยที่ถูกแสดงผลบนหน้าต่างส่วนต่อประสานผู้ใช้ เพื่อเป็นตัวสัญลักษณ์กำหนดว่าเป็นส่วนประกอบย่อยบนส่วนต่อประสานผู้ใช้และสำหรับการเชื่อมโยงแบบแผนภาพนำทางหน้าต่างเข้าด้วยกัน ก็ต้องกำหนดส่วนสแตทริโอไทป์ด้วยเช่นกันและกำหนดตัวสัญลักษณ์กำหนดว่าเป็นส่วนการเชื่อมโยงระหว่างส่วนต่อประสานกับส่วนต่อประสาน

## 2.) ส่งออกแบบแผนภาพนำทางหน้าต่างรูปแบบไฟล์เอ็กซ์เอ็มแอล

เมื่อผู้ใช้งานระบบทำการออกแบบระบบเสร็จจะต้องแบบแผนภาพนำทางหน้าต่างออกมาให้อยู่ในรูปแบบไฟล์เอ็กซ์เอ็มแอลด้วยเครื่องมือส่วนต่อขยายที่ถูกติดตั้งในสตาร์ยูเอ็มแอล เพื่อให้ผู้ดูแลระบบใช้ไฟล์เอ็กซ์เอ็มแอลในการนำเข้าสู่ระบบในขั้นตอนต่อไป

## 3.) นำเข้าไฟล์ข้อมูลเอ็กซ์เอ็มแอล

ผู้ดูแลระบบต้องนำไฟล์รูปแบบเอ็กซ์เอ็มแอลที่มีรายละเอียดเกี่ยวกับการออกแบบภาพนำทางหน้าต่างที่ได้จากขั้นตอนก่อนหน้านี้ นำไปเข้าสู่ระบบขั้นตอนการแปลงเนื้อหาภายในไฟล์ โดยจะทำการอ่านไฟล์เพื่อดึงข้อมูลรูปแบบเอ็กซ์เอ็มแอลออกมา จากนั้นจะทำการแปลงรูปแบบข้อมูลให้อยู่ในรูปแบบข้อมูลเจสัน ข้อดีของรูปแบบเจสันคือ จะสามารถช่วยให้ระบบสามารถเข้าถึงเนื้อหา รายละเอียดไฟล์ค่าต่าง ๆ ที่ประกอบอยู่ภายในได้สะดวกและง่ายขึ้น

## 4.) สร้างต้นแบบแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชัน

ผู้ดูแลระบบจำเป็นต้องสร้างโปรเจกต์แอนดรอยด์แอปพลิเคชัน เพื่อที่จะทำการพัฒนาและกำหนดรายละเอียดแต่ละแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้ให้มีความสวยงามและมีความหลากหลายเกิดขึ้น หลังจากที่สามารถพัฒนาแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้เสร็จแล้ว ผู้ดูแลระบบจำเป็นต้องพัฒนาโค้ดและระบุการทำงานพื้นฐานเล็กน้อยในแต่ละหน้าจอเพื่อให้สอดคล้องกับสิ่งที่ต้องการที่ต้องการนำเสนอออกทางส่วนต่อประสานผู้ใช้

## 5.) กำหนดส่วนควบคุมแอนดรอยด์แอปพลิเคชันด้วยโครงสร้างไฟล์เอ็กซ์เอ็มแอล

หลังจากที่ผู้ดูแลระบบสร้างแม่แบบจัดหน้าส่วนต่อประสานผู้ใช้เสร็จเรียบร้อยแล้ว ผู้ดูแลระบบจำเป็นต้องสร้างไฟล์เอ็กซ์เอ็มแอลขึ้นภายในโปรเจกต์แอนดรอยด์แอปพลิเคชัน เพื่อให้ไฟล์เป็นส่วนควบคุมเนื้อหาที่จะถูกนำเสนอออกทางหน้าต่างส่วนต่อประสานผู้ใช้ โดยรายละเอียดเนื้อหา

ภายในไฟล์จำเป็นต้องสอดคล้องเข้ากับส่วนสตอรี่ไอโพบ์ที่ถูกออกแบบในขั้นตอนออกแบบแผนภาพนำทางหน้าต่าง ต่อมาผู้ดูแลระบบจำเป็นต้องแก้ไขในส่วนโค้ดบางส่วนในส่วนประกอบย่อยของแต่ละหน้าจอส่วนต่อประสานผู้ใช้ เพื่อให้โค้ดที่ถูกพัฒนาในขั้นตอนก่อนหน้าสนับสนุนการทำงานของไฟล์เอ็กซ์เอ็มแอลที่เพิ่งถูกสร้างขึ้นและเพื่อสนับสนุนให้สามารถนำหน้าจอแม่แบบจัดหน้าส่วนต่อประสานกลับมาใช้ใหม่ได้เมื่อนำเอาถูกออกแบบใหม่ในขั้นตอนแรก

#### 4.3) สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือสนับสนุน

สภาพแวดล้อมที่ถูกใช้ในการพัฒนาระบบจะอ้างอิงมาจากเครื่องมือคอมพิวเตอร์ที่ใช้ในการพัฒนาเครื่องมือสนับสนุนชิ้นนี้ โดยประกอบไปด้วยทั้งฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) ที่มีส่วนเกี่ยวข้องกับการพัฒนาระบบเครื่องมือสนับสนุน ซึ่งรายละเอียดจะประกอบด้วยดังต่อไปนี้

##### 4.3.1) ฮาร์ดแวร์

เครื่องคอมพิวเตอร์ที่นำมาใช้พัฒนาเครื่องมือสนับสนุนชิ้นนี้ควรมีประสิทธิภาพของฮาร์ดแวร์ขั้นต่ำดังต่อไปนี้.

##### 1.) หน่วยประมวลผลโปรเซสเซอร์

หน่วยประมวลผลโปรเซสเซอร์ Intel Core i5 แบบ Dual-core ความเร็ว 2.8 GHz พร้อม Turbo Boost สูงสุดถึง 3.3 GHz พร้อมแคช L3 แบบใช้ร่วมกัน ขนาด 3MB

##### 2.) หน่วยความจำหลักของคอมพิวเตอร์ (หน่วยความจำชั่วคราว)

8 GB of 1600 MHz LPDDR3

##### 3.) หน่วยความจำถาวร

พื้นที่จัดเก็บข้อมูล ขนาด 512 GB เป็น flash storage แบบ PCIe

##### 4.) หน่วยประมวลผลภาพ การ์ดจอ

Intel Iris Graphics

##### 4.3.2) ซอฟต์แวร์

เครื่องคอมพิวเตอร์ที่นำมาใช้พัฒนาเครื่องมือสนับสนุนชิ้นนี้ควรมีซอฟต์แวร์ดังต่อไปนี้

- 1.) OS X El Capita
- 2.) Eclipse Mars (64-bit), Eclipse IDE for Java Developers for macOS
- 3.) Android Studio 1.5.2, install SDK for macOS (64-bit)
- 4.) Virtual Box V 5.0.14 r10527 for macOS (64-bit)
- 5.) Genymotion V 2.6.0 for macOS (64-bit)
- 6.) StarUML V 2.6.0 for macOS (64-bit)

#### 4.3.3) การติดตั้งซอฟต์แวร์

ทำการติดตั้งซอฟต์แวร์ที่ช่วยในการพัฒนาระบบทั้งหมดลงในเครื่องคอมพิวเตอร์เครื่องที่จะนำมาพัฒนาระบบโดยเริ่มติดตั้งตามลำดับขั้นตอนต่อไปนี้

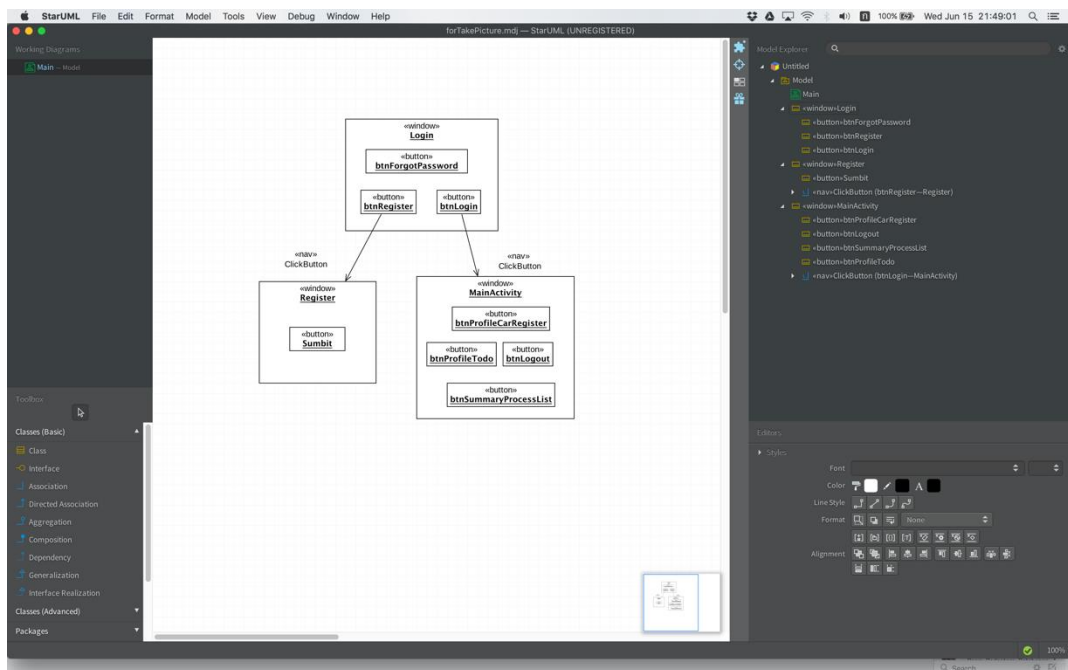
- 1.) ติดตั้งโปรแกรม Android Studio
- 2.) ติดตั้ง VirtualBox
- 3.) ติดตั้ง Genymotion
  - a. ติดตั้ง Google Nexus 5 V 5.1.0 API 22 (1080x1920) ขึ้นไป
- 4.) ติดตั้ง Eclipse Mars
  - a. ติดตั้ง Eclipse IDE for Java Developers
  - b. ติดตั้ง Swing Designer Documentation
  - c. ติดตั้ง SWT Designer
  - d. ติดตั้ง SWT Designer Core
  - e. ติดตั้ง SWT Designer Documentation
  - f. ติดตั้ง SWT Designer SWT\_AWT Support
  - g. ติดตั้ง SWT Designer XWT Support (requires Eclipse WTP/WST)
  - h. ติดตั้ง WindowBuilder Core Documentation
  - i. ติดตั้ง WindowBuilder XML Core (requires Eclipse WTP/WST)
- 5.) ติดตั้ง StarUML

- a. ติดตั้ง StarUML Extension Manager ใน ส่วน ของ XMI (XML Metadata Interchange) Import and Export.

#### 4.4) ขั้นตอนการทำงานของเครื่องมือ

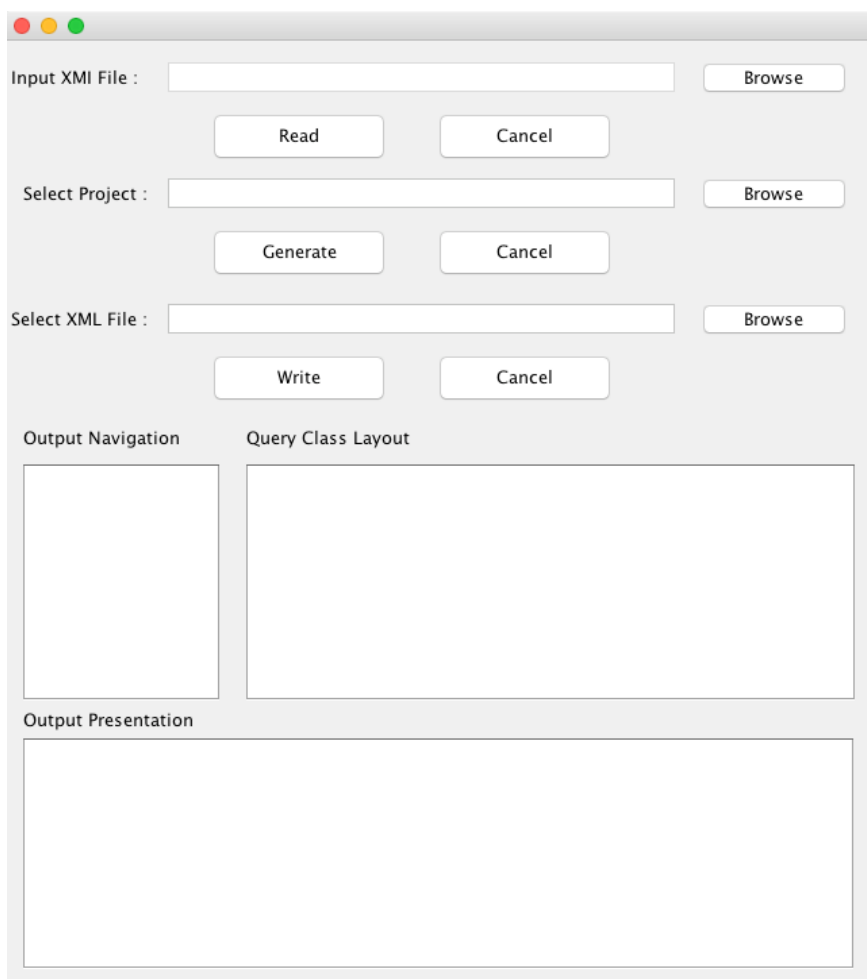
เครื่องมือมีเนื้อหาขั้นตอนการทำงานและภาพประกอบส่วนการทำงานดังต่อไปนี้

- 1) ผู้ใช้งานระบบต้องทำการสร้างแผนภาพนำทางหน้าต่างเพื่อกำหนดการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันด้วยเครื่องมือสตาร์ยูเอ็มแอล ดังรูปที่ 22



รูปที่ 22 ลักษณะการออกแบบแผนภาพนำทางหน้าต่างด้วยเครื่องมือสตาร์ยูเอ็มแอล

- 2) ผู้ใช้งานระบบต้องทำการส่งออกแผนภาพนำทางหน้าต่างผ่านเครื่องมือส่วนต่อขยายด้วยเครื่องมือสตาร์ยูเอ็มแอล เพื่อให้ได้ไฟล์รูปแบบเอ็กซ์เอ็มไอและนำเข้าไปสู่เครื่องมือสนับสนุน ดังรูปที่ 23



รูปที่ 23 ระบบเครื่องมือสนับสนุน

จุฬาลงกรณ์มหาวิทยาลัย

โดยผู้ใช้งานระบบจำเป็นต้องเลือกไฟล์เอ็กซ์เอ็มแอลเพื่อเป็นไฟล์นำเข้าเครื่องมือสนับสนุนนี้ เนื่องจากได้ทำการออกแบบระบบเครื่องมือสนับสนุนนี้ให้อ่านไฟล์เอ็กซ์เอ็มแอล เพื่อที่จะทราบรายละเอียดข้อมูลของแผนภาพนำทางหน้าต่าง วิธีการเข้าถึงไฟล์ ไฟล์จะผ่านเข้าช่องเลือกไฟล์เอ็กซ์เอ็มแอลซึ่งจำเป็นต้องทำการกดปุ่มอ่าน เพื่อทำการอ่านเนื้อหาไฟล์และทำการแปลงข้อมูลเป็นรูปแบบเจสัน ในงานวิจัยชิ้นนี้จะรูปแบบการทำงานการแปลงรูปแบบข้อมูลเอ็กซ์เอ็มแอลเป็นเจสันดังรูปที่ 24

```

public JSON getXMLformJSON(String Path)
{
    JSON json = null;
    try
    {
        File file = new File(Path);
        InputStream input = new FileInputStream(file);
        String xml = IOUtils.toString(input);
        json = new XMLSerializer().read(xml);
        System.out.println(json);
    }
    catch (Exception e)
    {
        // TODO: handle exception
        JOptionPane.showMessageDialog(null,
            e, "XMLToJSONHelper getXMLformJSON"
            , JOptionPane.ERROR_MESSAGE);
    }
    return json;
}
}

```

รูปที่ 24 รูปแบบการทำงานแปลงรูปแบบข้อมูลเอกซ์เอ็มแอลไปเจสัน

หลังจากใช้รูปแบบการทำงานแปลงรูปแบบข้อมูลเอกซ์เอ็มแอลไปเจสัน ซึ่งจะช่วยในการวิเคราะห์ทำงาน จะช่วยให้เข้าถึงรายละเอียดและองค์ประกอบของแอทริบิวต์ของวัตถุได้ง่ายและสะดวกขึ้น ตัวอย่างเช่นดังรูปที่ 25 และ รูปที่ 26

```

<?xml version="1.0" encoding="UTF-8"?>
<xml:XML xmlns:xm1="http://schema.omg.org/spec/UML/2.1" xmlns:xm2="http://schema.omg.org/spec/XMI/2.1">
  <xm1:Documentation exporter="StarUML" exporterVersion="2.0"/>
  <xm1:Model xmi:id="AAAAAFVWUvFsduofB=" xmi:type="uml:Model" name="RootModel">
    <packagedElement xmi:id="AAAAAFAFF+qBWK6M3Z8Ye=" name="Model" visibility="public" xmi:type="uml:Model">
      <packagedElement xmi:id="AAAAAFAFVedvksrSURI=" name="Login" visibility="public" xmi:type="uml:InstanceSpecification">
        <ownedMember xmi:id="AAAAAFAFVeq80TChsQ=" name="btnForgotPassword" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="button"/>
          </xm1:Extension>
        </ownedMember>
        <ownedMember xmi:id="AAAAAFAFVehTuTT9+8H=" name="btnRegister" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="button"/>
          </xm1:Extension>
        </ownedMember>
        <ownedMember xmi:id="AAAAAFAFVehzGzUbpwA=" name="btnLogin" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="button"/>
          </xm1:Extension>
        </ownedMember>
        <xm1:Extension extender="StarUML">
          <stereotype value="window"/>
        </xm1:Extension>
      </packagedElement>
      <packagedElement xmi:id="AAAAAFAFVeenPFzVQtmB=" name="Register" visibility="public" xmi:type="uml:InstanceSpecification">
        <ownedMember xmi:id="AAAAAFAFVeePTVl2K=" name="ClickButton" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="nav"/>
            <linkEnd1 value="AAAAAFAFVehTuTT9+8H="/>
            <linkEnd2 value="AAAAAFAFVeenPFzVQtmB="/>
          </xm1:Extension>
        </ownedMember>
        <ownedMember xmi:id="AAAAAFAFVeeqrTel97g=" name="Submit" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="button"/>
          </xm1:Extension>
        </ownedMember>
        <xm1:Extension extender="StarUML">
          <stereotype value="window"/>
        </xm1:Extension>
      </packagedElement>
      <packagedElement xmi:id="AAAAAFAFVeePsvzZyYos=" name="MainActivity" visibility="public" xmi:type="uml:InstanceSpecification">
        <ownedMember xmi:id="AAAAAFAFVeePsvzZyYos=" name="ClickButton" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="nav"/>
            <linkEnd1 value="AAAAAFAFVehzGzUbpwA="/>
            <linkEnd2 value="AAAAAFAFVeePsvzZyYos="/>
          </xm1:Extension>
        </ownedMember>
        <ownedMember xmi:id="AAAAAFAFVeeKnTf3szB=" name="btnProfileCarRegister" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="button"/>
          </xm1:Extension>
        </ownedMember>
        <ownedMember xmi:id="AAAAAFAFVeeT4VLC/r4=" name="btnLogout" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="button"/>
          </xm1:Extension>
        </ownedMember>
        <ownedMember xmi:id="AAAAAFAFVeevOhj2Ik=" name="btnSummaryProcessList" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="button"/>
          </xm1:Extension>
        </ownedMember>
        <ownedMember xmi:id="AAAAAFAFVeeWzhzRzRA=" name="btnProfileTodo" visibility="public" xmi:type="uml:InstanceSpecification">
          <xm1:Extension extender="StarUML">
            <stereotype value="button"/>
          </xm1:Extension>
        </ownedMember>
        <xm1:Extension extender="StarUML">

```

รูปที่ 25 เนื้อหาข้อมูลเอกซ์เอ็มแอลจากไฟล์เอกซ์เอ็มไอ





```

public JSONArray ReadJSON(JSONArray jsonArray)
{
    dataJSONArray = new JSONArray();
    matchDataJSONArray = new JSONArray();
    dataJSONArrayNotUseElement = new JSONArray();

    JSONArray newJSONArraySuccessFindNav = new JSONArray();
    JSONArray newJSONArraySuccessFindElement = new JSONArray();
    try
    {
        for (int i = 0; i < jsonArray.size(); i++)
        {
            JSONObject jsonObject = jsonArray.optJSONObject(i);
            String screenID = jsonObject.getString("@xmi:id");
            String screenName = jsonObject.getString("@name");
            JSONArray jsonArrayReaderElement = jsonObject.getJSONArray("ownedMember");
            System.out.println(jsonArrayReaderElement);
            JSONArray check = new JSONArray();
            check = FindNavigationFromComponent(jsonArrayReaderElement);
            JSONArray resultElementNotUsed = new JSONArray();
            resultElementNotUsed = FindElement(jsonArrayReaderElement, screenName);
            if (check.size() != 0)
            {
                newJSONArraySuccessFindNav = check;
            }
            if (resultElementNotUsed.size() != 0)
            {
                newJSONArraySuccessFindElement = resultElementNotUsed;
            }
        }
        FindMainComponentByNavPath(newJSONArraySuccessFindNav, jsonArray);
    }
    catch (Exception e)
    {
        // TODO: handle exception
        JOptionPane.showMessageDialog(null,
            e, "NavigationModelGenerator ReadJSON" ,
            JOptionPane.ERROR_MESSAGE);
    }
    FilterElementWithoutNav(newJSONArraySuccessFindElement, matchDataJSONArray);
    return matchDataJSONArray;
}

```

### รูปที่ 28 ภาพรวมการทำงานของแบบจำลองนำทาง

เพื่อที่จะทราบถึงเส้นทางกิจกรรมความสัมพันธ์การเชื่อมโยงระหว่างแบบจำลองนำทางหน้าต่างกับแผนภาพนำทางหน้าต่างซึ่งในงานวิจัยชิ้นนี้ได้ทำการออกแบบความสัมพันธ์ของแบบจำลองนำทางหน้าต่าง โดยเลือกใช้ลักษณะการเชื่อมโยงแบบลิงค์ (Link) ซึ่งจะสามารถนำเสนอความสัมพันธ์ที่เกิดขึ้นได้ระหว่างแผนภาพนำทางหน้าต่างกับแผนภาพนำทางหน้าต่างดังรูปที่ 28

```

public JSONArray FindNavigationFromComponent(JSONArray jsonArray)
{
    System.out.println(" ");
    for (int i = 0; i < jsonArray.size(); i++)
    {
        JSONObject object = jsonArray.optJSONObject(i);
        JSONObject getObject = object.getJSONObject("xmi:Extension");
        JSONObject getObject1 = getObject.getJSONObject("stereotype");
        String objectJsonString = getObject1.getString("@value");
        if (objectJsonString.equals("nav") )
        {
            System.out.println("Can Nav Find Path");
            dataJSONArray.add(GetLinkPath(getObject));
        }
        else
        {
            System.out.println("No Can't Nav Find Path");
        }
    }
    return dataJSONArray;
}

```

รูปที่ 29 รูปแบบการหารายละเอียดที่มีส่วนเกี่ยวข้องกับการเชื่อมโยงระหว่างแผนภาพนำทาง



หน้าต่าง

```

public JSONArray FindElement(JSONArray jsonArray, String ScreenName)
{
    for (int i = 0; i < jsonArray.size(); i++)
    {
        JSONObject object = jsonArray.optJSONObject(i);
        JSONObject getObject = object.getJSONObject("xmi:Extension");
        JSONObject getObject1 = getObject.getJSONObject("stereotype");
        String objectJsonString = getObject1.getString("@value");
        if (objectJsonString.equals("button") )
        {
            System.out.println("Yes Can Element Not Used");
            dataJSONArrayNotUseElement.add(GetNotUseElement(object, ScreenName));
        }
    }
    return dataJSONArrayNotUseElement;
}

```

รูปที่ 30 รูปแบบการหารายละเอียดที่ไม่มีส่วนเกี่ยวข้องกับการเชื่อมโยงระหว่างแผนภาพนำทาง

หน้าต่าง

เนื่องจากต้องการที่จะลดความซับซ้อนระหว่างองค์ประกอบที่ถูกเลือกในการเชื่อมโยงและไม่ถูกเลือกใช้ออกจากกัน จากรูปที่ 29 และ 30 ตามลำดับ เป็นการสกัดคัดเลือกองค์ประกอบที่ถูกเลือกใช้ในการเชื่อมโยงและองค์ประกอบที่ไม่ถูกเลือกใช้ออกจากกัน เพื่อเป็นประโยชน์ เพิ่มความสะดวกในการวิเคราะห์และการสร้างวัตถุ

```

public void FindMainComponentByNavPath(JSONArray jsonArray, JSONArray MainArray)
{
    try
    {
        for (int i = 0; i < jsonArray.size(); i++)
        {
            JSONObject object = jsonArray.optJSONObject(i);
            String RootMain = object.getString("Root Main");
            String NavEnd = object.getString("Nav End");
            JSONObject parentObject = FindRootMainComponent(RootMain, MainArray);
            JSONObject childObject = FindNavEndComponent(NavEnd, MainArray);
            JSONArray jsonArray2 = new JSONArray();
            jsonArray2.add(parentObject);
            jsonArray2.add(childObject);
            JSONObject object2 = new JSONObject();
            object2.put("Data", jsonArray2);
            matchDataJSONArray.add(object2);
        }
    }
    catch (Exception e)
    {
        // TODO: handle exception
        JOptionPane.showMessageDialog(null,
            e, "NavigationModelGenerator FindMainComponentByNavPath" ,
            JOptionPane.ERROR_MESSAGE);
    }
}

```

รูปที่ 31 รูปแบบการสกัดข้อมูลแผนภาพนำทางหน้าต่างมีส่วนเกี่ยวข้องกับการเชื่อมโยงระหว่าง  
แผนภาพนำทางหน้าต่าง

```

public JSONObject FindRootMainComponent(String string, JSONArray MainArray)
{
    JSONObject mainObject = new JSONObject();
    try
    {
        for (int i = 0; i < MainArray.size(); i++)
        {
            JSONObject mainJSONObject = MainArray.optJSONObject(i);
            String screenName = mainJSONObject.getString("@name");
            JSONArray mapData = mainJSONObject.getJSONArray("ownedMember");
            // Scan element In Screen.
            for (int j = 0; j < mapData.size(); j++)
            {
                JSONObject eachJSONObject = mapData.optJSONObject(j);
                String checkIDElement = eachJSONObject.getString("@xml:id");
                if (string.equals(checkIDElement))
                {
                    String elementName = eachJSONObject.getString("@name");
                    JSONObject getDetailElementName = eachJSONObject.getJSONObject("xml:Extension");
                    JSONObject getStereoTypeOfElement = getDetailElementName.getJSONObject("stereotype");
                    String typeOfElement = getStereoTypeOfElement.getString("@value");
                    // create json element
                    mainObject.put("RootScreen", screenName);
                    mainObject.put("Name", elementName);
                    mainObject.put("Element", typeOfElement);
                    mainObject.put("ElementID", checkIDElement);
                    break;
                }
            }
        }
    }
    catch (Exception e)
    {
        // TODO: handle exception
        JOptionPane.showMessageDialog(null,
            e, "NavigationModelGenerator FindRootMainComponent" ,
            JOptionPane.ERROR_MESSAGE);
    }
    return mainObject;
}

```

รูปที่ 32 รูปแบบการสกัดข้อมูลเฉพาะโดยเน้นที่แผนภาพนำทางหน้าต่าง

```

public JSONObject FindNavEndComponent(String string, JSONArray MainArray)
{
    JSONObject childObject = new JSONObject();
    try
    {
        System.out.println(MainArray);
        for (int i = 0; i < MainArray.size(); i++)
        {
            JSONObject mainJSONObject = MainArray.optJSONObject(i);
            String screenID = mainJSONObject.getString("@xml:id");
            String screenName = mainJSONObject.getString("@name");
            if (string.equals(screenID))
            {
                childObject.put("ChildScreen", screenName);
            }
        }
    }
    catch (Exception e)
    {
        // TODO: handle exception
        JOptionPane.showMessageDialog(null,
            e, "NavigationModelGenerator FindNavEndComponent",
            JOptionPane.ERROR_MESSAGE);
    }
    return childObject;
}

```

รูปที่ 33 รูปแบบการสกัดข้อมูลเฉพาะโดยเน้นที่แบบจำลองนำทางหน้าต่างปลายทาง



```

public void FilterElementWithOutNav(JSONArray ElementArray, JSONArray ScreenArray)
{
    JSONArray elementArray = ElementArray;
    for (int i = 0; i < ScreenArray.size(); i++)
    {
        JSONObject object = ScreenArray.optJSONObject(i);
        JSONArray dataJSONArray = object.getJSONArray("Data");
        JSONObject getJson = dataJSONArray.optJSONObject(0);
        String getElementId = getJson.getString("ElementID");
        for (int j = 0; j < elementArray.size(); j++)
        {
            JSONObject elementObject = elementArray.optJSONObject(j);
            String findElementId = elementObject.getString("ElementID");
            if (getElementId.equals(findElementId))
            {
                elementArray.remove(elementObject);
                break;
            }
        }
    }
    for (int i = 0; i < elementArray.size(); i++)
    {
        JSONObject jsonObject = new JSONObject();
        jsonObject.put("ChildScreen", "");
        JSONArray jsonArray2 = new JSONArray();
        jsonArray2.add(elementArray.opt(i));
        jsonArray2.add(jsonObject);
        JSONObject object2 = new JSONObject();
        object2.put("Data", jsonArray2);
        matchDataJSONArray.add(object2);
    }
}

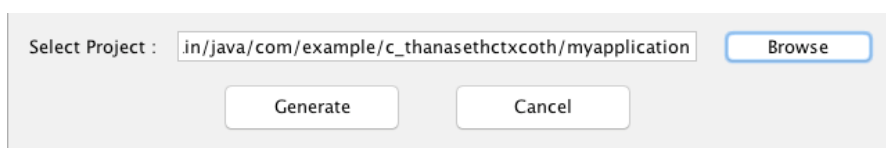
```

รูปที่ 34 ภาพรวมรูปแบบการสกัดรายละเอียดที่ไม่มีส่วนเกี่ยวกับการนำทาง

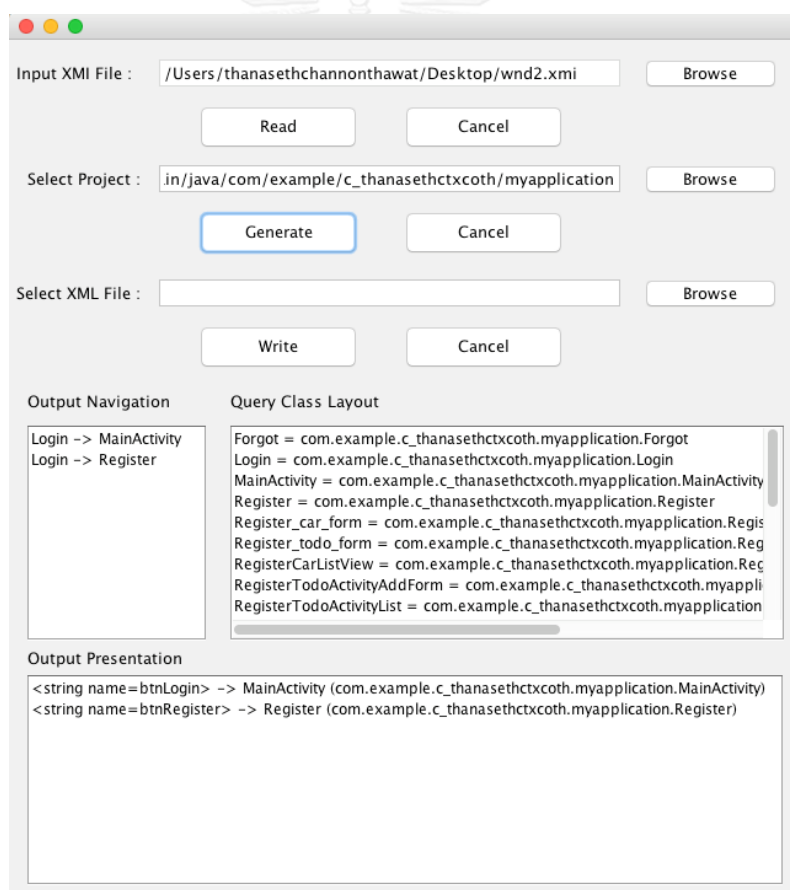
จากรูปที่ 31 32 และ 33 ตามลำดับ เป็นรูปแบบการพัฒนาที่สร้างขึ้นมาโดยให้ความมุ่งเน้นไปที่การสกัดข้อมูลแต่ละส่วนที่เกี่ยวข้องออกมาและสร้างเป็นแอร์เรย์แบบวัตถุไว้ อย่างเช่น เมื่อเราทราบแล้วว่าแผนภาพนำทางหน้าต่างอันไหนที่มีความสัมพันธ์กัน ขั้นตอนแรกแผนภาพนำทางหน้าต่างที่เป็นต้นทางจะถูกคัดเลือก สกัดข้อมูลและนำมาสร้างเป็นวัตถุใหม่ แล้วแผนภาพนำทางหน้าต่างที่เป็นปลายทางก็จะถูก คัดเลือก สกัดข้อมูลและนำมาสร้างเป็นวัตถุใหม่เช่นกัน ข้อดีของการทำแบบนี้เพื่อสามารถลดความซับซ้อนของแอสทริบิวท์ที่สำคัญหรือไม่สำคัญของขั้นตอนถัดไป จากรูปที่

34 เป็นการสกัดรายละเอียดข้อมูลและสร้างเป็นวัตถุเก็บไว้เหมือนกัน แต่ข้อดีของการสกัดข้อมูลที่ไม่เกี่ยวข้องกับการนำทางประเภทนี้คือจะช่วยสนับสนุนเพิ่มความสะดวกในการวิเคราะห์ในขั้นตอนถัดไป

- 5) วิเคราะห์เนื้อหาแบบจำลองทางความคิดเพื่อนำมาสร้างแบบจำลองนำเสนอ กระบวนการวิเคราะห์แบบจำลองนำเสนอจะมีลำดับขั้นตอนดังรูปที่ 35, 36, และ 37 ตามลำดับ



รูปที่ 35 การเลือกไฟล์โปรเจกต์แอนดรอยด์แอปพลิเคชัน



รูปที่ 36 ภาพรวมเครื่องมือสนับสนุนหลังจากกดปุ่มเจเนอเรท

```

public ArrayList<String> GetAllClass(String PathFile)
{
    File folder = new File(PathFile);
    File[] listOfFile = folder.listFiles();

    ArrayList<String> result = new ArrayList<String>();
    for (int i = 0; i < listOfFile.length; i++)
    {
        if (listOfFile[i].isFile())
        {
            result.add(listOfFile[i].getName());
        }
    }
    return result;
}

```

รูปที่ 37 รูปแบบวิธีการค้นหาคลาสในโปรเจกแอนดรอยด์แอปพลิเคชันที่เกี่ยวข้อง

จากรูปภาพที่ 35 36 และ 37 ตามลำดับ เป็นขั้นตอนเริ่มของการทำงานในส่วนการวิเคราะห์แบบจำลองนำเสนอ เริ่มต้นจากการเลือกโพลเดอรัลคลาสที่เกี่ยวข้องที่เป็นคลาสภาษาจาวาจากบนโปรเจกแอนดรอยด์แอปพลิเคชัน เพื่อทราบจำนวนคลาสที่เกี่ยวข้องภายในโปรเจกแอนดรอยด์แอปพลิเคชัน อันนี้ เมื่อทราบจำนวนไฟล์ที่เกี่ยวข้องหลังจากนั้นจะทำการสร้างเส้นทางการเข้าถึงไฟล์ของตัวเอง แต่ข้อจำกัดคือไฟล์นั้นต้องอยู่ภายใต้โปรเจกแอนดรอยด์แอปพลิเคชันเดียวกัน อย่างเช่น พบไฟล์แรกที่มีชื่อว่า Forgot จากนั้นเครื่องมือสนับสนุนจะทำการวิเคราะห์และเจนเนอเรทชื่อไฟล์ออกมาที่มีชื่อว่า com.example.c\_thanasethctxcoth.myapplication.Forgot

```

public void FindInitialScreen(JSONArray navigationJSON)
{
    for (int i = 0; i < navigationJSON.size(); i++)
    {
        JSONObject jsonObject = navigationJSON.optJSONObject(i);
        JSONArray jsonDetailNavigation = jsonObject.getJSONArray("Data");
        // find rootScreen
        JSONObject jsonObject3 = jsonDetailNavigation.optJSONObject(1);
        String childScreen = jsonObject3.getString("ChildScreen");
        // Separate Have ChildrenScreen or Not
        // For Identify in XML File
        if (!childScreen.equals(""))
        {
            CheckHaveChildrenScreenForUseScreen(jsonDetailNavigation);
        }
        else
        {
            CheckNotHaveChildrenScreenForNotUsingScreen(jsonDetailNavigation);
        }
    }
}

```

รูปที่ 38 รูปแบบค้นหาส่วนต่อประสานผู้ใช้ที่ถูกแสดงผลและไม่แสดงผลออกทางแอนดรอยด์แอปพลิเคชัน

```

public void CheckHaveChildernScreenForUseScreen(JSONArray dataScreen)
{
    JSONObject jsonObject = dataScreen.optJSONObject(0);
    String nameString = jsonObject.getString("Name");
    JSONObject jsonObject1 = dataScreen.optJSONObject(1);
    String childrenScreen = jsonObject1.getString("ChildScreen");
    JSONObject jsonObject2 = new JSONObject();
    jsonObject2.put("Element", nameString);
    jsonObject2.put("toScreen", childrenScreen);
    haveUsed.add(jsonObject2);
}

```

รูปที่ 39 รูปแบบค้นหาส่วนต่อประสานผู้ใช้ที่แสดงผลในการจำลองแอนดรอยด์แอปพลิเคชัน

```

public void CheckNotHaveChildrenScreenForNotUsingScreen(JSONArray dataScreen)
{
    JSONObject jsonObject = dataScreen.optJSONObject(0);
    String nameString = jsonObject.getString("Name");
    JSONObject jsonObject1 = dataScreen.optJSONObject(1);
    String childrenScreen = jsonObject1.getString("ChildScreen");
    JSONObject jsonObject2 = new JSONObject();
    jsonObject2.put("Element", nameString);
    jsonObject2.put("toScreen", childrenScreen);
    haventUsed.add(jsonObject2);
}

```

รูปที่ 40 รูปแบบค้นหาส่วนต่อประสานผู้ใช้ที่ไม่แสดงผลในการจำลองแอนดรอยด์แอปพลิเคชัน

```

public void InsertName(JSONArray jArray)
{
    for (int i = 0; i < haveUsed.size(); i++)
    {
        JSONObject jsonObject = haveUsed.optJSONObject(i);
        String string = jsonObject.getString("toScreen");
        for (int j = 0; j < jArray.size(); j++)
        {
            JSONObject listJson = jArray.optJSONObject(j);
            Iterator<?> keys = listJson.keys();
            String keyString = (String)keys.next();
            if (string.equals(keyString))
            {
                String detailString = listJson.getString(keyString);
                String string2 = jsonObject.getString("Element");
                JSONObject dObject = new JSONObject();
                dObject.put("screen", string);
                dObject.put("element", string2);
                dObject.put("data", detailString);
                dataJSONArray.add(dObject);
            }
        }
    }
}

```

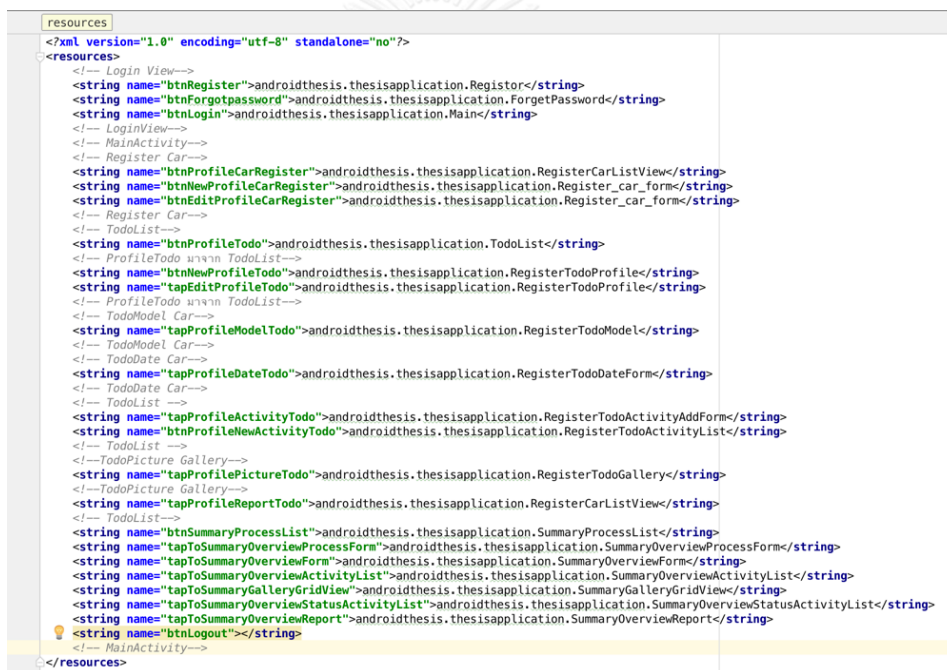
รูปที่ 41 รูปแบบการทวนสอบค่าข้อมูล

จากรูปที่ 38 39 และ 40 จะทำงานตามลำดับ ในขั้นตอนแรกจะทำการสกัดและคัดเลือกข้อมูลออกเป็นสองประเภทคือ 1) ส่วนต่อประสานผู้ใช้ต้นทางที่มีส่วนต่อประสานผู้ใช้ปลายทาง 2) ส่วนต่อประสานผู้ใช้ต้นทางที่ไม่มีส่วนต่อประสานผู้ใช้ปลายทาง ข้อดีของการสกัดและแยกข้อมูลสองประเภทนี้ คือสามารถช่วยลดความซับซ้อนและสนับสนุนการกำหนดชื่อไฟล์ให้แก่แต่ละแผนภาพนำทางหน้าต่าง จากนั้นจะนำข้อมูลที่มีข้อมูลทั้งต้นทางและปลายทางมากำหนดชื่อไฟล์ ทั้งนี้ชื่อไฟล์แต่ละชื่อ



ไฟล์ต้องสอดคล้องกับชื่อภายในข้อมูลต้นทางและข้อมูลปลายทาง จากนั้นมีรูปแบบตรวจสอบค่าข้อมูลก่อนที่จะทำการโอนข้อมูล เมื่อสามารถจับคู่ได้แล้วจะทำการถ่ายโอนข้อมูลไปวัตถุประสงค์และจะนำข้อมูลที่ได้นำมาเสนอออกทางเครื่องมือสนับสนุนดังรูปที่ 36 และ 41

- 6) การสร้างไฟล์ส่วนควบคุมและเตรียมส่วนควบคุมในโปรเจกต์โปรเจกต์แอนดรอยด์แอปพลิเคชัน ในขั้นตอนนี้ต้องทำในโปรเจกต์แอนดรอยด์แอปพลิเคชันที่ถูกพัฒนาด้วยซอฟต์แวร์ Android Studio ต้องการสร้างไฟล์เอกซ์เอ็มแอลในส่วนเพิ่มข้อมูลที่สามารถรองรับและรันทดสอบไฟล์นี้ได้ แอทริบิวต์ภายในไฟล์เอกซ์เอ็มแอลจะประกอบไปด้วยชื่อของการทำงานและองค์ประกอบที่สามารถช่วยให้นำทางไปยังอีกส่วนต่อประสานลำดับถัดไปได้ ดังรูปที่ 42



```

resources
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<resources>
  <!-- Login View-->
  <string name="btnRegister">androidthesis.thisisapplication.Register</string>
  <string name="btnForgotpassword">androidthesis.thisisapplication.ForgetPassword</string>
  <string name="btnLogin">androidthesis.thisisapplication.Main</string>
  <!-- LoginView-->
  <!-- MainActivity-->
  <!-- Register Car-->
  <string name="btnProfileCarRegister">androidthesis.thisisapplication.RegisterCarListView</string>
  <string name="btnNewProfileCarRegister">androidthesis.thisisapplication.Register_car_form</string>
  <string name="btnEditProfileCarRegister">androidthesis.thisisapplication.Register_car_form</string>
  <!-- Register Car-->
  <!-- TodoList-->
  <string name="btnProfileTodo">androidthesis.thisisapplication.TodoList</string>
  <!-- ProfileTodo มาจาก TodoList-->
  <string name="btnNewProfileTodo">androidthesis.thisisapplication.RegisterTodoProfile</string>
  <string name="tapEditProfileTodo">androidthesis.thisisapplication.RegisterTodoProfile</string>
  <!-- ProfileTodo มาจาก TodoList-->
  <!-- TodoModel Car-->
  <string name="tapProfileModelTodo">androidthesis.thisisapplication.RegisterTodoModel</string>
  <!-- TodoModel Car-->
  <!-- TodoDate Car-->
  <string name="tapProfileDateTodo">androidthesis.thisisapplication.RegisterTodoDateForm</string>
  <!-- TodoDate Car-->
  <!-- TodoList -->
  <string name="tapProfileActivityTodo">androidthesis.thisisapplication.RegisterTodoActivityAddForm</string>
  <string name="btnProfileNewActivityTodo">androidthesis.thisisapplication.RegisterTodoActivityList</string>
  <!-- TodoList -->
  <!--TodoPicture Gallery-->
  <string name="tapProfilePictureTodo">androidthesis.thisisapplication.RegisterTodoGallery</string>
  <!--TodoPicture Gallery-->
  <string name="tapProfileReportTodo">androidthesis.thisisapplication.RegisterCarListView</string>
  <!-- TodoList-->
  <string name="btnSummaryProcessList">androidthesis.thisisapplication.SummaryProcessList</string>
  <string name="tapToSummaryOverviewProcessForm">androidthesis.thisisapplication.SummaryOverviewProcessForm</string>
  <string name="tapToSummaryOverviewForm">androidthesis.thisisapplication.SummaryOverviewForm</string>
  <string name="tapToSummaryOverviewActivityList">androidthesis.thisisapplication.SummaryOverviewActivityList</string>
  <string name="tapToSummaryGalleryGridView">androidthesis.thisisapplication.SummaryGalleryGridView</string>
  <string name="tapToSummaryOverviewStatusActivityList">androidthesis.thisisapplication.SummaryOverviewStatusActivityList</string>
  <string name="tapToSummaryOverviewReport">androidthesis.thisisapplication.SummaryOverviewReport</string>
  <string name="btnLogout"></string>
  <!-- MainActivity-->
</resources>

```

รูปที่ 42 เนื้อหาภายในไฟล์เอกซ์เอ็มแอลส่วนควบคุม

```

public static Class<?> startIntentActivity(String name)
{
    String myClass = name;
    Class<?> classsss = null;

    if (myClass != null)
    {
        try
        {
            classsss = Class.forName(myClass);
        }
        catch (ClassNotFoundException e)
        {
            e.printStackTrace();
        }
    }
    return classsss;
}

```

รูปที่ 43 รูปแบบเนื้อหาหาคลาสตัวช่วยส่วนควบคุมการนำเสนอ

```

Boolean isAuthenPass;
isAuthenPass = authentication(emailEditText.getText().toString(), passwordEditText.getText().toString());

if (isAuthenPass)
{
    // IntentHelper
    String myString = getResources().getString(R.string.btnLogin);
    if (myString != null)
    {
        Class<?> myClass = null;
        myClass = IntentHelper.startIntentActivity(myString);
        Intent i = new Intent(Login.this, myClass);
        startActivity(i);
    }
}
else
{
}
}

```

รูปที่ 44 รูปแบบการนำคลาสตัวช่วยส่วนควบคุมการนำเสนอไปประยุกต์ใช้

จากรูปที่ 43 และ 44 เป็นตัวอย่างการพัฒนาที่นำไปใช้พัฒนาในแต่ละส่วนของส่วนควบคุมการนำเสนอ โดยที่รูปที่ 43 เป็นการพัฒนารูปแบบคลาสตัวช่วยสนับสนุนที่ไม่ว่าจะเป็นแอนดรอยด์วิเทจประเภทกดปุ่มหรือการแตะ ก็จะสามารถอ้างอิงถึงคลาสนี้และเอาเป็นตัวช่วยสนับสนุนในการนำทางไปอีกส่วนต่อประสานได้ รูปที่ 44 เป็นตัวอย่างที่นำคลาสตัวช่วยมาพัฒนาในไฟล์จาวาที่รับผิดชอบในส่วนการพัฒนาโปรแกรมของแต่ละหน้าจอส่วนต่อประสานผู้ใช้

- 7) วิเคราะห์เนื้อหาแบบจำลองนำทางเข้ากับเนื้อหาแบบจำลองนำเสนอเพื่อเตรียมการแปลงข้อมูลในส่วนควบคุมการแสดงผลของแอนดรอยด์แอปพลิเคชัน ดังรูปที่ 45 และ

46



รูปที่ 45 การเลือกไฟล์ส่วนควบคุมเพื่อนำมาวิเคราะห์

```

private void DomXMLParsing(String PathFile, JSONArray resultPresentation)
{
    JSONArray presentationJSONArray = resultPresentation;

    try
    {
        // DOM Parser
        File fileXML = new File(filepath);
        DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
        Document doc = docBuilder.parse(fileXML);
        doc.getDocumentElement().normalize();

        // Read Resource
        // Get original Document element
        System.out.println("Root element : " + doc.getDocumentElement().getNodeName());

        NodeList nList = doc.getElementsByTagName("string");
        // Loop for Find Tag <string >
        for (int i = 0; i < nList.getLength(); i++)
        {
            // Read Tag <string >
            org.w3c.dom.Node nNode = nList.item(i);
            System.out.println("\nCurrent Element : " + nNode.getNodeName());
            if (nNode.getNodeType() == Node.ELEMENT_NODE)
            {
                // get btnRegister
                // Update <string name='btnRegister'> ..this data... </string> attribute
                // Check <string name=""> attribute 'name' is same result presentation data
                Element eElement = (Element) nNode;
                System.out.println("\nElement no : " + eElement.getAttribute("name"));

                for (int j = 0; j < presentationJSONArray.size(); j++)
                {
                    JSONObject object = presentationJSONArray.optJSONObject(j);
                    String element = object.getString("element");
                    System.out.println("\nName : " + element);
                    if (eElement.getAttribute("name").equals(element))
                    {
                        System.out.println("\nMatch");
                        String data = object.getString("data");
                        eElement.setTextContent(data);
                        // remove if already insert data
                        presentationJSONArray.remove(object);
                    }
                }
            }
        }
    }
}

```

รูปที่ 46 รูปแบบการอ่านและเขียนไฟล์เอกซ์เอ็มแอล

จากรูปที่ 45 และ 46 เป็นกระบวนการเลือกไฟล์เอกซ์เอ็มแอลเพื่อที่จะทำการค้นหาอ่านแต่ละแท็ก เพื่อที่จะทราบรายละเอียดแต่ละแอทริบิวต์ เมื่อทราบรายละเอียดแต่ละแอทริบิวต์แล้วเราจะทำการค้นหาแอทริบิวต์ที่มีเนื้อหาซ้ำกันกับชุดข้อมูลที่ถูกนำเข้ามา หลังจากนั้นจะใช้หลักการ DOM Parsing ในการเขียนข้อมูลลงในไฟล์เอกซ์เอ็มแอลเมื่อทำการเขียนข้อมูลครบแล้วจะทำการบันทึกข้อมูล

## บทที่ 5

### การทดสอบและการวิเคราะห์ผล

#### 5.1) วัตถุประสงค์ของการทดสอบ

วัตถุประสงค์ของการทดสอบระบบ เพื่อเป็นแนวทางในการสนับสนุนการสร้างต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติจากแผนภาพนำทางหน้าต่างที่ถูกออกแบบและพัฒนาเครื่องมือช่วยสนับสนุนแนวทางจากในบทที่ 4 โดยเนื้อหาประกอบไปด้วยการทดสอบระบบ ตั้งแต่การออกแบบแผนภาพนำทางหน้าต่างการนำข้อมูลไปสกัด การนำเข้าของแบบจำลองแผนภาพนำทางหน้าต่าง การวิเคราะห์ผลของโมเดลนำทาง การวิเคราะห์ผลของโมเดลนำเสนอ การพาร์ทข้อมูลตลอดจนผลลัพธ์ของหน้าส่วนต่อประสานผู้ใช้บนแอนดรอยด์แอปพลิเคชัน ผลลัพธ์สุดท้ายของการทดลองทดสอบจำลองรันแอนดรอยด์แอปพลิเคชันที่ถูกต้องและครบถ้วนตามความต้องการ ที่มีการแสดงผลข้อมูลตรงตามที่ได้ออกแบบไว้ถูกต้องครบถ้วนสมบูรณ์

#### 5.2) การทดสอบระบบ

หลักการทดสอบระบบที่นำมาใช้ทดสอบแนวทางการสร้างต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติจากแผนภาพนำทางหน้าต่างคือ การทดสอบแบบกล่องดำ (Black Box Testing) ซึ่งจะเป็นการทดสอบแบบที่จะดูผลลัพธ์ที่ได้ออกมา นั้นถูกต้องตามที่คาดหวังเมื่อเทียบกับตามความต้องการ โดยจะทำการทดสอบแต่ละในระบบจะมีรายการดังต่อไปนี้

##### 5.2.1) การทดสอบการนำเข้าข้อมูลเข้าเครื่องมือสนับสนุนด้วยรูปแบบไฟล์เอ็กซ์เอ็มแอล

การทดสอบการนำเข้ารูปแบบไฟล์เอ็กซ์เอ็มแอล มีวัตถุประสงค์เพื่อนำข้อมูลเข้าเครื่องมือสนับสนุน หลังจากนั้นเครื่องมือสนับสนุนจะนำข้อมูลดังกล่าวไปวิเคราะห์เพื่อจำลองต้นแบบแอนดรอยด์แอปพลิเคชันได้อย่างถูกต้อง

##### 5.2.2) การทดสอบเครื่องมือการแปลงข้อมูลจากรูปแบบเอ็กซ์เอ็มแอลไปรูปแบบเจสัน

การทดสอบผลการแปลงรูปแบบข้อมูลจากเอ็กซ์เอ็มแอลไปยังเจสัน มีวัตถุประสงค์เพื่อใช้รับส่งข้อมูลระหว่างโมเดกับโมเดลภายในเครื่องมือสนับสนุนได้อย่างถูกต้อง

##### 5.2.3) การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองทางความคิด

การทดสอบผลการสร้างโมเดลแบบจำลองนำทาง มีวัตถุประสงค์เพื่อใช้สร้างเนื้อหาการนำทางไปยังโมเดลแบบจำลองนำทางและจะถูกนำไปแสดงผลบนส่วนต่อประสานผู้ใช้จำลองแอนดรอยด์แอปพลิเคชันได้อย่างถูกต้อง

#### 5.2.4) การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำทาง

การทดสอบผลการสร้างโมเดลแบบจำลองนำทาง มีวัตถุประสงค์เพื่อใช้สร้างเนื้อหาการนำทางไปยังโมเดลแบบจำลองนำทางและจะถูกนำไปแสดงผลบนส่วนต่อประสานแอนดรอยด์แอปพลิเคชันได้อย่างถูกต้อง

#### 5.2.5) การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำเสนอ

การทดสอบผลการสร้างโมเดลแบบจำลองนำเสนอ มีวัตถุประสงค์เพื่อที่ทำการค้นหาส่วนประกอบที่เกี่ยวข้องกับการแสดงผล รวมถึงการควบคุมแม่แบบเค้าร่างส่วนต่อประสานผู้ใช้ที่จะถูกแสดงผลออกทางส่วนต่อประสานแอนดรอยด์แอปพลิเคชันต้องทำงานได้ถูกต้อง

#### 5.2.6) การทดสอบเครื่องมือการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติ

การทดสอบผลการจำลองต้นแบบแอนดรอยด์แอปพลิเคชัน มีวัตถุประสงค์เพื่อทำการทดสอบการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติ

วัตถุประสงค์ของการทดสอบเพื่อเป็นการยืนยันความถูกต้องของการทำงานของเครื่องมือสนับสนุน โดยการทดสอบจะกระทำตามรายการดังต่อไปนี้

ตารางที่ 14 การทดสอบการนำข้อมูลเข้าเครื่องมือสนับสนุนด้วยไฟล์รูปแบบเอ็กซ์เอ็มไอ

รหัส	การทดสอบ	ผลที่คาดหวัง	ผลลัพธ์
TC01	การนำไฟล์เข้าในรูปแบบเอ็กซ์เอ็มไอ	เครื่องมือสนับสนุนจะต้องสามารถเลือกไฟล์เอ็กซ์เอ็มไอได้	ถูกต้อง
TC02	ตรวจสอบว่าไฟล์เอ็กซ์เอ็มไอสามารถเข้าถึงและอ่านได้	เครื่องมือสนับสนุนจะต้องสามารถเข้าถึงและสามารถอ่านไฟล์เอ็กซ์เอ็มไอได้	ถูกต้อง
TC03	ตรวจสอบว่าไม่สามารถอ่านไฟล์ผิดประเภทได้	เครื่องมือสนับสนุนจะต้องมีข้อความแจ้งเตือน	ถูกต้อง

ตารางที่ 15 การทดสอบเครื่องมือการแปลงข้อมูลจากรูปแบบเอ็กซ์เอ็มแอลไปรูปแบบเจสัน

รหัส	การทดสอบ	ผลที่คาดหวัง	ผลลัพธ์
TC04	ทำการแปลงค่าข้อมูลนำเข้าจากรูปแบบเอ็กซ์เอ็มแอลไปยังรูปแบบเจสัน	เครื่องมือสนับสนุนจะสามารถทำการแปลงรูปแบบข้อมูลให้เป็นเจสันได้	ถูกต้อง

ตารางที่ 16 การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองทางความคิด

รหัส	การทดสอบ	ผลที่คาดหวัง	ผลลัพธ์
TC05	สกัดข้อมูลที่ได้จากการแปลงเจสันส่วนที่ไม่สำคัญทิ้ง เพื่อสร้างโมเดลจำลองทางความคิด	เครื่องสนับสนุนสามารถจะรับรู้ถึงข้อมูลเจสันที่ได้ และจะยังคงสัมพันธ์ข้อมูลเดิมอยู่	ถูกต้อง
TC06	สกัดข้อมูลที่ไม่มีความสัมพันธ์เกี่ยวกับแผนภาพนำทางหน้าต่าง	เครื่องมือสนับสนุนจะสามารถแสดงข้อความแจ้งเตือนได้	ถูกต้อง

ตารางที่ 17 การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำทาง

รหัส	การทดสอบ	ผลที่คาดหวัง	ผลลัพธ์
TC07	สกัดข้อมูลโมเดลจำลองทางความคิดเพื่อหาความสัมพันธ์ระหว่างแผนภาพนำทางหน้าต่าง	เครื่องมือสนับสนุนจะต้องสามารถหาความสัมพันธ์ที่เกิดขึ้นระหว่างแผนภาพนำทางหน้าต่างได้	ถูกต้อง
TC08	สกัดข้อมูลที่ไม่เกี่ยวข้องกับความสัมพันธ์ระหว่างแผนภาพนำทางหน้าต่างได้	เครื่องมือสนับสนุนจะต้องทราบรายละเอียดความสัมพันธ์ที่ไม่เกิดขึ้นระหว่างหน้าจอส่วนต่อ	ถูกต้อง

		ประสานผู้ใช้ได้	
TC09	สกรีนข้อมูลโมเดลจำลองทางความคิดเพื่อหาความสัมพันธ์ระหว่างแผนภาพนำทางหน้าต่างได้	เครื่องมือสนับสนุนสามารถสร้างความสัมพันธ์ที่เกิดจากแผนภาพนำทางหน้าต่างที่ลึกไม่เกิน 2 ระดับได้	ถูกต้อง
TC10	สกรีนข้อมูลโมเดลจำลองทางความคิดเพื่อหาความสัมพันธ์ระหว่างแผนภาพนำทางหน้าต่าง	เครื่องมือสนับสนุนสามารถสร้างความสัมพันธ์ที่เกิดจากแผนภาพนำทางหน้าต่างที่กว้างไม่เกิน 2 ระดับได้	ถูกต้อง

ตารางที่ 18 การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำเสนอ

รหัส	การทดสอบ	ผลที่คาดหวัง	ผลลัพธ์
TC11	ตรวจสอบว่าสามารถค้นหาคลาสแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้ในแฟ้มแพ็คเกจของไฟล์แอนดรอยด์แอปพลิเคชันเมื่อรู้สถานที่เก็บไฟล์ได้	เครื่องมือสำหรับสนับสนุนการค้นหาไฟล์คลาสในแฟ้มแพ็คเกจแอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC12	ตรวจสอบว่าสามารถเรียกหาชื่อคลาสจากแฟ้มข้อมูลแพ็คเกจแอนดรอยด์แอปพลิเคชันได้	เครื่องมือสนับสนุนสามารถค้นหาชื่อคลาสจากแฟ้มข้อมูลแพ็คเกจแอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC13	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส login ได้	เมื่อทำการจำลองส่วนต่อประสานคลาส login ที่อยู่ในแฟ้มแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถ	ถูกต้อง

		แสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	
TC14	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส register ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส register ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC15	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส forgot ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส forgot ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC16	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส mainactivity ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส mainactivity ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC17	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส register_car_form ที่อยู่	ถูกต้อง



	register_car_form ได้	ในแพ็คเกจแอนดรอยด์ แอปพลิเคชันจะสามารถ แสดงรายละเอียด ส่วนประกอบย่อยได้อย่าง ถูกต้องและครบถ้วน	
TC18	หน้าส่วนต่อประสานผู้ใช้จะ สามารถระบุถึงการแสดงผล ส่วนประกอบย่อยของคลาส register_todo_form ได้	เมื่อทำการจำลองส่วนต่อ ประสาน คลาส ผู้ใช้ register_todo_form ที่ อยู่ในแพ็คเกจแอนดรอยด์ แอปพลิเคชันจะสามารถ แสดงรายละเอียด ส่วนประกอบย่อยได้อย่าง ถูกต้องและครบถ้วน	ถูกต้อง
TC19	หน้าส่วนต่อประสานผู้ใช้จะ สามารถระบุถึงการแสดงผล ส่วนประกอบย่อยของคลาส registercarlistivew ได้	เมื่อทำการจำลองส่วนต่อ ประสาน ผู้ใช้ คลาส registercarlistivew ที่อยู่ ในแพ็คเกจแอนดรอยด์ แอปพลิเคชันจะสามารถ แสดงรายละเอียด ส่วนประกอบย่อยได้อย่าง ถูกต้องและครบถ้วน	ถูกต้อง
TC20	หน้าส่วนต่อประสานผู้ใช้จะ สามารถระบุถึงการแสดงผล ส่วนประกอบย่อยของคลาส registertodoactivityaddfor m ได้	เมื่อทำการจำลองส่วนต่อ ประสาน ผู้ใช้ คลาส registertodoactivityadd form ที่อยู่ในแพ็คเกจแอน ดรอยด์แอปพลิเคชันจะ สามารถแสดงรายละเอียด ส่วนประกอบย่อยได้อย่าง ถูกต้องและครบถ้วน	ถูกต้อง

TC21	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส registertodoactivitylist ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส registertodoactivitylist ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC22	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส registertododateform ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส registertododateform ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC23	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส registertodogallery ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส registertodogallery ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC24	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส registertodomodel ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส registertodomodel ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียด	ถูกต้อง

		ส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	
TC25	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส registertodoprofile ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส registertodoprofile ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC26	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส summarygallerycustomcolumn ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส summarygallerycustomcolumn ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC27	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส summarygallerygridview ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส summarygallerygridview ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC28	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผล	เมื่อทำการจำลองส่วนต่อ	ถูกต้อง

	ส่วนประกอบย่อยของคลาส summaryoverviewactivitylist ได้	summaryoverviewactivitylist ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	
TC29	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส summaryoverviewform ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส summaryoverviewform ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC30	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส summaryoverviewprocessform ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส summaryoverviewprocessform ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC31	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส summaryoverviewreport ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส summaryoverviewreport ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียด	ถูกต้อง

		ส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	
TC32	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส summaryoverviewstatusactivity ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส summaryoverviewstatusactivity ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC33	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส summaryprocesslist ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส summaryprocesslist ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง
TC34	หน้าส่วนต่อประสานผู้ใช้จะสามารถระบุถึงการแสดงผลส่วนประกอบย่อยของคลาส todolist ได้	เมื่อทำการจำลองส่วนต่อประสานผู้ใช้คลาส todolist ที่อยู่ในแพ็คเกจแอนดรอยด์แอปพลิเคชันจะสามารถแสดงรายละเอียดส่วนประกอบย่อยได้อย่างถูกต้องและครบถ้วน	ถูกต้อง

ตารางที่ 19 การทดสอบเครื่องมือการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติ

รหัส	การทดสอบ	ผลที่คาดหวัง	ผลลัพธ์
TC35	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.Register ให้กับคุณลักษณะ btnRegister	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชันและสามารถผลออกทางหน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC36	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.Forgot ให้กับคุณลักษณะ btnForgotpassword	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชันและสามารถผลออกทางหน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC37	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.Main ให้กับคุณลักษณะ btnlogin	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชันและสามารถผลออกทางหน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC38	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.RegisterCarListView ให้กับคุณลักษณะ btnProfileCarRegister	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชันและสามารถผลออกทางหน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC39	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplica	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชันและสามารถผลออกทาง	ถูกต้อง

	tion.Register_car_form ให้ กับ คุณ ลั ก ษ ณะ btnNewProfileCarRegister	หน้าจอลงส่วนต่อ ประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	
TC40	หน้าแสดงส่วนต่อประสานผู้ ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplica tion.Register_car_form ให้ กับ คุณ ลั ก ษ ณะ btnEditProfileCarRegister	สามารถจำลองต้นแบบ แอนดรอยด์แอปพลิเคชัน และสามารถผลออกทาง หน้าจอลงส่วนต่อ ประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	ถูกต้อง
TC41	หน้าแสดงส่วนต่อประสานผู้ ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplica tion.TODOList ให้ กับ คุณลักษณะ btnProfileTodo	สามารถจำลองต้นแบบ แอนดรอยด์แอปพลิเคชัน และสามารถผลออกทาง หน้าจอลงส่วนต่อ ประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	ถูกต้อง
TC42	หน้าแสดงส่วนต่อประสานผู้ ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplica tion.RegisterTodoProfile ให้ กับ คุณ ลั ก ษ ณะ btnNewProfileTodo	สามารถจำลองต้นแบบ แอนดรอยด์แอปพลิเคชัน และสามารถผลออกทาง หน้าจอลงส่วนต่อ ประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	ถูกต้อง
TC43	หน้าแสดงส่วนต่อประสานผู้ ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplica tion.RegisterTodoProfile ให้ กับ คุณ ลั ก ษ ณะ tapEditProfileTodo	สามารถจำลองต้นแบบ แอนดรอยด์แอปพลิเคชัน และสามารถผลออกทาง หน้าจอลงส่วนต่อ ประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	ถูกต้อง
TC44	หน้าแสดงส่วนต่อประสานผู้ ใช้ได้หลังจากกำหนดค่า	สามารถจำลองต้นแบบ แอนดรอยด์แอปพลิเคชัน	ถูกต้อง

	androidthesis.thesisapplication.RegisterToDoModel ให้ กับ คุณ ลัก ษ ณ ะ tapProfileModelToDo	และสามารถผลออกทาง หน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	
TC45	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.RegisterToDoDateForm ให้ กับ คุณ ลัก ษ ณ ะ tapProfileDateToDo	สามารถจำลองต้นแบบ แอนดรอยด์แอปพลิเคชัน และสามารถผลออกทาง หน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	ถูกต้อง
TC46	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.RegisterToDoActivityAddForm ให้ กับ คุณ ลัก ษ ณ ะ tapProfileActivityToDo	สามารถจำลองต้นแบบ แอนดรอยด์แอปพลิเคชัน และสามารถผลออกทาง หน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	ถูกต้อง
TC47	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.RegisterToDoActivityList ให้ กับ คุณ ลัก ษ ณ ะ btnProfileNameActivityToDo	สามารถจำลองต้นแบบ แอนดรอยด์แอปพลิเคชัน และสามารถผลออกทาง หน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	ถูกต้อง
TC48	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.RegisterToDoGallery ให้ กับ คุณ ลัก ษ ณ ะ tapProfilePictureToDo	สามารถจำลองต้นแบบ แอนดรอยด์แอปพลิเคชัน และสามารถผลออกทาง หน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์ แอปพลิเคชันได้	ถูกต้อง



TC49	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.RegisterCarListView ให้กับคุณลักษณะ tapProfileReportTodo	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชัน และสามารถผลออกทางหน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC50	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.SummaryProcessList ให้กับคุณลักษณะ btnSummaryProcessList	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชัน และสามารถผลออกทางหน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC51	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.SummaryOverviewProcessForm ให้กับคุณลักษณะ tapToSummaryOverviewProcessForm	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชัน และสามารถผลออกทางหน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC52	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.SummaryOverviewForm ให้กับคุณลักษณะ tapToSummaryOverviewForm	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชัน และสามารถผลออกทางหน้าจอจำลองส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC53	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisaplica	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชัน และสามารถผลออกทาง	ถูกต้อง

	tion.SummaryOverviewActivityList ให้กับคุณลักษณะ tapToSummaryOverviewActivityList	หน้าจอลงส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	
TC54	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.SummaryGalleryGridView ให้กับคุณลักษณะ tapToSummaryGalleryGridView	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชัน และสามารถผลออกทางหน้าจอลงส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC55	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.SummaryOverviewStatusActivityList ให้กับคุณลักษณะ tapToSummaryOverviewStatusActivityList	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชัน และสามารถผลออกทางหน้าจอลงส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง
TC56	หน้าแสดงส่วนต่อประสานผู้ใช้ได้หลังจากกำหนดค่า androidthesis.thesisapplication.SummaryOverviewReport ให้กับคุณลักษณะ tapToSummaryOverviewReport	สามารถจำลองต้นแบบแอนดรอยด์แอปพลิเคชัน และสามารถผลออกทางหน้าจอลงส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้	ถูกต้อง

### 5.3) สรุปผลการทดลอง

การทดลองในขั้นตอนที่ 5.2 ผู้วิจัยจะขอสรุปผลการทดสอบดังต่อไปนี้

#### 1) การทดสอบการนำข้อมูลเข้าเครื่องมือสนับสนุนด้วยไฟล์รูปแบบเอ็กซ์เอ็มไอ

สามารถนำเข้าและอ่านไฟล์เอ็กซ์เอ็มไอได้อย่างถูกต้อง ทำให้สามารถนำเนื้อหาข้อมูลไฟล์เอ็กซ์เอ็มไอไปใช้ในขั้นตอนการแปลงข้อมูลและสกัดข้อมูลได้ถูกต้องและครบถ้วน

#### 2) การทดสอบเครื่องมือการแปลงรูปแบบข้อมูลจากเอ็กซ์เอ็มแอลไปเจสัน

สามารถทำการแปลงรูปแบบข้อมูลเอ็กซ์เอ็มแอลไปยังข้อมูลเจสันได้ถูกต้องและครบถ้วน ทำให้เครื่องมือสนับสนุนเกิดความสะดวกในการเข้าถึงข้อมูลและนำไปใช้ในขั้นตอนถัดไป

#### 3) การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองทางความคิด

สามารถทำการสกัดข้อมูลดิบรูปแบบเจสันและสร้างโมเดลแบบจำลองทางความคิดขึ้นได้ถูกต้องและครบถ้วน และเพื่อให้เป็นโมเดลต้นแบบของโมเดลแบบจำลองนำทางและโมเดลแบบจำลองนำเสนอ

#### 4) การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำทาง

สามารถทำการสกัดโมเดลแบบจำลองทางความคิดเพื่อสร้างความสัมพันธ์เส้นทางกิจกรรมระหว่างแผนภาพนำทางหน้าต่างได้ถูกต้องและครบถ้วน

#### 5) การทดสอบเครื่องมือการสกัดข้อมูลเพื่อสร้างโมเดลแบบจำลองนำเสนอ

สามารถทำการค้นหาคลาสแม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชันได้อย่างถูกต้องและครบถ้วนตามความต้องการที่ได้ถูกออกแบบไว้รวมถึงการกำหนดรายละเอียดส่วนต่อประสานย่อยแสดงผลได้อย่างถูกต้องและครบถ้วนเช่นกัน

#### 6) การทดสอบเครื่องมือการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติ

สามารถทำการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติได้อย่างถูกต้องและครบถ้วนรวมถึงการสร้างส่วนควบคุมในขั้นตอนพาร์ทข้อมูลก็ถูกต้องและครบถ้วนด้วยเช่นกัน

## บทที่ 6

### สรุปผลการวิจัย

บทที่ 6 กล่าวถึงผลสรุปงานวิจัย ข้อจำกัดของงานวิจัยชิ้นนี้ งานวิจัยในอนาคตและผล งานตีพิมพ์จากวิทยานิพนธ์ โดยแต่ละส่วนที่กล่าวมานั้นมีรายละเอียดดังต่อไปนี้

#### 6.1) สรุปผลการวิจัย

การพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองของต้นแบบแอนดรอยด์แอปพลิเคชันจากแผนภาพ นำทางหน้าต่างได้ถูกแบบขึ้นเพื่อเป็นอีกหนึ่งแนวทางเพื่อนำเสนอความต้องการออกทางส่วนต่อ ประสานผู้ใช้ ทำให้ทราบถึงความต้องการและชัดเจนมากขึ้น แต่บางครั้งหลังจากสร้างส่วนต่อ ประสานผู้ใช้แอนดรอยด์แอปพลิเคชันนั้นในบางครั้งไม่ได้ถูกนำมาใช้ในซอฟต์แวร์นั้นหรือไม่ได้ถูก นำมาใช้ประโยชน์ต่อและการพัฒนาแอนดรอยด์แอปพลิเคชันนั้นเป็นขั้นตอนที่ใช้เวลารวมถึง สิ้นเปลืองทรัพยากร ซึ่งในงานวิจัยนี้ได้นำเสนอการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันแบบ อัตโนมัติด้วยเครื่องมือสตาร์ยูเอ็มแอล จากนั้นนำข้อมูลเข้าสู่การสกัดข้อมูลหาความสัมพันธ์ที่เกิดขึ้น โดยงานวิจัยนี้ผู้วิจัยได้ออกแบบและพัฒนาเครื่องมือสนับสนุนขึ้นมาเพื่อทำการสกัด การหา ความสัมพันธ์ การค้นหาคลาสส่วนต่อประสานผู้ใช้แอนดรอยด์แอปพลิเคชัน การสร้างส่วนควบคุม เพื่อให้สามารถสร้างต้นแบบแอนดรอยด์แอปพลิเคชันแบบอัตโนมัติขึ้นได้ จากที่กล่าวมาเครื่องมือ สนับสนุนงานวิจัยได้ถูกออกแบบและใช้วิธีการในการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันเพื่อช่วย ในการปรับปรุงกระบวนการพัฒนาซอฟต์แวร์ ช่วยลดเวลาในการพัฒนา และช่วยส่งผลให้ องค์กรประกอบภาพรวมของโครงการพัฒนาซอฟต์แวร์ให้มีความชัดเจนขึ้นและไปในทิศทางที่ดี

#### 6.2) ข้อจำกัดของงานวิจัย

- 1) เครื่องมือสนับสนุนที่ใช้สร้างแผนภาพนำทางหน้าต่างต้องสร้างจากเครื่องมือสตาร์ยูเอ็มแอลเท่านั้น
- 2) การนำไฟล์เอ็กซ์เอ็มไอต้องเป็นไฟล์ที่ถูกส่งออกมาจากเครื่องมือสตาร์ยูเอ็มแอลที่ถูกติดตั้ง ส่วนต่อขยายเอ็กซ์เอ็มไอเท่านั้น
- 3) ไฟล์เอ็กซ์เอ็มไอที่นำเข้าเนื้อหาองค์ประกอบภายในต้องเป็นเอ็กซ์เอ็มแอลเท่านั้น
- 3) การนำเข้าคลาสของแอนดรอยด์แอปพลิเคชันและที่อยู่ของโปรเจกแอนดรอยด์แอปพลิเคชันต้องเป็นโปรเจกที่ถูกสร้างด้วยเครื่องมือ Android Studio เท่านั้น

4) การนำเข้าไฟล์เอ็กซ์เอ็มแอลส่วนควบคุมการแสดงผล ไฟล์เอ็กซ์เอ็มแอลต้องถูกสร้างอยู่ในส่วนที่เครื่องมือ Android Studio สามารถเข้าถึงและอ่านไฟล์ได้

5) ไฟล์เอ็กซ์เอ็มแอลส่วนควบคุมการแสดงผล รูปแบบเนื้อหาต้องถูกกำหนดให้ตรงกับเงื่อนไขรูปแบบที่กำหนดไว้เท่านั้น

6) ไฟล์จาวาที่เป็นส่วนพัฒนาการทำงานของคลาสแอนดรอยด์ในส่วนในส่วนการพัฒนาส่วนประกอบย่อยที่สามารถนำไปยังคลาสอื่นได้ ต้องพัฒนาให้อยู่ในรูปแบบที่กำหนดเท่านั้น

7) แม่แบบการจัดหน้าส่วนต่อประสานผู้ใช้ที่ใช้ในงานวิจัยนี้ ต้องถูกสร้างด้วยเครื่องมือ Android Studio เท่านั้น

8) .ในการจำลองแอนดรอยด์แอปพลิเคชันต้องใช้ Genymotion ช่วยในการจำลองแอปพลิเคชันเท่านั้น

### 6.3) งานวิจัยในอนาคต

1) นำแนวทางหลักการในการพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองของต้นแบบแอนดรอยด์แอปพลิเคชันจากแผนภาพนำทางหน้าต่างนำไปประยุกต์ใช้กับต้นแบบอื่น ๆ เพื่อให้มีคุณภาพที่ดีและเพื่อช่วยให้เวลาของการพัฒนาระบบลดลง

2) นำแนวทางหลักการในการพัฒนาเชิงขับเคลื่อนด้วยแบบจำลองของต้นแบบแอนดรอยด์แอปพลิเคชันจากแผนภาพนำทางหน้าต่างนำไปประยุกต์ใช้กับต้นแบบพัฒนาแอปพลิเคชันมือถือแบบอื่นด้วยได้

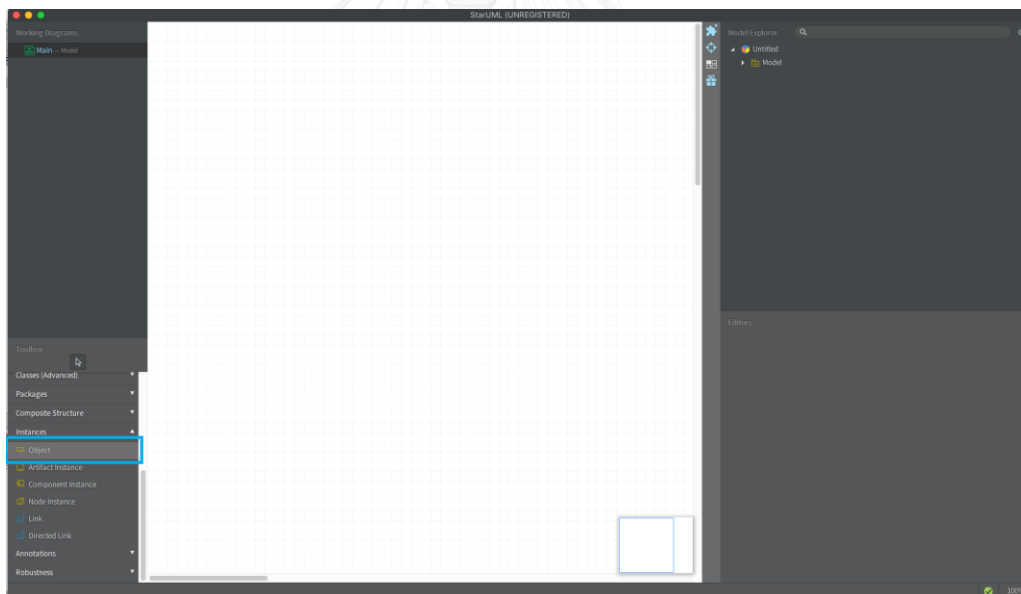
## รายการอ้างอิง

- [1] (2014-2015, June 25). *StarUML*. Available: <http://staruml.io/>
- [2] (1999, June 25). *JSON*. Available: <http://www.json.org/>
- [3] (2013-2015, June 25). *Extensible Markup Language*. Available: <https://www.w3.org/XML/>
- [4] (1997-2005, June 25). *Document Object Model*. Available: <https://www.w3.org/DOM/>
- [5] (1999-2016, June 25). *XML DOM Parser*. Available: [http://www.w3schools.com/xml/dom\\_parser.asp](http://www.w3schools.com/xml/dom_parser.asp)
- [6] (1997-2016, June 25). *MDA*. Available: <http://www.omg.org/mda/>
- [7] (1997-2016, June 25). *Object Management Group*. Available: <http://www.omg.org/>
- [8] (1997-2016, June 25). *Welcome to UML Web Site*. Available: <http://www.uml.org>
- [9] (2016, June 25). *Unified Modeling Language*. Available: [https://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language)
- [10] A. Sabraoui, M. E. Koutbi, and I. Khriess, "GUI code generation for Android applications using a MDA approach," in *Complex Systems (ICCS), 2012 International Conference on*, 2012, pp. 1-6.
- [11] M. Lachgar and A. Abdali, "Generating Android graphical user interfaces using an MDA approach," in *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, 2014, pp. 80-85.
- [12] (1997-2016, June 25). *XMI*. Available: <http://www.omg.org/spec/XMI/>
- [13] (2016, June 25). *Download Android Studio*. Available: <https://developer.android.com/studio/index.html>

## ภาคผนวก การสร้างแผนภาพหน้าต่างต่าง

การสร้างแบบจำลองแผนภาพหน้าต่างต่างด้วยเครื่องมือสตาร์ยูเอ็มแอล เป็นการนำเสนอแผนภาพหน้าต่างต่างที่ใช้สำหรับการจำลองเหตุการณ์ที่เกิดขึ้นในหนึ่งเหตุการณ์ที่จะแสดงอยู่บนแต่ละแผนภาพหน้าต่างต่าง และเกี่ยวกับความสัมพันธ์ความเชื่อมโยงที่อยากให้เกิดระหว่างแผนภาพหน้าต่างต่างด้วย เหตุการณ์ที่เกิดขึ้นและรวมถึงความสัมพันธ์ความเชื่อมโยงกันจะส่งผลลัพธ์ไปยังการจำลองต้นแบบแอนดรอยด์แอปพลิเคชันโดยตรง โดยรายละเอียดการวาดภาพจำลองแผนภาพหน้าต่างต่างจะเป็นดังต่อไปนี้

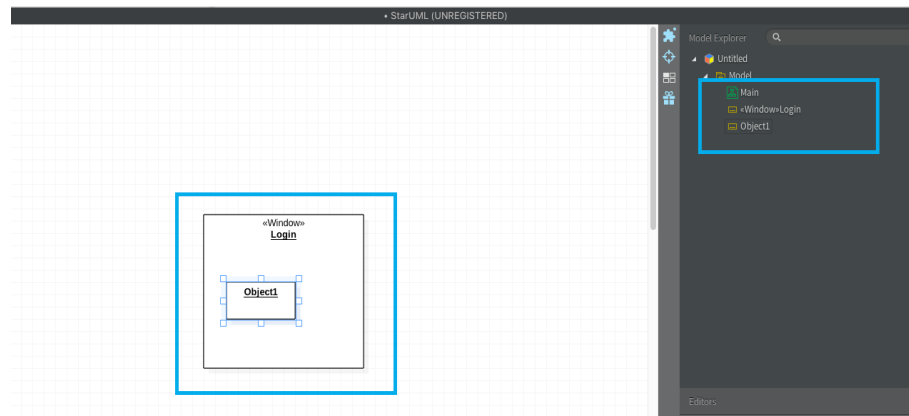
- 1) ติดตั้งเครื่องมือ StarUML for macOS
- 2) เปิดเครื่องมือและเลือกแม่แบบประเภท Class Diagram และเลือกแผนภาพที่ชื่อว่า Object ภายใต้คุณสมบัติ Instances แสดงดังรูป 47



รูปที่ 47 ภาพรวมเครื่องมือสตาร์ยูเอ็มแอลที่เลือกแม่แบบประเภท Class Diagram

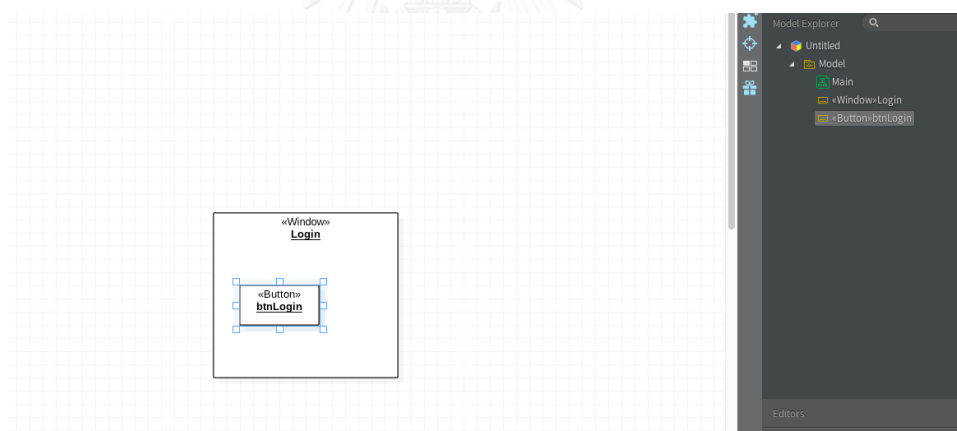
จากรูปที่ 47 ภายในรูปจะเห็นกล่องสี่ฟาด้านซ้ายมือ เป็นการแสดงให้เห็นถึงแผนภาพ Object ที่เป็นแผนภาพตัวแทนสำหรับวาดรูปแผนภาพหน้าต่างต่าง

- 3) หลังจากเลือกแผนภาพ Object ที่เป็นตัวแทนสำหรับวาดรูปแผนภาพหน้าต่างต่างแล้ว จะทำการเลือกกล่องพื้นที่สีขาวเพื่อทำการวาดรายละเอียดลงไป



รูปที่ 48 ทดสอบการวาดแผนภาพหน้าต่างต่าง

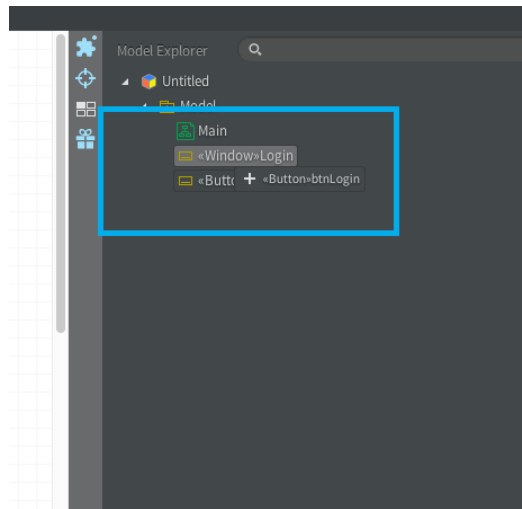
จากรูปที่ 48 แสดงให้เห็นว่าจากกล่องสี่เหลี่ยมสีฟ้ากลางรูป เป็นการวาดแบบจำลองแผนภาพหน้าต่างต่างที่มีชื่อว่า Login และมี Stereotype ที่มีชื่อว่า Window โดยจะสังเกตจากกล่องสี่เหลี่ยมด้านบนขวามือจะพบ <<Window>>Login เนื้อหาส่วนนี้จะสอดคล้องกับสิ่งที่ได้วาดรูปและกำหนดชื่อลงในแผนภาพ



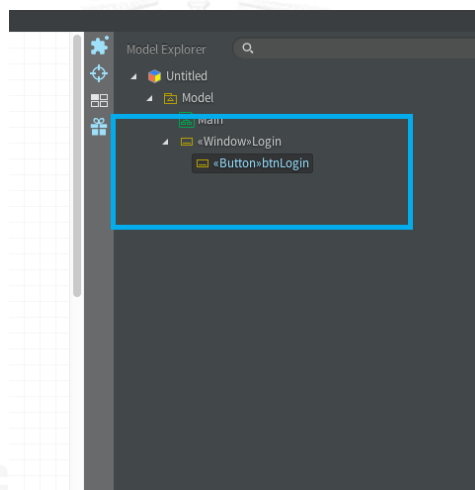
รูปที่ 49 ทดสอบการเปลี่ยนชื่อแผนภาพ <<Button>> btnLogin

- 4) ทำการย้ายแผนภาพ <<Button>> btnLogin ให้อยู่ภายในแผนภาพ <<Window>> Login เพื่อทำการกำหนดให้ว่าแผนภาพ <<Button>> btnLogin จะอยู่ภายใต้แผนภาพ <<Window>> Login ดังรูปที่ 50 และ 51





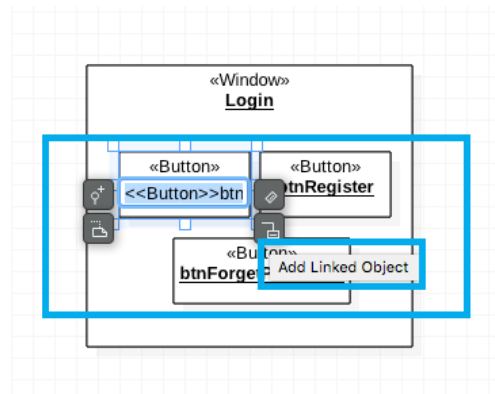
รูปที่ 50 ภาพการซ้อนแผนภาพกับแผนภาพ



รูปที่ 51 ภาพหลังจากทำการซ้อนแผนภาพ

จากรูปที่ 50 และ 51 จะแสดงให้เห็นวิธีการย้าย <<Button>> btnLogin ให้ข้อมูลอยู่ภายใต้แผนภาพ <<Window>> Login

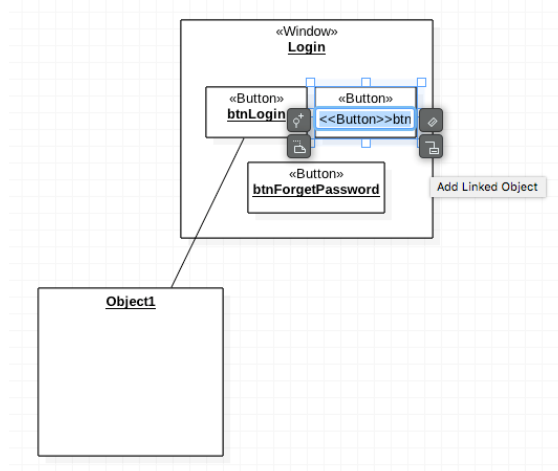
- 5) เมื่อทำการซ้อนแผนภาพแล้วต้องการที่จะสร้างความสัมพันธ์เชื่อมโยงระหว่างแผนภาพนำทางหน้าต่าง จะการเลือกแผนภาพ <<Button>> btnLogin แล้วทำการคลิกสองครั้งเพื่อให้ส่วน Shorthand แสดงภาพขึ้นมาดังรูป 52



รูปที่ 52 ภาพเครื่องมือ Shorthand

จากรูปที่ 52 ใช้เครื่องมือสนับสนุนการสร้างแผนภาพแบบรวดเร็ว เมื่อทำการกด Add Linked Object

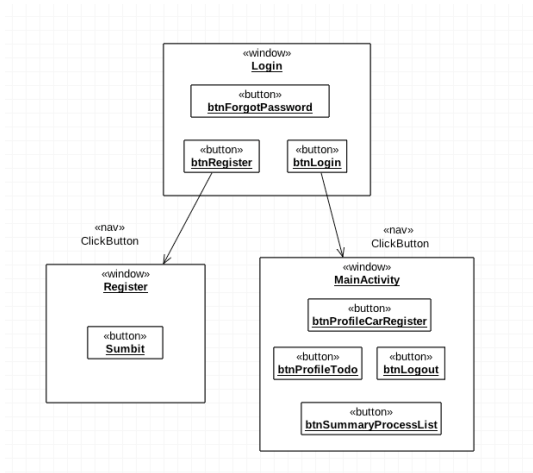
6) หลังจากทำการกดเลือก Add Linked Object เรียบร้อยแล้ว จะแสดงดังรูปที่ 53



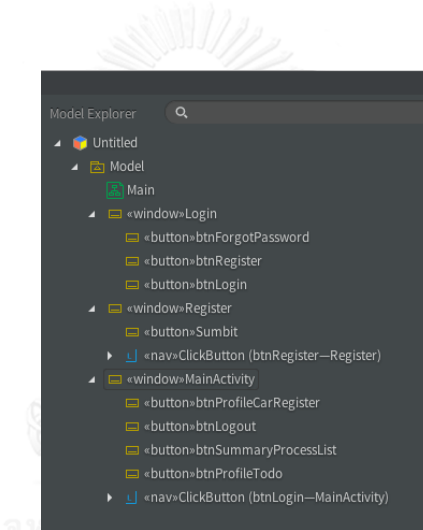
รูปที่ 53 ภาพผลลัพธ์หลังจากใช้เครื่องมือ Shorthand ช่วยการสร้างความสัมพันธ์การเชื่อมโยงระหว่างแผนภาพหน้าต่างต่าง

จากรูปที่ 53 แสดงให้เห็นผลลัพธ์จากการใช้เครื่องมือสนับสนุนการสร้างแผนภาพแบบรวดเร็ว ผลลัพธ์ที่ได้จะได้แผนภาพที่ชื่อว่า Object1 ที่แสดงอยู่ทางซ้ายของรูปที่ 53

7) หลังจากสร้างความสัมพันธ์การเชื่อมโยงระหว่างแผนภาพหน้าต่างต่างเสร็จผลลัพธ์จะเป็นดังรูปที่ 54 และ 55



รูปที่ 54 ภาพรวมการกำหนดรายละเอียดแผนภาพหน้าต่างหน้าต่าง



รูปที่ 55 รายละเอียดภาพรวมเมื่อทำการกำหนดรายละเอียดเสร็จเรียบร้อยแล้ว

จากรูปที่ 54 และ 55 แสดงให้ผลลัพธ์สุดท้ายหลังจากการกำหนดรายละเอียดเรียบร้อยแล้ว โดยจะสังเกตว่า รายละเอียดที่อยู่ภายใต้แต่ละแผนภาพหน้าต่างหน้าต่าง จากรูปที่ 54 ต้องสอดคล้องกับรูปที่ 55

## ประวัติผู้เขียนวิทยานิพนธ์

นาย ธนเสฏฐ์ ชาญนนทวัฒน์ เกิดเมื่อวันที่ 1 มกราคม พ.ศ. 2535 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีหลักสูตรวิศวกรรมศาสตรบัณฑิต (วศ.บ.) สาขาวิชา วิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง ในปีการศึกษา 2557 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขา วิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์ มหาวิทยาลัย ในปีการศึกษา 2557

