



การพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์
บนระบบปฏิบัติการแอนดรอยด์

โดย

ว่าที่ร้อยตรีอนันต์ชัย อู่คล้าย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาการจัดการงานวิศวกรรม

ภาควิชาวิศวกรรมอุตสาหกรรมและการจัดการ

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2557

ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

การพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์
บนระบบปฏิบัติการแอนดรอยด์

โดย

ว่าที่ร้อยตรีอนันต์ชัย อุ้มคล้าย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาการจัดการงานวิศวกรรม

ภาควิชาวิศวกรรมอุตสาหกรรมและการจัดการ

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2557

ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

THE DEVELOPMENT OF THE PROGRAM ERGONOMICS WORKLOAD ASSESSMENT
FOR OPERATION ANDROID

By

Acting Sub Lt. Ananchai Uklai

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree

Master of Engineering Program in Engineering Management

Department of Industrial Engineering and Management

Graduate School, Silpakorn University

Academic Year 2014

Copyright of Graduate School, Silpakorn University

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร อนุมัติให้วิทยานิพนธ์เรื่อง“การพัฒนาโปรแกรม
ประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์” เสนอโดย ว่าที่ร้อยตรีอนันต์ชัย
อู่คล้ายเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิตสาขาวิชา
การจัดการงานวิศวกรรม

.....
(รองศาสตราจารย์ ดร.ปานใจ ชารัทสนวงศ์)
คณบดีบัณฑิตวิทยาลัย
วันที่.....เดือน..... พ.ศ.....

อาจารย์ที่ปรึกษาวิทยานิพนธ์
ผู้ช่วยศาสตราจารย์ ดร.ประจวบ กล่อมจิตร

คณะกรรมการตรวจสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.สิทธิชัย แซ่เหล่ม)
...../...../.....

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ไพฑูรย์ ศิริโอพาจ)
...../...../.....

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ประจวบ กล่อมจิตร)
...../...../.....

55405326 : สาขาวิชาการจัดการงานวิศวกรรม

คำสำคัญ: การยศาสตร์ /ระบบปฏิบัติการแอนดรอยด์

อนันต์ชัย อุ่คล้าย : การพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์. อาจารย์ที่ปรึกษาวิทยานิพนธ์ : ผศ. ดร. ประจวบ กล่อมจิตร. 185 หน้า.

การศึกษานี้มีวัตถุประสงค์เพื่อศึกษาและพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์ เป็นการศึกษาเกี่ยวกับการนำวิธีการประเมินภาระงานทางการยศาสตร์ต่างๆ ได้แก่ การประเมินท่าทางการทำงานโดยใช้วิธีการ RULA และการประเมินงานยกของด้วยมือโดยสมการ NIOSH 1991 มาเป็นหลักการพื้นฐานเพื่อนำไปใช้ในการออกแบบและพัฒนาเป็นแอปพลิเคชันแอนดรอยด์โดยการศึกษาวิธีการประเมินภาระงานวิธีต่างๆ เพื่อนำมาใช้เป็นแนวทางในการเขียนโปรแกรม จากนั้นจึงทำการออกแบบและพัฒนาโปรแกรม และวัดผล ตลอดจนเปรียบเทียบประสิทธิภาพของโปรแกรมพร้อมก็นำไปประยุกต์ใช้กับอุตสาหกรรมตัวอย่าง โดยผลการศึกษาพบว่า โปรแกรมที่พัฒนาสามารถนำไปใช้งานได้จริง ค่าของข้อมูลนำเข้าและข้อมูลสรุปผลการประเมินที่ได้มีความถูกต้อง สามารถนำผลที่ได้จากการโปรแกรมไปใช้งานต่อได้จากการนำโปรแกรมไปทดสอบใช้งานกับโรงงานอุตสาหกรรมกรณีศึกษา ปรากฏว่าการใช้งานโปรแกรมในการประเมินภาระงานเมื่อเปรียบเทียบกับวิธีการการคำนวณมือโปรแกรม “RulaSU” ช่วยลดระยะเวลาในการทำการประเมินภาระงานลง 2.07 นาที หรือคิดเป็น 47.36%

ภาควิชาวิศวกรรมอุตสาหกรรมและการจัดการ

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ลายมือชื่อนักศึกษา

ปีการศึกษา 2557

ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์

55405326 : MAJOR : ENGINEERING MANAGEMENT

KEYWORDS : ERGONOMICS / ANDROID OPERATION SYSTEM

ANANCHAI UKLAI : THE DEVELOPMENT OF THE PROGRAM
ERGONOMICS WORKLOAD ASSESSMENT FOR ANDROID OPERATION SYSTEM.
THESIS ADVISOR : ASST. PROF. PRACHUAB KLOMJIT, PH.D., 185 pp.

The objective of this study is investigate and develop the ergonomics assessment of the workload on the Android operating system. A study on the introduction of assessment workload ergonomics include assessing posture work using RULA and assessing raised by hand by the equation NIOSH 1991 the basic principles to be used in the design and develop the applications with Android. By studying how to assess the workload of different ways to implement as a guideline to programming. Then design and program development and evaluation. As well as the performance of the program, will be applied to industrial samples. The study found that program developed can be applied practically. The value of the input data and concluded that it is accurate. The results of the program can be used to. From the adoption program to try to industrial case studies, it appears that the use of assessment workload compared with the calculation method used. Program "RulaSU" reduces the duration of the assessment workload by 2.07 minutes, or 47.36%.

Department of Industrial Engineering and Management Graduate School, Silpakorn University
Student's signature Academic Year 2014
Thesis Advisor's signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จลุล่วงได้ด้วยความช่วยเหลือจากบุคคลหลายท่าน ผู้วิจัยจึงขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. ประจวบ กล่อมจิตร อาจารย์ที่ปรึกษาวิทยานิพนธ์เป็นอย่างสูงที่กรุณาให้คำแนะนำ ข้อคิดเห็น ถ่ายทอดความรู้ต่างๆ อันเป็นประโยชน์ต่อผู้วิจัยเป็นอย่างยิ่ง อีกทั้งท่านยังคงยึดติดตามความก้าวหน้าตลอดระยะเวลาในการทำวิจัย และยังเสียสละเวลาในการตรวจสอบวิทยานิพนธ์ด้วยความตั้งใจและเต็มใจอย่างยิ่งตลอดมา รวมทั้งขอขอบพระคุณ คณะกรรมการสอบวิทยานิพนธ์ทุกท่านที่กรุณาให้คำแนะนำและข้อเสนอแนะที่มีประโยชน์ยิ่งต่อการปรับปรุงแก้ไขวิทยานิพนธ์ฉบับนี้ ให้สมบูรณ์ยิ่งขึ้น

ขอขอบคุณอาจารย์ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากรทุกท่านที่ให้ข้อมูลในการทำวิทยานิพนธ์ อีกทั้งความร่วมมือต่างๆ ที่เป็นประโยชน์ยิ่งสำหรับการทำวิทยานิพนธ์ฉบับนี้

สุดท้ายนี้ ขอกราบขอบพระคุณพ่อและแม่ รวมถึงทุกคนในครอบครัวที่เป็นกำลังใจ และให้การสนับสนุนช่วยเหลือจนทำให้วิทยานิพนธ์ฉบับนี้ สำเร็จเสร็จสิ้นลงได้ด้วยดี

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฅ
สารบัญรูป.....	ฉ
บทที่	หน้า
1 บทนำ.....	1
ความเป็นมาและความสำคัญของปัญหา.....	1
กรอบแนวคิดของการวิจัย.....	3
วัตถุประสงค์ของการวิจัย.....	4
ขอบเขตของการวิจัย.....	4
ประโยชน์ที่จะได้รับจากการวิจัย.....	4
2 ทบทวนวรรณกรรม.....	6
ทฤษฎีที่เกี่ยวข้อง.....	6
งานวิจัยที่เกี่ยวข้อง.....	20
3 วิธีดำเนินการวิจัย.....	22
การศึกษาเบื้องต้นเกี่ยวกับหลักการและทฤษฎีที่เกี่ยวข้องต่างๆ ของการของการ ประเมินภาระงานตามหลักการยศาสตร์.....	23
การศึกษางานวิจัยที่เกี่ยวข้อง.....	23
ออกแบบและพัฒนาโปรแกรม.....	23
สอบถามการออกแบบพัฒนา โปรแกรม.....	24
การทดลองการใช้งานโปรแกรม.....	24
ประเมินผลการใช้งานของโปรแกรม.....	24
สรุปผลการดำเนินงาน.....	26

บทที่	หน้า
จัดทำรายงานสรุปผลการศึกษ.....	26
4 ผลการวิเคราะห์ข้อมูลและอภิปรายผล.....	28
ผลการออกแบบและการสร้างโปรแกรม.....	28
ผลการประเมินประสิทธิภาพและคุณภาพของโปรแกรม.....	35
การประเมินข้อมูลด้านเวลาเชิงสถิติ.....	48
ผลการนำโปรแกรมไปทดลองใช้งานในโรงงานอุตสาหกรรมกรณีศึกษา.....	54
5 สรุปผลการศึกษาและข้อเสนอแนะ.....	56
สรุปผลการศึกษา.....	56
ปัญหาและอุปสรรค.....	57
ข้อเสนอแนะ.....	58
รายการอ้างอิง.....	59
ภาคผนวก.....	60
ภาคผนวก ก.....	61
ภาคผนวก ข.....	66
ภาคผนวก ค.....	77
ภาคผนวก ง.....	82
ภาคผนวก จ.....	91
ประวัติผู้วิจัย.....	185

สารบัญตาราง

ตารางที่		หน้า
2.1	ค่า C ในสมการ NIOSH 1991.....	9
2.2	ค่า F ในสมการ NIOSH 1991.....	9
2.3	คะแนน A โดยใช้ท่าทางของส่วนของร่างกายในกลุ่ม A.....	13
2.4	คะแนน B โดยใช้ท่าทางของส่วนร่างกายในกลุ่ม B.....	16
2.5	คะแนนผลลัพธ์โดยใช้คะแนน C และ D.....	17
3.1	แผนการปฏิบัติงาน.....	27
4.1	การเปรียบเทียบคุณสมบัติของโปรแกรมที่ใช้สำหรับการประเมินท่าทางในการ ทำงานโดยใช้วิธีการRULA.....	30
4.2	ผลการเก็บข้อมูลและเปรียบเทียบข้อมูลระหว่างการใช้งานโปรแกรมRulaSUกับ การใช้งานโปรแกรม ErgoIntelligence.....	36
4.3	ผลการเก็บข้อมูลและเปรียบเทียบข้อมูลระหว่างการใช้งานโปรแกรมRulaSUกับ วิธีการคำนวณมือ.....	36
4.4	ผลการประเมินคุณภาพในแต่ละด้านของโปรแกรมRulaSU.....	38
4.5	ผลการประเมินด้านความง่ายต่อการใช้งานระบบ (Usability Test).....	39
4.6	ผลการประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test).....	41
4.7	ผลการประเมินด้านการตรงความต้องการของผู้ใช้ระบบ(Functional Requirement Test).....	42
4.8	ผลการประเมินคุณภาพในแต่ละด้านของโปรแกรมErgoIntelligence.....	43
4.9	ผลการประเมินด้านความง่ายต่อการใช้งานระบบ (Usability Test).....	44
4.10	ผลการประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test).....	46
4.11	ผลการประเมินด้านการตรงความต้องการของผู้ใช้ระบบ(Functional Requirement Test).....	47
4.12	การเก็บข้อมูลเวลารวมการทำงานรวมทั้งขั้นตอน ของการประเมินท่าทางการ ทำงานโดยใช้วิธีการRULA.....	48
4.13	การจัดเรียงข้อมูลสำหรับหาค่าProbability ของการใช้โปรแกรมRulaSU.....	49

ตารางที่		หน้า
4.14	การจัดเรียงข้อมูลสำหรับหาค่าProbability ของการใช้โปรแกรม ErgoIntelligence.....	49
4.15	การจัดเรียงข้อมูลสำหรับหาค่าProbability ของวิธีการคำนวณมือเพื่อประเมินท่าทาง ในการทำงานโดยใช้วิธีการRULA.....	50
4.16	ผลการใช้ MINITAB โดยเลือก t-Test: Two – Sample.....	52
4.17	ผลการใช้ MINITAB โดยเลือก t-Test: Two – Sample.....	53
4.18	ผลข้อมูลเวลาเฉลี่ยที่ได้จากการทำการประเมินท่าทางในการทำงานโดยใช้วิธีการ RULA.....	55

สารบัญรูป

รูปที่		หน้า
1.1	กรอบแนวความคิดของการวิจัย.....	3
2.1	การให้คะแนนท่าทางการทำงานของแขนส่วนบน.....	11
2.2	การให้คะแนนท่าทางการทำงานของแขนส่วนล่าง.....	11
2.3	การให้คะแนนของมือและการบิดของข้อมือ.....	12
2.4	การให้คะแนนท่าทางการทำงานในส่วนของคอ.....	14
2.5	การให้คะแนนท่าทางการทำงานในส่วนของลำตัว.....	15
4.1	ผังงานของโปรแกรมRulaSU.....	29
4.2	หน้าตาเริ่มต้นของโปรแกรมRulaSU.....	31
4.3	หน้าตาสำหรับข้อมูลของผู้ทำการประเมินและรูปท่าทางการทำงาน.....	31
4.4	หน้าตาสำหรับรับข้อมูลของการให้คะแนนท่าทางในส่วนต่างๆของร่างกาย.....	32
4.5	หน้าตาสำหรับการรับข้อมูลการประเมินการใช้แรงและกล้ามเนื้อในการทำงานของ ส่วนแขนและมือ.....	34
4.6	หน้าตาสำหรับการรับข้อมูลการประเมินการใช้แรงและกล้ามเนื้อในการทำงานของ ส่วนคอ ลำตัวและเท้า.....	34
4.7	หน้าตาสำหรับแสดงผลการประเมินท่าทางในการทำงาน.....	35
4.8	แผนภูมิแท่งแสดงเวลาที่ใช้ในการประเมินท่าทางในการทำงานโดยใช้วิธีการRULA (วินาที).....	37
4.9	ผลการประเมินคุณภาพของโปรแกรม RulaSU.....	38
4.10	ผลการประเมินด้านความง่ายต่อการใช้งานระบบ (Usability Test) ของการใช้งาน โปรแกรม RulaSU.....	40
4.11	ผลการประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test) ของการใช้ งานโปรแกรม RulaSU.....	41
4.12	ผลการประเมินด้านกฏตรงความต้องการของผู้ใช้ระบบ(Functional Requirement Test)ของใช้งานโปรแกรม RulaSU.....	42
4.13	ผลการประเมินคุณภาพของโปรแกรม ErgoIntelligence.....	43

รูปที่	หน้า
4.14 ผลการประเมินด้านความง่ายต่อการใช้งานระบบ(Usability Test) ของการใช้งาน โปรแกรม ErgoIntelligence.....	45
4.15 ผลการประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test) ของการใช้ งานโปรแกรม ErgoIntelligence.....	46
4.16 ผลการประเมินด้านการตรงความต้องการของผู้ใช้ระบบ(Functional Requirement Test)ของการใช้งานโปรแกรม ErgoIntelligence.....	47
4.17 กราฟการกระจายของข้อมูลด้านเวลาของการใช้โปรแกรมRulaSU.....	50
4.18 กราฟการกระจายของข้อมูลด้านเวลาของการใช้โปรแกรมErgoIntelligence.....	51
4.19 กราฟการกระจายของข้อมูลด้านเวลาของวิธีการคำนวณมือเพื่อประเมินท่าทางในการ ทำงานโดยใช้วิธีการRULA.....	51
4.20 ภาพขณะทำการการอบรมให้ควมรู้เกี่ยวกับการประเมินภาระงานโดยใช้วิธีการ ทางการยศาสตร์แก่ผู้ทดลองใช้งาน โปรแกรม.....	54
4.21 ผลข้อมูลเวลาเฉลี่ยที่ใช้ในการประเมินท่าทางในการทำงานโดยใช้วิธีการRULA.....	55

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบัน โรงงานอุตสาหกรรมในประเทศไทยมีอยู่มากมายหลายประเภท ซึ่งในแต่ละประเภทนั้นได้มีการนำเอาเทคโนโลยีที่ทันสมัยต่างๆ เข้ามาประยุกต์ใช้เพื่อเพิ่มประสิทธิภาพในการผลิตและความสะดวกสบายในการทำงาน เช่น เครื่องจักรอัตโนมัติ อุปกรณ์ที่ควบคุมโดยใช้ระบบคอมพิวเตอร์ หุ่นยนต์ เป็นต้น แม้ว่าการนำเทคโนโลยีเข้ามามีใช้ในการผลิตนั้นกำลังเป็นที่แพร่หลายและมีการนำไปใช้เกือบทุกส่วนงานของโรงงานอุตสาหกรรม แต่ในบางส่วนงานนั้น การใช้แรงงานก็ยังเป็นสิ่งจำเป็นต่อการดำเนินงานของโรงงาน ไม่ว่าจะเป็นการใช้แรงงานในการผลิตหรือการใช้แรงงานในการควบคุมเครื่องจักรต่างๆ แรงงานจึงเป็นปัจจัยสำคัญอย่างหนึ่งที่เกิดผลผลิตของโรงงานอุตสาหกรรม

การประเมินภาระงานให้เหมาะสมกับความสามารถของผู้ปฏิบัติงานจึงเป็นมีความสำคัญต่อประสิทธิภาพในการผลิตและการดำเนินงานของโรงงาน ภาระงานที่เหมาะสมกับผู้ปฏิบัติงานนั้นจะส่งผลต่อประสิทธิภาพของระบบงานทั้งทางด้านผลผลิต ทางด้านร่างกายและทางด้านจิตใจของผู้ปฏิบัติงานให้ดีขึ้น ในทางตรงกันข้ามหากภาระงานที่มากเกินไปนั้นจะทำให้ร่างกายเกิดความเมื่อยล้าจากการทำงาน ส่งผลให้ประสิทธิภาพงานโดยรวมลดลงหรือหากภาระงานที่น้อยเกินไปก็จะทำให้โรงงานนั้นเกิดการล้มละลายจากการขาดทุนได้

โรงงานอุตสาหกรรมที่มีการใช้แรงงานในการดำเนินงานนั้น ยังคงมีการศึกษาถึงการประเมินสภาพภาระงานของแรงงานอยู่น้อย ทำให้ขาดความรู้ความตระหนักถึงปัญหาภาระงานที่แรงงานนั้นต้องแบกรับไว้ในการทำงาน ซึ่งส่งผลให้แรงงานได้รับบาดเจ็บจากการทำงานได้ อาการบาดเจ็บส่วนใหญ่มักเกิดจากท่าทางการทำงานที่ไม่เหมาะสม ซึ่งอาจมาจากการที่แรงงานไม่มีความรู้ความเข้าใจในเรื่องการปฏิบัติงานในท่าทางการทำงานที่ถูกต้อง สถานที่ เครื่องจักรหรืออุปกรณ์การทำงาน ไม่เอื้ออำนวยต่อการปฏิบัติงานประเภทนั้น อาการบาดเจ็บที่เกิดขึ้นจะส่งผลกระทบต่อการทำงานของสถานประกอบการ เช่น การสูญเสียต่อผลผลิตจากการที่แรงงานไม่

สามารถที่จะปฏิบัติงานได้ ทรัพย์สินของสถานประกอบการเกิดความเสียหายขึ้นจากอุบัติเหตุใน
การทำงาน สูญเสียระยะเวลาในการดำเนินการสอนงานให้แก่ผู้ที่มาทำการปฏิบัติงานแทน รวมไปถึง
ถึงสภาพจิตใจของผู้ที่ได้รับบาดเจ็บจากการปฏิบัติงาน

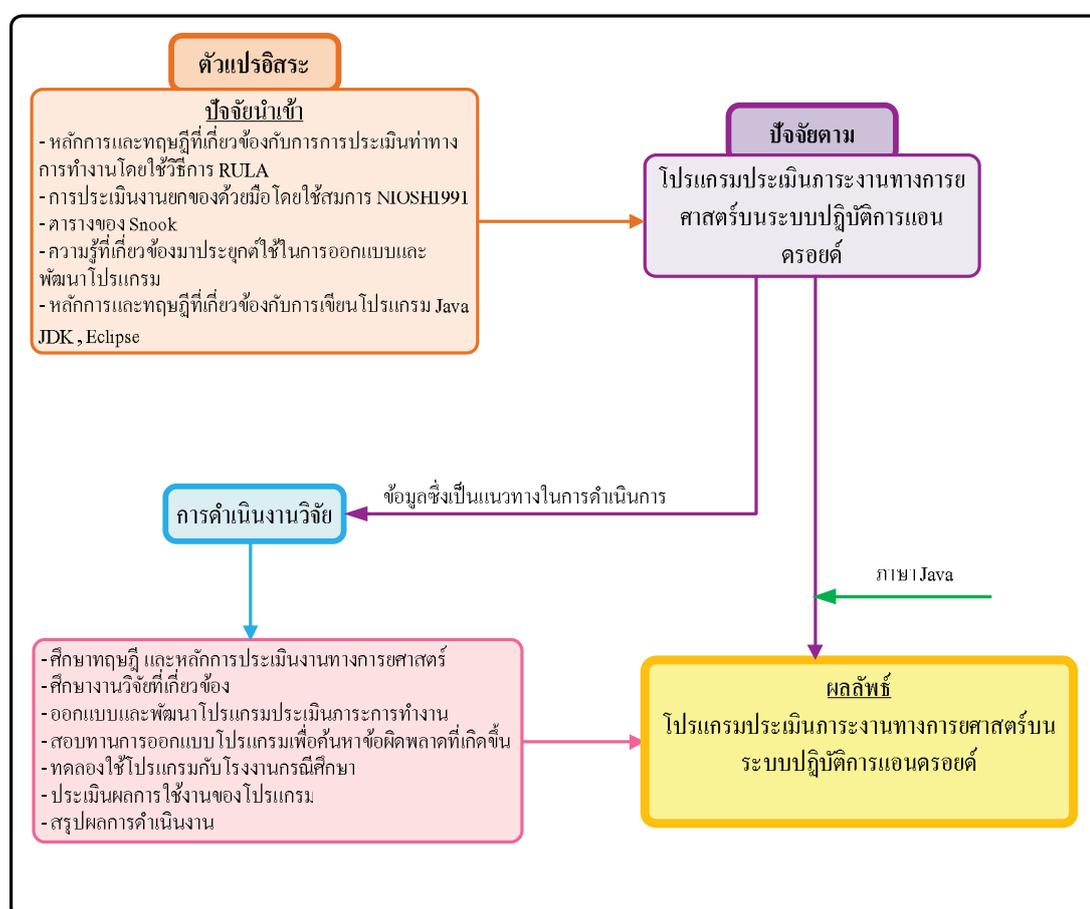
การยศาสตร์ (Ergonomics) เป็นความรู้ที่เกี่ยวข้องกับการประยุกต์หลักการ วิธีการและ
ข้อมูล ที่ได้จากความรู้จากหลากหลายสาขา เช่น ความรู้ทางการแพทย์ ความรู้ทางวิทยาศาสตร์ต่างๆ
ความรู้ทางวิศวกรรม ความรู้ทางสิ่งแวดล้อม ความรู้ทางจิตวิทยา เป็นต้น สำหรับใช้ในการศึกษา
และวิเคราะห์งานเพื่อปรับปรุงงาน โดยเน้นความสัมพันธ์ระหว่างผู้ปฏิบัติงานกับงานและ
สิ่งแวดล้อมในการทำงาน เพื่อให้ผู้ที่เกี่ยวข้องกับระบบการทำงานเหล่านั้น สามารถใช้งานหรือทำ
การปฏิบัติงานในระบบงาน ได้อย่างมีประสิทธิภาพและมีความปลอดภัย

การยศาสตร์จึงมีส่วนสำคัญต่อการทำให้มีการปรับปรุงรูปแบบภาระการทำงานในสถาน
ประกอบการให้เกิดความเหมาะสม เป็นศาสตร์ที่ใช้เพื่อให้เกิดความปลอดภัยในการทำงานโดย
ประยุกต์ใช้ศาสตร์จากหลายแขนงวิชาในการสำรวจขั้นตอนการทำงานเดิม เพื่อประเมินและทำการ
ควบคุมหรือปรับปรุงให้เกิดความปลอดภัยต่อพนักงานผู้ปฏิบัติงานและทรัพย์สินของสถาน
ประกอบการ การนำหลักการของการยศาสตร์ไปใช้นั้น อาจเป็นการออกแบบระบบงานใหม่หรือ
เป็นการปรับปรุงรูปแบบการทำงาน เช่น การออกแบบสถานีงานใหม่ให้มีความสะดวกต่อการ
ทำงานมากขึ้น ท่าทางในการทำงานนั้นมีความเหมาะสมมากยิ่งขึ้น ออกแบบเครื่องมือเพื่อช่วยลด
ความเมื่อยล้าในการทำงาน เพื่อให้สามารถทำงานได้ง่ายขึ้นและลดความเสี่ยงต่อการเกิดความ
ผิดพลาดในการทำงานได้ รวมไปถึงการออกแบบสภาพแวดล้อมในการทำงานใหม่เพื่อส่งผลดีต่อ
สภาพจิตใจของพนักงานได้ ส่งผลให้สามารถทำงานได้อย่างมีประสิทธิภาพสูงและเป็นการลดความ
เสี่ยงต่อในการทำงานของพนักงานด้วย

เครื่องมือสำหรับการประเมินภาระการทำงานตามหลักการทางการยศาสตร์ในปัจจุบันนั้น มี
หลากหลายรูปแบบให้เลือกใช้ เช่น การประเมินท่าทางการทำงานด้วย RULA (Rapid Upper Limb
Assessment) การประเมินภาระงานต้น งานคิงและงานถือ โดยใช้ตารางของ Snook เป็นต้น
เครื่องมือดังกล่าวนี้ มีความสะดวกต่อการใช้งาน มีหลักการประเมินที่เข้าใจได้ง่าย และสามารถนำ
ไปใช้งานได้จริง จึงทำให้ปัจจุบันมีผู้สนใจในการนำเครื่องมือดังกล่าวไปใช้ในการปรับปรุงรูปแบบ
วิธีการทำงานให้เหมาะสมตามหลักการทางการยศาสตร์มากยิ่งขึ้น

ผู้วิจัยจึงพิจารณาจัดทำงานวิจัยเรื่อง การพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์ โดยมุ่งเน้นที่จะพัฒนาโปรแกรมสำเร็จรูปบนระบบปฏิบัติการแอนดรอยด์สำหรับใช้ในการประเมินภาระงานทางการยศาสตร์ที่เกี่ยวข้องกับท่าทางในการทำงาน เป็นการเผยแพร่และส่งเสริมการนำหลักการดังกล่าวไปใช้ เพื่อเป็นประโยชน์ในการประเมินสภาวะแวดล้อมและพฤติกรรมในการทำงานซึ่งจะนำไปสู่การปรับปรุงลักษณะท่าทางการทำงานและสภาพแวดล้อม และเพื่อลดความเสี่ยงต่อการบาดเจ็บสะสมจากการทำงานได้

1.2 กรอบแนวความคิดของการวิจัย



รูปที่ 1.1 แสดงกรอบแนวความคิดของการวิจัย

1.3 วัตถุประสงค์ของงานวิจัย

1.3.1 เพื่อศึกษาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์

1.3.2 เพื่อพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์โดยใช้ภาษาJava

1.3.3 นำโปรแกรมที่ได้ทำการออกแบบและพัฒนา มาใช้ในการประเมินผลลัพท์งานทางการยศาสตร์ในโรงงานอุตสาหกรรมกรณีศึกษา

1.4 ขอบเขตของการวิจัย

งานวิจัยเรื่อง การพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์ จัดทำขึ้นเพื่อศึกษาและพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์ โดยใช้โปรแกรม Java JDK , Eclipse , Android SDK และ Android Developer Tools ในการออกแบบและพัฒนา แอปพลิเคชันแอนดรอยด์สำหรับการประเมินภาระการทำงานทางการยศาสตร์ที่จะทำการพัฒนานั้น ได้แก่ แอปพลิเคชันแอนดรอยด์สำหรับการประเมินท่าทางการทำงานโดยใช้วิธีการ RULA โดยดำเนินการสอบทานการออกแบบโปรแกรมเพื่อค้นหาข้อผิดพลาดที่เกิดขึ้นในขั้นตอนของการพัฒนาโปรแกรม และมีการเปรียบเทียบผลการใช้งานโปรแกรมอีกด้วย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 แอปพลิเคชันแอนดรอยด์ที่พัฒนาขึ้นสามารถนำไปใช้ในการประเมินภาระการทำงานทางการยศาสตร์ได้จริง มีรูปแบบการแสดงผลที่ง่ายต่อความเข้าใจและข้อมูลที่ได้จากการใช้โปรแกรมดังกล่าวนี้ จะต้องมีความถูกต้อง

1.5.2 สามารถนำแอปพลิเคชันแอนดรอยด์สำหรับการประเมินภาระงานทางการยศาสตร์ดังกล่าวไปทดลองใช้งานในการศึกษาและปรับปรุงขั้นตอนวิธีการทำงานในโรงงานอุตสาหกรรมตัวอย่างได้

1.5.3 ผู้ที่มีความสนใจในหลักการประเมินภาระงานทางการยศาสตร์ สามารถนำแอปพลิเคชันแอนดรอยด์ดังกล่าวไปใช้ในการศึกษาหาความรู้เพิ่มเติมได้

1.5.4 เป็นแนวทางในการพัฒนา แอปพลิเคชันแอนดรอยด์ที่เกี่ยวข้องกับการประยุกต์ใช้ความรู้ทางด้านการจัดการงานวิศวกรรมต่อไป

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ความหมายของการยศาสตร์ [1], [2], [3]

การยศาสตร์ (Ergonomics) มาจากคำภาษากรีก 2 คำมารวมกันคือ Ergon หมายถึงงาน (Work) และ Nomos หมายถึงกฎ (Law) เมื่อรวมแล้วจึงเป็น Ergonomics หรือกฎของการทำงาน (Law of work) เป็นศาสตร์ที่ว่าด้วยการออกแบบสถานที่ในการทำงาน วิธีการทำงาน อุปกรณ์ เครื่องจักรกล ผลิตภัณฑ์ สิ่งแวดล้อมและระบบ โดยนำความรู้ที่เกี่ยวข้องกับการประยุกต์หลักการ วิธีการและข้อมูลที่ได้จากความรู้ในสาขาต่างๆ เช่น ความรู้ทางการแพทย์ ความรู้ทางวิทยาศาสตร์ สาขาต่างๆ ความรู้ทางวิศวกรรม ความรู้ทางสิ่งแวดล้อม ความรู้ทางจิตวิทยา เป็นต้น มาใช้ศึกษาความสามารถของมนุษย์ในแง่มุมลักษณะทางกายภาพ สรีรวิทยา กลศาสตร์ชีวภาพและจิตวิทยา เพื่อเป็นปัจจัยสำคัญในการพัฒนาระบบการทำงานที่ของมนุษย์หรือมีมนุษย์เข้าไปเกี่ยวข้องทำให้ให้ผู้เกี่ยวข้องเหล่านั้นสามารถใช้งานหรือทำงานได้อย่างมีประสิทธิภาพและประสิทธิผลในระบบงาน ในขณะที่เกี่ยวกับการออกแบบนั้นก็ต้องคำนึงถึงความปลอดภัยสุขภาพอนามัยและความเป็นอยู่ที่ดีของผู้ปฏิบัติงานนั้นๆพร้อมกันไปในเวลาเดียวกัน

การยศาสตร์นี้ มีความสำคัญต่อความสบายและประสิทธิภาพในระบบการทำงาน ได้สรุปคำนิยามของการยศาสตร์ (Ergonomics) ไว้ดังนี้

“การศึกษากฎหรือวิธีการทำงานที่เป็นพื้นฐานในการออกแบบและพัฒนาเครื่องมือ เครื่องจักร ลักษณะงานและสภาพแวดล้อมที่เกี่ยวข้อง เช่น แสง สี เสียง บรรยากาศ เป็นต้น โดยมีวัตถุประสงค์เพื่อให้มนุษย์สามารถทำงานหรือใช้งานกับสิ่งต่างๆ เหล่านี้ได้มีประสิทธิภาพ และปลอดภัย”

ในหลายทศวรรษที่ผ่านมา มีผู้รู้คนหนึ่งได้ให้ความหมายของคำว่า “การยศาสตร์” ในเชิงปฏิบัติว่าเป็น “การศึกษาเกี่ยวกับการปฏิสัมพันธ์กันหรืออิทธิพลระหว่างมนุษย์และเครื่องมือ

อุปกรณ์ภายใต้สิ่งแวดล้อมที่มนุษย์ทำงานอยู่ ความหมายนี้ดูเหมือนจะครอบคลุมองค์ประกอบที่สำคัญที่สุดไว้แล้ว คือ มนุษย์ เครื่องมืออุปกรณ์ สิ่งแวดล้อมและอุทกิริยาที่ซับซ้อนระหว่างปัจจัยทั้งสามนี้

การยศาสตร์ให้ความสำคัญในเรื่องพฤติกรรมของมนุษย์และปฏิสัมพันธ์ของมนุษย์ ที่มีต่อการออกแบบผลิตภัณฑ์ การใช้เครื่องมือ อุปกรณ์อำนวยความสะดวกและสิ่งแวดล้อม ตลอดจนกระบวนการที่มนุษย์ใช้ในการทำงานและใช้ในชีวิตประจำวัน เพื่อเพิ่มประสิทธิภาพหรือสมรรถนะให้กับการทำงานตามที่กำหนด ลดข้อผิดพลาด ลดความตึงเครียด ลดความเมื่อยล้า เพิ่มความปลอดภัยและเพิ่มความพึงพอใจในการทำงานของมนุษย์

2.1.2 การประยุกต์ใช้ความรู้ทางด้านการยศาสตร์ [1]

หากพิจารณาความหมายของการยศาสตร์ สามารถพิจารณาได้ว่า การประยุกต์ใช้ความรู้ทางด้านการยศาสตร์จะประกอบด้วยส่วนสำคัญสองส่วนคือ

1. การใช้ความรู้สึกเป็นเหตุเป็นผล (Common Sense)
2. การใช้หลักฐานทางวิทยาศาสตร์ (Sophisticated Science)

ความสำเร็จของการประยุกต์ใช้ความรู้ทางการยศาสตร์จะเป็นไปได้ยากถ้าขาดส่วนหนึ่งส่วนใด เพราะทั้งสองส่วนมีความสำคัญที่จะช่วยในการสนับสนุนซึ่งกันและกัน ความสำคัญของทั้งสองส่วนสามารถอธิบายได้ดังนี้

2.1.3 การประเมินงานยกของด้วยมือ โดยสมการ NIOSH 1991 [2]

ในรายงานของผู้ปฏิบัติงานจากการศึกษาของสถาบันอาชีวอนามัยและความปลอดภัยแห่งชาติสหรัฐอเมริกา หรือ NIOSH (National Institute of Safety and Health) พบว่า อัตราความถี่ของการบาดเจ็บที่หลัง (Frequency rates of back injuries) และอัตราความรุนแรงของการบาดเจ็บที่หลัง (Severity rates of back injuries) มีแนวโน้มเพิ่มขึ้นมากกว่าอัตราการบาดเจ็บปกติทั่วไป ซึ่งพบว่าลักษณะงาน 5 ประการดังต่อไปนี้ ส่งผลต่อการบาดเจ็บที่กระดูกสันหลัง

1. ยกของที่หนักมากเกินไป
2. ยกของที่มีรูปร่างใหญ่โต เทอะทะมากเกินไป
3. ยกของจากพื้นขึ้นสู่ที่สูงในแนวตั้ง
4. ยกของบ่อยๆ มากเกินไป

5. ยกของในท่าที่ไม่สมมาตรหรือไม่สมดุล เช่น ยกของหนักด้วยมือเดียวหรือเอี้ยวตัวยกของ เป็นต้น

สมการ NIOSH 1991 ได้ทำการศึกษาและเสนอแนะสมการที่ใช้ในกรคำนวณน้ำหนักที่เหมาะสมกับการทำงานและน้ำหนักสูงสุดที่ยอมให้ทำงานได้ โดยพิจารณาจากปัจจัยต่างๆ เช่น ความสูงในการยก น้ำหนักของที่ยก เป็นต้น

$$RWL = LC \times HM \times VM \times DM \times AM \times FM \times CM \quad (1)$$

$$LI = \frac{L}{RWL} \quad (2)$$

โดย

RWL = Recommended Weight Limit เป็นน้ำหนักมากที่สุดที่สามารถยกได้โดยไม่เกิดการปวดเมื่อย

L = น้ำหนักที่ยกขณะนั้น

LI = Lifting Index เป็นดัชนีการยก ถ้ามากกว่า 1 ต้องปรับปรุง

LC = Load Constant เป็นค่าคงที่ มีค่า 23 กิโลกรัม

HM = Horizontal Multiplier ($25/H$); H เป็นระยะจากข้อมือในแนวราบถึงจุดกึ่งกลางของสิ่งของที่ยก

VM = Vertical Multiplier ; V เป็นระยะจากข้อมือในแนวตั้งถึงพื้นที่จุดยก

DM = Distance Multiplier ; D เป็นระยะของวัตถุที่ยกในแนวตั้งถึงพื้นที่จุดวาง

AM = Asymmetric Multiplier ; A เป็นมุมของหัวไหล่ที่เอียงไปจากปกติในแนวราบ

FM = Frequency Multiplier

CM = Coupling Multiplier

2.1.4 การประเมินท่าทางโดยวิธีการ RULA (Rapid Upper Limb Assessment) [1], [3]

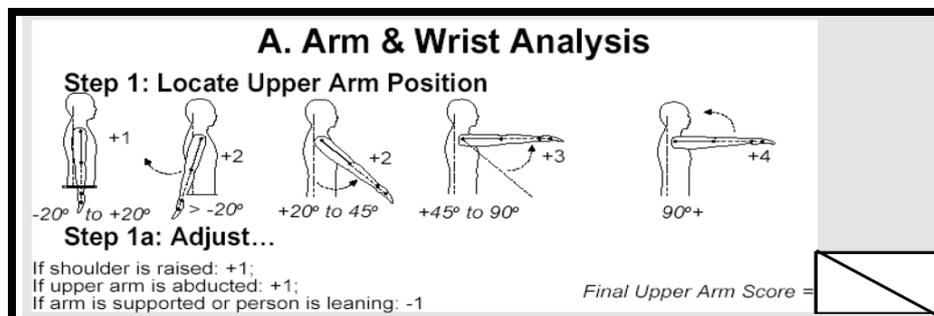
เป็นวิธีที่ออกแบบเพื่อใช้ประเมินความรุนแรงของท่าทาง โดยเฉพาะงานที่นั่งทำงานพัฒนา โดย Lynn Mcatamney และ Nigel Corlett ในปี ค.ศ.1993 ใช้กรณีตัวอย่างของอุตสาหกรรมตัดเย็บเสื้อผ้าโดยพิจารณางานต่างๆ ในกระบวนการผลิต วิธีนี้ใช้ประเมินความเสี่ยงของแต่ละบุคคลจากลักษณะท่าทางการทำงาน โดยพิจารณาดำแหน่งและลักษณะของการเคลื่อนไหว การพิจารณาจะแบ่งร่างกายออกเป็นสองกลุ่ม กลุ่มแรกประกอบด้วยกลุ่มของข้อมือแขนส่วนล่างแขนส่วนบนและหัวไหล่ หลังจากนั้นนำคะแนนมาพิจารณาร่วมกับการใช้แรง ซึ่งอาจเกิดจากน้ำหนักของที่ยก ลักษณะการออกแรงผลักหรือดึงการใช้แรงแบบอยู่กับที่หรือเคลื่อนไหวและปริมาณการทำซ้ำ คะแนนการประเมินร่างกายในกลุ่มที่สองเป็นการประเมินตำแหน่งของร่างกายส่วนของคอลำตัวและขา โดยพิจารณามุมหรือการหมุนของข้อต่อจากนั้นนำคะแนนมาพิจารณาร่วมกับการใช้แรงของกล้ามเนื้อแล้วนำคะแนนการประเมินที่ได้จากกลุ่มแรกและกลุ่มที่สองมารวมกันนำคะแนนที่ได้ไปเปิดตารางเพื่อพิจารณาระดับความเสี่ยงโดยคะแนนสูงสุดของการประเมินจะเท่ากับ 7 แสดงว่าลักษณะงานที่ทำให้มีความเสี่ยงต่อการเกิดกลุ่มอาการผิดปกติทางระบบโครงร่างและกล้ามเนื้อ ควรมีการดำเนินการปรับปรุงแก้ไขโดยเร็ว

การประเมินท่าทางโดยใช้วิธีการ RULA นี้ จะทำการประเมินส่วนต่างๆของร่างกาย ดังนี้

1. ศีรษะและคอ (Head and Neck)
2. ลำตัว(Trunk)
3. ไหล่ (Shoulder)
4. แขนส่วนบน (Upper Arm)
5. แขนส่วนล่าง (Lower Arm: Forearm)
6. มือและข้อมือ (Hand and Wrist)
7. ขาส่วนบน (Upper Leg: Thigh)
8. ขาส่วนล่าง (Lower Leg)
9. เท้า (Foot)

ซึ่งมีขั้นตอนวิธีการในการประเมิน ดังนี้

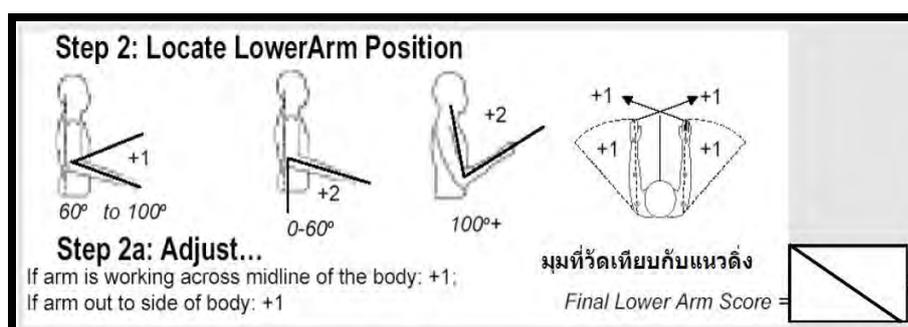
ขั้นตอนที่1 การประเมินตำแหน่งแขนส่วนบน (Upper Arm)



รูปที่ 2.1 แสดงการให้คะแนนท่าทางการทำงานของแขนส่วนบน

- 1.1 ระดับของแขนการยกที่สูงขึ้นระดับคะแนนที่ให้มากขึ้น คะแนนอยู่ระหว่าง-4
- 1.2 ถ้ามีการยกของไหลให้บวกคะแนนเพิ่มอีก+1
- 1.3 ถ้ามีการกางแขนให้บวกคะแนนเพิ่มอีก+1
- 1.4 ถ้าแขนมีที่รองรับหรือวางพาดอยู่ให้ลบคะแนน-1
- 1.5 คะแนนสูงสุดของขั้นตอนนี้ จะมีค่าไม่เกิน6 คะแนน
- 1.6 ให้แยกการประเมินแขนซ้ายและขวา

ขั้นตอนที่2 การประเมินตำแหน่งแขนส่วนล่าง (Lower Arm หรือ Forearm)

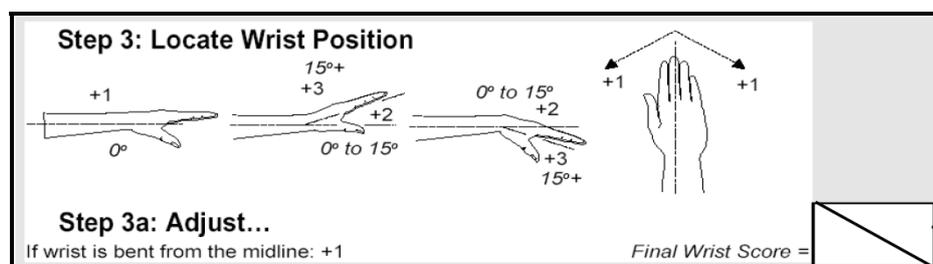


รูปที่ 2.2 แสดงการให้คะแนนท่าทางการทำงานของแขนส่วนล่าง

2.1 ระดับของแขนส่วนล่างควรอยู่ในแนวระดับขณะทำงานหรืออยู่ในช่วงประมาณ 60-100 องศาจากแนวตั้งถ้ามุมของแขนส่วนล่างอยู่นอกช่วงดังกล่าวให้คะแนนตามรูปที่ 1 และ 2 จากซ้าย

- 2.2 ถ้ามีการทำงานไขว้แขนเลยแกนกลางลำตัวให้บวกคะแนนเพิ่มอีก 1
- 2.3 ถ้ามีการทำงานในลักษณะกางแขนออกไปด้านข้างลำตัวให้บวกคะแนนเพิ่มอีก 1
- 2.4 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน 4 คะแนน
- 2.5 ให้แยกประเมินระหว่างแขนซ้ายและขวา

ขั้นตอนที่ 3 การประเมินตำแหน่งของมือและข้อมือ (Hand และ Wrist)



รูปที่ 2.3 แสดงการให้คะแนนของมือและการบิดของข้อมือ

3.1 ขณะทำงานข้อมือไม่ควรอยู่ในลักษณะตรงไม่บิดงอ ดังแสดงในรูปที่ 1 จากซ้าย ถ้าข้อมือมีการบิดงอจะให้คะแนนตามรูปที่ 1 (flexion) และ 2 (extension) จากซ้าย

3.2 ถ้ามีการทำงานที่เกิดการเบี่ยงข้อมือออก (deviation) ดังแสดงในรูปที่ 4 ทางซ้าย ให้บวกคะแนนเพิ่มอีก +1

- 3.3 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน 4 คะแนน
- 3.4 ให้แยกประเมินระหว่างแขนซ้ายและขวา

ขั้นตอนที่ 4 การประเมินตำแหน่งของมือและข้อมือ (Wrist Twist)

- 4.1 ขณะทำงานข้อมือไม่ควรหมุนถ้ามีการหมุนข้อมือให้คะแนนเป็น
- 4.2 ถ้ามีการทำงานที่หมุนข้อมือมากเกือบสุดให้คะแนนเป็น 2
- 4.3 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน 2 คะแนน

4.4 ให้แยกประเมินระหว่างแขนซ้ายและขวา

ขั้นตอนที่ 5 สรุปผลจากขั้นตอนที่ 1-4 โดยใช้ตาราง A

นำข้อมูลจากขั้นตอนที่ 1-4 ซึ่งเป็นผลจากการวิเคราะห์ท่าทางของแขนและมือในขณะทำงานมาเปิดค่าคะแนนรวมในตาราง A

ตารางที่ 2.3 ตารางคะแนน A โดยใช้ท่าทางของส่วนของร่างกายในกลุ่ม A

		คะแนนข้อมือ							
แขนส่วนบน	แขนส่วนล่าง	1		2		3		4	
		บิดข้อมือ		บิดข้อมือ		บิดข้อมือ		บิดข้อมือ	
		1	2	1	2	1	2	1	2
1	1	1	2	2	2	2	3	3	3
	2	2	2	2	2	3	3	3	3
	3	2	3	3	3	3	3	4	4
2	1	2	3	3	3	3	4	4	4
	2	3	3	3	3	3	4	4	4
	3	3	4	4	4	4	4	5	5
3	1	3	3	4	4	4	4	5	5
	2	3	4	4	4	4	4	5	5
	3	4	4	4	4	4	5	5	5
4	1	4	4	4	4	4	5	5	5
	2	4	4	4	4	4	5	5	5
	3	4	4	4	5	5	5	6	6
5	1	5	5	5	5	5	6	6	7
	2	5	6	6	6	6	7	7	7
	3	6	6	6	7	7	7	7	8
6	1	7	7	7	7	7	8	8	9
	2	8	8	8	8	8	9	9	9
	3	9	9	9	9	9	9	9	9

ขั้นตอนที่ 6 ประเมินระดับการใช้แรงของกล้ามเนื้อในการทำงาน

6.1 ถ้าการทำงานดังกล่าวมีลักษณะการใช้แรงจากกล้ามเนื้อแบบสถิตเช่นมีการใช้แรงโดยเกร็งกล้ามเนื้อต่อเนื่องนานกว่า 1 นาทีให้ใส่คะแนนเป็น 1

6.2 ถ้าการทำงานเป็นแบบซ้ ๆ โดยมีมีการเคลื่อนไหวกลับไปกลับมาเกินกว่าครั้งต่อนาทีหรือมากกว่าให้บวกคะแนนเพิ่มอีก

6.3 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน 2 คะแนน

ขั้นตอนที่7 ประเมินภาระงานที่ทำ

7.1 ภาระงานที่ทำให้แก่แรงที่ใช้หรือน้ำหนักที่ถือถ้าน้อยกว่า 1 กิโลกรัมให้คะแนนเป็น 0

7.2 ถ้าภาระงานอยู่ระหว่าง 2-10 กก. ถือหรือใช้แรงนานๆ ครั้งให้คะแนนเป็น 1

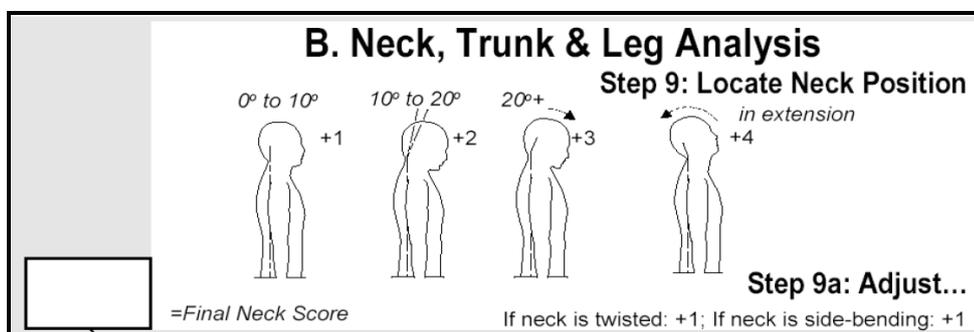
7.3 ถ้าภาระงานอยู่ระหว่าง 2-10 กก. ถือหรือใช้แรงตลอดเวลาหรือทำซ้ำๆ ไปมาบ่อยๆ ให้คะแนนเป็น 2

7.4 ถ้าภาระงานมากกว่า 10 กก. ถือหรือใช้แรงแบบสลับหรือเคลื่อนที่ซ้ำๆ ไปมาบ่อยๆ หรือมีการใช้แรงทำงานดังกล่าวอย่างรวมเร็วให้คะแนนเป็น 3

ขั้นตอนที่8 สรุปผลคะแนนการวิเคราะห์แขนและข้อมือ

รวมผลคะแนนจากขั้นตอนที่ 5 – 7 ไว้ในขั้นตอนนี้เพื่อใช้เปิดตาราง C ในการประเมินผลร่วมกับร่างกายส่วนที่เหลือ

ขั้นตอนที่9 การวิเคราะห์ท่าทางของศีรษะและคอ (Head and Neck)



รูปที่ 2.4 แสดงการให้คะแนนท่าทางการทำงานในส่วนของคอ

9.1 ถ้ามุมก้มอยู่ระหว่าง 0-10 องศาให้คะแนนเป็น 1

9.2 ถ้ามุมก้มอยู่ระหว่าง 10-20 องศาให้คะแนนเป็น 2

9.3 ถ้ามุมก้มมากกว่า 20 องศาขึ้นไปให้คะแนนเป็น 3

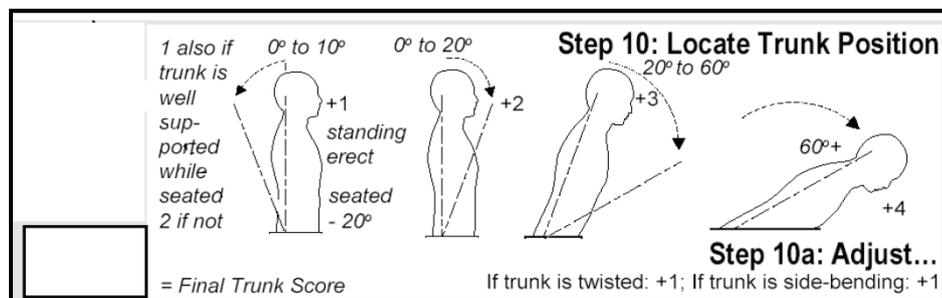
9.4 ถ้ามีการเงยศีรษะให้คะแนนเป็น 4

9.5 ถ้ามีการหมุน (twist) ศีรษะด้วยให้คะแนนเพิ่มอีก +1

9.6 ถ้ามีการเอียงศีรษะไปด้านข้างให้คะแนนเพิ่มอีก +1

9.7 คะแนนสูงสุดในขั้นตอนนี้ มีค่าไม่เกิน 6 คะแนน

ขั้นตอนที่10 การวิเคราะห์ตำแหน่งของลำตัว(Trunk)



รูปที่ 2.5 แสดงการให้คะแนนท่าทางการทำงานในส่วนของลำตัว

10.1 ลำตัวควรอยู่ในลักษณะที่ตั้งตรงเมื่อยืนหรือในกรณีการนั่งมีพนักพิงรองรับอย่างดีที่มุมเอียงไม่เกิน -20 องศาให้คะแนนเป็น 1

10.2 ลำตัวโน้มไปด้านหน้าระหว่าง-20 องศาให้คะแนนเป็น 2

10.3 ลำตัวโน้มไปด้านหน้าระหว่าง 21-60 องศาให้คะแนนเป็น 3

10.4 ลำตัวโน้มไปด้านหน้าระหว่างมากกว่า 60 องศาให้คะแนนเป็น 4

10.5 ลำตัวมีการหมุนให้คะแนนเพิ่มอีก+1

10.6 ลำตัวมีการเอียงไปด้านข้างให้คะแนนเพิ่มอีก +1

10.7 คะแนนสูงสุดในขั้นตอนนี้ มีค่าไม่เกิน 6 คะแนน

ขั้นตอนที่11 การประเมินท่าทางของขาและเท้า(Leg and Foot)

11.1 ขาอยู่ในลักษณะสมดุลช่วยขาโดยเท้าสามารถวางบนพื้นที่มีการรองรับดีให้คะแนนเป็น 1

11.2 ถ้าไม่สมดุลหรือพื้นรองรับเท้าไม่ดีให้คะแนนเป็น 2

11.3 คะแนนสูงสุดในขั้นตอนนี้ ไม่เกิน 2 คะแนน

ขั้นตอนที่12 สรุปผลการให้คะแนนท่าทางของขั้นตอนที่9 – 11 โดยใช้ตาราง B

ตาราง B เป็นการสรุปผลท่าทางของศีรษะลำตัวขาและเท้าโดยใช้ข้อมูลที่ได้จากขั้นตอนที่ 9 10 และ 11 มาเปิดตาราง B

ตารางที่ 2.4 ตารางคะแนน B โดยใช้ท่าทางของส่วนร่างกายในกลุ่ม B

คะแนนลำตัว													
ลำดับ	1		2		3		4		5		6		
	ขา		ขา		ขา		ขา		ขา		ขา		
	1	2	1	2	1	2	1	2	1	2	1	2	
1	1	3	2	3	3	4	5	5	6	6	7	7	
2	2	3	2	3	4	5	5	5	6	7	7	7	
3	3	3	3	4	4	5	5	6	6	7	7	7	
4	5	5	5	6	6	7	7	7	7	7	8	8	
5	7	7	7	7	7	8	8	8	8	8	8	8	
6	8	8	8	8	8	9	9	9	9	9	9	9	

ขั้นตอนที่13 ประเมินระดับการใช้แรงของกล้ามเนื้อในการทำงาน

13.1 ถ้าการทำงานดังกล่าวมีลักษณะการใช้แรงจากกล้ามเนื้อแบบสถิต เช่น มีการใช้แรงโดยเกร็งกล้ามเนื้อต่อเนื่องนานกว่า1 นาทีให้ใส่คะแนนเป็น 1

13.2 ถ้าการทำงานเป็นแบบซ้ำ ๆ โดยมีการเคลื่อนไหวกลับไปกลับมาเกินกว่า 4 ครั้งต่อ นาทีหรือมากกว่าให้บวกคะแนนเพิ่มอีก

13.3 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน2 คะแนน

ขั้นตอนที่14 ประเมินภาระงานที่ทำ

14.1 ภาระงานที่ทำได้แก่แรงที่ใช้หรือน้ำหนักที่ถือถ้าน้อยกว่ากิโลกรัมให้คะแนนเป็น 0

14.2 ถ้าภาระงานอยู่ระหว่าง 2-10 กก. ถือหรือใช้แรงนานๆ ครั้งให้คะแนนเป็น1

14.3 ถ้าภาระงานอยู่ระหว่าง2-10 กก. ถือหรือใช้แรงตลอดเวลาหรือทำซ้ำๆไปมาบ่อยๆ ให้คะแนนเป็น 2

14.4 ถ้าภาระงานมากกว่า10 กก. ถือหรือใช้แรงแบบสถิตหรือเคลื่อนที่ซ้ำๆไปมาบ่อยๆ หรือมีการใช้แรงทำงานดังกล่าวอย่างรวมน้ำหนักให้คะแนนเป็น 3

ขั้นตอนที่15 สรุปผลการวิเคราะห์ศรียะ คอ ลำตัว และเท้า

เป็นผลรวมคะแนนจากขั้นตอนที่12 ซึ่งได้จากการเปิดตารางB รวมกับคะแนนในขั้นตอนที่ 13 และ 14 ซึ่งเป็นลักษณะการใช้งานกล้ามเนื้อและภาระงานที่ต้องทำคะแนนรวมที่ได้ใส่ไว้ในขั้นตอนนี้เพื่อนำไปเปิดตารางสรุปผลของ RULA ในตาราง C

3ตารางที่ 2.5 ตารางคะแนนผลลัพธ์โดยใช้คะแนนC และ D

คะแนนสรุปจากขั้นตอนที่15							
คะแนนสรุปจากขั้นตอนที่8	1	2	3	4	5	6	7
1	1	2	3	3	4	5	5
2	2	2	3	4	4	5	5
3	3	3	3	4	4	5	6
4	3	3	3	4	5	6	6
5	4	4	4	5	6	7	7
6	4	4	5	6	6	7	7
7	5	5	6	6	7	7	7
8	5	5	6	7	7	7	7

ขั้นตอนที่16 หรือขั้นสุดท้ายคือการสรุปผลระดับคะแนนของRULA ในตาราง C

16.1 นำค่าที่ได้ในขั้นตอนที่8 และคะแนนที่ได้ในขั้นตอนที่15 ไปใช้ในการเปิดตาราง C

16.2 โดยคะแนนในขั้นตอนที่8 ใช้เลือกตำแหน่งของแถวส่วนคะแนนในขั้นตอนที่ 15 ใช้เลือกตำแหน่งของคอลัมน์ช่องที่ตัดกันระหว่างคะแนนทั้งสองในตารางC เป็นระดับคะแนนสุดท้ายของ RULA

16.3 คะแนน RULA จะมีค่าอยู่ระหว่าง 1-7 คะแนนที่สูงกว่าหมายถึงความเสี่ยงต่อปัญหาทางด้านกายศาสตร์มีสูงด้วย

การสรุปผลการประเมินท่าทางการทำงานโดยใช้วิธีกร RULA สามารถแบ่งออกได้เป็น 4 ระดับ ดังนี้

ระดับ 1: คะแนนอยู่ที่ 1-2 งานนี้ยอมรับได้แต่อาจเป็นมีปัญหาด้านกายศาสตร์ได้ ถ้ามีการทำงานดังกล่าวซ้ำ ๆ ต่อเนื่องเป็นเวลานานกว่าเดิม

ระดับ 2: คะแนนอยู่ที่ 3-4 งานนี้ควรได้รับการพิจารณาการศึกษาละเอียดขึ้นและติดตามวัดผลอย่างต่อเนื่องการออกแบบงานใหม่อาจมีความจำเป็น

ระดับ 3: คะแนนอยู่ที่ 5-6 งานนี้เริ่มเป็นปัญหาควรทำการศึกษาเพิ่มเติมและควรรีบ
ดำเนินการปรับปรุงลักษณะงานดังกล่าว

ระดับ 4: คะแนนตั้งแต่ 7 ขึ้นไปงานนี้มีปัญหาด้านการยศาสตร์ที่ต้องได้รับการปรับปรุง
โดยทันที

2.1.5 วิธีการประเมินภาระการต้น ดิ่ง โดยใช้ตารางของ Snook [4], [5], [6], [7]

Snook ได้ศึกษาและจัดทำตาราง การผลักรถ การแบก-หิ้ว ขึ้น โดยเมื่อพิจารณาจากตาราง
ของ Snook จะสามารถเปิดตารางหาค่าแรงที่เหมาะสมกับการทำงานในลักษณะดังกล่าวได้ซึ่งมี
รายละเอียดที่แตกต่างกันไปในแต่ละงาน โดยขั้นตอนในการประเมินความเสี่ยงในงานแบก ผลัก
ลาก โดยใช้ตารางของ Snook มีดังนี้

ขั้นตอนที่ 1 เลือกงานที่ต้องการประเมิน

งานผลัก (PUSH)

งานลาก (PULL)

งานแบก (CARRY)

ขั้นตอนที่ 2 เลือกเพศของพนักงาน

ผู้ชาย (MALE)

ผู้หญิง (FEMALE)

ขั้นตอนที่ 3 เปิดตารางเพื่อหาค่าแรงที่เหมาะสมในการทำงาน

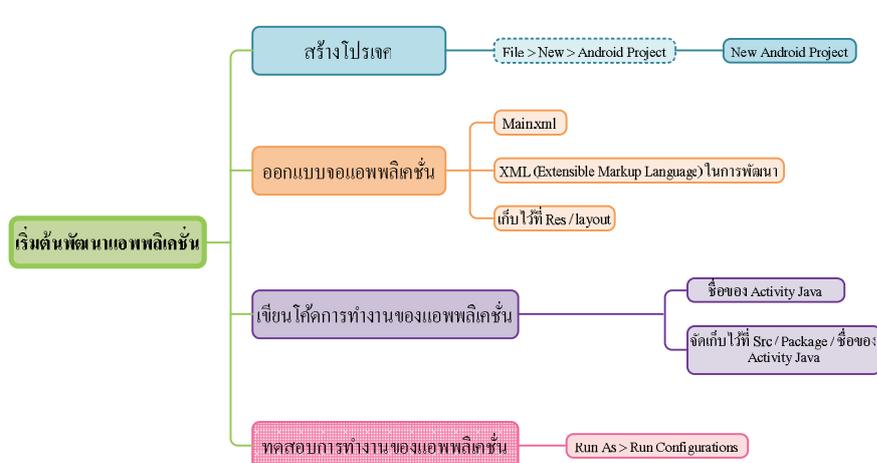
งานผลักและงานลาก (Push and Pull)

1. วัดระดับความสูงจากพื้นถึงจุดออกแรง
2. เลือกระยะทางในการปฏิบัติงาน
3. หาค่าแรงในตารางตามความสูง ระยะทาง เพศและจำนวนรอบทำงาน
4. หาร้อยละของกลุ่มประชากรที่ต้องการให้ปฏิบัติงานนี้ได้โดยไม่เกิดความเครียด

งานหิ้ว ถือ แบก (Carry)

5. วัดระดับความสูงจากพื้นถึงจุดที่ยก
6. เลือกระยะทางในการปฏิบัติงาน
7. หาค่าน้ำหนักในตารางตามความสูง ระยะทาง เพศและจำนวนรอบทำงาน
8. หาร้อยละของกลุ่มประชากรที่ต้องการให้ปฏิบัติงานนี้ โดยไม่เกิดความเครียด

2.1.6 การสร้างแอปพลิเคชันบนแอนดรอยด์



ขั้นตอนของการพัฒนาแอปพลิเคชันบนแอนดรอยด์ มี ขั้นตอน ดังนี้

1. สร้างโปรเจกต์ ในการพัฒนาแอปพลิเคชัน จะต้องมียกที่หนึ่งสำหรับ เก็บไฟล์โฟลเดอร์ต่างๆ ที่ โค้ดแอนดรอยด์ ที่จะกำลังจะเขียนขึ้นต้องการ
2. ออกแบบจอแอปพลิเคชัน ก่อนที่เราจะเข้าไป Coding ต้องทำการออกแบบ user Interface หรือ ที่เรียกกันว่า UI ก่อน ภาษาที่ใช้ในการพัฒนาก็จะใช้ Extensible Markup Language หรือที่เราคุ้นเคยเรียกว่า XML โดยจะมีไฟล์ที่ชื่อว่า main.xml ที่เก็บไว้ที่ res/layout เป็นตัวควบคุม layout ต่างๆ
3. เขียนโค้ดการทำงานของแอปพลิเคชัน ส่วนนี้เป็นขั้นตอนการ Coding ให้แอปพลิเคชันบนแอนดรอยด์ ทำงานตามที่ต้องการ โดยจะเก็บโค้ดไว้ใน src/ package / ไฟล์.java โดย จะใช้ Java เป็นภาษาในการพัฒนา

4. ทดสอบโค้ดที่เขียนได้จะทดสอบทั้งใน AVD (android virtual device) และ บนมือถือแอนดรอยด์จริงด้วย

2.2 งานวิจัยที่เกี่ยวข้อง

ชัยยุทธ วงศ์อัจฉริยา(พ.ศ.2550) งานวิจัยนี้ได้ศึกษาการเปรียบเทียบวิธีการประเมินภาระทางการยศาสตร์ในสภาวะแวดล้อมการทำงานในวิสาหกิจขนาดกลางและขนาดย่อม โดยทำการประเมินวิเคราะห์ท่าทางการทำงานของพนักงาน โดยใช้วิธีทางการยศาสตร์แบบต่างๆ เพื่อที่จะเปรียบเทียบผลการประเมินจากวิธีการประเมินในแต่ละวิธี โดยผลการศึกษาชี้ให้เห็นว่าวิธีการประเมินทางการยศาสตร์ในแต่ละวิธีนี้มีผลการประเมินที่แตกต่างออกไปตามหลักการของแต่ละวิธีและมีความเหมาะสมในการที่จะนำมาใช้ในการประเมินลักษณะการทำงานในสภาวะแวดล้อมการทำงานในวิสาหกิจขนาดกลางและขนาดย่อมและเมื่อพิจารณาเปรียบเทียบเรื่องของปัจจัยด้านความน่าเชื่อถือและความถูกต้องแม่นยำของผลการประเมินสามารถสรุปได้ว่า วิธีการประเมิน RULA REBA และ Strain Index มีความเหมาะสมในการนำมาใช้ประเมินการทำงานของวิสาหกิจไทยเพื่อเกิดการตระหนักถึงภาระทางการยศาสตร์ในการทำงานที่เกิดขึ้นและนำไปสู่การปรับปรุงท่าทางการทำงาน เพื่อลดความเสี่ยงต่อการบาดเจ็บจากการทำงานลงได้

ธนิยา เกตุเทศ (พ.ศ.2540) งานวิจัยนี้ได้ศึกษาการวิเคราะห์ท่าทางการทำงานโดยใช้วิธีการ RULA ในอุตสาหกรรมสุรา เพื่อศึกษาปัญหาและสภาพโดยทั่วไปในการทำงานของคนงานและประเมินท่าทางการทรงตัวด้วยวิธีการ RULA ซึ่งจากการศึกษาพบว่าระดับคะแนน RULA ของกิจกรรมบ่อล้าง กิจกรรมเรียงขวดใส่กระสอบ กิจกรรมเหยียงเหล้าและกิจกรรมยกถังเข้าเครื่องล้างอยู่ในระดับที่ต้องให้ความสนใจและปรับปรุงวิธีการทำงานในเรื่องการเอื้ออำนวยของกิจกรรมต่างๆ ที่อยู่เกินระยะเอื้อของแขน ควรปรับปรุงให้ระยะการเอื้อมอยู่ในระดับที่เหมาะสมและปลอดภัยในเรื่องของการก้มตัวคนงานไม่ควรพับเอวลงมา ควรจะย่อจากระดับสะโพกหรือย่อลงมา ซึ่งจะช่วยให้การทรงตัวนั้นเกิดความสมดุลและสามารถที่จะรับภาระที่จะเกิดขึ้นกับส่วนต่างๆ ของร่างกายได้

ประจวบ กล่อมจิตร (พ.ศ.2548) งานวิจัยนี้ ได้ศึกษาการสำรวจงานยกของด้วยมือโดยใช้ สมการ NIOSH 1991 ซึ่งสามารถใช้ประเมินงานที่เป็นการยกของในเงื่อนไขต่างๆ ได้ เพื่อนำค่าที่ได้ไปใช้ในการวิเคราะห์ความเหมาะสมท่าทางในการยกของเพื่อเสนอเป็นแนวทางในการปรับปรุงผลการสำรวจแล้วนำไปวิเคราะห์ในสมการ NIOSH 1991 ทำให้ได้แนวทางในการปรับปรุงงาน โดยแนวทางในการปรับปรุงงานด้านการเปลี่ยนแปลงสถานงาน ท่าทางในการทำงานที่เหมาะสม อุปกรณ์ที่ช่วยในการผ่อนแรงการยกของซึ่งพิจารณาจากค่าใช้จ่ายในการสร้างเครื่องและน้ำหนักสูงสุดที่พนักงานจะสามารถยกได้สูงสุด โดยไม่เกิดอันตรายในการยก

เสกสิทธิ์ ขำวิเศษ(พ.ศ.2541) งานวิจัยนี้ ได้ศึกษาปัญหาและสภาพทั่วไปในการทำงานของผู้ปฏิบัติงานทอผ้า โดยการวิเคราะห์และปรับปรุงท่าทางการทำงาน โดยวิธีการ RULA กรณีศึกษาในอุตสาหกรรมทอผ้าพื้นเมืองภาคเหนือ โดยมี 3 จังหวัด คือ สุโขทัย ลำพูนและเชียงใหม่ โดยวิเคราะห์ท่าทางการทำงานและอุปกรณ์ในการใช้ในงานทอผ้าที่ไม่เหมาะสมกับขนาดสัดส่วนของร่างกาย ซึ่งส่งผลกระทบต่อความเจ็บปวดของกล้ามเนื้อที่ใช้ในการทำงาน โดยเสนอแนะวิธีการทำงานรวมทั้งการออกแบบอุปกรณ์ที่ใช้ในการทอผ้าให้มีความถูกต้องและมีเหมาะสมกับขนาดสัดส่วนของผู้ปฏิบัติงานทอผ้าตามหลักการยศาสตร์

นายชัยพร วราวุฒิ (พ.ศ.2554) งานวิจัยนี้ ได้ศึกษาพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการประเมินภาระงานทางการยศาสตร์ โดยรูปแบบของโปรแกรมจะแตกต่างกันไปตามวิธีการที่ใช้ในการประเมินภาระงาน ซึ่งได้แก่ โปรแกรม RulaSU สำหรับการประเมินท่าทางในการปฏิบัติงาน โดยใช้วิธีการ RULA โปรแกรม NioshSU สำหรับการประเมินภาระงานยกของด้วยมือโดยใช้สมการ NIOSH 1991 และโปรแกรม SnookSU สำหรับการประเมินภาระงานผลัก งานลาก และงานแบก โดยใช้ตารางของ Snook

บทที่ 3

วิธีดำเนินการวิจัย

การจัดทำวิทยานิพนธ์เรื่อง การพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์นี้ เป็นการศึกษาเกี่ยวกับการนำวิธีการประเมินภาระงานทางการยศาสตร์ต่างๆ ได้แก่ การประเมินท่าทางการทำงานโดยใช้วิธีการRULA การประเมินงานยกของด้วยมือโดยสมการ NIOSH 1991 มาเป็นหลักการในการพื้นฐานเพื่อนำไปใช้ในการออกแบบและพัฒนาเป็นแอปพลิเคชันแอนดรอยด์ โดยใช้โปรแกรมJava JDK , Eclipse , Android SDK และ Android Developer Tools ในการออกแบบและพัฒนาโปรแกรม

การพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์มีขั้นตอนการดำเนินงานหลากหลายขั้นตอน ดังนั้นการคำนึงถึงวัตถุประสงค์ของการดำเนินงาน เพื่อให้การดำเนินงานเป็นไปอย่างมีประสิทธิภาพนั้น การกำหนดแผนงานและวิธีการให้เป็นขั้นเป็นตอน นับว่ามีความสำคัญมากอย่างหนึ่งดังนั้นขั้นตอนการดำเนินงานในการออกแบบและพัฒนาโปรแกรมการประเมินภาระงานตามหลักการทางการยศาสตร์นั้น จึงประกอบไปด้วยขั้นตอนในการทำงาน 8 ขั้นตอน ดังนี้

3.1 ศึกษาเบื้องต้นเกี่ยวกับหลักการและทฤษฎีที่เกี่ยวข้องต่างๆของการประเมินภาระงานตามหลักการทางการยศาสตร์

3.2 ศึกษางานวิจัยที่เกี่ยวข้อง

3.3 ออกแบบและพัฒนาโปรแกรมโดยใช้โปรแกรมJava JDK , Eclipse , Android SDK และ Android Developer Tools

3.4 สอบทานการออกแบบโปรแกรมเพื่อค้นหาข้อผิดพลาดที่เกิดขึ้น

3.5 การทดลองใช้งานโปรแกรม

3.6 ประเมินผลการใช้งานของโปรแกรม

3.7 สรุปผลการดำเนินงาน

3.8 จัดทำรายงานสรุปผลการศึกษา

3.1 ศึกษาเบื้องต้นเกี่ยวกับหลักการและทฤษฎีที่เกี่ยวข้องต่างๆ ของการประเมินภาระงานตามหลักการทางกายศาสตร์

3.1.1 ศึกษาหลักการและทฤษฎีที่เกี่ยวข้องกับการการประเมินท่าทางการทำงานโดยใช้วิธีการ RULA การประเมินงานยกของด้วยมือโดยใช้สมการ NIOSH1991 เพื่อเป็นแนวทางในการนำทฤษฎีและความรู้ที่เกี่ยวข้องมาประยุกต์ใช้ในการออกแบบและพัฒนาโปรแกรม

3.1.2 ศึกษาหลักการและทฤษฎีที่เกี่ยวข้องกับการเขียนโปรแกรมJava JDK , Eclipse ทั้งในส่วนของการขั้นตอนการสร้างโปรแกรม รวมถึงเทคนิคต่างๆ ที่เป็นประโยชน์ในการออกแบบและพัฒนาโปรแกรม

3.2 ศึกษางานวิจัยที่เกี่ยวข้อง

ทำการค้นคว้าหางานวิจัยที่เกี่ยวข้องเพื่อทำการศึกษารูปแบบและขั้นตอนต่างๆในการนำเสนอ ซึ่งในส่วนของงานวิจัยที่เกี่ยวข้อง จะครอบคลุมเนื้อหา ดังต่อไปนี้

3.2.1 งานวิจัยที่มีเนื้อหาครอบคลุมในส่วนของการประเมินภาระงานโดยใช้วิธีการตามหลักการทางกายศาสตร์

3.2.2 งานวิจัยที่มีเนื้อหาครอบคลุมในส่วนของการออกแบบและพัฒนาโปรแกรม

โดยงานวิจัยแต่ละงานวิจัยนั้น จะใช้ประกอบเป็นรูปแบบและแนวทางในการทำวิจัยควบคู่กันไปโดยตลอด

3.3 ออกแบบและพัฒนาโปรแกรม

ทำการออกแบบลักษณะการทำงานของโปรแกรม เพื่อเป็นการวางแผนและกำหนดรูปแบบของโปรแกรมที่จะใช้ในการประเมินภาระงานตามหลักการทางกายศาสตร์ โดยแนวทางในการออกแบบของโปรแกรมนั้นจะมีดังนี้

3.3.1 โปรแกรมดังกล่าวจะมีความง่ายต่อการใช้งานต่อผู้ที่สนใจ

3.3.2 ผลการประเมินของโปรแกรมนั้นจะต้องมีความถูกต้องสมบูรณ์เมื่อเทียบกับวิธีการคำนวณมือ

3.3.3 ผู้ที่เกี่ยวข้องจะสามารถปรับปรุงรูปแบบการทำงานของโปรแกรมได้ง่าย

3.4 สอบทานการออกแบบพัฒนาโปรแกรม

ทำการประเมินการทำงานของโปรแกรมที่พัฒนาขึ้นว่ามีความถูกต้องของผลการประเมินภาระงานมากน้อยเพียงใด ค้นหาข้อผิดพลาดที่กิดขึ้นเปรียบเทียบกับผลที่ได้จากการคำนวณมือกับการใช้งานแอปพลิเคชัน เพื่อให้แน่ใจได้ผลการประเมินที่ได้จากโปรแกรมมีความถูกต้อง พร้อมต่อการนำโปรแกรมไปทดลองใช้งานได้จริง

3.5 การทดลองการใช้งานโปรแกรม

ทำการทดสอบการใช้งาน โปรแกรมดังกล่าวโดยการนำไปใช้ในการประเมินภาระการทำงานจริง เพื่อศึกษาความเหมาะสมของรูปแบบของโปรแกรมในการนำไปใช้งานและศึกษาประสิทธิภาพในการใช้งานของโปรแกรมในการประเมินสภาพแวดล้อมการทำงาน โดยนำโปรแกรมดังกล่าวไปทดลองใช้งานกับกรณีศึกษาและนำข้อมูลดังกล่าวมาใช้ในการปรับปรุงรูปแบบการทำงานของโปรแกรมให้มีความเหมาะสมมากยิ่งขึ้น โดยมีขั้นตอนในการนำโปรแกรมไปทดลองใช้งานในโรงงานอุตสาหกรรมกรณีศึกษา ดังนี้

3.5.1 ให้การอบรมความรู้เบื้องต้นเกี่ยวกับความหมายและวิธีการประเมินภาระงานทางกายศาสตร์

3.5.2 สอนวิธีการทำการประเมินภาระงานทางกายศาสตร์ โดยใช้วิธีการคำนวณมือและการใช้โปรแกรมคอมพิวเตอร์ที่ได้ทำการออกแบบและพัฒนาช่วยในการประเมินภาระงาน

3.5.3 ทำการเก็บข้อมูลการใช้งาน โปรแกรมของพนักงาน

3.5.4 นำผลข้อมูลที่ได้มาสรุปผล

3.6 ประเมินผลการใช้งานของโปรแกรม

ทำการประเมินการใช้งานโปรแกรมดังกล่าวในการประเมินภาระงานทางกายศาสตร์ เมื่อเทียบกับวิธีการอื่นๆ เช่น การใช้การคำนวณมือ หรือการใช้โปรแกรมอื่นๆ ที่มีอยู่แล้ว เพื่อเปรียบเทียบรูปแบบการประเมิน โดยวิธีการประเมินว่ามีความเหมือนหรือแตกต่างกันอย่างไร เพื่อใช้ข้อมูลดังกล่าวในการปรับปรุงการทำงานของโปรแกรมต่อไป

3.6.1 เครื่องมือที่ใช้ในการศึกษา

การศึกษาโครงการครั้งนี้ คณะผู้จัดทำโครงการได้ออกแบบและสร้างเครื่องมือเพื่อใช้ในการเก็บข้อมูล เครื่องมือที่นำมาใช้เพื่อให้การเก็บข้อมูลเป็นไปอย่างมีประสิทธิภาพมีดังนี้

3.6.1.1 ฟังการไหล

เป็นการเก็บข้อมูลขั้นตอนในการทำงาน เพื่อให้ทราบถึงระยะเวลาและขั้นตอนในการปฏิบัติงาน

3.6.1.2 แบบฟอร์มประเมินโปรแกรม

เป็นการเก็บข้อมูลการประเมินโปรแกรม เพื่อให้ทราบถึงระดับความพึงพอใจของผู้ใช้งาน โดยมีเกณฑ์การให้คะแนนดังนี้

คะแนนที่ 5 หมายถึง มากที่สุด

คะแนนที่ 4 หมายถึง มาก

คะแนนที่ 3 หมายถึง ปานกลาง

คะแนนที่ 2 หมายถึง น้อย

คะแนนที่ 1 หมายถึง น้อยที่สุด

3.6.2 การสร้างแบบฟอร์มที่ใช้ในการเก็บข้อมูล

เครื่องมือที่ใช้ในการศึกษาโครงการในครั้งนี้ คือแบบฟอร์มประเมินโปรแกรมขั้นตอนในการสร้างเครื่องมือมีขั้นตอนดังนี้

3.6.2.1 ขั้นตอนการจัดสร้างแบบฟอร์มการประเมิน

1. ศึกษาขั้นตอนในการทำงาน
2. ออกแบบแบบฟอร์มการประเมิน โปรแกรม
3. พบผู้เชี่ยวชาญด้านการศึกษาเพื่อขอคำแนะนำเกี่ยวกับการออกแบบฟอร์มการประเมินโปรแกรม
4. ตรวจสอบแบบฟอร์มการประเมิน โปรแกรม
5. นำแบบฟอร์มการประเมินโปรแกรมที่ได้ไปทดลองเก็บข้อมูล และปรับปรุงแก้ไข

3.6.2.2 นำแบบฟอร์มการประเมิน โปรแกรมไปทำการบันทึกข้อมูล

3.6.4 การวิเคราะห์ข้อมูล

จะทำการแบ่งการวิเคราะห์ออกเป็น 2 ส่วนคือ

3.6.4.1 การวิเคราะห์ฝั่งการไหล

1. รวบรวมข้อมูลจากฝั่งการไหล
2. ศึกษาข้อมูลด้านเวลาจากฝั่งการไหล
3. สรุปผลการวิเคราะห์เพื่อเปรียบเทียบการทำงานในแต่ละวิธี

3.6.4.2 การวิเคราะห์แบบประเมินโปรแกรม

1. รวบรวมข้อมูลจากแบบประเมินโปรแกรม
2. สรุปผลการประเมินโปรแกรม

3.7 สรุปผลการดำเนินงาน

เมื่อทำการสร้างโปรแกรมและทดสอบแอปพลิเคชันเป็นที่เรียบร้อยแล้ว ผลการทดสอบโปรแกรม รวมถึงผลที่ประเมินออกมาที่มีประสิทธิภาพที่ยอมรับได้ จึงทำการรวบรวมข้อมูลต่างๆ เพื่อที่จะนำมาทำรายงานสรุปผลการศึกษา

3.8 จัดทำรายงานสรุปผลการศึกษา

จัดทำรูปเล่มปริชญานิพนธ์ที่ได้ดำเนินการพัฒนาแอปพลิเคชันตามแผนงานที่กำหนดไว้ พร้อมทั้งให้ข้อเสนอแนะต่างๆตลอดจนปัญหาในการทำงานเพื่อที่จะเป็นประโยชน์ต่อทางผู้ที่สนใจและเป็นข้อมูลในการทำวิจัยครั้งต่อไป

ตารางที่ 3.1 แสดงแผนการปฏิบัติงาน

หัวข้อกิจกรรม	ระยะเวลาดำเนินการ									
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
	56	56	56	56	56	56	56	57	57	57
1. ศึกษาทฤษฎี และหลักการประเมินงานทางการยศาสตร์										
2. ศึกษางานวิจัยที่เกี่ยวข้อง										
3. ออกแบบและพัฒนาโปรแกรมประเมินภาระการทำงาน										
4. สอบทานการออกแบบโปรแกรมเพื่อค้นหาข้อผิดพลาดที่เกิดขึ้น										
5. ทดลองใช้โปรแกรมดังกล่าวกับโรงงานในกรณีศึกษา										
6. ประเมินผลการใช้งานของโปรแกรม										
7. สรุปผลการดำเนินงาน										
8. จัดทำรายงานสรุปผลการวิจัย										

บทที่ 4

การวิเคราะห์ข้อมูลและอภิปรายผล

การจัดทำโครงการวิจัยได้ทำการศึกษาถึงการพัฒนาโปรแกรมประเมินภาระงานทางกายศาสตร์บนระบบปฏิบัติการแอนดรอยด์สำหรับการประเมินภาระงานตามหลักการทางกายศาสตร์ซึ่งได้แก่ โปรแกรม RulaSU สำหรับการประเมินท่าทางในการทำงานโดยใช้วิธีการ RULA สำหรับการประเมินภาระงานของโดยใช้สมการ NIOSH1991 การวิเคราะห์ผลการใช้งานโปรแกรมเพื่อนำไปสู่การออกแบบและจัดทำโปรแกรมให้ตรงกับลักษณะการใช้งานจึงมีความสำคัญต่อการออกแบบและพัฒนาโปรแกรมซึ่งมีการวิเคราะห์ข้อมูลดังต่อไปนี้

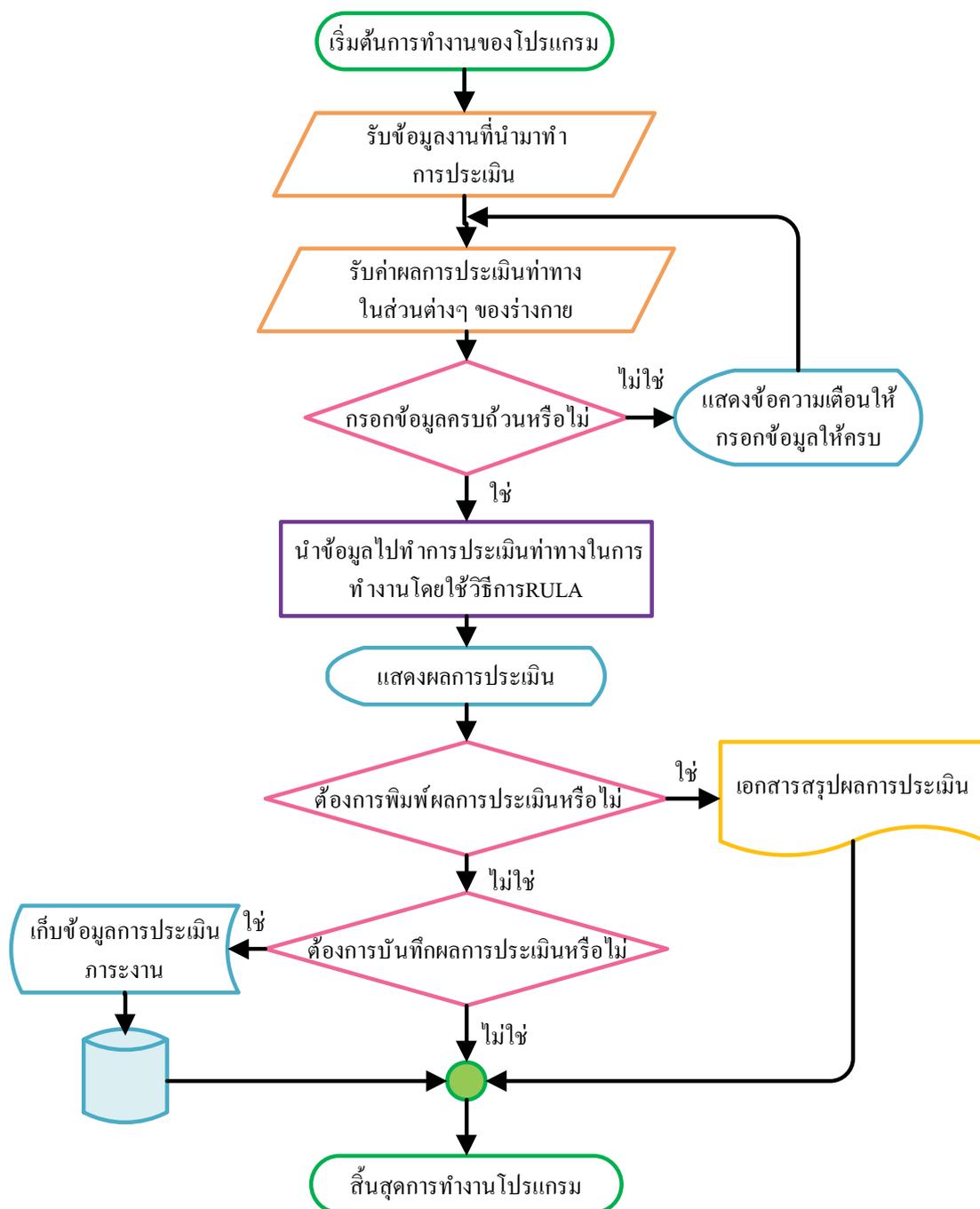
- 4.1 การออกแบบและการสร้างโปรแกรม
- 4.2 การประเมินประสิทธิภาพและคุณภาพของโปรแกรม
- 4.3 การประเมินข้อมูลด้านเวลาเชิงสถิติ
- 4.4 การนำโปรแกรมไปทดลองใช้งานในโรงงานอุตสาหกรรมกรณีศึกษา

4.1 ผลการออกแบบและการสร้างโปรแกรม

ผลจากการศึกษาข้อมูลต่างๆที่เกี่ยวข้องเพื่อใช้ในการออกแบบโครงสร้างการทำงานของโปรแกรม มีวัตถุประสงค์หลักคือ โปรแกรมจะต้องมีความถูกต้องของข้อมูลที่ได้จากการใช้งานและง่ายต่อการใช้งานของผู้ที่สนใจโดยทำการแยกโปรแกรมออกตามลักษณะการนำไปใช้งานในการประเมินภาระงานต่างๆ ดังนี้

- 4.1.1 โปรแกรม RulaSU สำหรับใช้ในการประเมินท่าทางในการทำงานโดยใช้วิธีการ RULA

4.1.1.1 ฟังก์ชันของโปรแกรม



รูปที่ 4.1 แสดงฟังก์ชันของโปรแกรมRulaSU

4.1.1.2 ลักษณะเปรียบเทียบของโปรแกรม

ตารางที่ 4.1 แสดงการเปรียบเทียบคุณสมบัติของโปรแกรมที่ใช้สำหรับการประเมินท่าทางในการทำงานโดยใช้วิธีการ RULA

โปรแกรม	Input	Output	หลักการที่ใช้ในการวิเคราะห์	ข้อจำกัดของโปรแกรม
RulaSU	1.ข้อมูลของผู้ทำการประเมิน 2.ค่าคะแนนการประเมินท่าทางในแต่ละส่วนของร่างกาย 3.ค่าคะแนนการใช้แรงและการใช้กล้ามเนื้อในการทำงาน 4.รูปภาพแสดงท่าทางการทำงานของงานที่นำมาประเมิน	ผลระดับการประเมินท่าทางในการทำงาน	RULA (Rapid Upper Limb Assessment)	ระบบปฏิบัติการต้องเป็นระบบปฏิบัติการแอนดรอยด์เท่านั้น
Ergo Intelligence	1.ข้อมูลของผู้ทำการประเมิน 2.ค่าคะแนนการประเมินท่าทางในแต่ละส่วนของร่างกาย 3.ค่าคะแนนการใช้แรงและการใช้กล้ามเนื้อในการทำงาน	ผลระดับการประเมินท่าทางในการทำงาน	RULA (Rapid Upper Limb Assessment)	1. มีการจำกัดระยะเวลาในการใช้งานโปรแกรม 2. มีปัญหาในการใช้งานบนระบบปฏิบัติการ Windows Vista / 7 หรือสูงกว่า

4.1.1.3 รายละเอียดของแบบฟอร์มหน้าต่างสำหรับแสดงผลของโปรแกรมRulaSU เมื่อได้ทำการออกแบบและสร้างโปรแกรมด้วยJava JDK , Eclipse , Android SDK และ Android Developer Tools เรียบร้อยแล้ว รูปร่างลักษณะของโปรแกรมจึงมีลักษณะดังต่อไปนี้

1. หน้าต่างเริ่มต้นของโปรแกรม



รูปที่ 4.2 แสดงหน้าต่างเริ่มต้นของโปรแกรมRulaSU

2. หน้าต่างรับข้อมูลของผู้ทำการประเมินและรูปท่าทางการทำงาน

รูปที่ 4.3 แสดงหน้าต่างรับข้อมูลของผู้ทำการประเมินและรูปท่าทางการทำงาน

3. หน้าต่างสำหรับการรับข้อมูลการให้คะแนนท่าทางในส่วนต่างๆของร่างกายเพื่อใช้ในการประเมินท่าทางในการทำงาน

การประเมินตำแหน่งแขนส่วนบน (หัวไหล่ถึงข้อศอก)

ยืนไหล่
 มีการทรงออกข้างลำตัวมาก
 แขนมีส่วนรับน้ำหนักหรือมีที่พัก

คะแนนที่ได้: 5

การประเมินตำแหน่งแขนส่วนล่าง (ข้อศอกถึงข้อมือ)

แขนทำงานเข้ามกลางลำตัว
 แขนกางออกข้างลำตัว

คะแนนที่ได้: 3

การประเมินตำแหน่งมือและข้อมือ

ประเมินข้อมือ

ไม่มีอาการของข้อมือ
 ข้อมือขึ้นหรือลง ไม่นเกิน 15
 ข้อมือขึ้นหรือลดต่ำกว่า 15

มือมีการเปลี่ยน โยจากแนวกลาง

ประเมินการยึดของข้อมือ

ข้อมือยึดน้อย
 ข้อมือยึดมาก

คะแนนที่ได้: 1

รูปที่ 4.4 แสดงหน้าต่างสำหรับรับข้อมูลของการให้คะแนนท่าทางในส่วนต่างๆของร่างกาย

การประเมินตำแหน่งคอ

0 ถึง 10°
 10° ถึง 20°
 มากกว่า 20°
 เตะศีรษะ

คอยืดหรือหมุน
 คอเอียงไปด้านข้าง

คะแนนที่ได้: 3

การประเมินตำแหน่งลำตัว

0 ถึง 10°
 10 ถึง 20°
 20 ถึง 60°
 มากกว่า 60°

คอยืดหรือหมุน
 คอเอียงไปด้านข้าง

คะแนนที่ได้: 3

มือ

การประเมินตำแหน่งขาและเท้า

รายการ: ตัวเลือก: **คะแนนที่ได้: 2**

ไบพีทีช่วยรองขาและเท้า

ประเมินการใช้แรงและกล้ามเนื้อ

ส่วนแขนและมือ
 ส่วนคอ ลำตัว และเท้า

ประเมินท่าทางการทำงานโดยใช้ FIMs
 อรรถาพยากรรมการทำงาน

รูปที่ 4.4 แสดงหน้าต่างสำหรับรับข้อมูลของการให้คะแนนท่าทางในส่วนต่างๆของร่างกาย(ต่อ)

4. หน้าต่างสำหรับการรับข้อมูลการประเมินการใช้แรงและกล้ามเนื้อในการทำงานของส่วนแขนและมือ

รูปที่ 4.5 แสดงหน้าต่างสำหรับการรับข้อมูลการประเมินการใช้แรงและกล้ามเนื้อในการทำงานของส่วนแขนและมือ

5. หน้าต่างสำหรับการรับข้อมูลการประเมินการใช้แรงและกล้ามเนื้อในการทำงานของส่วนคอ ลำตัวและเท้า

รูปที่ 4.6 แสดงหน้าต่างสำหรับการรับข้อมูลการประเมินการใช้แรงและกล้ามเนื้อในการทำงานของส่วนคอ ลำตัวและเท้า

6. หน้าต่างสำหรับแสดงผลการประเมินท่าทางในการทำงาน



รูปที่ 4.7 แสดงหน้าต่างสำหรับแสดงผลการประเมินท่าทางในการทำงาน

4.1.1.4 คู่มือการติดตั้งโปรแกรม

แสดงรายละเอียดอยู่ในภาคภาคผนวกข.

4.1.1.5 คู่มือการใช้งานโปรแกรม

แสดงรายละเอียดอยู่ในภาคภาคผนวกข.

4.2 ผลการประเมินประสิทธิภาพและคุณภาพของโปรแกรม

ทำการศึกษาถึงผลที่ได้รับจากการใช้งาน โปรแกรมหลังจากที่ได้มีการนำไปทดลองใช้งาน โดยทำการประเมินการใช้งานโปรแกรมทั้งในด้านประสิทธิภาพและคุณภาพใช้งานของโปรแกรม

โดยมีรายละเอียดของการศึกษาดังต่อไปนี้

4.2.1 การประเมินประสิทธิภาพของโปรแกรม

ทำการประเมินผลประสิทธิภาพในการใช้งานโปรแกรมด้วยวิธีการศึกษาการทำงาน โดยการเก็บข้อมูลขั้นตอนการใช้งานโปรแกรมแล้วจึงนำมาเปรียบเทียบกับขั้นตอนการทำงานกับโปรแกรมหรือวิธีการอื่นๆ ในการประเมินภาระงานตามหลักการทางกายศาสตร์ ซึ่งรายละเอียดการประเมินประสิทธิภาพของโปรแกรมจะแบ่งออกตามหลักการที่ใช้ในการประเมินดังนี้

4.2.1.1 ผลการประเมินการใช้งานโปรแกรม RulaSU

ตารางที่ 4.2 แสดงผลการเก็บข้อมูลและเปรียบเทียบข้อมูลระหว่างการใช้งาน โปรแกรม RulaSU
กับการใช้งานโปรแกรมErgoIntelligence

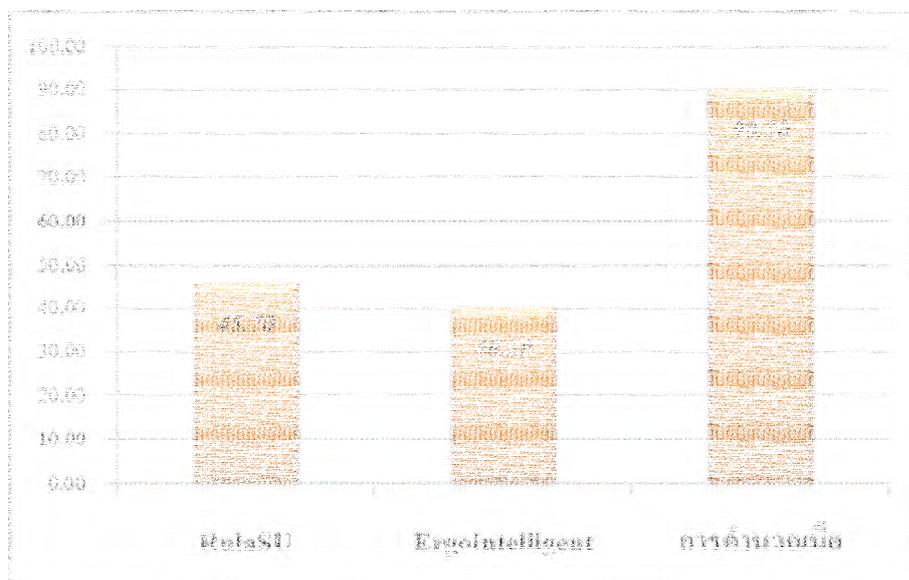
Activity	RulaSU	ErgoIntelligence	Saving
Operation 	10	8	-2
Transport 	0	0	0
Delay 	0	0	0
Inspection 	1	1	0
Storage 	0	0	0
Time (sec)	45.7	40.2	-5.5

ตารางที่ 4.3 แสดงผลการเก็บข้อมูลและเปรียบเทียบข้อมูลระหว่างการใช้งาน โปรแกรม RulaSU
กับวิธีการคำนวณมือ

Activity	RulaSU	วิธีการคำนวณมือ	Saving
Operation 	10	12	2
Transport 	0	0	0
Delay 	0	0	0
Inspection 	1	4	3
Storage 	0	0	0
Time (sec)	45.7	90.3	44.6

จากตารางที่ 4.2 เป็นการเก็บข้อมูลเพื่อทำการเปรียบเทียบระหว่างการใช้งานโปรแกรม RulaSU กับโปรแกรม ErgoIntelligence ในการประเมินท่าทางการทำงานโดยใช้วิธีการRULAซึ่งจากการใช้งานโปรแกรมRulaSU ใช้เวลาเฉลี่ยในการใช้งานโปรแกรมเพื่อประเมินงาน คือ 45.7 วินาที ส่วน โปรแกรม ErgoIntelligence นั้นใช้เวลาเฉลี่ย 40.2 วินาที ต่อกรณีศึกษาที่นำมาศึกษา 1 งาน

จากตารางที่ 4.3เป็นการเก็บข้อมูลเพื่อทำการเปรียบเทียบระหว่างการใช้งานโปรแกรม RulaSUกับวิธีการคำนวณมือในการประเมินท่าทางการทำงานโดยใช้วิธีการ RULA ซึ่งจากการใช้งานโปรแกรมRulaSUใช้เวลาเฉลี่ยในการใช้งานโปรแกรมเพื่อประเมินงาน คือ45.7วินาที ส่วนวิธีการคำนวณมือนั้นใช้เวลาเฉลี่ย90.3วินาที ต่อกรณีศึกษาที่นำมาศึกษา 1 งาน



รูปที่ 4.8 แสดงแผนภูมิแท่งแสดงเวลาที่ใช้ในการประเมินท่าทางในการทำงาน
โดยใช้วิธีการ RULA (วินาที)

4.2.2 การประเมินคุณภาพของโปรแกรม

ทำการประเมินคุณภาพของโปรแกรม โดยใช้ข้อมูลจากการทำแบบประเมินความพึงพอใจของผู้ใช้งาน โปรแกรมซึ่งข้อมูลก็นำมาพิจารณาแบ่งออกเป็น 3 ด้าน ดังนี้

1. ด้านความง่ายต่อการใช้งานระบบ (Usability Test)
2. ด้านการทำงานได้ตามฟังก์ชันงานของระบบ (Function Test)
3. ด้านการตรงตามความต้องการของผู้ใช้ระบบ (Functional Requirement Test)

โดยมีรายละเอียดผลการประเมินคุณภาพแบ่งแยกตามแต่ละโปรแกรมที่ได้ทำการพัฒนาดังต่อไปนี้

4.2.2.1 ผลการประเมินคุณภาพของโปรแกรม RulaSU

ตารางที่ 4.4 แสดงผลการประเมินคุณภาพในแต่ละด้านของโปรแกรม RulaSU

ปัจจัย	คะแนน
1. ด้านความง่ายต่อการใช้งานระบบ(Usability Test)	3.81
2. ด้านการทำงานได้ตามฟังก์ชันงานของระบบ(Function Test)	4.24
3. ด้านการตรงตามความต้องการของผู้ใช้ระบบ(Functional Requirement Test)	4.09
รวม	4.04

Usability test

Function

Requirement

Test

Function

Test

รูปที่ 4.9 แสดงผลการประเมินคุณภาพของ โปรแกรม RulaSU

1.การประเมินด้านความง่ายต่อการใช้งานระบบ(Usability Test)

ด้านความง่ายต่อการใช้งานระบบ จะแบ่งออกเป็น 2 ส่วน ดังนี้

1.การติดตั้งโปรแกรมสามารถทำการติดตั้งได้ง่ายไม่ซับซ้อน

2.การใช้งานโปรแกรม

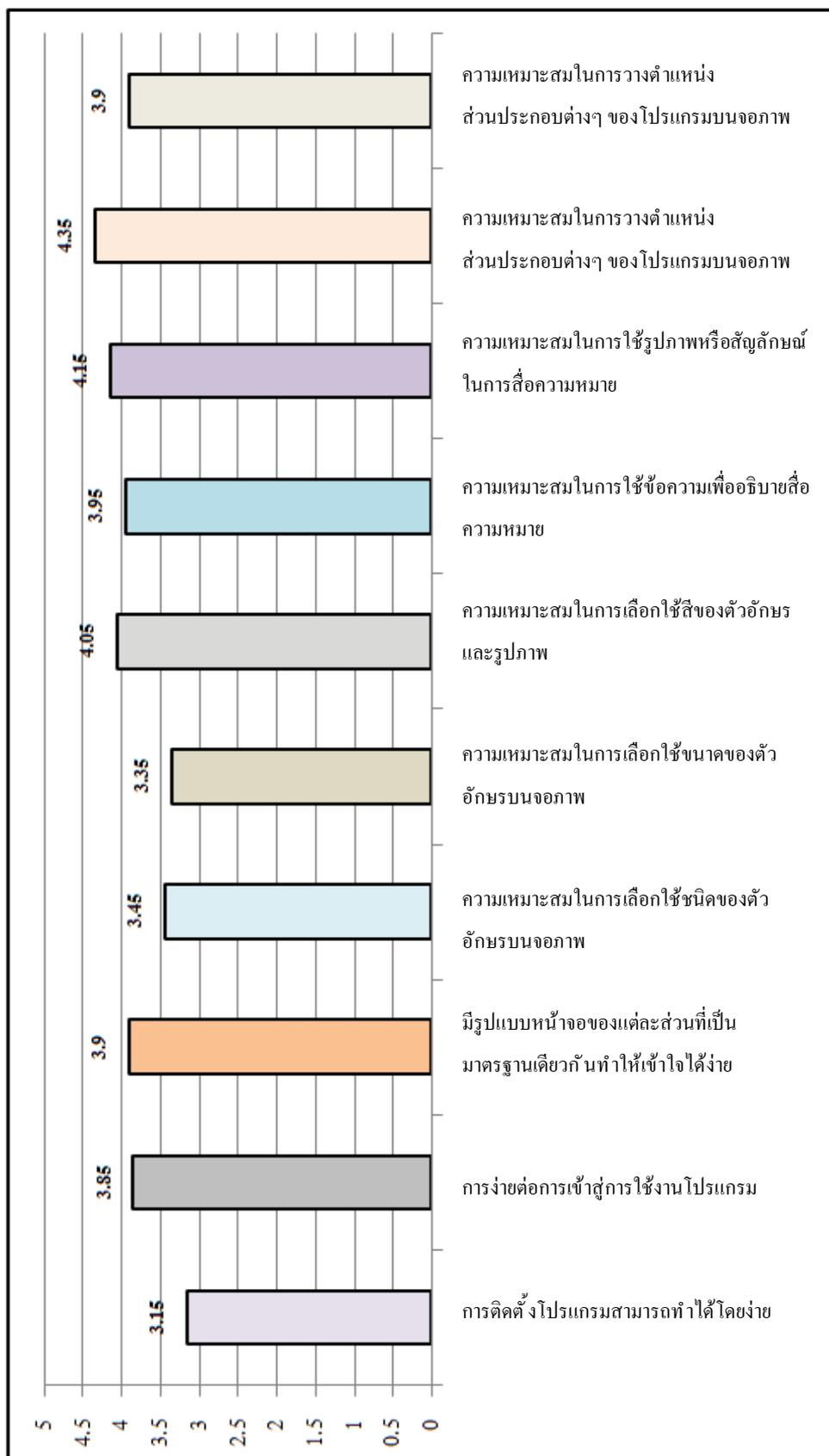
ในการประเมินด้านความง่ายต่อการใช้งานระบบในส่วนของการใช้งานโปรแกรม นั้น จะแบ่งออกเป็นปัจจัยต่างๆ คือ การง่ายต่อการเข้าสู่การใช้งานโปรแกรมมีรูปแบบหน้าจอของแต่ละส่วนที่เป็นมาตรฐานเดียวกันทำให้เข้าใจได้ง่ายเหมาะสมในการเลือกใช้นิพจน์ของตัวอักษรบนจอภาพความเหมาะสมในการเลือกใช้นิพจน์ของตัวอักษรบนจอภาพความเหมาะสมในการเลือกใช้นิพจน์ของตัวอักษรและรูปภาพความเหมาะสมในการใช้ข้อความเพื่ออธิบายสื่อความหมาย

เหมาะสมในการใช้รูปภาพหรือสัญลักษณ์ในการสื่อความหมายความเหมาะสมในการวางตำแหน่ง ส่วนประกอบต่างๆของโปรแกรมบนจอภาพและความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆของโปรแกรมบนจอภาพโดยได้กำหนดค่าน้ำหนักต่างๆ ดังแสดงในตารางที่ 4.10

ตารางที่ 4.5 แสดงผลการประเมินด้านความง่ายต่อการใช้งานระบบ(Usability Test)

เรื่อง	น้ำหนัก	คะแนน
ความคิดเห็นด้านความพึงพอใจต่อการใช้งานระบบ (Usability Test)		
1.การติดตั้งโปรแกรม	5	
1.1การติดตั้งโปรแกรมสามารถทำได้โดยง่าย	5	3.15
2.การใช้งานโปรแกรม	35	
2.1การง่ายต่อการเข้าสู่การใช้งานโปรแกรม	3	3.85
2.2มีรูปแบบหน้าจอของแต่ละส่วนที่เป็นมาตรฐานเดียวกันทำให้เข้าใจได้ง่าย	4	3.90
2.3ความเหมาะสมในการเลือกใช้นิพจน์ของตัวอักษรบนจอภาพ	4	3.45
2.4ความเหมาะสมในการเลือกใช้นาฬิกาของตัวอักษรบนจอภาพ	4	3.35
2.5ความเหมาะสมในการเลือกใช้สีของตัวอักษรและรูปภาพ	4	4.05
2.6ความเหมาะสมในการใช้ข้อความเพื่ออธิบายสื่อความหมาย	4	3.95
2.7ความเหมาะสมในการใช้รูปภาพหรือสัญลักษณ์ในการสื่อความหมาย	4	4.15
2.8ความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆของโปรแกรมบนจอภาพ	4	4.35
2.9ความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆของโปรแกรมบนจอภาพ	4	3.90
รวม	40	3.81

โดยผลของการประเมินแสดงดังรูปที่ 4.10



รูปที่ 4.10 แสดงผลการประเมินด้านความง่ายต่อการใช้งานระบบ (Usability Test) ของการใช้งานโปรแกรม RuLaSU

2. การประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test)

ด้านการทำงานได้ตามฟังก์ชันของระบบปัจจัยของการประเมินคือความถูกต้องในการจัดเก็บข้อมูลนำเข้าเพื่อใช้ในการประเมินความถูกต้องในการแก้ไขหรือเปลี่ยนแปลงข้อมูลนำเข้า ความถูกต้องในการแสดงผลลัพธ์ที่ได้จากการใช้งานโปรแกรมการป้องกันข้อผิดพลาดที่อาจเกิดขึ้นในการใช้งานและความรวดเร็วในการประมวลผลของโปรแกรมโดยได้กำหนดค่าน้ำหนักต่างๆ ดังแสดงในตารางที่4.11

ตารางที่ 4.6 แสดงผลการประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test)

เรื่อง	น้ำหนัก	คะแนน
ความคิดเห็นด้านการทำงานได้ตามฟังก์ชันของระบบ (Function Test)		
1. ความถูกต้องในการจัดเก็บข้อมูลนำเข้าเพื่อใช้ในการประเมิน	4	4.65
2. ความถูกต้องในการแก้ไขหรือเปลี่ยนแปลงข้อมูลนำเข้า	4	4.10
3. ความถูกต้องในการแสดงผลลัพธ์ที่ได้จากการใช้งานโปรแกรม	4	4.25
4. การป้องกันข้อผิดพลาดที่อาจเกิดขึ้นในการใช้งาน	4	3.65
5. ความรวดเร็วในการประมวลผลของโปรแกรม	4	4.55
รวม	20	4.24

โดยผลของการประเมินแสดงดังรูปที่4-24

รูปที่ 4.11 แสดงผลการประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test)
ของการใช้งานโปรแกรม RulaSU

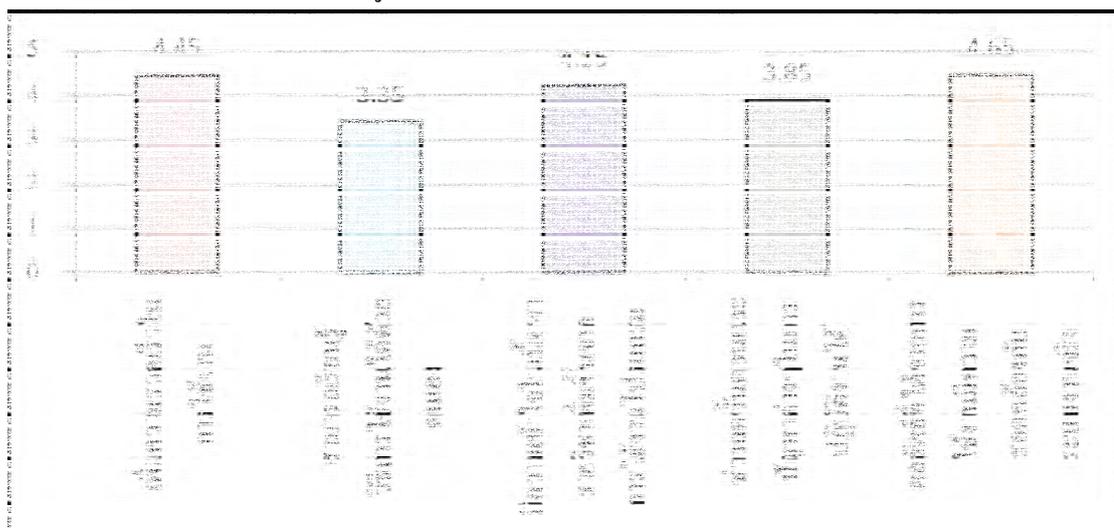
3. การประเมินด้านการตรงความต้องการของผู้ใช้ระบบ(Functional Requirement Test)

ด้านการทำงานได้ตามฟังก์ชันของระบบ ปัจจัยของการประเมิน คือเพิ่มความรวดเร็วในการใช้งานสามารถเรียนรู้วิธีการใช้งานได้ด้วยตนเองมีความเข้าใจในเนื้อหารายวิชาเพิ่มขึ้นจากการใช้งานโปรแกรมช่วยลดขั้นตอนและปัญหาต่างๆในการปฏิบัติงานได้และผลลัพธ์ที่ได้จากการใช้งานมีความสอดคล้องกับระบบงานจริงโดยได้กำหนดค่าน้ำหนักต่างๆดังแสดงในตารางที่.12

ตารางที่ 4.7แสดงผลการประเมินด้านการตรงความต้องการของผู้ใช้ระบบ
(Functional Requirement Test)

เรื่อง	น้ำหนัก	คะแนน
ความคิดเห็นด้านการทำงานได้ตามฟังก์ชันของระบบ (Function Test)		
1. เพิ่มความรวดเร็วในการใช้งาน	4	4.45
2. สามารถเรียนรู้วิธีการใช้งานได้ด้วยตนเอง	4	3.35
3. มีความเข้าใจในเนื้อหารายวิชาเพิ่มขึ้นจากการใช้งาน โปรแกรม	4	4.15
4. ช่วยลดขั้นตอนและปัญหาต่างๆในการปฏิบัติงานได้	4	3.85
5. ผลลัพธ์ที่ได้จากการใช้งานมีความสอดคล้องกับระบบงานจริง	4	4.65
รวม	20	4.09

โดยผลของการประเมินแสดงดังรูปที่ 4.12

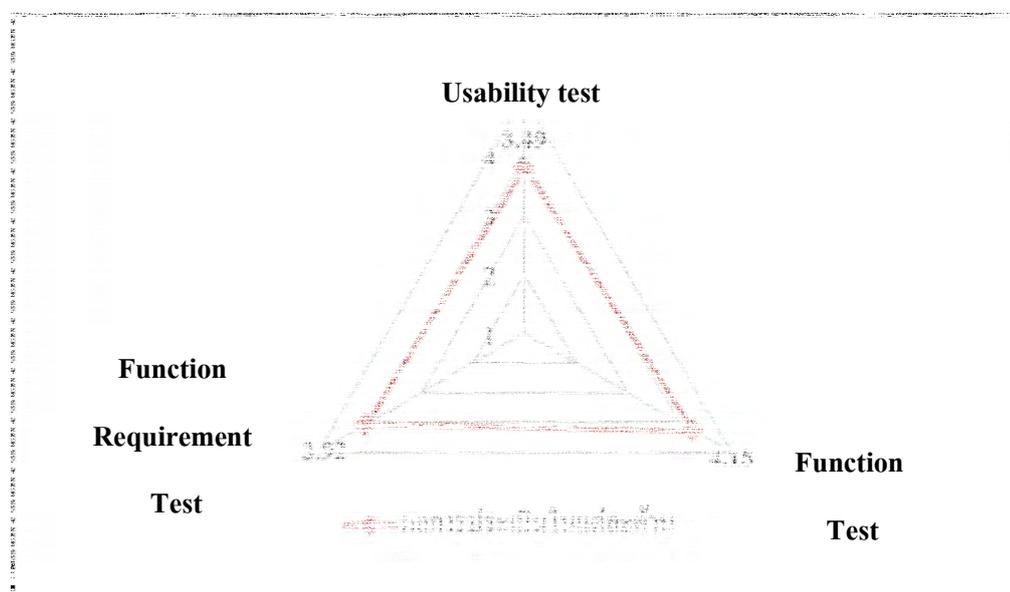


รูปที่ 4.12 แสดงผลการประเมินด้านการตรงความต้องการของผู้ใช้ระบบ
(Functional Requirement Test)ของการใช้งานโปรแกรม RulaSU

4.2.2.1 ผลการประเมินคุณภาพของโปรแกรมErgoIntelligence

ตารางที่ 4.8 แสดงผลการประเมินคุณภาพในแต่ละด้านของโปรแกรมErgoIntelligence

ปัจจัย	คะแนน
1) ด้านความง่ายต่อการใช้งานระบบ(Usability Test)	3.49
2) ด้านการทำงานได้ตามฟังก์ชันงานของระบบ(Function Test)	4.15
3) ด้านการตรงตามความต้องการของผู้ใช้ระบบ(Functional Requirement Test)	3.92
รวม	3.85



รูปที่ 4.13 แสดงผลการประเมินคุณภาพของโปรแกรม ErgoIntelligence

4. การประเมินด้านความง่ายต่อการใช้งานระบบ(Usability Test)

ด้านความง่ายต่อการใช้งานระบบ จะแบ่งออกเป็น 2 ส่วน ดังนี้

- 1.การติดตั้งโปรแกรมสามารถทำการติดตั้งได้ง่าย ไม่ซับซ้อน
- 2.การใช้งาน โปรแกรม

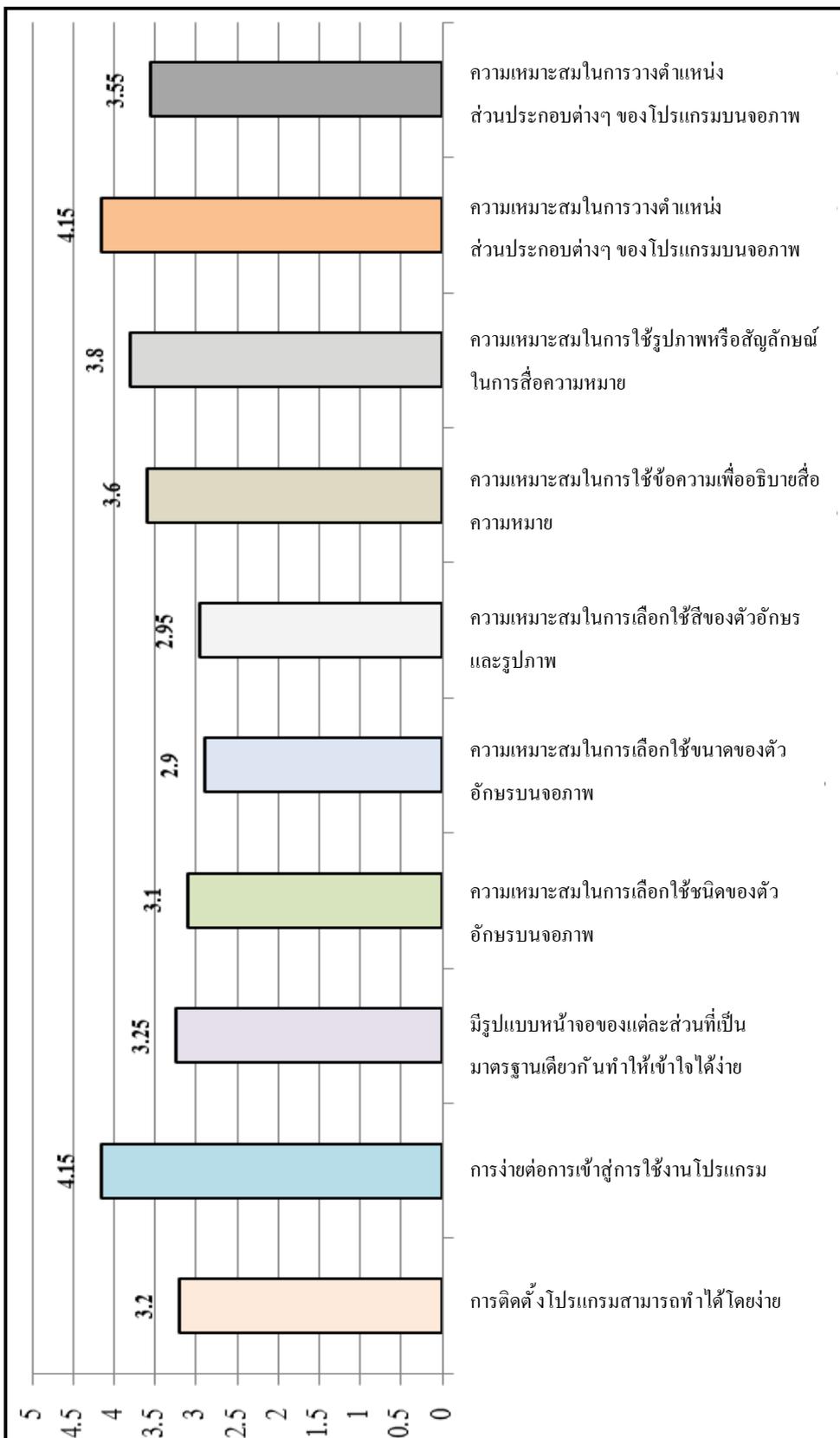
ในการประเมินด้านความง่ายต่อการใช้งานระบบในส่วนของการใช้งานโปรแกรมนั้น จะแบ่งออกเป็นปัจจัยต่างๆ คือ การง่ายต่อการเข้าสู่การใช้งานโปรแกรมมีรูปแบบหน้าจอของแต่ละส่วนที่เป็นมาตรฐานเดียวกันทำให้เข้าใจได้ง่ายเหมาะสมในการเลือกใช้นิพจน์ของตัวอักษรบนจอภาพความเหมาะสมในการเลือกใช้นิพจน์ของตัวอักษรบนจอภาพความเหมาะสมในการเลือกใช้นิพจน์

ของตัวอักษรและรูปภาพ ความเหมาะสมในการใช้ข้อความเพื่ออธิบายสื่อความหมาย ความเหมาะสมในการใช้รูปภาพหรือสัญลักษณ์ในการสื่อความหมาย ความเหมาะสมในการวางตำแหน่ง ส่วนประกอบต่างๆของโปรแกรมบนจอภาพและความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆของโปรแกรมบนจอภาพโดยได้กำหนดค่าน้ำหนักต่างๆ ดังแสดงในตารางที่.14

ตารางที่ 4.9 แสดงผลการประเมินด้านความง่ายต่อการใช้งานระบบ(Usability Test)

เรื่อง	น้ำหนัก	คะแนน
ความคิดเห็นด้านความพึงพอใจต่อการใช้งานระบบ (Usability Test)		
1.การติดตั้งโปรแกรม	5	
1.1) การติดตั้งโปรแกรมสามารถทำได้โดยง่าย	5	3.45
2.การใช้งานโปรแกรม	35	
2.1)การง่ายต่อการเข้าสู่การใช้งานโปรแกรม	3	4.15
2.2) มีรูปแบบหน้าจอของแต่ละส่วนที่เป็นมาตรฐานเดียวกันทำให้เข้าใจได้ง่าย	4	3.25
2.3) ความเหมาะสมในการเลือกใช้ชนิดของตัวอักษรบนจอภาพ	4	3.10
2.4) ความเหมาะสมในการเลือกใช้ขนาดของตัวอักษรบนจอภาพ	4	2.90
2.5) ความเหมาะสมในการเลือกใช้สีของตัวอักษรและรูปภาพ	4	2.95
2.6) ความเหมาะสมในการใช้ข้อความเพื่ออธิบายสื่อความหมาย	4	3.60
2.7) ความเหมาะสมในการใช้รูปภาพหรือสัญลักษณ์ในการสื่อความหมาย	4	3.80
2.8) ความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆของโปรแกรมบนจอภาพ	4	4.15
2.9) ความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆของโปรแกรมบนจอภาพ	4	3.55
รวม	40	3.49

โดยผลของการประเมินแสดงดังรูปที่ 4.14



รูปที่ 4.14 แสดงผลการประเมินด้านความง่ายต่อการใช้งานระบบ (Usability Test) ของการใช้งานโปรแกรม ErgoIntelligence

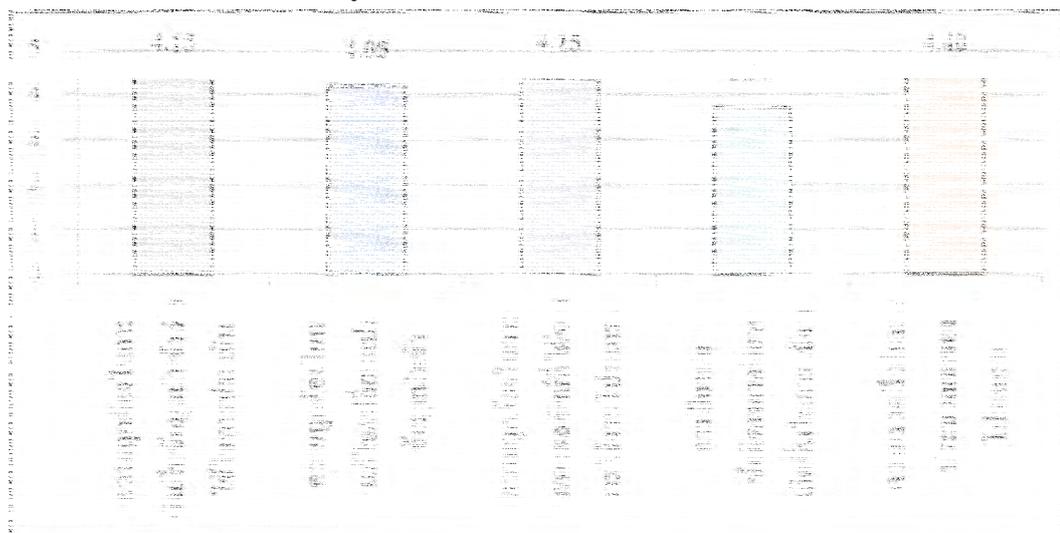
5. การประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test)

ด้านการทำงานได้ตามฟังก์ชันของระบบ ปัจจัยของการประเมิน คือความถูกต้องในการจัดเก็บข้อมูลนำเข้าเพื่อใช้ในการประเมินความถูกต้องในการแก้ไขหรือเปลี่ยนแปลงข้อมูลนำเข้า ความถูกต้องในการแสดงผลลัพธ์ที่ได้จากการใช้งาน โปรแกรมการป้องกันข้อผิดพลาดที่อาจเกิดขึ้นในการใช้งานและความรวดเร็วในการประมวลผลของโปรแกรมโดยได้กำหนดค่าน้ำหนักต่างๆดังแสดงในตารางที่ 4.10

ตารางที่ 4.10 แสดงผลการประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test)

เรื่อง	น้ำหนัก	คะแนน
ความคิดเห็นด้านการทำงานได้ตามฟังก์ชันของระบบ (Function Test)		
1. ความถูกต้องในการจัดเก็บข้อมูลนำเข้าเพื่อใช้ในการประเมิน	4	4.15
2. ความถูกต้องในการแก้ไขหรือเปลี่ยนแปลงข้อมูลนำเข้า	4	4.05
3. ความถูกต้องในการแสดงผลลัพธ์ที่ได้จากการใช้งานโปรแกรม	4	4.25
4. การป้องกันข้อผิดพลาดที่อาจเกิดขึ้นในการใช้งาน	4	3.90
5. ความรวดเร็วในการประมวลผลของโปรแกรม	4	4.40
รวม	20	4.15

โดยผลของการประเมินแสดงดังรูปที่ 4.15



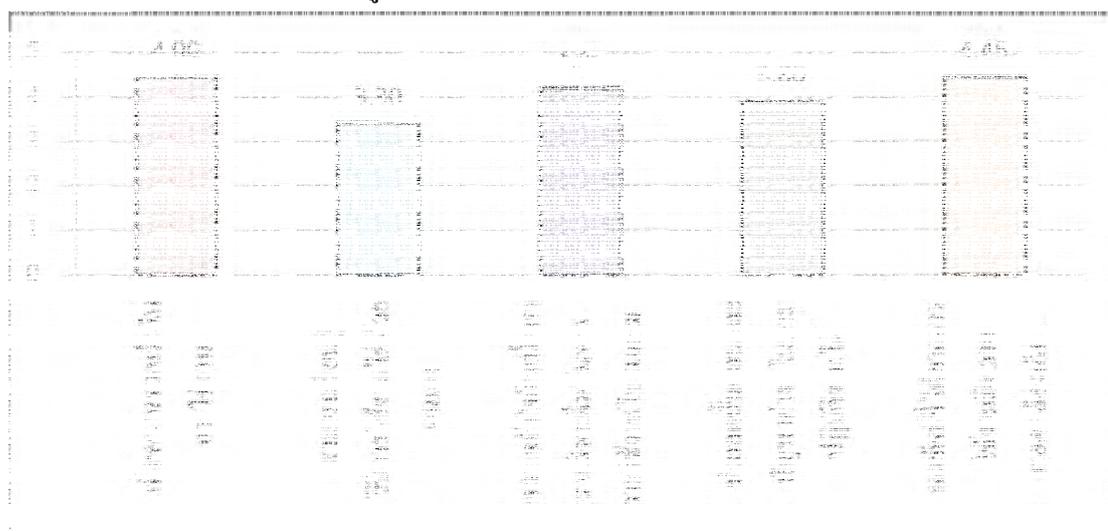
รูปที่ 4.15 แสดงผลการประเมินด้านการทำงานได้ตามฟังก์ชันของระบบ(Function Test) ของการใช้งานโปรแกรม ErgoIntelligence

6. การประเมินด้านการตรงความต้องการของผู้ใช้ระบบ(Functional Requirement Test) ด้านการทำงานได้ตามฟังก์ชันของระบบ ปัจจัยของการประเมิน คือเพิ่มความรวดเร็วในการใช้งานสามารถเรียนรู้วิธีการใช้งานได้ด้วยตนเอง มีความเข้าใจในเนื้อหารายวิชาเพิ่มขึ้นจากการใช้งานโปรแกรมช่วยลัดขั้นตอนและปัญหาต่างๆในการปฏิบัติงานได้และผลลัพธ์ที่ได้จากการใช้งานมีความสอดคล้องกับระบบงานจริงโดยได้กำหนดค่าน้ำหนักต่างๆดังแสดงในตารางที่.11

ตารางที่ 4.11 แสดงผลการประเมินด้านการตรงความต้องการของผู้ใช้ระบบ
(Functional Requirement Test)

เรื่อง	น้ำหนัก	คะแนน
ความคิดเห็นด้านการทำงานได้ตามฟังก์ชันของระบบ (Function Test)		
1. เพิ่มความรวดเร็วในการใช้งาน	4	4.00
2. สามารถเรียนรู้วิธีการใช้งานได้ด้วยตนเอง	4	3.30
3. มีความเข้าใจในเนื้อหารายวิชาเพิ่มขึ้นจากการใช้งานโปรแกรม	4	4.05
4. ช่วยลัดขั้นตอนและปัญหาต่างๆในการปฏิบัติงานได้	4	3.80
5. ผลลัพธ์ที่ได้จากการใช้งานมีความสอดคล้องกับระบบงานจริง	4	4.45
รวม	20	3.92

โดยผลของการประเมินแสดงดังรูปที่ 4.16



รูปที่ 4.16 แสดงผลการประเมินด้านการตรงความต้องการของผู้ใช้ระบบ
(Functional Requirement Test)ของการใช้งานโปรแกรม ErgoIntelligence

4.3 การประเมินข้อมูลด้านเวลาเชิงสถิติ

4.3.1 การประเมินข้อมูลด้านเวลาเชิงสถิติของโปรแกรมRulaSU

ตารางที่ 4.12 ตารางการเก็บข้อมูลเวลารวมการทำงานรวมทุกชั้นตอน ของการประเมินท่าทางการทำงานโดยใช้วิธีการRULA

ครั้งที่	RulaSU	ErgoIntelligence	การคำนวณมือ	หน่วย
1	47.5	40.6	90.7	วินาที
2	45.3	39.7	90.9	วินาที
3	45.8	40.8	90.8	วินาที
4	45.6	40.3	89.8	วินาที
5	45.0	40.2	90.6	วินาที
6	45.0	40.4	90.7	วินาที
7	45.6	40.3	90.3	วินาที
8	45.3	40.2	89.6	วินาที
9	45.6	40.5	89.8	วินาที
10	45.6	40.0	90.5	วินาที

ขั้นตอนการทดสอบ

1. ทำการทดสอบข้อมูลทั้งสองกลุ่มเสียก่อนว่ามีการกระจายตัวเป็นแบบ Normal Distribution หรือไม่ โดยใช้วิธี Probability Plot ซึ่งมีขั้นตอนการทำงาน ดังนี้

- 1.1 ทำการเรียงลำดับค่าของข้อมูลจากน้อยไปหามาก
 - 1.2 ใ้หมายเลขกำกับ(Rank: i) ตั้งแต่ $i = 1, 2, 3, \dots, 10$
 - 1.3 คำนวณหาค่าProbability ตามสูตร $p = (i - 0.5) / n$
- โดยมีผลการคำนวณดังตาราง

ตารางที่ 4.13 ตารางการจัดเรียงข้อมูลสำหรับหาค่าProbability ของการใช้โปรแกรม RulaSU

Data	Rank	Data(Sorted)	(i – 0.5) / n
47.5	1	45.0	0.05
45.3	2	45.0	0.15
45.8	3	45.3	0.25
45.6	4	45.3	0.35
45.0	5	45.6	0.45
45.0	6	45.6	0.55
45.6	7	45.6	0.65
45.3	8	45.6	0.75
45.6	9	45.8	0.85
45.6	10	47.5	0.95

ตารางที่ 4.14 ตารางการจัดเรียงข้อมูลสำหรับหาค่าProbability ของการใช้โปรแกรม

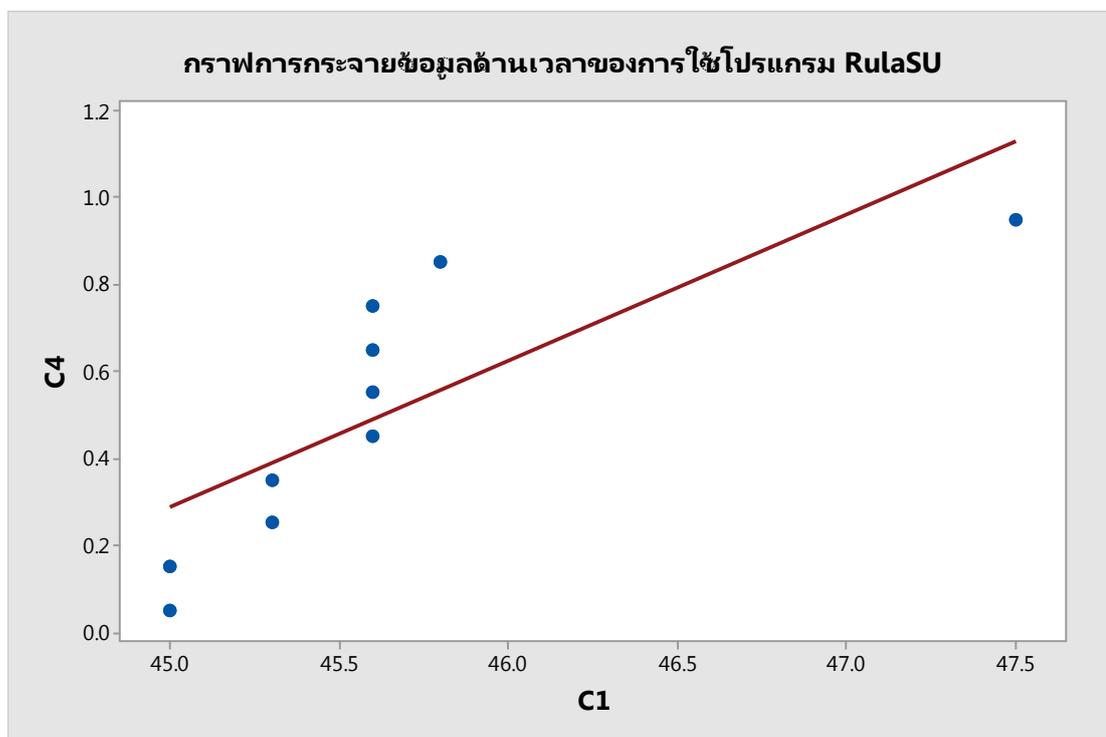
ErgoIntelligence

Data	Rank	Data(Sorted)	(i – 0.5) / n
40.6	1	39.7	0.05
39.7	2	40.0	0.15
40.8	3	40.2	0.25
40.3	4	40.2	0.35
40.2	5	40.2	0.45
40.4	6	40.3	0.55
40.3	7	40.4	0.65
40.2	8	40.5	0.75
40.5	9	40.6	0.85
40.0	10	40.8	0.95

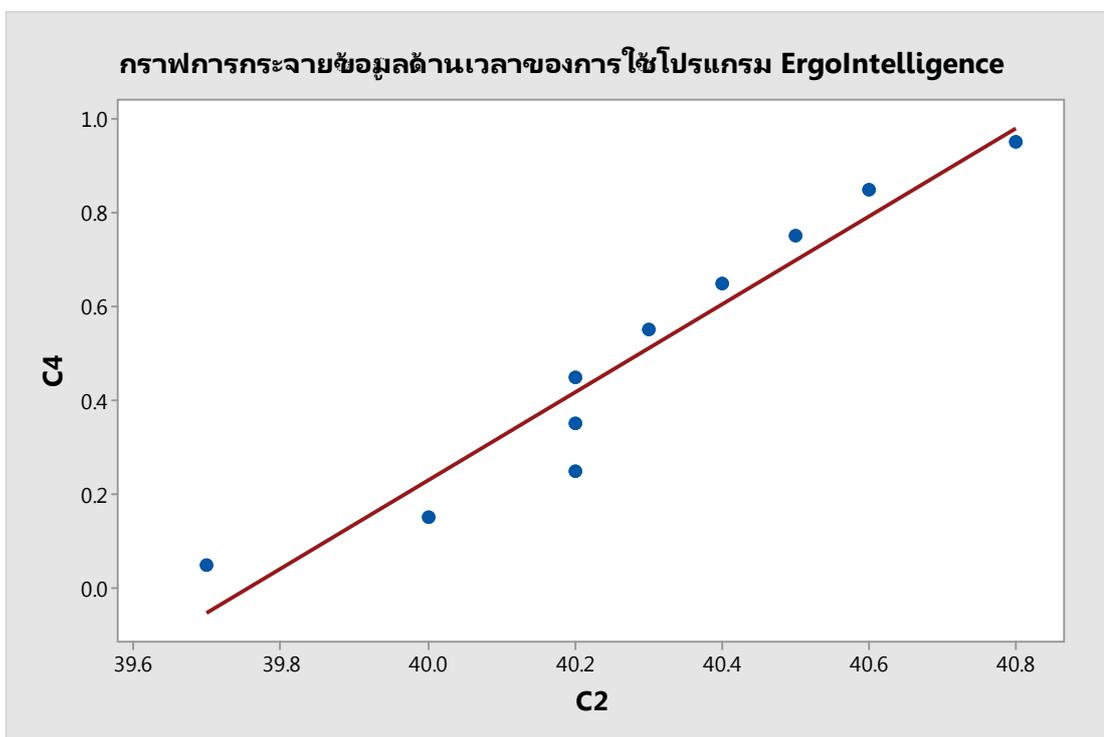
ตารางที่ 4.15 ตารางการจัดเรียงข้อมูลสำหรับหาค่าProbability ของวิธีการคำนวณมือเพื่อประเมิน
 ทำทางในการทำงานโดยใช้วิธีการRULA

Data	Rank	Data(Sorted)	$(i - 0.5) / n$
90.7	1	89.6	0.05
90.9	2	89.8	0.15
90.8	3	89.8	0.25
89.8	4	90.3	0.35
90.6	5	90.5	0.45
90.7	6	90.6	0.55
90.3	7	90.7	0.65
89.6	8	90.7	0.75
89.8	9	90.8	0.85
90.5	10	90.9	0.95

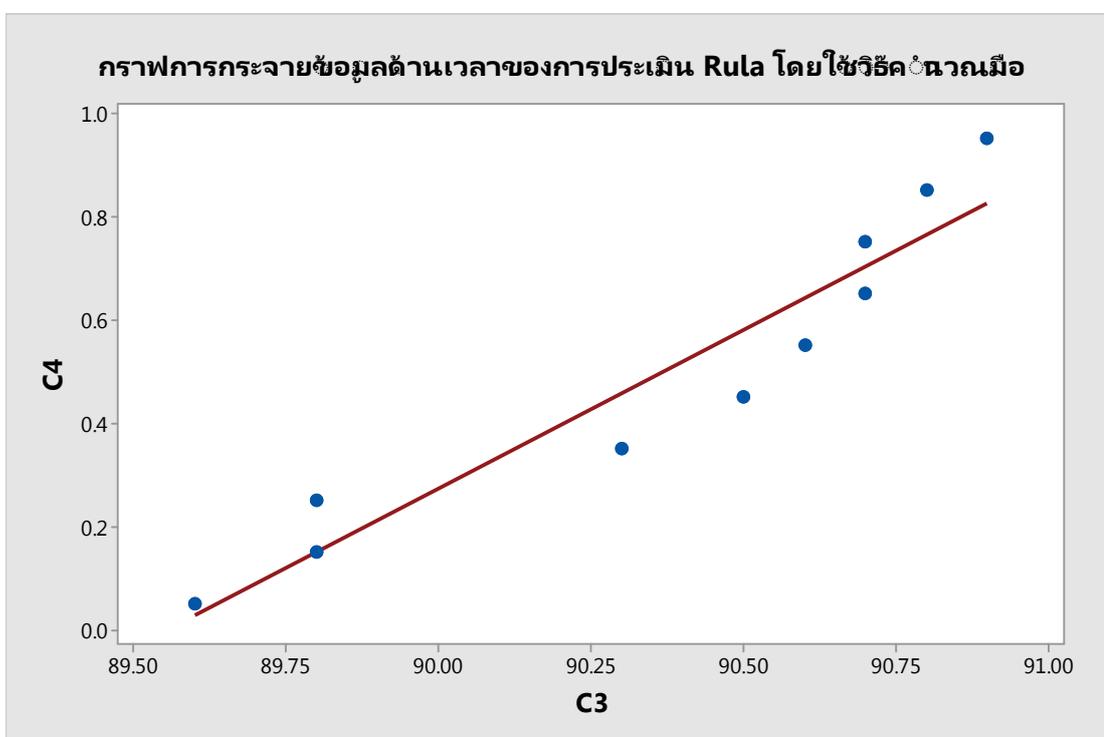
- ทำการ Scatter plot เพื่อทำการ Plot จุดตัดระหว่างProbability และ Data ที่ได้เรียงลำดับ
 เรียบร้อยแล้ว โดยให้แสดงเส้นตรงด้วย



รูปที่ 4.17 แสดงกราฟการกระจายของข้อมูลด้านเวลาของการใช้โปรแกรมRulaSU



รูปที่ 4.18 แสดงกราฟการกระจายของข้อมูลด้านเวลาของการใช้โปรแกรม ErgoIntelligence



รูปที่ 4.19 แสดงกราฟการกระจายของข้อมูลด้านเวลาของวิธีการคํานวณมือ
เพื่อประเมินท่าทางในการทำงานโดยใช้วิธีการ RULA

จากกราฟที่ 3 จะเห็นได้ว่า จุดตัดเรียงกันเป็นแนวเส้นตรง ลักษณะการเกิดจุดไม่กระจุกเป็นกลุ่มๆ และความห่างระหว่างแต่ละจุดใกล้เคียงกันเป็นส่วนใหญ่ ซึ่งกราฟที่ 3 พอจะประมาณว่าเป็นการกระจายแบบ Normal Distribution

2. ทำการทดสอบสมมติฐานโดยการใช้หลักสถิติ-T-Test ซึ่งได้ทำการตั้งค่าสมมติฐานหลักและสมมติฐานรอง ดังนี้

2.1 ค่าความสัมพันธ์ของข้อมูลด้านเวลาระหว่างการใช้งาน โปรแกรม RulaSU และการใช้งานโปรแกรม ErgoIntelligence เพื่อประเมินท่าทางการทำงานโดยใช้วิธีการ RULA

$$\text{สมมติฐานหลัก } H_0: \mu_1 = \mu_2$$

$$\text{สมมติฐานรอง } H_1: \mu_1 > \mu_2$$

โดยกำหนดให้ μ_1 คือ ค่าเฉลี่ยของกลุ่มข้อมูลด้านเวลาของการใช้งานโปรแกรม RulaSU

μ_2 คือค่าเฉลี่ยของกลุ่มข้อมูลด้านเวลาของการใช้งานโปรแกรม ErgoIntelligence

การทดสอบค่าเฉลี่ย t-Test โดยใช้โปรแกรม MINITAB (Trial Version) โดยเลือก t-Test: Two – Sample จะให้ผลดังตารางที่ 4.16

ตารางที่ 4.16 แสดงผลการใช้ MINITAB โดยเลือก t-Test: Two – Sample

t-Test: Two-Sample		
	RulaSU	ErgoIntelligence
Mean	45.70	40.30
Variance	0.505	0.0966
Observations	10	10
Hypothesized Mean Difference	5.34	
DF	12	
T-test difference	0	
T-value	21.78	
P-value	0.00...	

จะสังเกตได้ว่าค่าความแปรปรวนของข้อมูล มีความแตกต่างกัน(σ_1^2 ไม่เท่ากับ σ_2^2)

- ทำการเปรียบเทียบ P-Value กับ 0.00 ถ้า P-Value มากกว่า 0 เราก็ยอมรับสมมติฐานหลัก $H_0: \sigma_1 = \sigma_2$

ในข้อนี้ ค่า P-Value ~ 0 ซึ่งน้อยกว่า 0.05 เราจึงปฏิเสธสมมติฐานหลักตั้งนั้นจึงสรุปได้ว่าค่าความสัมพันธ์ของข้อมูลด้านเวลาระหว่างการใช้งานโปรแกรม RulaSU และการใช้งานโปรแกรม ErgoIntelligence เพื่อประเมินท่าทางการทำงานโดยใช้วิธีการ RULA มีความแตกต่างกันอย่างมีนัยสำคัญที่ระดับความเชื่อมั่น 95%

2.2 ค่าความสัมพันธ์ของข้อมูลด้านเวลาระหว่างการใช้งาน โปรแกรม RulaSU กับวิธีการคำนวณมือเพื่อประเมินท่าทางการทำงานโดยใช้วิธีการ RULA

สมมติฐานหลัก $H_0: \sigma_1 = \sigma_2$

สมมติฐานรอง $H_1: \sigma_1 > \sigma_2$

โดยกำหนดให้ σ_1 คือ ค่าเฉลี่ยของกลุ่มข้อมูลด้านเวลาของวิธีการคำนวณมือ

σ_2 คือ ค่าเฉลี่ยของกลุ่มข้อมูลด้านเวลาของการใช้งาน โปรแกรม RulaSU

การทดสอบค่าเลือก t-Test โดยใช้โปรแกรม MINITAB (Trial Version) โดยเลือก t-Test: Two – Sample จะให้ผลดังตารางที่ 4.17

ตารางที่ 4.17 แสดงผลการใช้ MINITAB โดยเลือก t-Test: Two – Sample

t-Test: Two-Sample		
	วิธีการคำนวณมือ	RulaSU
Mean	90.30	45.70
Variance	0.222	0.505
Observations	10	10
Hypothesized Mean Difference	44.74	
DF	15	
T-test difference	0	
T-value	165.94	
P-value	0.00...	

จะสังเกตได้ว่าค่าความแปรปรวนของข้อมูล มีความแตกต่างกัน(σ_1^2 ไม่เท่ากับ σ_2^2)

- ทำการเปรียบเทียบ P-Value กับ 0.000 ถ้า P-Value มากกว่า 0.000 เราก็ยอมรับสมมติฐานหลัก $H_0: \mu_1 = \mu_2$

ในข้อนี้ ค่า P-Value ~ 0 ซึ่งน้อยกว่า 0.05 เราจึงปฏิเสธสมมติฐานหลักตั้งนั้น จึงสรุปได้ว่าค่าความสัมพันธ์ของข้อมูลด้านเวลาระหว่างการใช้งานโปรแกรม RulaSU กับวิธีการคำนวณมือเพื่อประเมินท่าทางการทำงาน โดยใช้วิธีการ RULA มีความแตกต่างกันอย่างมีนัยสำคัญที่ระดับความเชื่อมั่น 95%

4.4 ผลการนำโปรแกรมไปทดลองใช้งานในโรงงานอุตสาหกรรมกรณีศึกษา

นำโปรแกรมที่ได้ทำการออกแบบและพัฒนาไปทดลองใช้กับโรงงานอุตสาหกรรมกรณีศึกษา ซึ่งเป็นโรงงานแบ่งบรรจุเคมีเกษตร โดยมีผลข้อมูลเวลาเฉลี่ยที่ได้จากการทดลองใช้งานโปรแกรม ดังนี้



รูปที่ 4.20 แสดงภาพขณะทำการการอบรมให้ความรู้เกี่ยวกับการประเมินภาระงาน

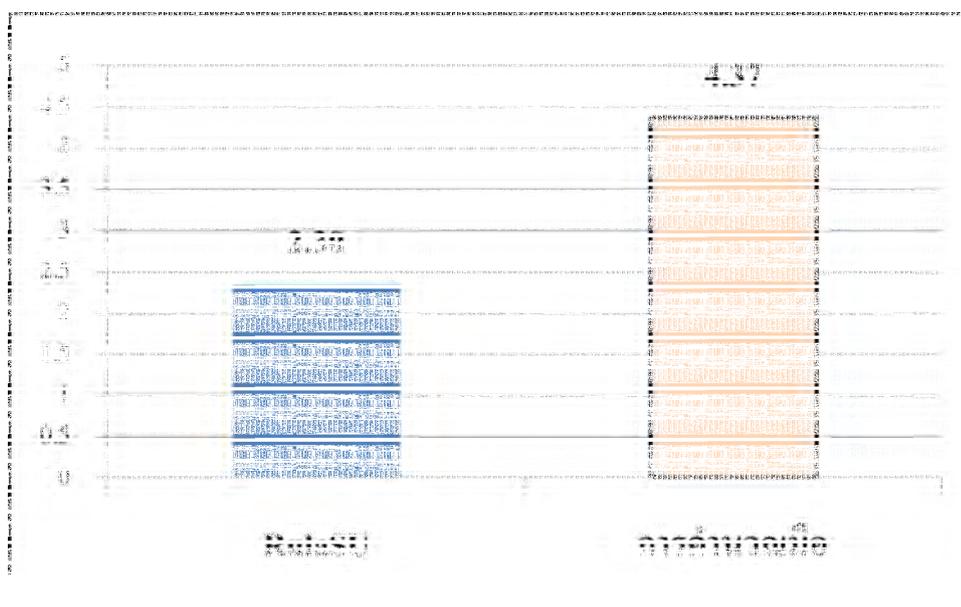
โดยใช้วิธีการทางกายศาสตร์แก่ผู้ทดลองใช้งานโปรแกรม

4.4.1 โปรแกรม RulaSU

ตารางที่ 4.18 แสดงผลข้อมูลเวลาเฉลี่ยที่ได้จากการทำการประเมินท่าทางในการทำงานโดยใช้วิธีการ RULA

ข้อมูลเวลาเฉลี่ยที่ได้จากการทำการประเมินท่าทางในการทำงานโดยใช้วิธีการRULA (นาที)	
RulaSU	2.30
การคำนวณมือ	4.37
ผลต่าง	2.07

จากการนำโปรแกรม RulaSU มาใช้ในการประเมินท่าทางการทำงานโดยใช้วิธีการ RULA จะใช้เวลาน้อยกว่าวิธีการคำนวณมือ 2.07 นาที หรือใช้เวลาในการทำงานลดลง 47.36%



รูปที่ 4.21 แสดงผลข้อมูลเวลาเฉลี่ยที่ใช้ในการประเมินท่าทางในการทำงาน โดยใช้วิธีการ RULA

บทที่ 5

สรุปผลการศึกษาและข้อเสนอแนะ

งานวิจัยเรื่องการพัฒนาโปรแกรมประเมินภาระงานทางการยศาสตร์บนระบบปฏิบัติการแอนดรอยด์มีวัตถุประสงค์ที่สำคัญ คือ เพื่อศึกษาแนวทางการประเมินภาระงานทางการยศาสตร์และการเขียนโปรแกรมโดยใช้ภาษา Java JDK , Eclipse , Android SDK และ Android Developer Tools สำหรับใช้ในการประเมินภาระงานทางการยศาสตร์และเพื่อประเมินผลลัพธ์ในการใช้งานโปรแกรมประเมินภาระงานทางการยศาสตร์ที่ได้ทำการออกแบบและพัฒนา จากการศึกษาสามารถสรุปผลโดยแยกออกเป็นหัวข้อดังนี้

1. สรุปผลการศึกษา
2. ปัญหาและอุปสรรค
3. ข้อเสนอแนะ

5.1 สรุปผลการศึกษา

5.1.1 การออกแบบและการสร้างโปรแกรม

จากการออกแบบและพัฒนาโปรแกรม สามารถพัฒนาโปรแกรมที่สร้างขึ้นด้วยภาษา Java ซึ่งเขียนโดยใช้โปรแกรม Eclipse , Android SDK และ Android Developer แต่ละโปรแกรมนั้นจะแบ่งแยกไปตามวิธีการที่ใช้ในการประเมินภาระงาน โปรแกรมที่พัฒนาสามารถนำไปใช้งานได้จริง ค่าของข้อมูลนำเข้าและข้อมูลสรุปผลการประเมินที่ได้มีความถูกต้องสามารถนำผลที่ได้จากการโปรแกรมไปใช้งานต่อได้

5.1.2 การประเมินประสิทธิภาพและคุณภาพของโปรแกรม

5.1.2.1 การประเมินประสิทธิภาพของโปรแกรม

จากการศึกษาเปรียบเทียบการใช้งานโปรแกรม RulaSU กับวิธีการคำนวณมือในการประเมินท่าทางในการทำงาน โดยใช้วิธีการ RULA พบว่าการใช้งานโปรแกรม RulaSU มีขั้นตอนการทำงานน้อยกว่าวิธีการคำนวณมือ ขั้นตอน เวลาที่ใช้ในการประเมินภาระงานน้อยกว่าวิธีการคำนวณมือ 44.6 วินาที (49.39%) ส่วนในการเปรียบเทียบประสิทธิภาพการใช้งานระหว่างโปรแกรม RulaSU กับโปรแกรม ErgoIntelligence โปรแกรม RulaSU มีขั้นตอนการทำงานมากกว่าขั้นตอนและใช้เวลาในการประเมินภาระงานมากกว่า 5.5 วินาที (12.03%) ซึ่งเป็นผลมาจาก

โปรแกรม RulaSU มีหน้าตาต่างการสรุปผลที่ต้องการข้อมูลมากกว่าโปรแกรม ErgoIntelligence เพื่อให้ผู้ใช้สามารถเข้าใจผลการประเมินได้มากยิ่งขึ้น จึงมีขั้นตอนในการกรอกข้อมูลที่มากกว่าและใช้เวลาเพิ่มขึ้นจากโปรแกรม ErgoIntelligence

5.1.2.2 การประเมินคุณภาพของโปรแกรม

ผลจากการทำแบบประเมินการใช้งานโปรแกรมที่ได้ทำการออกแบบและพัฒนา ด้านการด้านความง่ายต่อการใช้งานระบบ (Usability Test) ด้านการทำงานได้ตามฟังก์ชันงานของระบบ (Function Test) และด้านการตรงตามความต้องการของผู้ใช้ระบบ (Functional Requirement Test) สามารถสรุปผลการประเมินได้ดังนี้

- โปรแกรม RulaSU ได้คะแนน 4.04 คะแนน จากคะแนนเต็ม 5 คะแนน ซึ่งอยู่ในระดับที่ดี และได้คะแนนมากกว่าโปรแกรม ErgoIntelligence ที่ได้คะแนน 3.85 คะแนน จากคะแนนเต็ม 5 คะแนน

5.1.3 การประเมินข้อมูลด้านเวลาเชิงสถิติ

จากการทำการทดสอบสมมติฐาน โดยใช้หลักสถิติ-Test เพื่อเปรียบเทียบค่าความสัมพันธ์ของข้อมูลด้านเวลาของกขใช้งานโปรแกรมสำหรับการประเมินภาระงานวิธีการต่างๆ ในวิธีการทำงานที่แตกต่างกัน พบว่า ค่าความสัมพันธ์ของข้อมูลด้านเวลาที่ได้ทำการเปรียบเทียบของทุกโปรแกรมนั้นมีความแตกต่างกันอย่างมีนัยสำคัญ ที่ระดับความเชื่อมั่น 5%

5.1.4 การนำไปทดลองใช้งานในโรงงานอุตสาหกรรมกรณีศึกษา

จากการนำโปรแกรมไปทดลองใช้ในโรงงานอุตสาหกรรมกรณีศึกษาเพื่อศึกษาการใช้งานโปรแกรมที่ได้ทำการออกแบบและพัฒนาต่อผู้ที่สนใจโดยมีผลของการศึกษาการใช้ของโปรแกรมของผู้ที่สนใจ ดังนี้

- โปรแกรม RulaSU ช่วยลดระยะเวลาในการทำการประเมินภาระงานลง 2.07 นาที หรือคิดเป็น 47.36% จากการใช้วิธีการคำนวณมือ

5.2 ปัญหาและอุปสรรค

5.2.1 เนื่องจากโปรแกรมที่ได้ออกแบบและพัฒนาในช่วงแรกนั้น แบ่งแยกออกเป็นหลายโปรแกรมย่อยแตกต่างกันไปตามแต่ละวิธีการประเมินภาระงาน จึงทำให้ขั้นตอนของการพัฒนาโปรแกรมและการประเมินผลลัพธ์การใช้งานมีความยุ่งยาก ดังนั้นเพื่อให้ง่ายต่อการดำเนินงาน จึงจำเป็นที่จะต้องแยกออกเป็นโปรแกรมที่แตกต่างกันไปแต่ละวิธีการประเมินภาระงาน

5.2.3 การที่จะให้โปรแกรมดังกล่าวออกมาสมบูรณ์และมีประสิทธิภาพต่อการใช้งานจริงมากที่สุด จำเป็นต้องมีการปรับเปลี่ยนรูปแบบหน้าตาและวิธีการทำงานของโปรแกรมอยู่บ่อยครั้ง ทำให้ต้องเสียเวลาแก้ไขโปรแกรมอยู่นานพอสมควร เพื่อให้โปรแกรมสอดคล้องต่อความต้องการมากที่สุด

5.3 ข้อเสนอแนะ

5.3.1 ก่อนการใช้งานโปรแกรม ควรศึกษาการประเมินภาระงานทางการยศาสตร์ในวิธีการต่างๆ พร้อมทั้งวิธีการติดตั้งและการใช้งาน โปรแกรมให้เรียบร้อย เพื่อให้สามารถทำงานได้ง่ายค่าของข้อมูลที่ได้มีความถูกต้องตรงตามงานที่ประเมินและสะดวกต่อการใช้งานโปรแกรมมากยิ่งขึ้น

5.3.2 ในอนาคตหากมีการแก้ไขข้อมูลการทำงานของโปรแกรมหรือเพิ่มฟังก์ชันการทำงานต่างๆ ให้กับโปรแกรม ควรศึกษารูปแบบการเขียนโปรแกรมให้เข้าใจก่อน เพื่อให้สามารถพัฒนาโปรแกรมได้อย่างมีประสิทธิภาพและโปรแกรมสามารถทำงานได้ตรงตามฟังก์ชันที่ต้องการ

รายการอ้างอิง

- [1] ประจวบ ก่ออมจิตร (2554). การประเมินและปรับปรุงงานในสถานประกอบการโดยหลัก
การยศาสตร์, นครปฐม: โรงพิมพ์มหาวิทยาลัยศิลปากร
- [2] กิตติ อินทรานนท์ (2548). การยศาสตร์: **Ergonomic**, แอคทีฟพริ้นท์ กรุงเทพฯ.
- [3] จักรชัย โสอินทร์(2555). **Android App Development**, บริษัท ไอซีดี พรีเมียร์ จำกัด
กรุงเทพฯ.
- [4] นริศ เจริญพร (2542). “การยศาสตร์” เอกสารประกอบการสอนวิชาการยศาสตร์คณะ
วิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์ กรุงเทพฯ.
- [5] ประจวบ ก่ออมจิตร, (2548) การประเมินงานยกของ ด้วยมือในโรงงานอุตสาหกรรมโดย
ใช้สมการ NIOSH 1991, วารสารงานวิจัยและพัฒนา มจร, ปีที่ 28 ฉบับที่ 3 ก.ค.-ก.ย.
- [6] ชัยยุทธ วงศ์อัจฉริยา(2550). “การเปรียบเทียบ วิธีการประเมินภาระทางการยศาสตร์ ใน
สถานะ แวดล้อมการทำงานในวิสาหกิจขนาดกลางและขนาดย่อม. วิศวกรรม
ศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมอุตสาหกรรม บัณฑิตวิทยาลัย
มหาวิทยาลัยเชียงใหม่
- [7] Sanders, M.S. and McCormick, E.J. (1987) **Human Factors in Engineering and
Design**, Singapore, McGraw-Hill.
- [8] Grandjean, E. (1988) **Fitting the task to the man**. Taylor & Francis, UK.
- [9] Bhattacharya, A. and McGlothlin, D.J. (1996) **Occupational Ergonomics**. Marcel
Dekker, Inc. New York.

ภาคผนวก

ภาคผนวก ก
แผนภูมิการไหลของโปรแกรม

ตารางที่ ก.1 ตารางแสดงแผนภูมิการไหลของโปรแกรม RulaSU

Operation Flow Chart							
CHART NO. 1	Summary						
SHEET NO. 1 OF 3	Activity	RulaSU	การคำนวณมือ	Saving			
Activity : การประเมินท่าทางในการทำงาน โดยใช้วิธีการ RULA	Operation 	10	12	2			
	Transport 	0	0	0			
Method : การใช้งานโปรแกรม RulaSU	Delay 	0	0	0			
	Inspection 	1	4	3			
Location :	Storage 	0	0	0			
	Time(sec)	45.7	90.3	44.6			
DESCRIPTION	Time(sec)	Symbol					Remarks
							
1. กรอกข้อมูลของผู้ทำการประเมิน	20.3	●					
2. ทำการประเมินท่าทางการทำงานใน ส่วนของแขนส่วนบน	2.5	●					
3. ทำการประเมินท่าทางการทำงานใน ส่วนของแขนส่วนล่าง	2.5	●					
4. ทำการประเมินการงอของมือ	2.0	●					
5. ทำการประเมินการบิดของมือ	1.5	●					
6. ทำการประเมินท่าทางการทำงานใน ของคอ	2.6	●					
7. ทำการประเมินท่าทางการทำงานใน ของลำตัว	3.0	●					
8. ทำการประเมินตำแหน่งของขาและเท้า	1.4	●					
9. การประเมินการใช้แรงและการใช้ กล้ามเนื้อในส่วนของแขนและมือ	3.6	●					
10. การประเมินการใช้แรงและการใช้ กล้ามเนื้อในส่วนของคอ ลำตัว และ ขา	3.8	●					
11. ตรวจสอบผลการประเมิน	2.5				■		
รวม	45.7	10	0	0	1	0	

ตารางที่ ก.2 ตารางแสดงแผนภูมิการไหลของโปรแกรม RulaSU

Operation Flow Chart							
CHART NO. 2	Summary						
SHEET NO. 2 OF 3	Activity	RulaSU	Ergointelligence	Saving			
Activity : การประเมินท่าทางในการทำงาน โดยใช้วิธีการ RULA	Operation 	10	8	-2			
	Transport 	0	0	0			
Method : การใช้งานโปรแกรม ErgoIntelligence	Delay 	0	0	0			
	Inspection 	1	1	0			
Location :	Storage 	0	0	0			
	Time(sec)	45.7	40.2	-5.5			
DESCRIPTION	Time(sec)	Symbol					Remarks
							
1. กรอกข้อมูลของผู้ทำการประเมิน	20.1	●					
2. ทำการประเมินตำแหน่งท่าทางของมือ และ การบิดของมือ	3.0	●					
3. ทำการประเมินท่าทางการทำงานใน ส่วนของแขนส่วนล่าง	2.1	●					
4. ทำการประเมินท่าทางการทำงานใน ส่วนของแขนส่วนบน	2.4	●					
5. ทำการประเมินท่าทางการทำงานใน ส่วนของคอ	2.0	●					
6. ทำการประเมินท่าทางการทำงานใน ส่วนของลำตัว	2.1	●					
7. ทำการประเมินท่าทางการทำงานใน ส่วนของขาและเท้า	1.9	●					
8. ทำการประเมินการใช้แรงและ กล้ามเนื้อในการทำงาน	2.3	●					
9. ตรวจสอบผลการประเมิน	4.3				■		
รวม	40.2	8	0	0	1	0	

ตารางที่ ก.3 ตารางแสดงแผนภูมิการไหลของวิธีการคำนวณมือเพื่อประเมินท่าทางในการทำงาน โดยใช้วิธีการ RULA

Operation Flow Chart							
CHART NO. 3	Summary						
SHEET NO. 3 OF 3	Activity	RulaSU	การคำนวณมือ	Saving			
Activity : การประเมินท่าทางในการทำงาน โดยใช้วิธีการ RULA	Operation ○	10	12	2			
	Transport ⇨	0	0	0			
Method : การคำนวณมือ	Delay D	0	0	0			
	Inspection □	1	4	3			
Location :	Storage ▽	0	0	0			
	Time(sec)	45.7	90.3	44.6			
DESCRIPTION	Time(sec)	Symbol					Remarks
		○	⇨	D	□	▽	
1. กรอกละเนนการประเมินท่าทางของแขนส่วนบนลงบนแบบฟอร์ม	3.9	●					
2. กรอกละเนนการประเมินท่าทางการทำงานในส่วนของแขนส่วนล่างลงบนแบบฟอร์ม	4.3	●					
3. กรอกละเนนการประเมินการงอของมือลงบนแบบฟอร์ม	4.4	●					
4. กรอกละเนนการประเมินการบิดของมือลงบนแบบฟอร์ม	3.4	●					
5. ตรวจสอบค่าคะแนนที่ได้จากตาราง A เพื่อสรุปคะแนนการประเมินท่าทางส่วนแขนและมือ	26.9				■		
6. กรอกผลคะแนนที่ได้ลงบนแบบฟอร์ม	3.8	●					
7. กรอกละเนนการประเมินท่าทางการทำงานในส่วนของคอลงบนแบบฟอร์ม	4.0	●					
8. กรอกละเนนการประเมินท่าทางการทำงานในส่วนของลำตัวลงบนแบบฟอร์ม	3.7	●					
9. กรอกละเนนการประเมินตำแหน่งของขาและเท้าลงบนแบบฟอร์ม	3.8	●					
10. ตรวจสอบค่าคะแนนที่ได้จากตาราง B เพื่อสรุปคะแนนการประเมินท่าทางส่วนคอ ลำตัวและขา	10.7				■		
11. กรอกผลคะแนนที่ได้ลงบนแบบฟอร์ม	3.9	●					
12. กรอกละเนนการประเมินการใช้แรงและการใช้กล้ามเนื้อในส่วนของแขนและมือลงบนแบบฟอร์ม	3.7	●					

ตารางที่ ก.3 ตารางแสดงแผนภูมิการไหลของวิธีการคำนวณมือเพื่อประเมินท่าทางในการทำงาน โดยใช้วิธีการ RULA (ต่อ)

DESCRIPTION	Time(sec)	Symbol					Remarks
		○	⇒	∩	□	▽	
13. กรอกระยะการประเมินการใช้แรงและการใช้กล้ามเนื้อในส่วนของคอ ลำตัว และขาลงบนแบบฟอร์ม	4.1	●					
14. ตรวจสอบค่าคะแนนที่ได้จากตาราง C เพื่อสรุปคะแนนการประเมินท่าทางการทำงานโดยใช้วิธีการRula	3.8				■		
15. กรอกระยะคะแนนที่ได้ลงบนแบบฟอร์ม	3.3	●					
16. ตรวจสอบการกำหนดระดับการปฏิบัติงานจากค่าคะแนนที่ได้	2.6				■		
รวม	90.3	12	0	0	4	0	

ภาคผนวก ข
วิธีการติดตั้งและใช้งานโปรแกรม

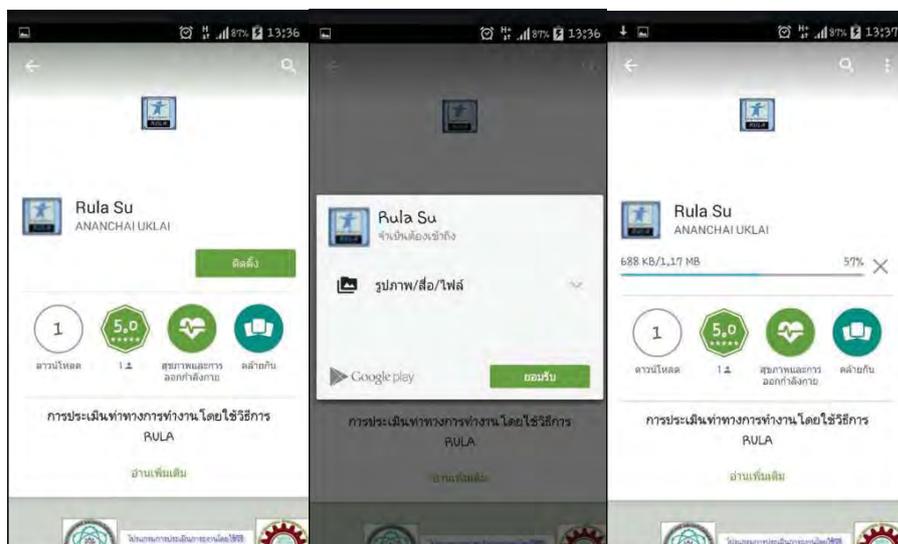
วิธีการติดตั้งโปรแกรมRulaSU

1. ทำการเข้าไปที่Play Store บน Smart Phone ที่มีระบบปฏิบัติการ Android จากนั้นพิมพ์คำว่าRula Su



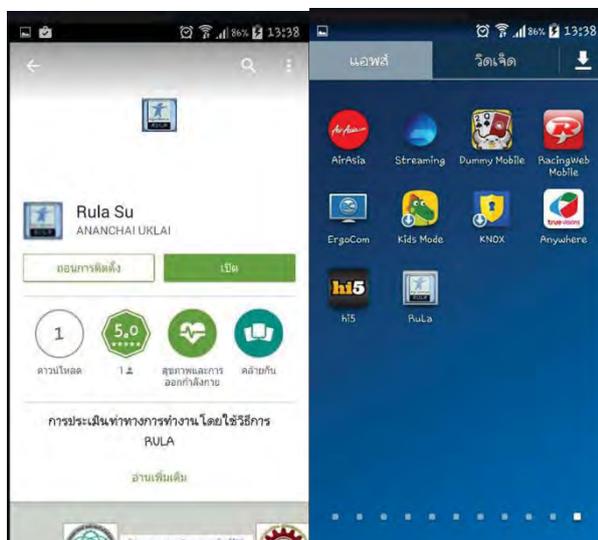
รูปที่ ข.1 แสดงตัวอย่างไฟล์สำหรับการติดตั้งโปรแกรม

2. ทำการเข้าไปที่Rula Su จากนั้นให้กดคำว่าติดตั้งจะมีค่าให้กดยอมรับ จากนั้นกดยอมรับแล้วเริ่มดาวน์โหลด



รูปที่ ข.2 แสดงตัวอย่างไฟล์สำหรับการติดตั้งโปรแกรม

3. เมื่อทำการดาวน์โหลดสำเร็จแล้ว การติดตั้งเสร็จจะมีแอปพลิเคชันอยู่บนSmart Phone



รูปที่ ข.3 แสดงตัวอย่างไฟล์สำหรับการติดตั้งโปรแกรม

คู่มือการใช้งานโปรแกรมRula SU ในการประเมินท่าทางการทำงานโดยวิธีการRULA (Rapid Upper Limb Assessment)

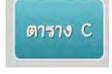
หลังจากที่ได้ทำการติดตั้งโปรแกรมเสร็จสิ้นแล้ว ทำการเข้าโปรแกรมโดยการเข้าไปที่
ICON ของโปรแกรม เมื่อเข้าสู่โปรแกรมจะพบหน้าจอเริ่มต้นของโปรแกรม ดังภาพ



รูปที่ ข.4 แสดงหน้าต่างเริ่มต้นของโปรแกรมRulaSU

ซึ่งในหน้าต่างเริ่มต้นของโปรแกรมนี้จะมีปุ่ม ล้างต่าง ๆ ดังนี้

ตารางที่ ข.1 แสดงการทำงานของปุ่ม ที่ปรากฏในหน้าต่างเริ่มต้นของโปรแกรมRulaSU

ปุ่ม	คำอธิบาย
	- เริ่มต้นทำการประเมินท่าทางในการทำงานโดยใช้โปรแกรมRulaSU
	- เข้าสู่โปรแกรมการเปิดค่าคะแนนจากตาราง A
	- เข้าสู่โปรแกรมการเปิดค่าคะแนนจากตาราง B
	- เข้าสู่โปรแกรมการเปิดค่าคะแนนจากตาราง C
	- เปิดคู่มือวิธีการใช้งานโปรแกรม
	- เปิดดูข้อมูลผู้พัฒนาโปรแกรม

ในการเริ่มต้นใช้งานโปรแกรม ให้คลิกที่ปุ่ม  จะปรากฏหน้าต่างสำหรับกรอกรายละเอียดข้อมูลการประเมินท่าทาง ดังภาพ



รูปที่ ข.5 แสดงหน้าต่างสำหรับการกรอกรายละเอียดสำหรับการประเมิน

ทำการกรอกข้อมูลรายละเอียดในแต่ละช่องให้ครบถ้วน หากว่าเรามีรูปทำทางการทำงานที่เราจะทำการประเมิน เราสามารถนำรูปดังกล่าวมาใส่ในโปรแกรมเพื่อแสดงรูปดังกล่าวในหน้าสรุปผลได้ โดยทำการกดปุ่ม  เพื่อเลือกรูปที่ต้องการ จากนั้นทำการกดปุ่ม  เพื่อเข้าสู่หน้าต่างสำหรับการทำการประเมินทำทางการทำงานโดยใช้วิธีการRULA

หากเราต้องการข้ามขั้นตอนนี้ สามารถกดปุ่ม  เพื่อผ่านขั้นตอนนี้และเข้าสู่หน้าต่างหับ การประเมินทำทางการทำงานโดยใช้วิธีการRULA ได้เลย

ซึ่งหน้าต่างสำหรับการประเมินทำทางการทำงานโดยใช้วิธีการRULA จะมีลักษณะดังภาพ



รูปที่ ข.6 แสดงตัวอย่างการกรอกรายละเอียดสำหรับการประเมิน

เราจะขออธิบายวิธีการทำการประเมินทำทางโดยใช้โปรแกรมRulaSU ตามวิธีการประเมินทำทางตามหลักการRULA ในแต่ละขั้นตอนนี้ ดังนี้

1. การประเมินตำแหน่งของแขนส่วนบน

ทำการคลิกเพื่อเลือกให้คะแนนทำทางตำแหน่งของแขนส่วนบน ซึ่งจะมีรายละเอียดการให้คะแนน ดังนี้



รูปที่ ข.7 แสดงหน้าต่างสำหรับการประเมินตำแหน่งของแขนส่วนบน

- 1.1 ระดับของแขนการยกที่สูงขึ้น ระดับคะแนนที่ให้มากขึ้นคะแนนอยู่ระหว่าง-4
- 1.2 ถ้ามีการยกของไหลให้บวกคะแนนเพิ่มอีก+1
- 1.3 ถ้ามีการกางแขนให้บวกคะแนนเพิ่มอีก+1
- 1.4 ถ้าแขนมีที่รองรับหรือวางพาดอยู่ที่ลดคะแนน-1
- 1.5 คะแนนสูงสุดของขั้นตอนนี้ จะมีค่าไม่เกิน6 คะแนน

2. การประเมินตำแหน่งของแขนส่วนล่าง

ทำการคลิกเพื่อเลือกให้คะแนนท่าทางตำแหน่งของแขนส่วนล่าง ซึ่งจะมีรายละเอียดการให้คะแนน ดังนี้



รูปที่ ข.8 แสดงหน้าต่างสำหรับการประเมินตำแหน่งของแขนส่วนล่าง

- 2.1 ระดับการยกของของแขนส่วนล่างที่แตกต่างกัน จะมีการให้คะแนนอยู่ระหว่าง-2 คะแนน
- 2.2 ถ้ามีการทำงานไขว้แขนเลยแกนกลางลำตัวให้บวกคะแนนเพิ่มอีก1
- 2.3 ถ้ามีการทำงานในลักษณะกางแขนออกไปด้านข้างลำตัวให้บวกคะแนนเพิ่มอีก1
- 2.4 คะแนนสูงสุดในขั้นตอนนี้ มีค่าไม่เกิน4 คะแนน

3. การประเมินตำแหน่งของมือ

ทำการคลิกเพื่อเลือกให้คะแนนท่าทางตำแหน่งของมือ ซึ่งจะมีรายละเอียดการให้คะแนน ดังนี้



รูปที่ ข.9 แสดงหน้าต่างสำหรับการประเมินตำแหน่งของมือ

- 3.1 ระดับการบิดงอของมือที่แตกต่างกันจะมีการให้คะแนนอยู่ระหว่าง 1-3 คะแนน
- 3.2 ถ้ามีการทำงานที่เกิดการเบี่ยงข้อมือออกไปจากแนวกลาง(deviation) คะแนนเพิ่มอีก 1
- 3.3 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน 4 คะแนน

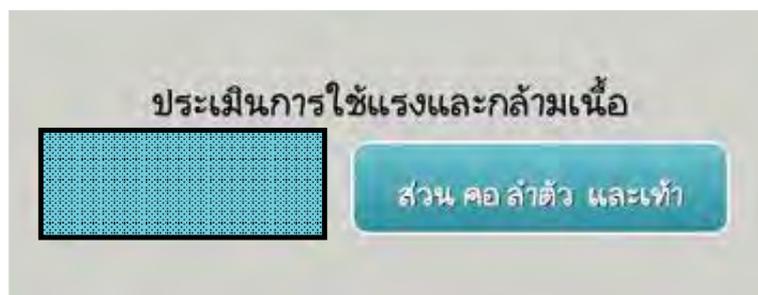
4. การประเมินการบิดของข้อมือ



รูปที่ ข.10 แสดงหน้าต่างสำหรับการประเมินการบิดของข้อมือ

- 4.1 ขณะทำงานข้อมือไม่ควรหมุนถ้ามีการหมุนข้อมือ(การบิดของข้อมือ) ให้คะแนนเป็น 1
- 4.2 ถ้ามีการทำงานที่หมุนข้อมือมากเกือบสุดให้คะแนนเป็น 2
- 4.3 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน 2 คะแนน

5. การประเมินการใช้แรงและการใช้กล้ามเนื้อในการทำงาน (ส่วนมือและส่วนแขน)



รูปที่ ข.11 แสดงหน้าต่างสำหรับการประเมินการใช้แรงและการใช้กล้ามเนื้อ
ในการทำงานของส่วนมือและส่วนแขน(1)



รูปที่ ข.12 แสดงหน้าต่างสำหรับการประเมินการใช้แรงและการใช้กล้ามเนื้อ
ในการทำงานของส่วนมือและส่วนแขน(2)

5.1 การประเมินการใช้แรงในการทำงาน

5.1.1 ภาระงานที่ทำได้แก่แรงที่ใช้หรือน้ำหนักที่ถือได้น้อยกว่า 2 กิโลกรัมให้คะแนนเป็น 0

5.1.2 ถ้าภาระงานอยู่ระหว่าง 2-10 กก. ถือหรือใช้แรงนานๆ ครั้งให้คะแนนเป็น 1

5.1.3 ถ้าภาระงานอยู่ระหว่าง 2-10 กก. ถือหรือใช้แรงตลอดเวลาหรือทำซ้ำๆ ไปมาบ่อยๆ ให้คะแนนเป็น 2

5.1.4 ถ้าภาระงานมากกว่า 10 กก. ถือหรือใช้แรงแบบสลับหรือเคลื่อนที่ซ้ำๆ ไปมาบ่อยๆ หรือมีการใช้แรงทำงานดังกล่าวอย่างรวดเร็วให้คะแนนเป็น 3

5.2 การประเมินการใช้กล้ามเนื้อในการทำงาน

5.2.1 ถ้าการทำงานคิดว่ามีลักษณะการใช้แรงจากกล้ามเนื้อแบบสลับเช่นมีการใช้แรงโดยเกร็งกล้ามเนื้อต่อเนื่องนานกว่า 1 นาที ให้คะแนนเป็น 1

5.2.2 ถ้าการทำงานเป็นแบบซ้ำ ๆ โดยมีการเคลื่อนไหวกลับไปกลับมาเกินกว่าครั้งต่อ นาทีหรือมากกว่าให้บวกคะแนนเพิ่มอีก

5.2.3 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน 2 คะแนน

6. การประเมินตำแหน่งของศีรษะและคอ

ทำการคลิกเพื่อเลือกให้คะแนนท่าทางตำแหน่งของศีรษะและคอ ซึ่งจะมีรายละเอียดการให้คะแนน ดังนี้



รูปที่ ข-13 แสดงหน้าต่างสำหรับการประเมินตำแหน่งของคอ

- 6.1 ถ้ามุมก้มอยู่ระหว่าง 0-10 องศา ให้คะแนนเป็น 1
- 6.2 ถ้ามุมก้มอยู่ระหว่าง 10-20 องศา ให้คะแนนเป็น 2
- 6.3 ถ้ามุมก้มมากกว่า 20 องศา ขึ้นไปให้คะแนนเป็น 3
- 6.4 ถ้ามีการเงยศีรษะให้คะแนนเป็น 4
- 6.5 ถ้ามีการหมุน (twist) ศีรษะด้วยให้คะแนนเพิ่มอีก +1
- 6.6 ถ้ามีการเอียงศีรษะไปด้านข้างให้คะแนนเพิ่มอีก +1
- 6.7 คะแนนสูงสุดในขั้นตอนนี้จะมิตค่าไม่เกิน 6 คะแนน

7. การประเมินตำแหน่งของลำตัว

ทำการคลิกเพื่อเลือกให้คะแนนท่าทางตำแหน่งของลำตัว ซึ่งจะมีรายละเอียดการให้คะแนน ดังนี้



รูปที่ ข.14 แสดงหน้าต่างสำหรับการประเมินตำแหน่งของลำตัว

7.1 ลำตัวควรอยู่ในลักษณะที่ตั้งขม่อมือยื่นหรือในกรณีการนั่งมีพนักพิงรองรับอย่างดีที่มุมเอียงไม่เกิน -20 องศา ให้คะแนนเป็น 1

7.2 ลำตัวโน้มไปด้านหน้าระหว่าง 0-20 องศา ให้คะแนนเป็น 2

7.3 ลำตัวโน้มไปด้านหน้าระหว่าง 21-60 องศา ให้คะแนนเป็น 3

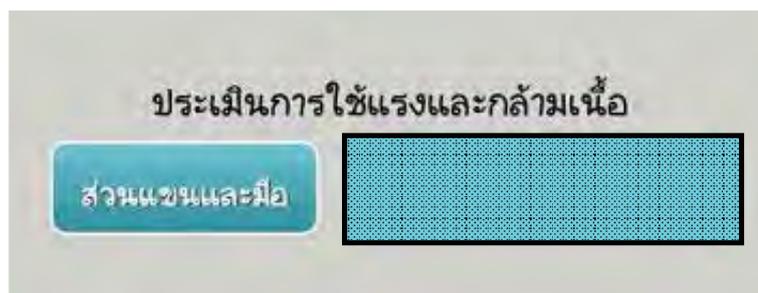
7.4 ลำตัวโน้มไปด้านหน้าระหว่างมากกว่า 60 องศา ให้คะแนนเป็น 4

7.5 ลำตัวมีการหมุนให้คะแนนเพิ่มอีก+1

7.6 ลำตัวมีการเอียงไปด้านข้างให้คะแนนเพิ่มอีก+1

7.7 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน 6 คะแนน

8.การประเมินการใช้แรงและการใช้กล้ามเนื้อในการทำงาน(ส่วนคอ ลำตัวและส่วนขา)



รูปที่ ข.15 แสดงปุ่มสำหรับการประเมินการใช้แรงและการใช้กล้ามเนื้อในการทำงานของส่วนคอ ลำตัวและส่วนขา



รูปที่ ข.14 แสดงหน้าต่างสำหรับการประเมินการใช้แรงและการใช้กล้ามเนื้อ
ในการทำงานของส่วนคอ ลำตัวและส่วนขา

8.1 การประเมินการใช้แรงในการทำงาน

8.1.1 ภาระงานที่ทำได้แก่แรงที่ใช้หรือน้ำหนักที่ถือน้อยกว่า 2 กิโลกรัม ให้คะแนนเป็น 0

8.1.2 ถ้าภาระงานอยู่ระหว่าง 2-10 กก. ถือหรือใช้แรงนานๆครั้ง ให้คะแนนเป็น 1

8.1.3 ถ้าภาระงานอยู่ระหว่าง 2-10 กก. ถือหรือใช้แรงตลอดเวลาหรือทำซ้ำๆไปมาบ่อยๆให้คะแนนเป็น 2

8.1.4 ถ้าภาระงานมากกว่า 10 กก. ถือหรือใช้แรงแบบสติดหรือเคลื่อนที่ซ้ำๆไปมาบ่อยๆ หรือมีการใช้แรงทำงานดังกล่าวอย่างรวดเร็วให้คะแนนเป็น 3

8.2 การประเมินการใช้กล้ามเนื้อในการทำงาน

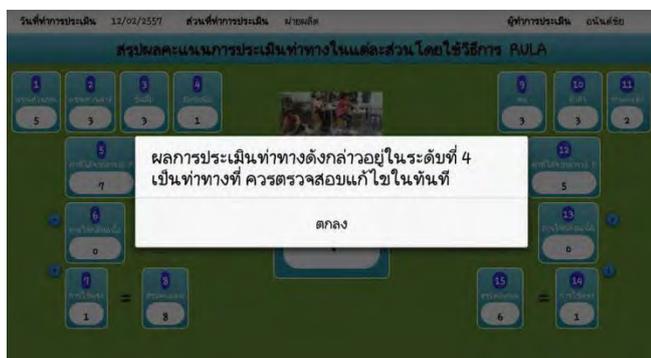
8.2.1 ถ้าการทำงานดังกล่าวมีลักษณะการใช้แรงจากกล้ามเนื้อแบบสติดเช่นมีการใช้แรงโดยเกร็งกล้ามเนื้อต่อเนื่องนานกว่า 1 นาทีให้คะแนนเป็น 1

8.2.2 ถ้าการทำงานเป็นแบบซ้ำๆโดยมีการเคลื่อนไหวกลับไปกลับมาเกินกว่า 4 ครั้งต่อนาทีหรือมากกว่าให้บวกคะแนนเพิ่มอีก

8.2.3 คะแนนสูงสุดในขั้นตอนนี้มีค่าไม่เกิน 2 คะแนน

เมื่อทำการประเมินครบแล้ว ให้ทำการกดที่ปุ่ม  เพื่อให้โปรแกรมประเมินค่าการทำงานโดยใช้วิธีการ RULA แล้วจะปรากฏหน้าจอสรุปผลการทำงาน ดังภาพ

หากเราลืมทำการประเมินในส่วนใดก็ตาม โปรแกรมจะแจ้งเตือนให้เราทำการประเมินให้เรียบร้อยก่อน จึงจะสามารถทำการประเมินผลโดยใช้โปรแกรมได้



รูปที่ ข.16 แสดงหน้าจอสรุปผลการประเมินของโปรแกรม Rula SU
หมายเหตุ ขณะเราทำการประเมิน เราสามารถเรียกดูรูปท่าทางการทำงานได้ โดยกดที่ปุ่ม



ภาคผนวก ค
ตารางการเก็บข้อมูลของเวลา

ตารางที่ ค.1 แสดงการเก็บเวลาของการประเมินท่าทางการทำงานโดยใช้วิธีการRULA จากการใช้งานโปรแกรม RulaSU

ขั้นตอน	เวลา (วินาที)										เฉลี่ย
	ครั้งที่										
	1	2	3	4	5	6	7	8	9	10	
1. กรอกข้อมูลของผู้ทำการประเมิน	21.1	19.8	20.2	20.4	19.9	20.1	20.1	19.8	19.9	21.7	20.3
2. ทำการประเมินท่าทางการทำงานในส่วน ของแขนส่วนบน	2.7	2.3	2.5	2.6	2.4	2.4	2.5	2.6	2.5	2.5	2.5
3. ทำการประเมินท่าทางการทำงานในส่วนของแขนส่วนล่าง	2.5	2.5	2.7	2.7	2.4	2.3	2.6	2.7	2.7	2.3	2.5
4. ทำการประเมินการงอของมือ	2.0	1.9	2.2	1.8	1.9	2.1	2.1	2.1	2.0	1.9	2.0
5. ทำการประเมินการบิดของมือ	1.7	1.6	1.4	1.4	1.4	1.5	1.6	1.4	1.5	1.5	1.5
6. ทำการประเมินท่าทางการทำงานในส่วน ของคอ	2.9	2.8	2.6	2.7	2.5	2.4	2.4	2.5	2.6	2.6	2.6
7. ทำการประเมินท่าทางการทำงานในส่วน ของลำตัว	3.3	3.0	2.9	2.8	3.1	2.9	3.0	3.1	3.1	2.8	3.0
8. ทำการประเมินตำแหน่งของขาและเท้า	1.4	1.5	1.5	1.3	1.3	1.3	1.5	1.4	1.4	1.4	1.4
9. การประเมินการใช้แรงและการใช้กล้ามเนื้อ ในส่วนของแขนและมือ	3.8	3.8	3.6	3.6	3.6	3.4	3.5	3.4	3.7	3.6	3.6
10. การประเมินการใช้แรงและการใช้กล้ามเนื้อในส่วนของคอ ลำตัว และขา	3.6	3.6	3.7	3.9	3.8	4.0	4.0	3.8	3.7	3.8	3.8
11. ตรวจสอบผลการประเมินภาระงาน	2.5	2.5	2.5	2.4	2.7	2.6	2.3	2.5	2.5	2.5	2.5
รวม	47.5	45.3	45.8	45.6	45	45	45.6	45.3	45.6	45.6	45.7

ตารางที่ ค.2 แสดงการเก็บเวลาของการประเมินท่าทางการทำงานโดยใช้วิธีการRULA จากการใช้งานโปรแกรม ErgoIntelligence

ขั้นตอน	เวลา (วินาที)										เฉลี่ย
	ครั้งที่										
	1	2	3	4	5	6	7	8	9	10	
1. กรอกข้อมูลของผู้ทำการประเมิน	20.4	20.0	20.2	20.3	20.3	20.1	20.2	20.1	20.2	20.2	20.2
2. ทำการประเมินตำแหน่งท่าทางของมือและ การบิดของมือ	3.1	3.1	3.0	2.9	3.0	2.9	3.0	3.0	3.1	2.9	3.0
3. ทำการประเมินท่าทางการทำงานในส่วน ของแขนส่วนล่าง	2.0	2.0	2.3	2.1	2.3	2.1	2.1	2.0	2.1	2.1	2.1
4. ทำการประเมินท่าทางการทำงานในส่วน ของแขนส่วนบน	2.4	2.4	2.5	2.4	2.3	2.3	2.5	2.4	2.5	2.3	2.4
5. ทำการประเมินท่าทางการทำงานในส่วน ของคอ	2.1	2.1	2.1	2.0	1.9	2.0	1.9	1.9	2.0	2.0	2.0
6. ทำการประเมินท่าทางการทำงานในส่วน ของลำตัว	2.2	2.1	2.1	2.2	2.0	2.1	2.1	2.0	2.1	2.1	2.1
7. ทำการประเมินท่าทางการทำงานในส่วน ของขาและเท้า	1.8	1.7	1.8	2.0	2.0	2.1	1.9	2.0	1.8	1.9	1.9
8. ทำการประเมินการใช้แรงและกล้ามเนื้อในการทำงาน	2.3	2.0	2.5	2.2	2.3	2.4	2.3	2.3	2.4	2.2	2.3
9. ตรวจสอบผลการประเมินภาระงาน	4.3	4.3	4.3	4.2	4.1	4.4	4.3	4.5	4.3	4.3	4.3
รวม	40.6	39.7	40.8	40.3	40.2	40.4	40.3	40.2	40.5	40.0	40.3

ตารางที่ ค.3 แสดงการเก็บเวลาของการประเมินท่าทางการทำงานโดยใช้วิธีการRULA จากการ
คำนวณมือ

ขั้นตอน	เวลา (วินาที)										เฉลี่ย
	ครั้งที่										
	1	2	3	4	5	6	7	8	9	10	
1. กรอกระดานการประเมินท่าทางการทำงานในส่วนของแขนส่วนบนลงบนแบบฟอร์ม	3.9	3.9	3.8	3.9	4.0	3.9	4.0	3.8	3.9	3.9	3.9
2. กรอกระดานการประเมินท่าทางการทำงานในส่วนของแขนส่วนล่างลงบนแบบฟอร์ม	4.3	4.2	4.3	4.3	4.4	4.5	4.3	4.1	4.3	4.3	4.3
3. กรอกระดานการประเมินการงอของมือลงบนแบบฟอร์ม	4.3	4.4	4.4	4.2	4.4	4.3	4.2	4.4	4.4	4.5	4.4
4. กรอกระดานการประเมินการบิดของมือลงบนแบบฟอร์ม	3.5	3.5	3.4	3.3	3.3	3.4	3.3	3.5	3.4	3.4	3.4
5. ตรวจสอบค่าคะแนนที่ได้จากตาราง A เพื่อสรุปคะแนนการประเมินท่าทางส่วนแขนและมือ	27.0	26.9	27.3	26.9	26.8	26.8	26.9	27.0	26.8	26.8	26.9
6. กรอกผลคะแนนที่ได้ลงบนแบบฟอร์ม	3.9	3.9	3.8	3.7	3.8	3.9	3.7	3.7	3.8	3.8	3.8
7. กรอกระดานการประเมินท่าทางการทำงานในส่วนของคอลงบนแบบฟอร์ม	4.0	4.0	4.1	4.0	4.1	3.9	4.1	3.9	3.9	4.0	4.0
8. กรอกระดานการประเมินท่าทางการทำงานในส่วนของลำตัวลงบนแบบฟอร์ม	4.0	3.8	3.7	3.5	3.8	3.7	3.8	3.8	3.7	3.6	3.7

ตารางที่ ค.3 แสดงการเก็บเวลาของการประเมินท่าทางการทำงานโดยใช้วิธีการRULA จากการ
คำนวณมือ(ต่อ)

ขั้นตอน	เวลา (วินาที)										เฉลี่ย
	ครั้งที่										
	1	2	3	4	5	6	7	8	9	10	
9. กรอกระดานการประเมิน ตำแหน่งของขาและเท้าลงบน แบบฟอร์ม	4.0	4.0	3.8	3.8	3.7	3.7	3.7	3.8	3.9	3.8	3.8
10. ตรวจสอบค่าคะแนนที่ได้จาก ตาราง B เพื่อสรุปคะแนนการ ประเมิน ท่าทางส่วนคอ ลำตัว และขา	10.5	10.7	10.7	10.8	10.9	10.9	10.7	10.7	10.8	10.7	10.7
11. กรอกระดานที่ได้ลงบน แบบฟอร์ม	3.8	3.8	3.9	3.7	4.0	4.1	4.0	3.8	3.7	3.8	3.9
12. กรอกระดานการประเมินการ ใช้แรงและการใช้กล้ามเนื้อใน ส่วนของแขนและมือลง บน แบบฟอร์ม	3.7	3.8	3.6	3.7	3.8	3.8	3.7	3.6	3.6	3.8	3.7
13. กรอกระดานการประเมินการ ใช้แรงและการใช้กล้ามเนื้อใน ส่วนของคอ ลำตัว และขา ลงบน แบบฟอร์ม	4.0	4.2	4.2	4.0	4.1	4.1	4.2	4.0	4.0	4.2	4.1
14. ตรวจสอบค่าคะแนนที่ได้จาก ตาราง C เพื่อสรุปคะแนนการ ประเมิน ท่าทางการทำงาน โดยใช้วิธีการ RULA	3.9	3.9	3.8	3.9	3.7	3.7	3.8	3.8	3.7	3.9	3.8
15. กรอกระดานที่ได้ลงบน แบบฟอร์ม	3.4	3.4	3.3	3.4	3.2	3.3	3.2	3.2	3.3	3.4	3.3
16. ตรวจสอบการกำหนดระดับ การปฏิบัติงาน จากค่าคะแนนที่ได้	2.5	2.5	2.7	2.7	2.6	2.7	2.7	2.5	2.6	2.6	2.6
รวม	90.7	90.9	90.8	89.8	90.6	90.7	90.3	89.6	89.8	90.5	90.3

ภาคผนวก ง
แบบฟอร์มการประเมินโปรแกรม

แบบประเมินผลความพึงพอใจในการใช้งาน
โปรแกรมการประเมินภาระงานโดยใช้หลักการทางกายศาสตร์

วันที่ทำการประเมิน / /

คำชี้แจง ให้ท่านทำเครื่องหมาย“□” ในช่อง หน้าข้อความที่เห็นว่าถูกต้องที่สุด ตามความเข้าใจของท่าน
 ข้อมูลทั่วไป

1.ข้อมูลเกี่ยวกับผู้ประเมิน

เพศ ชาย หญิง

2.ข้อมูลของโปรแกรมที่ทำการประเมิน

โปรแกรมที่ใช้งาน

RulaSU

3.ข้อมูลการประเมินผลการใช้งานโปรแกรมการประเมินภาระงานทางกายศาสตร์

คำชี้แจง ให้ท่านทำการประเมินผลการใช้งานและเลือกข้อมูลตามระดับความพึงพอใจ

รายละเอียด	ระดับการประเมินผล				
	มากที่สุด 5	มาก 4	ปานกลาง 3	น้อย 2	น้อยที่สุด 1
ความคิดเห็นด้านความพึงพอใจต่อการใช้งานระบบ (Usability Test)					
1. การติดตั้งโปรแกรม					
1.1) การติดตั้งโปรแกรมสามารถทำได้โดยง่าย					
2. การใช้งานโปรแกรม					
2.1) การง่ายต่อการเข้าสู่การใช้งานโปรแกรม					
รายละเอียด	ระดับการประเมินผล				
	มากที่สุด 5	มาก 4	ปานกลาง 3	น้อย 2	น้อยที่สุด 1
2.2) มีรูปแบบหน้าจอของแต่ละส่วนที่เป็นมาตรฐานเดียวกันทำให้เข้าใจได้ง่าย					

รายละเอียด	ระดับการประเมินผล				
	มากที่สุด 5	มาก 4	ปานกลาง 3	น้อย 2	น้อยที่สุด 1
2.3) ความเหมาะสมในการเลือกใช้ชนิดของตัวอักษรบนจอภาพ					
2.4) ความเหมาะสมในการเลือกใช้ขนาดของตัวอักษรบนจอภาพ					
2.5) ความเหมาะสมในการเลือกใช้สีของตัวอักษรบนและรูปภาพ					
2.6) ความเหมาะสมในการใช้ข้อความเพื่ออธิบายสื่อความหมาย					
2.7) ความเหมาะสมในการใช้รูปภาพหรือสัญลักษณ์ในการสื่อความหมาย					
2.8) ความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆของโปรแกรมบนจอภาพ					
2.9) ความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆของโปรแกรมบนจอภาพ					
แบบสอบถามความพึงพอใจด้านการทำงานได้ตามฟังก์ชันงานของระบบ (Function Test)					
1. ความถูกต้องในการจัดเก็บข้อมูลนำเข้าเพื่อใช้ในการประเมิน					
2. ความถูกต้องในการแก้ไข/เปลี่ยนแปลงข้อมูลนำเข้า					
3. ความถูกต้องในการแสดงผลลัพธ์ที่ได้จากการใช้งาน โปรแกรม					
4. การป้องกันข้อผิดพลาดที่อาจเกิดขึ้นในการใช้งาน					
5. ความรวดเร็วในการประมวลผลของโปรแกรม					
แบบสอบถามความพึงพอใจด้านการตรงตามความต้องการของผู้ใช้ระบบ (Functional Requirement Test)					
1. เพิ่มความรวดเร็วในการใช้งาน					
2. สามารถเรียนรู้วิธีการใช้งานได้ด้วยตนเอง					
3. มีความเข้าใจในเนื้อหาวิชาเพิ่มขึ้นจา					

รายละเอียด	ระดับการประเมินผล				
	มากที่สุด 5	มาก 4	ปานกลาง 3	น้อย 2	น้อยที่สุด 1
การใช้งานโปรแกรม					
4. ช่วยลดขั้นตอนและปัญหาต่างๆในการปฏิบัติงานได้					
5. ผลลัพธ์ที่ได้จากการใช้งานมีความสอดคล้องกับระบบงานจริง					
6. เป็นประโยชน์ต่อการใช้งานของผู้ที่สนใจ					

ข้อเสนอแนะเกี่ยวกับการใช้งานโปรแกรม

.....

.....

.....

.....

.....

.....

.....

เกณฑ์การให้คะแนนของแบบประเมินผลการใช้งานโปรแกรมการประเมินภาระงานทาง การยศาสตร์

ตารางที่ ง.1 ตารางแสดงรายละเอียดการให้คะแนนการประเมินผลความพึงพอใจด้านความง่ายต่อ
การใช้งานระบบ (Usability Test)

เรื่อง	รายละเอียด
ความง่ายต่อการใช้งานระบบ (Usability Test)	1. การติดตั้งโปรแกรม 1.1) การติดตั้งโปรแกรมสามารถทำได้โดยง่าย

เกณฑ์การให้คะแนน :

- 5 คะแนนสำหรับ สามารถติดตั้งโปรแกรมได้ด้วยตนเอง
- 4 คะแนนสำหรับ อ่านคู่มือการใช้งาน ขั้นตอนการติดตั้งโปรแกรม และสามารถติดตั้งโปรแกรมได้ด้วยตนเอง
- 3 คะแนนสำหรับ อ่านคู่มือการใช้งาน ขั้นตอนการติดตั้งโปรแกรม และได้รับคำแนะนำขั้นตอนการติดตั้งต่างๆเพิ่มเติม
- 2 คะแนนสำหรับ ผู้จัดทำ สอนการติดตั้งโปรแกรม
- 1 คะแนนสำหรับ ไม่สามารถทำการติดตั้งโปรแกรมด้วยตนเอง

ตารางที่ 1.1 ตารางแสดงรายละเอียดการให้คะแนนการประเมินผลความพึงพอใจด้านความง่ายต่อการใช้งานระบบ (Usability Test) (ต่อ)

เรื่อง	รายละเอียด
ความง่ายต่อการใช้งานระบบ (Usability Test)	2.2) มีรูปแบบหน้าจอของแต่ละส่วนที่เป็นมาตรฐานเดียวกันทำให้เข้าใจได้ง่าย
	2.3) ความเหมาะสมในการเลือกใช้ชนิดของตัวอักษรบนจอภาพ
	2.4) ความเหมาะสมในการเลือกใช้ขนาดของตัวอักษรบนจอภาพ
	2.5) ความเหมาะสมในการเลือกใช้สีของตัวอักษรบนและรูปภาพ
	2.6) ความเหมาะสมในการใช้ข้อความเพื่ออธิบายสื่อความหมาย
	2.7) ความเหมาะสมในการใช้รูปภาพหรือสัญลักษณ์ในการสื่อความหมาย
	2.8) ความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆของโปรแกรมบนจอภาพ
	2.9) ความเหมาะสมในการวางตำแหน่งส่วนประกอบต่างๆ ของโปรแกรมบนจอภาพ

เกณฑ์การให้คะแนน :

5 คะแนนสำหรับ การออกแบบการแสดงผลต่างๆของโปรแกรมเข้ากันเป็นอย่างดี ทำให้มองสบายตา

4 คะแนนสำหรับ การออกแบบการแสดงผลต่างๆของโปรแกรมมีความเหมาะสม

3 คะแนนสำหรับ การออกแบบการแสดงผลต่างๆของโปรแกรม เรียบ ไม่โดดเด่น

2 คะแนนสำหรับ การออกแบบการแสดงผลต่างๆของโปรแกรมยังไม่เหมาะสม มองแล้วไม่สบายตา

1 คะแนนสำหรับ การออกแบบการแสดงผลต่างๆของโปรแกรมมีความยุ่งยาก และทำให้เกิดความสับสน

ตารางที่ 2.2 ตารางแสดงรายละเอียดการให้คะแนนการประเมินผลความพึงพอใจด้านการทำงานได้
ตามฟังก์ชันงานของระบบ (Function Test)

เรื่อง	รายละเอียด
การทำงานได้ตามฟังก์ชันงานของระบบ (Function Test)	1. ความถูกต้องในการจัดเก็บข้อมูลนำเข้าเพื่อใช้ในกาประเมิน
	2. ความถูกต้องในการแก้ไข/เปลี่ยนแปลงข้อมูลนำเข้า
	3. ความถูกต้องในการแสดงผลลัพธ์ที่ได้จากการใช้งานโปรแกรม
	4. การป้องกันข้อผิดพลาดที่อาจเกิดขึ้นในการใช้งาน
	5. ความรวดเร็วในการประมวลผลของโปรแกรม

เกณฑ์การให้คะแนน :

5 คะแนนสำหรับ การทำงานของโปรแกรมสามารถแสดงผลข้อมูลของขั้นตอนต่างๆได้
เป็นเป็นอย่างดี

4 คะแนนสำหรับ การทำงานของโปรแกรมสามารถแสดงผลข้อมูลของขั้นตอนต่างๆได้
อย่างเหมาะสม ครบตามความต้องการของผู้ใช้งาน

3 คะแนนสำหรับ การทำงานของโปรแกรมสามารถแสดงผลข้อมูลของขั้นตอนต่างๆได้
อย่างเพียงพอ

2 คะแนนสำหรับ โปรแกรมมีการแสดงข้อมูลที่ต่างๆ ไม่เหมาะสมต่อความต้องการของ
ผู้ใช้งาน

1 คะแนนสำหรับ การทำงานของโปรแกรมมีความผิดพลาด

ตารางที่ 3.3 ตารางแสดงรายละเอียดการให้คะแนนการประเมินผลความพึงพอใจด้านการตรงตามความต้องการของผู้ใช้ระบบ(Function Requirement Test)

เรื่อง	รายละเอียด
ด้านการตรงตามความต้องการของผู้ใช้ระบบ (Functional Requirement Test)	1. เพิ่มความรวดเร็วในการใช้งาน
	2. สามารถเรียนรู้วิธีการใช้งานได้ด้วยตนเอง
	3. มีความเข้าใจในเนื้อหาวิชาเพิ่มขึ้นจากการใช้งานโปรแกรม
	4. ช่วยลดขั้นตอนและปัญหาต่างๆในการปฏิบัติงานได้
	5. ผลลัพธ์ที่ได้จากการใช้งานมีความสอดคล้องกับระบบงานจริง

เกณฑ์การให้คะแนน :

5 คะแนนสำหรับ ได้รับประโยชน์จากการใช้งานโปรแกรม สามารถช่วยลดเวลา ขั้นตอนในการทำงานต่างๆ รวมถึงเข้าใจหลักการวิธีการทางกายศาสตร์ได้เป็นอย่างดี

4 คะแนนสำหรับ ได้รับประโยชน์จากการใช้งานโปรแกรม สามารถช่วยลดเวลา ขั้นตอนในการทำงานต่างๆ รวมถึงเข้าใจหลักการวิธีการทางกายศาสตร์ได้เพิ่มขึ้นเพียงเล็กน้อย

3 คะแนนสำหรับ ไม่ได้ได้รับประโยชน์จากการใช้งานโปรแกรม วิธีการทำงานเดิมการคำนวณมือ) กับการใช้งานโปรแกรมไม่มีความแตกต่างกัน

2 คะแนนสำหรับ การใช้งานโปรแกรมเพิ่มขึ้นขั้นตอนการทำงาน และดูข้อมูลต่างๆ ไม่สะดวก

1 คะแนนสำหรับ การใช้งานโปรแกรมเพิ่มขึ้นขั้นตอนการทำงาน เวลาในการทำงานมากขึ้น ไม่เอื้อประโยชน์ เกิดความวุ่นวายและสับสน

ภาคผนวก จ
Code การทำงานโปรแกรม

โปรแกรม RulaSU

Code แสดงการทำงานของโปรแกรม เมื่อทำการกดปุ่ม 'ประเมินภาระงาน

ชื่อฟอร์ม ArmAndHandTestForceAndMuscle.java

```
package com.itsada.rula;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.EditText;

import android.widget.RadioButton;

import com.itsada.Business.Config;

import com.itsada.Business.IAcitivity;

public class ArmAndHandTestForceAndMuscle extends BaseActivity implements

        IAcitivity, OnClickListener {

    private Navigation navigation;

    private Button buttonOK;

    private CheckBox checkBoxForce1;
```

```
private CheckBox checkBoxForce2;

private CheckBox checkBoxForce3;

private CheckBox checkBoxForce4;

private RadioButton checkBoxMuscle1;

private RadioButton checkBoxMuscle2;

private EditText etScoreForce;

private EditText etScoreMuscle;

private int scoreForce = -1;

private int scoreMuscle = -1;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.arm_hand_force_muscle);

    navigation = new Navigation();

    Initial();

    Listener();

}

@SuppressWarnings("NewApi")

@Override

public void onClick(View v) {
```

```
etScoreForce.setError(null);

etScoreMuscle.setError(null);

switch (v.getId()) {

case R.id.checkBoxForce1:

    if (((CheckBox) v).isChecked()) {

        checkBoxForce2.setChecked(false);

        checkBoxForce3.setChecked(false);

        checkBoxForce4.setChecked(false);

        scoreForce = 0;

        etScoreForce.setText(String.valueOf(scoreForce));

    }

    break;

case R.id.checkBoxForce2:

    if (((CheckBox) v).isChecked()) {

        checkBoxForce1.setChecked(false);

        checkBoxForce3.setChecked(false);

        checkBoxForce4.setChecked(false);

        scoreForce = 1;

        etScoreForce.setText(String.valueOf(scoreForce));

    }

}
```

```
        break;

    case R.id.checkBoxForce3:

        if (((CheckBox) v).isChecked()) {

            checkBoxForce1.setChecked(false);

            checkBoxForce2.setChecked(false);

            checkBoxForce4.setChecked(false);

            scoreForce = 2;

            etScoreForce.setText(String.valueOf(scoreForce));

        }

        break;

    case R.id.checkBoxForce4:

        if (((CheckBox) v).isChecked()) {

            checkBoxForce1.setChecked(false);

            checkBoxForce2.setChecked(false);

            checkBoxForce3.setChecked(false);

            scoreForce = 3;

            etScoreForce.setText(String.valueOf(scoreForce));

        }

        break;

    case R.id.checkBoxMuscle1:
```

```
        if (((RadioButton) v).isChecked()) {

            scoreMuscle += 1;

            etScoreMuscle.setText(String.valueOf(scoreMuscle));

        }else{

            scoreMuscle -= 1;

            etScoreMuscle.setText(String.valueOf(scoreMuscle));

        }

        break;

    case R.id.checkBoxMuscle2:

        if (((RadioButton) v).isChecked()) {

            scoreMuscle += 1;

            etScoreMuscle.setText(String.valueOf(scoreMuscle));

        }else{

            scoreMuscle -= 1;

            etScoreMuscle.setText(String.valueOf(scoreMuscle));

        }

        break;

    case R.id.ButtonOK:

        if (validate(scoreForce, etScoreForce, "กรุณาประเมินการใช้แรง")

            && validate(scoreMuscle, etScoreMuscle,
```

```

        "กรุณาประเมินการใช้กล้ามเนื้อ") {
            if (scoreForce > 6) {
                etScoreForce
                    .setError("กรุณาตรวจสอบ เนื่องจาก
คะแนนที่ได้มากกว่า6 คะแนน");
            } else if (scoreMuscle > 2) {
                etScoreForce
                    .setError("กรุณาตรวจสอบ เนื่องจาก
คะแนนที่ได้มากกว่า2 คะแนน");
            }

            Intent intentOk = new Intent(
                getBaseContext(),
                navigation
                    .getPrevious(navigation.ArmAndHandTestForceAndMuscle));

            saveData(Config.FORCE_A, scoreForce);

            saveData(Config.MUSCLE_A, scoreMuscle);

            startActivity(intentOk);

            this.finish();
        }

        break;

```

```
        default:
            break;
    }
}

@Override

public void Initial() {

    etScoreForce = (EditText) findViewById(R.id.etScoreForce);

    etScoreMuscle = (EditText) findViewById(R.id.etScoreMuscle);

    checkBoxForce1 = (CheckBox) findViewById(R.id.checkBoxForce1);

    checkBoxForce2 = (CheckBox) findViewById(R.id.checkBoxForce2);

    checkBoxForce3 = (CheckBox) findViewById(R.id.checkBoxForce3);

    checkBoxForce4 = (CheckBox) findViewById(R.id.checkBoxForce4);

    checkBoxMuscle1 = (RadioButton) findViewById(R.id.checkBoxMuscle1);

    checkBoxMuscle2 = (RadioButton) findViewById(R.id.checkBoxMuscle2);

    this.buttonOK = (Button) findViewById(R.id.ButtonOK);

}

@Override

public void Listener() {

    this.buttonOK.setOnClickListener(this);

    checkBoxForce1.setOnClickListener(this);
```

```

        checkBoxForce2.setOnClickListener(this);

        checkBoxForce3.setOnClickListener(this);

        checkBoxForce4.setOnClickListener(this);

        checkBoxMuscle1.setOnClickListener(this);

        checkBoxMuscle2.setOnClickListener(this);

    }

    @Override

    public void onBackPressed() {

        super.onBackPressed();

    }

}

```

ชื่อฟอร์ม Assets.java

```

package com.itsada.rula;

import com.itsada.framework.Image;

public class Assets {

    public static Image splash;

    public static String testDate = "วันที่ทำการประเมิน";

    public static String testPlace = "สถานที่ทำการประเมิน";

    public static String testName = "ผู้ทำการประเมิน";

    // public static String Label1 = "-20 ถึง 20";

```

```

//      public static String Label2 = "มากกว่า -20";

//      public static String Label3 = "20 ถึง 45";

//      public static String Label4 = "45 ถึง 90";

//      public static String Label5 = "มากกว่า 90";

//

public static String Neck = "คอ";

public static String Hand = "มือ";

public static String Body = "ลำตัว";

public static String UpperArm = "แขนส่วนบน (หัวไหล่ถึงข้อศอก)";

public static String LowerArm = "แขนส่วนล่าง (ข้อศอกถึงข้อมือ)";

public static String Leg = "ขาและเท้า";

public static String LekAndFoot;

}

```

ชื่อฟอร์ม BaseActivity.java

```

package com.itsada.rula;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.content.pm.ActivityInfo;

import android.os.Bundle;

import android.view.Window;

```

```
import android.view.WindowManager;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.TextView;

import com.itsada.Business.AndroidEntity;

import com.itsada.Business.AndroidException;

public class BaseActivity extends AndroidEntity {

    protected TextView title;

    protected ImageButton home;

    protected ImageButton left;

    protected ImageButton right;

    protected AndroidException exceptionHandle;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        requestWindowFeature(Window.FEATURE_NO_TITLE);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,

                               WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
```

```
//rula = new Rula();

exceptionHandle = new AndroidException(getApplicationContext());

}

protected boolean validate(int value) {

    if (value == -1)

        return false;

    return true;

}

protected boolean validate(int value, EditText et, String message) {

    if (value == -1) {

        et.setError(message);

        et.requestFocus();

        return false;

    }

    return true;

}

protected boolean validate(String value) {

    if (value == "")

        return false;

    return true;

}
```

```
}  
  
protected boolean checkValidate(EditText et, String message) {  
  
    boolean validt = false;  
  
    if (et.getText().length() == 0) {  
  
        et.setError(message);  
  
        validt = false;  
  
        if (et.getText().length() != 0) {  
  
            et.requestFocus();  
  
        }  
  
    } else {  
  
        validt = true;  
  
    }  
  
    return validt;  
  
}  
  
protected TextView SetTextView(TextView textView, int id, String text) {  
  
    textView = (TextView) findViewById(id);  
  
    textView.setText(text);  
  
    return textView;  
  
}
```

```

protected Button SetButton(Button button, int id) {

    button = (Button) findViewById(id);

    return button;

}

public void onBackPressed() {

    new AlertDialog.Builder(this)

        .setMessage("Are you sure you want to exit?")

        .setCancelable(false)

        .setPositiveButton("Yes",

            new DialogInterface.OnClickListener() {

                public void onClick(DialogInterface

dialog, int id) {

                    BaseActivity.this.finish();

                }

            })

        .setNegativeButton("No", null).show();

    }

}

```

ชื่อฟอร์ม Body.java

```

package com.itsada.rula;

import android.annotation.SuppressLint;

```

```
import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.LinearLayout;

import com.itsada.Business.Config;

import com.itsada.Business.IAcitivity;

public class Body extends BaseActivity implements IAcitivity,

        OnClickListener {

    private Navigation navigation;

    private View image1;

    private View image2;

    private View image3;

    private View image4;

    private LinearLayout layout;

    private CheckBox checkBox1;
```

```
private CheckBox checkBox2;

private Button button1;

private Button button2;

private Button button3;

private Button button4;

private EditText etScore;

private int score = -1;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.body);

    navigation = new Navigation();

    Initial();

    Listener();

}

@SuppressWarnings("NewApi")

@Override

public void onClick(View v) {

    etScore.setError(null);

    switch (v.getId()) {
```

```
case R.id.imageViewBody1:
```

```
case R.id.buttonBody_1:
```

```
    score = 1;
```

```
    setCheck(false);
```

```
    setEnable(0);
```

```
    etScore.setText(String.valueOf(score));
```

```
    break;
```

```
case R.id.imageViewBody2:
```

```
case R.id.buttonBody_2:
```

```
    score = 2;
```

```
    setCheck(false);
```

```
    setEnable(0);
```

```
    etScore.setText(String.valueOf(score));
```

```
    break;
```

```
case R.id.imageViewBody3:
```

```
case R.id.buttonBody_3:
```

```
    score = 3;
```

```
    setCheck(false);
```

```
    setEnable(0);
```

```
    etScore.setText(String.valueOf(score));
```

```
        break;

    case R.id.imageViewBody4:

    case R.id.buttonBody_4:

        score = 4;

        setCheck(false);

        setEnable(0);

        etScore.setText(String.valueOf(score));

        break;

    case R.id.checkBoxBody_1:

        if (((CheckBox) v).isChecked()) {

            score += 1;

            etScore.setText(String.valueOf(score));

        } else {

            score -= 1;

            etScore.setText(String.valueOf(score));

        }

        break;

    case R.id.checkBoxBody_2:

        if (((CheckBox) v).isChecked()) {

            score += 1;
```

```

        etScore.setText(String.valueOf(score));

    } else {

        score -= 1;

        etScore.setText(String.valueOf(score));

    }

    break;

case R.id.event_next:

    if (validate(score, etScore, "กรุณาประเมินค่าตัว") {

        if(score > Config.body_Max){

            etScore.setError("กรุณาตรวจสอบ เนื่องจากคะแนนที่
ได้มากกว่า 6 คะแนน");

        }

        else{

            Intent intentNext = new Intent(getBaseContext(),

                navigation.getNext(navigation.Body));

            saveData(Config.BODY, score);

            startActivity(intentNext);

            this.finish();

        }

    }
}

```

```
        break;

    case R.id.event_previous:

        Intent intentPrevious = new Intent(getBaseContext(),

            navigation.getPreviuse(navigation.Body));

        startActivity(intentPrevious);

        this.finish();

        break;

    case R.id.event_home:

        Intent intentHome = new Intent(getBaseContext(), MainMenu.class);

        startActivity(intentHome);

        this.finish();

        break;

    default:

        break;

    }

}

private void setCheck(boolean isCheck) {

    checkBox1.setChecked(isCheck);

    checkBox2.setChecked(isCheck);

}
```

```
private void setEnable(int isEnabled) {  
  
    layout.setVisibility(isEnabled);  
  
    checkBox1.setVisibility(isEnabled);  
  
    checkBox2.setVisibility(isEnabled);  
  
}  
  
@Override  
  
public void Initial() {  
  
    right = (ImageButton) findViewById(R.id.event_next);  
  
    home = (ImageButton) findViewById(R.id.event_home);  
  
    layout = (LinearLayout) findViewById(R.id.llcheckBoxBody);  
  
    etScore = (EditText) findViewById(R.id.etScoreBody);  
  
    checkBox1 = (CheckBox) findViewById(R.id.checkBoxBody_1);  
  
    checkBox2 = (CheckBox) findViewById(R.id.checkBoxBody_2);  
  
    button1 = (Button) findViewById(R.id.buttonBody_1);  
  
    button2 = (Button) findViewById(R.id.buttonBody_2);  
  
    button3 = (Button) findViewById(R.id.buttonBody_3);  
  
    button4 = (Button) findViewById(R.id.buttonBody_4);  
  
    this.image1 = (View) findViewById(R.id.imageViewBody1);  
  
    this.image2 = (View) findViewById(R.id.imageViewBody2);  
  
    this.image3 = (View) findViewById(R.id.imageViewBody3);  
  
}
```

```
        this.image4 = (View) findViewById(R.id.imageViewBody4);
    }

    @Override

    public void Listener() {

        this.image1.setOnClickListener(this);

        this.image2.setOnClickListener(this);

        this.image3.setOnClickListener(this);

        this.image4.setOnClickListener(this);

        checkBox1.setOnClickListener(this);

        checkBox2.setOnClickListener(this);

        button1.setOnClickListener(this);

        button2.setOnClickListener(this);

        button3.setOnClickListener(this);

        button4.setOnClickListener(this);

        home.setOnClickListener(this);

        right.setOnClickListener(this);

    }

    public void onBackPressed() {

        super.onBackPressed();

    }
```

```
}
```

ชื่อฟอร์ม HandAndWrist.java

```
package com.itsada.rula;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.LinearLayout;

import com.itsada.Business.Config;

import com.itsada.Business.IAcitivity;

public class HandAndWrist extends BaseActivity implements IAcitivity,

        OnClickListener {

    private Navigation navigation;

    private View image1;

    private View image2;
```

```
private View image3;

private LinearLayout layout;

private Button button1;

private Button button2;

private Button button3;

private Button buttonWristTwist_1;

private Button buttonWristTwist_2;

private CheckBox checkBox1;

private EditText etScoreHandAndWrist;

private int scoreHandAndWrist = -1;

private EditText etScoreWristTwist;

private int scoreWristTwist = -1;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.handandwrist);

    navigation = new Navigation();

    Initial();

    Listener();

}
```

```
@SuppressWarnings("NewApi")

@Override

public void onClick(View v) {

    etScoreHandAndWrist.setError(null);

    etScoreWristTwist.setError(null);

    switch (v.getId()) {

        case R.id.imageViewWrist1:

        case R.id.buttonWrist_1:

            scoreHandAndWrist = 1;

            setCheck(false);

            setEnable(0);

            etScoreHandAndWrist.setText(String.valueOf(scoreHandAndWrist));

            break;

        case R.id.imageViewWrist2:

        case R.id.buttonWrist_2:

            scoreHandAndWrist = 2;

            setCheck(false);

            setEnable(0);

            etScoreHandAndWrist.setText(String.valueOf(scoreHandAndWrist));

            break;
```

```
case R.id.imageViewWrist3:

case R.id.buttonWrist_3:

    scoreHandAndWrist = 3;

    setCheck(false);

    setEnable(0);

    etScoreHandAndWrist.setText(String.valueOf(scoreHandAndWrist));

    break;

case R.id.checkBoxWrist:

    if (((CheckBox) v).isChecked()) {

        scoreHandAndWrist += 1;

etScoreHandAndWrist.setText(String.valueOf(scoreHandAndWrist));

    } else {

        scoreHandAndWrist -= 1;

etScoreHandAndWrist.setText(String.valueOf(scoreHandAndWrist));

    }

    break;

case R.id.buttonWristTwist_1:

    scoreWristTwist = 1;

etScoreWristTwist.setText(String.valueOf(scoreWristTwist));
```

```

        break;

    case R.id.buttonWristTwist_2:

        scoreWristTwist = 2;

        etScoreWristTwist.setText(String.valueOf(scoreWristTwist));

        break;

    case R.id.event_next:

        if (validate(scoreHandAndWrist, etScoreHandAndWrist,

                    "กรุณาประเมินข้อมือ")

            && validate(scoreWristTwist, etScoreWristTwist,

                    "กรุณาประเมินการบิดของข้อมือ")) {

            Intent intentNext = new Intent(getBaseContext(),

navigation.getNext(navigation.HandAndWrist));

            saveData(Config.WRIST, scoreHandAndWrist);

            saveData(Config.WRIST_TWRIST, scoreWristTwist);

            startActivity(intentNext);

            this.finish();

        }

        break;

    case R.id.event_home:

```

```
        Intent intentHome = new Intent(getBaseContext(), MainMenu.class);

        startActivity(intentHome);

        finish();

        break;

    default:

        break;

    }

}

private void setCheck(boolean isCheck) {

    checkBox1.setChecked(isCheck);

}

private void setEnable(int isEnabled) {

    layout.setVisibility(isEnabled);

    checkBox1.setVisibility(isEnabled);

}

@Override

public void Initial() {

    home = (ImageButton) findViewById(R.id.event_home);

    right = (ImageButton) findViewById(R.id.event_next);

    etScoreHandAndWrist = (EditText) findViewById(R.id.etScoreWrist);
```

```
etScoreWristTwist = (EditText) findViewById(R.id.etScoreWristTwist);

layout = (LinearLayout) findViewById(R.id.llcheckBoxHand);

checkBox1 = (CheckBox) findViewById(R.id.checkBoxWrist);

this.image1 = (View) findViewById(R.id.imageViewWrist1);

this.image2 = (View) findViewById(R.id.imageViewWrist2);

this.image3 = (View) findViewById(R.id.imageViewWrist3);

this.button1 = (Button) findViewById(R.id.buttonWrist_1);

this.button2 = (Button) findViewById(R.id.buttonWrist_2);

this.button3 = (Button) findViewById(R.id.buttonWrist_3);

this.buttonWristTwist_1 = (Button) findViewById(R.id.buttonWristTwist_1);

this.buttonWristTwist_2 = (Button) findViewById(R.id.buttonWristTwist_2);

}

@Override

public void Listener() {

    this.image1.setOnClickListener(this);

    this.image2.setOnClickListener(this);

    this.image3.setOnClickListener(this);

    this.button1.setOnClickListener(this);

    this.button2.setOnClickListener(this);

    this.button3.setOnClickListener(this);
```

```

        this.buttonWristTwist_1.setOnClickListener(this);

        this.buttonWristTwist_2.setOnClickListener(this);

        checkBox1.setOnClickListener(this);

        right.setOnClickListener(this);

        home.setOnClickListener(this);

    }

    public void onBackPressed() {

        super.onBackPressed();

    }

}

```

ชื่อฟอร์ม **Information.java**

```

package com.itsada.rula;

import java.io.FileNotFoundException;

import android.content.Intent;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

import com.itsada.Business.Config;

```

```

import com.itsada.Business.IAcitivity;

public class Information extends BaseActivity implements IAcitivity,
        OnClickListener {
    private Navigation navigation;
    private TextView tvTestDate;
    private TextView tvTestPlace;
    private TextView tvTestName;
    private TextView tvError;
    private EditText etDay;
    private EditText etMonth;
    private EditText etYear;
    private EditText etTestName;
    private EditText etTestPlace;
    private Button buttonUploadPicture;
    private Button buttonOK;
    private Button buttonNext;
    private ImageView picture;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try {
            setContentView(R.layout.information);
            navigation = new Navigation();
            Initial();
            Listerner();
        } catch (Exception e) {
        }
    }
    @Override
    public void onClick(View v) {

```

```

switch (v.getId()) {
case R.id.buttonOK:
    if (checkValidate(etDay, "กรุณากรอก วันที่")
        && checkValidate(etMonth, "กรุณากรอก เดือน")
        && checkValidate(etYear, "กรุณากรอก ปี")
        && checkValidate(etTestName, "กรุณากรอก ชื่อผู้
ประเมิน")
        && checkValidate(etTestPlace, "กรุณากรอก สถานที่
ประเมิน")) {
        Intent intentOk = new Intent(getApplicationContext(),
            navigation.getNext(navigation.Information));

        if (etDay != null)
            saveData(Config.TEST_DAY,
etDay.getText().toString());
        if (etMonth != null)
            saveData(Config.TEST_MONTH,
etMonth.getText().toString());
        if (etYear != null)
            saveData(Config.TEST_YEAR,
etYear.getText().toString());
        if (etTestName != null)
            saveData(Config.TEST_NAME,
etTestName.getText().toString());
        if (etTestPlace != null)
            saveData(Config.TEST_PLACE, etTestPlace.getText()
                .toString());
        startActivity(intentOk);
        this.finish();
    } else {

```

```

        if (tvError != null)
            tvError.setText("กรุณากรอกข้อมูลให้ครบ");
    }
    break;
case R.id.buttonCross:
    Intent intentNext = new Intent(getBaseContext(),
        navigation.getNext(navigation.Information));
    startActivity(intentNext);
    this.finish();
    break;
case R.id.event_home:
    Intent intentHome = new Intent(getBaseContext(), MainMenu.class);
    startActivity(intentHome);
    this.finish();
    break;
case R.id.buttonUploadPicture:
    uploadImage();
    break;
}
}
private void uploadImage() {
    Intent intent = new Intent(Intent.ACTION_PICK,
        android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, 0);
}
@Override
public void Initial() {
    try {
        tvTestDate = SetTextView(tvTestDate, R.id.tvTestDate,
            Assets.testDate);
    }
}

```

```

        tvTestPlace = SetTextView(tvTestPlace, R.id.tvTestPlace,
                                Assets.testPlace);

        tvTestName = SetTextView(tvTestName, R.id.tvTestName,
                                Assets.testName);

        picture = (ImageView) findViewById(R.id.imageView1);
        tvError = (TextView) findViewById(R.id.tvError);
        etDay = (EditText) findViewById(R.id.etDay);
        etMonth = (EditText) findViewById(R.id.etMonth);
        etYear = (EditText) findViewById(R.id.etYear);
        etTestName = (EditText) findViewById(R.id.etTestName);
        etTestPlace = (EditText) findViewById(R.id.etTestPlace);
        buttonUploadPicture = (Button)
findViewById(R.id.buttonUploadPicture);

        buttonOK = (Button) findViewById(R.id.buttonOK);
        buttonNext = (Button) findViewById(R.id.buttonCross);

    } catch (Exception e) {
    }
}

@Override
public void Listener() {
    try {
        if (buttonOK != null)
            buttonOK.setOnClickListener(this);
        if (buttonUploadPicture != null)
            buttonUploadPicture.setOnClickListener(this);
        if (buttonUploadPicture != null)
            buttonNext.setOnClickListener(this);
    } catch (Exception e) {
    }
}

```

```

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        // TODO Auto-generated method stub
        super.onActivityResult(requestCode, resultCode, data);
        try {
            if (resultCode == RESULT_OK) {
                Uri targetUri = data.getData();
                saveData(Config.TEST_IMAGE_URI, targetUri.toString());
                // textTargetUri.setText(targetUri.toString());
                Bitmap bitmap;
                try {
                    bitmap =
BitmapFactory.decodeStream(getContentResolver()
                                .openInputStream(targetUri));
                    picture.setImageBitmap(bitmap);
                } catch (FileNotFoundException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
        } catch (Exception e) {
        }
    }

    public void onBackPressed() {
        super.onBackPressed();
    }
}

```

ชื่อฟอร์ม LekAndFoot.java

```
package com.itsada.rula;
import java.util.ArrayList;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Spinner;
import com.itsada.Business.Config;
import com.itsada.Business.IAcitivity;
public class LekAndFoot extends BaseActivity implements IAcitivity,
        OnClickListener, OnItemSelectedListener {
    private Navigation navigation;
    private Button buttonFoceArmAndHand;
    private Button buttonFoceNeckAndBodyAndFoot;
    private Button buttonCalculateRula;
    private Button buttonPriview;
    private Spinner spinner;
    private ArrayList<String> forceArmAndHand;
    private EditText etScoreForceArmAndHand;
    private EditText etArmAndHandOK;
    private EditText etNeckAndBodyAndFootOK;
    private int scoreForceArmAndHand = -1;
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.lek_foot);
    navigation = new Navigation();
    Initial();
    Listener();
}

private void setSpinner() {
    spinner = (Spinner) findViewById(R.id.spinnerForceArmAndHand);
    forceArmAndHand = new ArrayList<String>();
    forceArmAndHand.add("");
    forceArmAndHand.add("มีที่ช่วยรองขาและเท้า");
    forceArmAndHand.add("ไม่มีที่ช่วยรองขาและเท้า");
    ArrayAdapter<String> choice = new ArrayAdapter<String>(this,
        R.layout.spinner, forceArmAndHand);
    choice.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(choice);
}

@SuppressWarnings("NewApi")
@Override
public void onClick(View v) {
    etScoreForceArmAndHand.setError(null);
    switch (v.getId()) {
    case R.id.buttonFoceArmAndHand:
        Intent intentFoceArmAndHand = new Intent(getBaseContext(),
            navigation.getChild1(navigation.LekAndFoot));
        startActivity(intentFoceArmAndHand);
        this.finish();
        break;
    case R.id.buttonFoceNeckAndBodyAndFoot:

```

```

Intent intentFoceNeckAndBodyAndFoot = new
Intent(getBaseContext(),
        navigation.getChild2(navigation.LekAndFoot));
startActivity(intentFoceNeckAndBodyAndFoot);
this.finish();
break;
case R.id.buttonCalculateRula:
    if (validate(getData(Config.ARM), etScoreForceArmAndHand,
        "กรุณาประเมินค่า เหน่งขาและเท้า")
        && validate(getData(Config.FORCE_A),
etArmAndHandOK,
        "กรุณาประเมินการใช้แรงและ
        กล้ามเนื้อในส่วนแขนและขา")
        && validate(getData(Config.MUSCLE_A),
etArmAndHandOK,
        "กรุณาประเมินการใช้แรงและ
        กล้ามเนื้อในส่วนแขนและขา")
        && validate(getData(Config.FORCE_B),
etNeckAndBodyAndFootOK,
        "กรุณาประเมินการใช้แรงและ
        กล้ามเนื้อในส่วนคอ ลำตัว และขา")
        && validate(getData(Config.MUSCLE_B),
etNeckAndBodyAndFootOK,
        "กรุณาประเมินการใช้แรงและ
        กล้ามเนื้อในส่วนคอ ลำตัว และขา")) {

        Intent intentReprt = new Intent(getBaseContext(),
                navigation.getNext(navigation.LekAndFoot));
startActivity(intentReprt);
this.finish();

```

```

        }
        break;
    case R.id.buttonPriview:
        break;
    case R.id.event_next:
        Intent intentNext = new Intent(getBaseContext(),
            navigation.getNext(navigation.LekAndFoot));
        startActivity(intentNext);
        this.finish();
        break;
    case R.id.event_home:
        Intent intentHome = new Intent(getBaseContext(), MainMenu.class);
        startActivity(intentHome);
        this.finish();
        break;
    default:
        break;
    }
}
}

```

@Override

```

public void onItemClick(AdapterView<?> parent, View view, int position,
    long id) {
    etScoreForceArmAndHand.setText("");
    if (parent.getItemAtPosition(position).toString() == "มีที่ช่วยรองขาและเท้า") {
        scoreForceArmAndHand = 1;
        saveData(Config.ARM, scoreForceArmAndHand);
        etScoreForceArmAndHand
            .setText(String.valueOf(scoreForceArmAndHand));
    }
}

```

```

    } else if (parent.getItemAtPosition(position).toString() == "ไม่มีที่ช่วยรองขาและ
เท้า") {

        scoreForceArmAndHand = 2;
        saveData(Config.ARM, scoreForceArmAndHand);
        etScoreForceArmAndHand
            .setText(String.valueOf(scoreForceArmAndHand));
    } else {
        etScoreForceArmAndHand.setText("");
    }
    saveData("selectId", position);
}

@Override
public void onNothingSelected(AdapterView<?> parent) {
    // TODO Auto-generated method stub
}

@Override
public void Initial() {
    home = (ImageButton) findViewById(R.id.event_home);
    etScoreForceArmAndHand = (EditText)
findViewById(R.id.etScoreForceLekAndHand);
    etArmAndHandOK = (EditText) findViewById(R.id.etArmAndHandOK);
    etNeckAndBodyAndFootOK = (EditText)
findViewById(R.id.etNeckAndBodyAndFootOK);
    int arm = getData(Config.ARM);
    int forceA = getData(Config.FORCE_A);
    int muscleA = getData(Config.MUSCLE_A);
    int forceB = getData(Config.FORCE_B);
    int muscleB = getData(Config.MUSCLE_B);
    if (arm != -1) {
        etScoreForceArmAndHand.setText(String.valueOf(arm));
    }
}

```

```

    } else
        etScoreForceArmAndHand.setText("");
    setSpinner();
    int selectId = getData("selectId");
    if (selectId != -1)
        spinner.setSelection(selectId);
    if (forceA != -1 && muscleA != -1)
        etArmAndHandOK.setText("OK");
    else
        etArmAndHandOK.setText("");
    if (forceB != -1 && muscleB != -1)
        etNeckAndBodyAndFootOK.setText("OK");
    else
        etNeckAndBodyAndFootOK.setText("");
    this.buttonFoceArmAndHand = (Button)
findViewById(R.id.buttonFoceArmAndHand);
    this.buttonFoceNeckAndBodyAndFoot = (Button)
findViewById(R.id.buttonFoceNeckAndBodyAndFoot);
    this.buttonCalculateRula = (Button) findViewById(R.id.buttonCalculateRula);
    this.buttonPriview = (Button) findViewById(R.id.buttonPriview);
}
@Override
public void Listerner() {
    this.buttonFoceArmAndHand.setOnClickListener(this);
    this.buttonFoceNeckAndBodyAndFoot.setOnClickListener(this);
    this.buttonCalculateRula.setOnClickListener(this);
    this.buttonPriview.setOnClickListener(this);
    spinner.setOnItemClickListener(this);
    home.setOnClickListener(this);
}

```

```

        public void onBackPressed() {
            super.onBackPressed();
        }
    }
}

```

ข้อพร้อม LookupTableA.java

```

package com.itsada.rula;

import java.util.ArrayList;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

import com.itsada.Business.IAcitivity;

import com.itsada.Business.TableA;

public class LookupTableA extends BaseActivity implements IAcitivity,
        OnClickListener, OnItemSelectedListener {

    private TableA tableA;

    private Button buttonGetScoreA;

    private Button buttonMenu;

    private Spinner spinnerUpperArm;

    private Spinner spinnerLowerArm;

    private Spinner spinnerWrist;

    private Spinner spinnerWristTwist;

    private ArrayList<String> upperArms;

```

```

private ArrayList<String> lowerArms;
private ArrayList<String> wrists;
private ArrayList<String> twists;
private int upperAram = -1;
private int lowerArm = -1;
private int wrist = -1;
private int twist = -1;
private EditText etScore;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.table_a);
    Initial();
    Listerner();
}

private void setSpnnier() {
    spinnerUpperArm = (Spinner) findViewById(R.id.SpinnerUpperArm);
    spinnerUpperArm.setAdapter(null);
    upperArms = new ArrayList<String>();
    upperArms.add("");
    upperArms.add("1");
    upperArms.add("2");
    upperArms.add("3");
    upperArms.add("4");
    upperArms.add("5");
    upperArms.add("6");
    ArrayAdapter<String> upperArmsAdapter = new ArrayAdapter<String>(this,
        R.layout.spinner, upperArms);
    upperArmsAdapter
.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

```

```

spinnerUpperArm.setAdapter(upperArmsAdapter);
spinnerLowerArm = (Spinner) findViewById(R.id.SpinnerLowerArm);
spinnerLowerArm.setAdapter(null);
lowerArms = new ArrayList<String>();
lowerArms.add("");
lowerArms.add("1");
lowerArms.add("2");
lowerArms.add("3");
ArrayAdapter<String> lowerArmsAdapter = new ArrayAdapter<String>(this,
        R.layout.spinner, lowerArms);
lowerArmsAdapter
.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinnerLowerArm.setAdapter(lowerArmsAdapter);
spinnerWrist = (Spinner) findViewById(R.id.SpinnerWrist);
spinnerWrist.setAdapter(null);
wrists = new ArrayList<String>();
wrists.add("");
wrists.add("1");
wrists.add("2");
wrists.add("3");
wrists.add("4");
ArrayAdapter<String> wristsAdapter = new ArrayAdapter<String>(this,
        R.layout.spinner, wrists);
wristsAdapter
.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinnerWrist.setAdapter(wristsAdapter);
spinnerWristTwist = (Spinner) findViewById(R.id.SpinnerWristTwist);
spinnerWristTwist.setAdapter(null);
twists = new ArrayList<String>();
twists.add("");

```

```

twists.add("1");
twists.add("2");
ArrayAdapter<String> twistsAdapter = new ArrayAdapter<String>(this,
    R.layout.spinner, twists);
twistsAdapter
.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinnerWristTwist.setAdapter(twistsAdapter);
}
@SuppressLint("NewApi")
@Override
public void onClick(View v) {
    etScore.setError(null);
    switch (v.getId()) {
    case R.id.buttonMenu:
        Intent intentHome = new Intent(getBaseContext(), MainMenu.class);
        startActivity(intentHome);
        this.finish();
        break;
    case R.id.buttonGetScoreA:
        if (validate(upperAram, etScore, "กรุณากรอกข้อมูลให้ครบ")
            && validate(lowerArm, etScore, "กรุณากรอกข้อมูลให้
ครบ")
            && validate(wrist, etScore, "กรุณากรอกข้อมูลให้
ครบ")
            && validate(twist, etScore, "กรุณากรอกข้อมูลให้
ครบ")) {
            tableA = new TableA();
            int score = tableA.getScore(upperAram, lowerArm, wrist,
twist);
            if (score != -1) {

```

```

        etScore.setError(null);
        etScore.setText(String.valueOf(score));
        upperAram = -1;
        lowerArm = -1;
        wrist = -1;
        twist = -1;
        setSpnnier();
    }
}
}
}
}
@Override
public void onItemSelected(AdapterView<?> parent, View view, int position,
    long id) {
    switch (parent.getId()) {
        case R.id.SpinnerUpperArm:
            if (parent.getItemAtPosition(position).toString() != "") {
                upperAram =
Integer.parseInt(parent.getItemAtPosition(position)
                .toString());
                etScore.setError(null);
                etScore.setText("0");
            }
            break;
        case R.id.SpinnerLowerArm:
            if (parent.getItemAtPosition(position).toString() != "") {
                lowerArm =
Integer.parseInt(parent.getItemAtPosition(position)
                .toString());
                etScore.setError(null);

```

```

        etScore.setText("0");
    }
    break;
case R.id.SpinnerWrist:
    if (parent.getItemAtPosition(position).toString() != "") {
        wrist = Integer.parseInt(parent.getItemAtPosition(position)
            .toString());
        etScore.setError(null);
        etScore.setText("0");
    }
    break;
case R.id.SpinnerWristTwist:
    if (parent.getItemAtPosition(position).toString() != "") {
        twist = Integer.parseInt(parent.getItemAtPosition(position)
            .toString());
        etScore.setError(null);
        etScore.setText("0");
    }
    break;
}
}
@Override
public void onNothingSelected(AdapterView<?> parent) {
    // TODO Auto-generated method stub
}
@Override
public void Initial() {
    buttonGetScoreA = (Button) findViewById(R.id.buttonGetScoreA);
    buttonMenu = (Button) findViewById(R.id.buttonMenu);
    etScore = (EditText) findViewById(R.id.etScore);
}

```

```

        setSpnnier();
    }

    @Override
    public void Listerner() {
        spinnerUpperArm.setOnItemSelectedListener(this);
        spinnerLowerArm.setOnItemSelectedListener(this);
        spinnerWrist.setOnItemSelectedListener(this);
        spinnerWristTwist.setOnItemSelectedListener(this);
        buttonGetScoreA.setOnClickListener(this);
        buttonMenu.setOnClickListener(this);
    }

    public void onBackPressed() {
        super.onBackPressed();
    }
}

```

ชื่อฟอร์ม **LookUpTableB.java**

```

package com.itsada.rula;

import java.util.ArrayList;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import com.itsada.Business.IAcitivity;

```

```
import com.itsada.Business.TableB;

public class LookUpTableB extends BaseActivity implements IAcitivity,
        OnClickListener, OnItemSelectedListener {

    private TableB tableB;
    private Button buttonGetScoreB;
    private Button buttonMenu;
    private Spinner spinnerNeck;
    private Spinner spinnerBody;
    private Spinner spinnerLek;
    private ArrayList<String> necks;
    private ArrayList<String> bodies;
    private ArrayList<String> leks;
    private int neck = -1;
    private int body = -1;
    private int lek = -1;
    private EditText etScore;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.table_b);
        Initial();
        Listerner();
    }

    private void setSpnnier() {
        spinnerNeck = (Spinner) findViewById(R.id.SpinnerNeck);
        spinnerNeck.setAdapter(null);
        necks = new ArrayList<String>();
        necks.add("");
        necks.add("1");
        necks.add("2");
    }
}
```

```
        necks.add("3");
        necks.add("4");
        necks.add("5");
        necks.add("6");
        ArrayAdapter<String> neckAdapter = new ArrayAdapter<String>(this,
                R.layout.spinner, necks);
        neckAdapter
        .setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinnerNeck.setAdapter(neckAdapter);
        spinnerBody = (Spinner) findViewById(R.id.SpinnerBody);
        spinnerBody.setAdapter(null);
        bodies = new ArrayList<String>();
        bodies.add("");
        bodies.add("1");
        bodies.add("2");
        bodies.add("3");
        bodies.add("4");
        bodies.add("5");
        bodies.add("6");
        ArrayAdapter<String> bodyAdapter = new ArrayAdapter<String>(this,
                R.layout.spinner, bodies);
        bodyAdapter
        .setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinnerBody.setAdapter(bodyAdapter);
        spinnerLek = (Spinner) findViewById(R.id.SpinnerLek);
        spinnerLek.setAdapter(null);
        leks = new ArrayList<String>();
        leks.add("");
        leks.add("1");
        leks.add("2");
```

```

        ArrayAdapter<String> lekAdapter = new ArrayAdapter<String>(this,
            R.layout.spinner, leks);

        lekAdapter
            .setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinnerLek.setAdapter(lekAdapter);
    }

    @SuppressWarnings("NewApi")
    @Override
    public void onClick(View v) {
        etScore.setError(null);
        switch (v.getId()) {
            case R.id.buttonMenu:
                Intent intentHome = new Intent(getApplicationContext(), MainMenu.class);
                startActivity(intentHome);
                this.finish();
                break;
            case R.id.buttonGetScoreB:
                if (validate(neck, etScore, "กรุณากรอกข้อมูลให้ครบ")
                    && validate(body, etScore, "กรุณากรอกข้อมูลให้
ครบ")
                    && validate(lek, etScore, "กรุณากรอกข้อมูลให้ครบ"))
                {
                    tableB = new TableB();
                    int score = tableB.getScore(neck, body, lek);
                    if (score != -1) {
                        etScore.setError(null);
                        etScore.setText(String.valueOf(score));
                        neck = -1;
                        body = -1;
                        lek = -1;
                    }
                }
            }
    }

```

```

        setSpnnier();
    }
}
}
}
}
@Override
public void onItemSelected(AdapterView<?> parent, View view, int position,
    long id) {
    switch (parent.getId()) {
    case R.id.SpinnerNeck:
        if (parent.getItemAtPosition(position).toString() != ""){
            neck = Integer.parseInt(parent.getItemAtPosition(position)
                .toString());
            etScore.setError(null);
            etScore.setText("0");
        }
        break;
    case R.id.SpinnerBody:
        if (parent.getItemAtPosition(position).toString() != ""){
            body = Integer.parseInt(parent.getItemAtPosition(position)
                .toString());
            etScore.setError(null);
            etScore.setText("0");
        }
        break;
    case R.id.SpinnerLek:
        if (parent.getItemAtPosition(position).toString() != ""){
            lek = Integer.parseInt(parent.getItemAtPosition(position)
                .toString());
            etScore.setError(null);

```

```

        etScore.setText("0");
    }
    break;
}
}
@Override
public void onNothingSelected(AdapterView<?> parent) {
    // TODO Auto-generated method stub
}
@Override
public void Initial() {
    buttonGetScoreB = (Button) findViewById(R.id.buttonGetScoreB);
    buttonMenu = (Button) findViewById(R.id.buttonMenu);
    etScore = (EditText) findViewById(R.id.etScore);
    setSpnnier();
}
@Override
public void Listerner() {
    spinnerNeck.setOnItemSelectedListener(this);
    spinnerBody.setOnItemSelectedListener(this);
    spinnerLek.setOnItemSelectedListener(this);
    buttonGetScoreB.setOnClickListener(this);
    buttonMenu.setOnClickListener(this);
}
public void onBackPressed() {
    super.onBackPressed();
}
}
}
ข้อฟอรัม LookUpTableC.java
package com.itsada.rula;

```

```
import java.util.ArrayList;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import com.itsada.Business.IAcityivity;
import com.itsada.Business.TableC;
public class LookUpTableC extends BaseActivity implements IAcityivity,
        OnClickListener, OnItemSelectedListener {
    private TableC tableC;
    private Button buttonGetScoreC;
    private Button buttonMenu;
    private Spinner spinnerEight;
    private Spinner spinnerFitteen;
    private ArrayList<String> eights;
    private ArrayList<String> fitteens;
    private int eight = -1;
    private int fitteen = -1;
    private EditText etScore;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.table_c);
    }
}
```

```
        Initial();
        Listener();
    }
    private void setSpinner() {
        spinnerEight = (Spinner) findViewById(R.id.SpinnerEight);
        spinnerEight.setAdapter(null);
        eights = new ArrayList<String>();
        eights.add("");
        eights.add("1");
        eights.add("2");
        eights.add("3");
        eights.add("4");
        eights.add("5");
        eights.add("6");
        eights.add("7");
        eights.add("8");
        ArrayAdapter<String> eightAdapter = new ArrayAdapter<String>(this,
            R.layout.spinner, eights);
        eightAdapter

    .setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinnerEight.setAdapter(eightAdapter);
        spinnerFifteen = (Spinner) findViewById(R.id.SpinnerFifteen);
        spinnerFifteen.setAdapter(null);
        fitteens = new ArrayList<String>();
        fitteens.add("");
        fitteens.add("1");
        fitteens.add("2");
        fitteens.add("3");
        fitteens.add("4");
```

```

        fitteens.add("5");
        fitteens.add("6");
        fitteens.add("7");
        ArrayAdapter<String> fitteenAdapter = new ArrayAdapter<String>(this,
                R.layout.spinner, fitteens);
        fitteenAdapter
        .setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinnerFitteen.setAdapter(fitteenAdapter);
    }
    @SuppressWarnings("NewApi")
    @Override
    public void onClick(View v) {
        etScore.setError(null);
        switch (v.getId()) {
            case R.id.buttonMenu:
                Intent intentHome = new Intent(getApplicationContext(), MainMenu.class);
                startActivity(intentHome);
                this.finish();
                break;
            case R.id.buttonGetScoreC:
                if (validate(eight, etScore, "กรุณากรอกข้อมูลให้ครบ")
                    && validate(fitteen, etScore, "กรุณากรอกข้อมูลให้
ครบ")) {
                    tableC = new TableC();
                    int score = tableC.getScore(eight, fitteen);
                    if (score != -1) {
                        etScore.setError(null);
                        etScore.setText(String.valueOf(score));
                        eight = -1;
                        fitteen = -1;

```

```

        setSpnnier();
    }
}
}
}
@Override
public void onItemSelected(AdapterView<?> parent, View view, int position,
    long id) {
    switch (parent.getId()) {
    case R.id.SpinnerEight:
        if (parent.getItemAtPosition(position).toString() != "") {
            eight = Integer.parseInt(parent.getItemAtPosition(position)
                .toString());
            etScore.setError(null);
            etScore.setText("0");
        }
        break;
    case R.id.SpinnerFitteen:
        if (parent.getItemAtPosition(position).toString() != "") {
            fitteen = Integer.parseInt(parent.getItemAtPosition(position)
                .toString());
            etScore.setError(null);
            etScore.setText("0");
        }
        break;
    }
}
@Override
public void onNothingSelected(AdapterView<?> parent) {
    // TODO Auto-generated method stub

```

```

    }
    @Override
    public void Initial() {
        buttonGetScoreC = (Button) findViewById(R.id.buttonGetScoreC);
        buttonMenu = (Button) findViewById(R.id.buttonMenu);
        etScore = (EditText) findViewById(R.id.etScore);
        setSpinner();
    }
    @Override
    public void Listener() {
        spinnerEight.setOnItemClickListener(this);
        spinnerFifteen.setOnItemClickListener(this);
        buttonGetScoreC.setOnClickListener(this);
        buttonMenu.setOnClickListener(this);
    }
    public void onBackPressed() {
        super.onBackPressed();
    }
}

```

ชื่อไฟล์ **LowerArm.java**

```

package com.itsada.rula;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageButton;

```

```

import android.widget.LinearLayout;
import com.itsada.Business.Config;
import com.itsada.Business.IAcitivity;
@SuppressLint("NewApi")
public class LowerArm extends BaseActivity implements IAcitivity,
        OnClickListener {
    private Navigation navigation;
    private View image1;
    private View image2;
    private View image3;
    private LinearLayout layout;
    private CheckBox checkBox1;
    private CheckBox checkBox2;
    private Button button1;
    private Button button2;
    private Button button3;
    private EditText etScore;
    private int score = -1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.lowerarm);
        navigation = new Navigation();
        Initial();
        Listerner();
    }
    @Override
    public void onClick(View v) {
        etScore.setError(null);
        switch (v.getId()) {

```

```
case R.id.imageView1:
case R.id.buttonLowerarm_1:
    score = 1;
    setCheck(false);
    setEnable(0);
    etScore.setText(String.valueOf(score));
    break;
case R.id.imageView2:
case R.id.buttonLowerarm_2:
    score = 2;
    setCheck(false);
    setEnable(0);
    etScore.setText(String.valueOf(score));
    break;
case R.id.imageView3:
case R.id.buttonLowerarm_3:
    score = 2;
    setCheck(false);
    setEnable(0);
    etScore.setText(String.valueOf(score));
    break;
case R.id.checkBoxLowerarm_1:
    if (((CheckBox) v).isChecked()) {
        score += 1;
        etScore.setText(String.valueOf(score));
    } else {
        score -= 1;
        etScore.setText(String.valueOf(score));
    }
    break;
```

```

case R.id.checkBoxLowerarm_2:
    if (((CheckBox) v).isChecked()) {
        score += 1;
        etScore.setText(String.valueOf(score));
    } else {
        score -= 1;
        etScore.setText(String.valueOf(score));
    }
    break;
case R.id.event_next:
    if (validate(score, etScore,
        "กรุณาประเมินตำแหน่งแขนส่วนล่าง(ข้อศอกถึง
    ข้อมือ)") {
        if (score > 4) {
            etScore.setError("กรุณาตรวจสอบ เนื่องจากคะแนนที่
    ได้มากกว่า 4 คะแนน");
        } else {
            Intent intentNext = new Intent(getBaseContext(),

navigation.getNext(navigation.LowerArm));

            saveData(Config.LOWER_ARM, score);
            startActivity(intentNext);
            this.finish();
        }
    }
    break;
case R.id.event_home:
    Intent intentHome = new Intent(getBaseContext(), MainMenu.class);
    startActivity(intentHome);
    finish();

```

```
                break;
            default:
                break;
        }
    }

    private void setCheck(boolean isCheck) {
        checkBox1.setChecked(isCheck);
        checkBox2.setChecked(isCheck);
    }

    private void setEnable(int isEnabled) {
        layout.setVisibility(isEnabled);
        checkBox1.setVisibility(isEnabled);
        checkBox2.setVisibility(isEnabled);
    }

    @Override
    public void Initial() {
        right = (ImageButton) findViewById(R.id.event_next);
        home = (ImageButton) findViewById(R.id.event_home);
        etScore = (EditText) findViewById(R.id.etScoreLowerarm);
        layout = (LinearLayout) findViewById(R.id.llcheckBoxLowerarm);
        checkBox1 = (CheckBox) findViewById(R.id.checkBoxLowerarm_1);
        checkBox2 = (CheckBox) findViewById(R.id.checkBoxLowerarm_2);
        button1 = (Button) findViewById(R.id.buttonLowerarm_1);
        button2 = (Button) findViewById(R.id.buttonLowerarm_2);
        button3 = (Button) findViewById(R.id.buttonLowerarm_3);
        this.image1 = (View) findViewById(R.id.imageView1);
        this.image2 = (View) findViewById(R.id.imageView2);
        this.image3 = (View) findViewById(R.id.imageView3);
    }

    @Override
```

```

public void Listener() {
    this.image1.setOnClickListener(this);
    this.image2.setOnClickListener(this);
    this.image3.setOnClickListener(this);
    checkBox1.setOnClickListener(this);
    checkBox2.setOnClickListener(this);
    button1.setOnClickListener(this);
    button2.setOnClickListener(this);
    button3.setOnClickListener(this);
    right.setOnClickListener(this);
    home.setOnClickListener(this);
}

public void onBackPressed() {
    super.onBackPressed();
}
}

```

ข้อพร้อม MainMenu.java

```

package com.itsada.rula;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
public class MainMenu extends BaseActivity implements OnClickListener {
    private Button btStart;
    private Button btTableA;
    private Button btTableB;
    private Button btTableC;
    private Button btManual;
    private Button btAbout;
}

```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    btStart = SetButton(btStart, R.id.btStart);
    btTableA = SetButton(btTableA, R.id.btTableA);
    btTableB = SetButton(btTableB, R.id.btTableB);
    btTableC = SetButton(btTableC, R.id.btTableC);
    btManual = SetButton(btManual, R.id.btManual);
    btAbout = SetButton(btAbout, R.id.btAbout);
    btStart.setOnClickListener(this);
    btTableA.setOnClickListener(this);
    btTableB.setOnClickListener(this);
    btTableC.setOnClickListener(this);
    btManual.setOnClickListener(this);
    btAbout.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btStart:
            Intent intentStrat = new Intent(getBaseContext(), Information.class);
            startActivity(intentStrat);
            this.finish();
            break;
        case R.id.btTableA:
            Intent intentTableA = new Intent(getBaseContext(),
                LookupTableA.class);
            startActivity(intentTableA);
            this.finish();
    }
}
```

```
        break;
    case R.id.btTableB:
        Intent intentTableB = new Intent(getBaseContext(),
            LookUpTableB.class);
        startActivity(intentTableB);
        this.finish();
        break;
    case R.id.btTableC:
        Intent intentTableC = new Intent(getBaseContext(),
            LookUpTableC.class);
        startActivity(intentTableC);
        this.finish();
        break;
    case R.id.btManual:
        Intent intentManual = new Intent(getBaseContext(),
            Information.class);
        startActivity(intentManual);
        this.finish();
        break;
    case R.id.btAbout:
        Intent intentAbout = new Intent(getBaseContext(), Information.class);
        startActivity(intentAbout);
        this.finish();
        break;
    }
}

public void onBackPressed() {
    super.onBackPressed();
}
}
```

Navigation.java

```
package com.itsada.rula;
import java.util.ArrayList;
public class Navigation {
    public String Information = "Information";
    public String UpperArm = "UpperArm";
    public String LowerArm = "LowerArm";
    public String HandAndWrist = "HandAndWrist";
    // public String WristTwist = "WristTwist";
    public String Neck = "Neck";
    public String Body = "Body";
    public String LekAndFoot = "LekAndFoot";
    public String ArmAndHandTestForceAndMuscle =
"ArmAndHandTestForceAndMuscle";
    public String NeckAndBodyTestForceAndMuscle =
"NeckAndBodyTestForceAndMuscle";
    public String Report = "Report";
    public Navigation() {
        Initial();
    }
    public Navigation(String name, Class<?> current, Class<?> next,
        Class<?> previuse) {
        this.name = name;
        this.current = current;
        this.next = next;
        this.previuse = previuse;
    }
    private String name;
    private Class<?> current;
```

```

private Class<?> next;
private Class<?> previuse;
private Class<?> Child1;
private Class<?> Child2;
private ArrayList<Navigation> activity = new ArrayList<Navigation>();
public ArrayList<Navigation> Initial() {
    Navigation information = new Navigation(Information, Information.class,
        UpperArm.class, MainMenu.class);
    Navigation upperArm = new Navigation(UpperArm, UpperArm.class,
        LowerArm.class, Information.class);
    Navigation lowerArm = new Navigation(LowerArm, LowerArm.class,
        HandAndWrist.class, UpperArm.class);
    Navigation handAndWrist = new Navigation(HandAndWrist,
        HandAndWrist.class, Neck.class, LowerArm.class);
//    Navigation wristTwist = new Navigation(WristTwist, WristTwist.class,
//        Neck.class, HandAndWrist.class);
    Navigation neck = new Navigation(Neck, Neck.class, Body.class,
        HandAndWrist.class);
    Navigation body = new Navigation(Body, Body.class, LekAndFoot.class,
        Neck.class);
    Navigation lekAndFoot = new Navigation(LekAndFoot, LekAndFoot.class,
        Report.class, Body.class);
    lekAndFoot.Child1 = ArmAndHandTestForceAndMuscle.class;
    lekAndFoot.Child2 = NeckAndFootTestForceAndMuscle.class;
    Navigation armAndHandTestForceAndMuscle = new Navigation(
        ArmAndHandTestForceAndMuscle,
        ArmAndHandTestForceAndMuscle.class, null,
LekAndFoot.class);
    Navigation neckAndFootTestForceAndMuscle = new Navigation(
        NeckAndBodyTestForceAndMuscle,

```

```

        ArmAndHandTestForceAndMuscle.class, null,
LekAndFoot.class);
        Navigation report = new Navigation(
            Report,
            Report.class, null, LekAndFoot.class);
        activity.add(information);
        activity.add(upperArm);
        activity.add(lowerArm);
        activity.add(handAndWrist);
//        activity.add(wristTwist);
        activity.add(neck);
        activity.add(body);
        activity.add(lekAndFoot);
        activity.add(armAndHandTestForceAndMuscle);
        activity.add(neckAndFootTestForceAndMuscle);
        activity.add(report);
        return activity;
    }
    public Class<?> getCurrent(String name) {
        for (Navigation n : activity) {
            if (n.name == name)
                return n.current;
        }
        return null;
    }
    public Class<?> getNext(String name) {
        for (Navigation n : activity) {
            if (n.name == name)
                return n.next;
        }
    }

```

```
        return null;
    }

    public Class<?> getPreviuse(String name) {
        for (Navigation n : activity) {
            if (n.name == name)
                return n.previuse;
        }
        return null;
    }

    public Class<?> getChild1(String name) {
        for (Navigation n : activity) {
            if (n.name == name)
                return n.Child1;
        }
        return null;
    }

    public Class<?> getChild2(String name) {
        for (Navigation n : activity) {
            if (n.name == name)
                return n.Child2;
        }
        return null;
    }
}
```

Neck.java

```
package com.itsada.rula;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
```

```
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import com.itsada.Business.Config;
import com.itsada.Business.IAcitivity;
public class Neck extends BaseActivity implements IAcitivity, OnClickListener {
    private Navigation navigation;
    private View image1;
    private View image2;
    private View image3;
    private View image4;
    private LinearLayout layout;
    private CheckBox checkBox1;
    private CheckBox checkBox2;
    private Button button1;
    private Button button2;
    private Button button3;
    private Button button4;
    private EditText etScore;
    private int score = -1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.neck);
        navigation = new Navigation();
        Initial();
        Listerner();
    }
}
```

```
}  
  
@SuppressWarnings("NewApi")  
  
@Override  
public void onClick(View v) {  
    etScore.setError(null);  
    switch (v.getId()) {  
        case R.id.imageViewNeck1:  
        case R.id.buttonNeck_1:  
            score = 1;  
            setCheck(false);  
            setEnable(0);  
            etScore.setText(String.valueOf(score));  
            break;  
        case R.id.imageViewNeck2:  
        case R.id.buttonNeck_2:  
            score = 2;  
            setCheck(false);  
            setEnable(0);  
            etScore.setText(String.valueOf(score));  
            break;  
        case R.id.imageViewNeck3:  
        case R.id.buttonNeck_3:  
            score = 3;  
            setCheck(false);  
            setEnable(0);  
            etScore.setText(String.valueOf(score));  
            break;  
        case R.id.imageViewNeck4:  
        case R.id.buttonNeck_4:  
            score = 4;
```

```

        setCheck(false);
        setEnable(0);
        etScore.setText(String.valueOf(score));
        break;
    case R.id.checkBoxNeck_1:
        if (((CheckBox) v).isChecked()) {
            checkBox2.setEnabled(false);
            score += 1;
        } else {
            checkBox2.setEnabled(true);
            score -= 1;
        }
        etScore.setText(String.valueOf(score));
        break;
    case R.id.checkBoxNeck_2:
        if (((CheckBox) v).isChecked()) {
            checkBox2.setEnabled(true);
            score += 1;
        } else {
            checkBox2.setEnabled(true);
            score -= 1;
        }
        etScore.setText(String.valueOf(score));
        break;
    case R.id.event_next:
        if (validate(score, etScore, "กรุณาประเมินตำแหน่งคอ") ) {
            if (score > Config.neck_Max) {
                etScore.setError("กรุณาตรวจสอบ เนื่องจากคะแนนที่
                ได้มากกว่า 6 คะแนน");
            } else {

```

```
Intent intentNext = new Intent(getBaseContext(),

navigation.getNext(navigation.Neck));

        saveData(Config.NECK, score);
        startActivity(intentNext);
        this.finish();
    }
}
break;
case R.id.event_home:
    Intent intentHome = new Intent(getBaseContext(), MainMenu.class);
    startActivity(intentHome);
    finish();
    break;
default:
    break;
}

}

private void setCheck(boolean isCheck) {
    checkBox1.setChecked(isCheck);
    checkBox2.setChecked(isCheck);
}

private void setEnable(int isEnabled) {
    layout.setVisibility(isEnabled);
    checkBox1.setVisibility(isEnabled);
    checkBox2.setVisibility(isEnabled);
}

@Override
public void Initial() {
```

```
right = (ImageButton) findViewById(R.id.event_next);
home = (ImageButton) findViewById(R.id.event_home);
etScore = (EditText) findViewById(R.id.etScoreNeck);
layout = (LinearLayout) findViewById(R.id.llcheckBoxNeck);
checkBox1 = (CheckBox) findViewById(R.id.checkBoxNeck_1);
checkBox2 = (CheckBox) findViewById(R.id.checkBoxNeck_2);
button1 = (Button) findViewById(R.id.buttonNeck_1);
button2 = (Button) findViewById(R.id.buttonNeck_2);
button3 = (Button) findViewById(R.id.buttonNeck_3);
button4 = (Button) findViewById(R.id.buttonNeck_4);
this.image1 = (View) findViewById(R.id.imageViewNeck1);
this.image2 = (View) findViewById(R.id.imageViewNeck2);
this.image3 = (View) findViewById(R.id.imageViewNeck3);
this.image4 = (View) findViewById(R.id.imageViewNeck4);
}

@Override
public void Listener() {
    this.image1.setOnClickListener(this);
    this.image2.setOnClickListener(this);
    this.image3.setOnClickListener(this);
    this.image4.setOnClickListener(this);
    checkBox1.setOnClickListener(this);
    checkBox2.setOnClickListener(this);
    button1.setOnClickListener(this);
    button2.setOnClickListener(this);
    button3.setOnClickListener(this);
    button4.setOnClickListener(this);
    right.setOnClickListener(this);
    home.setOnClickListener(this);
}
```

```

        public void onBackPressed() {
            super.onBackPressed();
        }
    }
}

```

ข้อพร้อม NeckAndFootTestForceAndMuscle.java

```

package com.itsada.rula;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;
import com.itsada.Business.Config;
import com.itsada.Business.IAcityivity;

public class NeckAndFootTestForceAndMuscle extends BaseActivity implements
        IAcityivity, OnClickListener {
    private Navigation navigation;
    private Button buttonOK;
    private CheckBox checkBoxForce1;
    private CheckBox checkBoxForce2;
    private CheckBox checkBoxForce3;
    private CheckBox checkBoxForce4;
    private RadioButton checkBoxMuscle1;
    private RadioButton checkBoxMuscle2;
    private EditText etScoreForce;
    private EditText etScoreMuscle;
    private int scoreForce = -1;

```

```
private int scoreMuscle = -1;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.neck_body_foot_force_muscle);
    navigation = new Navigation();
    Initial();
    Listener();
}

@SuppressLint("NewApi")
@Override
public void onClick(View v) {
    etScoreForce.setError(null);
    etScoreMuscle.setError(null);
    switch (v.getId()) {
        case R.id.checkBoxForce1:
            if (((CheckBox) v).isChecked()) {
                checkBoxForce2.setChecked(false);
                checkBoxForce3.setChecked(false);
                checkBoxForce4.setChecked(false);
                scoreForce = 1;
                etScoreForce.setText(String.valueOf(scoreForce));
            }
            break;
        case R.id.checkBoxForce2:
            if (((CheckBox) v).isChecked()) {
                checkBoxForce1.setChecked(false);
                checkBoxForce3.setChecked(false);
                checkBoxForce4.setChecked(false);
                scoreForce = 2;
            }
            break;
    }
}
```

```
        etScoreForce.setText(String.valueOf(scoreForce));
    }
    break;
case R.id.checkBoxForce3:
    if (((CheckBox) v).isChecked()) {
        checkBoxForce1.setChecked(false);
        checkBoxForce2.setChecked(false);
        checkBoxForce4.setChecked(false);
        scoreForce = 3;
        etScoreForce.setText(String.valueOf(scoreForce));
    }
    break;
case R.id.checkBoxForce4:
    if (((CheckBox) v).isChecked()) {
        checkBoxForce1.setChecked(false);
        checkBoxForce2.setChecked(false);
        checkBoxForce3.setChecked(false);
        scoreForce = 4;
        etScoreForce.setText(String.valueOf(scoreForce));
    }
    break;
case R.id.checkBoxMuscle1:
    if (((RadioButton) v).isChecked()) {
        scoreMuscle = 0;
        etScoreMuscle.setText(String.valueOf(scoreMuscle));
    }
    break;
case R.id.checkBoxMuscle2:
    if (((RadioButton) v).isChecked()) {
        scoreMuscle = 1;
```

```

        etScoreMuscle.setText(String.valueOf(scoreMuscle));
    }
    break;
case R.id.ButtonOK:
    if (validate(scoreForce, etScoreForce, "กรุณาประเมินการใช้แรง")
        && validate(scoreMuscle, etScoreMuscle,
            "กรุณาประเมินการใช้กล้ามเนื้อ")) {
        if (scoreForce > 6) {
            etScoreForce
                .setError("กรุณาตรวจสอบ เนื่องจาก
คะแนนที่ได้มากกว่า6 คะแนน");
        } else if (scoreMuscle > 2) {
            etScoreForce
                .setError("กรุณาตรวจสอบ เนื่องจาก
คะแนนที่ได้มากกว่า2 คะแนน");
        }
        Intent intentOk = new Intent(
            getBaseContext(),
            navigation
                .getPrevious(navigation.NeckAndBodyTestForceAndMuscle));
        saveData(Config.FORCE_B, scoreForce);
        saveData(Config.MUSCLE_B, scoreMuscle);
        startActivity(intentOk);
        this.finish();
    }
    break;
default:
    break;
}
}

```

```

@Override
public void Initial() {
    etScoreForce = (EditText) findViewById(R.id.etScoreForce);
    etScoreMuscle = (EditText) findViewById(R.id.etScoreMuscle);
    checkBoxForce1 = (CheckBox) findViewById(R.id.checkBoxForce1);
    checkBoxForce2 = (CheckBox) findViewById(R.id.checkBoxForce2);
    checkBoxForce3 = (CheckBox) findViewById(R.id.checkBoxForce3);
    checkBoxForce4 = (CheckBox) findViewById(R.id.checkBoxForce4);
    checkBoxMuscle1 = (RadioButton) findViewById(R.id.checkBoxMuscle1);
    checkBoxMuscle2 = (RadioButton) findViewById(R.id.checkBoxMuscle2);
    this.buttonOK = (Button) findViewById(R.id.ButtonOK);
}

@Override
public void Listener() {
    this.buttonOK.setOnClickListener(this);
    checkBoxForce1.setOnClickListener(this);
    checkBoxForce2.setOnClickListener(this);
    checkBoxForce3.setOnClickListener(this);
    checkBoxForce4.setOnClickListener(this);
    checkBoxMuscle1.setOnClickListener(this);
    checkBoxMuscle2.setOnClickListener(this);
}

public void onBackPressed() {
    super.onBackPressed();
}
}

```

ข้อพ่อรวม Popup.java

```

package com.itsada.rula;
import android.app.AlertDialog;
import android.app.Dialog;

```

```

import android.app.DialogFragment;
import android.content.DialogInterface;
import android.os.Bundle;
public class Popup extends DialogFragment {
    private String measge;
    public AlertDialog.Builder builder;
    public Popup() {
        // TODO Auto-generated constructor stub
    }
    public Popup(String m) {
        measge = m;
    }
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setMessage(measge)
            .setPositiveButton("ตกลง", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    // FIRE ZE MISSILES!
                }
            });
        // Create the AlertDialog object and return it
        return builder.create();
    }
}

```

ข้อพ่อรวม Report.java

```

package com.itsada.rula;
import java.io.FileNotFoundException;
import android.annotation.SuppressLint;

```

```
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import com.itsada.Business.Config;
import com.itsada.Business.Rula;
public class Report extends BaseActivity implements OnClickListener {
    private Rula rula;
    private TextView tvDate;
    private TextView tvTestName;
    private TextView tvTestPlace;
    private ImageView piture;
    private EditText tvUpperArm;
    private EditText tvLowerArm;
    private EditText tvWrist;
    private EditText tvWristTwist;
    private EditText tvNeck;
    private EditText tvBody;
    private EditText tvLekAndFoot;
    private EditText tvTableA;
    private EditText tvMuscleA;
    private EditText tvForceA;
    private EditText tvleft;
```

```
private EditText tvTableB;
private EditText tvMuscleB;
private EditText tvForceB;
private EditText tvRight;
private EditText tvfinalScore;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.report);
    tvDate = (TextView) findViewById(R.id.textView4);
    tvTestName = (TextView) findViewById(R.id.textView9);
    tvTestPlace = (TextView) findViewById(R.id.textView6);
    pivture = (ImageView) findViewById(R.id.imageView1);
    String day = getStringData(Config.TEST_DAY);
    String month = getStringData(Config.TEST_MONTH);
    String year = getStringData(Config.TEST_YEAR);
    String testName = getStringData(Config.TEST_NAME);
    String testPlace = getStringData(Config.TEST_PLACE);
    String imageUri = getStringData(Config.TEST_IMAGE_URI);
    if (day != "" && month != "" && year != "") {
        tvDate.setText(getStringData(day + "/" + month + "/" + year));
    }
    if (testName != "") {
        tvTestName.setText(testName);
    }
    if (testPlace != "") {
        tvTestPlace.setText(testPlace);
    }
    if (imageUri != "") {
        Bitmap bitmap;
```

```
        try {  
            Uri uri = Uri.parse(imageUri);  
            bitmap = BitmapFactory.decodeStream(getContentResolver()  
                .openInputStream(uri));  
            pivture.setImageBitmap(bitmap);  
        } catch (FileNotFoundException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
    }  
  
    tvUpperArm = (EditText) findViewById(R.id.tvUpperArm);  
    tvLowerArm = (EditText) findViewById(R.id.tvLowerArm);  
    tvWrist = (EditText) findViewById(R.id.tvWrist);  
    tvWristTwist = (EditText) findViewById(R.id.tvWristTwist);  
    tvNeck = (EditText) findViewById(R.id.tvNeck);  
    tvBody = (EditText) findViewById(R.id.tvBody);  
    tvLekAndFoot = (EditText) findViewById(R.id.tvLekAndFoot);  
    tvTableA = (EditText) findViewById(R.id.tvTableA);  
    tvMuscleA = (EditText) findViewById(R.id.tvMuscleA);  
    tvForceA = (EditText) findViewById(R.id.tvForceA);  
    tvleft = (EditText) findViewById(R.id.tvLeft);  
    tvTableB = (EditText) findViewById(R.id.tvTableB);  
    tvMuscleB = (EditText) findViewById(R.id.tvMuscleB);  
    tvForceB = (EditText) findViewById(R.id.tvForceB);  
    tvRight = (EditText) findViewById(R.id.tvRight);  
    tvfinalScore = (EditText) findViewById(R.id.tvScore);  
    rula = new Rula();  
    rula.upperArm = getData(Config.UPPER_ARM);  
    rula.lowerArm = getData(Config.LOWER_ARM);  
    rula.handAndWrist = getData(Config.WRIST);
```

```

rula.wristTwist = getData(Config.WRIST_TWIST);
rula.neck = getData(Config.NECK);
rula.body = getData(Config.BODY);
rula.arm = getData(Config.ARM);
rula.forceA = getData(Config.FORCE_A);
rula.muscleA = getData(Config.MUSCLE_A);
rula.forceB = getData(Config.FORCE_B);
rula.muscleB = getData(Config.MUSCLE_B);
tvUpperArm.setText(String.valueOf(rula.upperArm));
tvLowerArm.setText(String.valueOf(rula.lowerArm));
tvWrist.setText(String.valueOf(rula.handAndWrist));
tvWristTwist.setText(String.valueOf(rula.wristTwist));
tvNeck.setText(String.valueOf(rula.neck));
tvBody.setText(String.valueOf(rula.body));
tvLekAndFoot.setText(String.valueOf(rula.arm));
tvTableA.setText(String.valueOf(rula.a.getScore(rula.upperArm,
        rula.lowerArm, rula.handAndWrist, rula.wristTwist)));
tvMuscleA.setText(String.valueOf(rula.muscleA));
tvForceA.setText(String.valueOf(rula.forceA));
tvleft.setText(String.valueOf(rula.left()));
int b = rula.b.getScore(rula.neck, rula.body, rula.arm);
tvTableB.setText(String.valueOf(b));
tvMuscleB.setText(String.valueOf(rula.muscleB));
tvForceB.setText(String.valueOf(rula.forceB));
tvRight.setText(String.valueOf(rula.right()));
int finalscore = rula.finalScore();
String text = null;
switch (finalscore) {
case 1:
case 2:

```

```

        text = "ผลการประเมินท่าทางดังกล่าวอยู่ในระดับที่1 เป็นท่าทางที่
ยอมรับได้";
        break;
    case 3:
    case 4:
        text = "ผลการประเมินท่าทางดังกล่าวอยู่ในระดับที่2 เป็นท่าทางที่ ควร
ตรวจสอบและอาจต้องมีการแก้ไข";
        break;
    case 5:
    case 6:
        text = "ผลการประเมินท่าทางดังกล่าวอยู่ในระดับที่3 เป็นท่าทางที่ ควร
ตรวจสอบแก้ไขโดยเร็ว";
        break;
    case 7:
    case 8:
        text = "ผลการประเมินท่าทางดังกล่าวอยู่ในระดับที่4 เป็นท่าทางที่ ควร
ตรวจสอบแก้ไขในทันที";
        break;
    }
    tvfinalScore.setText(String.valueOf(finalscore));
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage(text)
        .setPositiveButton("ตกลง", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                // FIRE ZE MISSILES!
                dialog.cancel();
            }
        });
    // Create the AlertDialog object and return it
    AlertDialog

```

```

alert = builder.create();
    alert.show();
    }
    @SuppressWarnings("NewApi")
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            }
        }
        public void onBackPressed() {
            super.onBackPressed();
        }
    }
}

```

ชื่อฟอร์ม Splash.java

```

package com.itsada.rula;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

import com.itsada.Business.AndroidEntity;
public class Splash extends AndroidEntity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.home_layout);
        clearData();
        Handler x = new Handler(); // Define and set up Handler
        x.postDelayed(new SplashHandler(), 2000);
    }
}

```

```

class SplashHandler implements Runnable {
    public void run() {
        startActivity(new Intent(getBaseContext(), MainMenu.class)); // yes
        Splash.this.finish(); // kill Splash screen
    }
}

```

```

}

```

ชื่อฟอร์ม UpperArm.java

```

package com.itsada.rula;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import com.itsada.Business.Config;
import com.itsada.Business.IAactivity;

public class UpperArm extends BaseActivity implements IAactivity,
    OnClickListener {
    private Navigation navigation;
    private View image1;
    private View image2;
    private View image3;
    private View image4;
    private View image5;
    private LinearLayout layout;

```

```
private CheckBox checkBox1;
private CheckBox checkBox2;
private CheckBox checkBox3;
private Button button1;
private Button button2;
private Button button3;
private Button button4;
private Button button5;
private EditText etScore;
private int score = -1;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    try {
        setContentView(R.layout.upperarm);
        navigation = new Navigation();
        Initial();
        Listener();
    } catch (Exception e) {
        // TODO: handle exception
    }
}

public void Initial() {
    image1 = (View) findViewById(R.id.imageView1);
    image2 = (View) findViewById(R.id.imageView2);
    image3 = (View) findViewById(R.id.imageView3);
    image4 = (View) findViewById(R.id.imageView4);
    image5 = (View) findViewById(R.id.imageView5);
    layout = (LinearLayout) findViewById(R.id.llcheckBoxUpperarm);
    checkBox1 = (CheckBox) findViewById(R.id.checkBoxUpperarm_1);
```

```
checkbox2 = (CheckBox) findViewById(R.id.checkboxUpperarm_2);
checkbox3 = (CheckBox) findViewById(R.id.checkboxUpperarm_3);
button1 = (Button) findViewById(R.id.buttonUpperarm_1);
button2 = (Button) findViewById(R.id.buttonUpperarm_2);
button3 = (Button) findViewById(R.id.buttonUpperarm_3);
button4 = (Button) findViewById(R.id.buttonUpperarm_4);
button5 = (Button) findViewById(R.id.buttonUpperarm_5);
home = (ImageButton) findViewById(R.id.event_home);
right = (ImageButton) findViewById(R.id.event_next);
etScore = (EditText) findViewById(R.id.etScoreUpperarm);
}
public void Listener() {
    try {
        if (image1 != null)
            image1.setOnClickListener(this);
        if (image2 != null)
            image2.setOnClickListener(this);
        if (image3 != null)
            image3.setOnClickListener(this);
        if (image4 != null)
            image4.setOnClickListener(this);
        if (image5 != null)
            image5.setOnClickListener(this);
        if (checkbox1 != null)
            checkbox1.setOnClickListener(this);
        if (checkbox2 != null)
            checkbox2.setOnClickListener(this);
        if (checkbox3 != null)
            checkbox3.setOnClickListener(this);
        if (button1 != null)
```

```

        button1.setOnClickListener(this);
    if (button2 != null)
        button2.setOnClickListener(this);
    if (button3 != null)
        button3.setOnClickListener(this);
    if (button4 != null)
        button4.setOnClickListener(this);
    if (button5 != null)
        button5.setOnClickListener(this);
    if (home != null)
        home.setOnClickListener(this);
    if (right != null)
        right.setOnClickListener(this);
    } catch (Exception e) {
    }
}
@SuppressLint("NewApi")
@Override
public void onClick(View v) {
    try {
        etScore.setError(null);
        switch (v.getId()) {
        case R.id.imageView1:
        case R.id.buttonUpperarm_1:
            score = 1;
            setCheck(false);
            setEnable(0);
            etScore.setText(String.valueOf(score));
            break;
        case R.id.imageView2:

```

```
case R.id.buttonUpperarm_2:
    score = 2;
    setCheck(false);
    setEnable(0);
    etScore.setText(String.valueOf(score));
    break;
case R.id.imageView3:
case R.id.buttonUpperarm_3:
    score = 2;
    setCheck(false);
    setEnable(0);
    etScore.setText(String.valueOf(score));
    break;
case R.id.imageView4:
case R.id.buttonUpperarm_4:
    score = 3;
    setCheck(false);
    setEnable(0);
    etScore.setText(String.valueOf(score));
    break;
case R.id.imageView5:
case R.id.buttonUpperarm_5:
    score = 4;
    setCheck(false);
    setEnable(0);
    etScore.setText(String.valueOf(score));
    break;
case R.id.checkBoxUpperarm_1:
    if (((CheckBox) v).isChecked() {
        score += 1;
```

```

        etScore.setText(String.valueOf(score));
    } else {
        score -= 1;
        etScore.setText(String.valueOf(score));
    }
    break;
case R.id.checkBoxUpperarm_2:
    if (((CheckBox) v).isChecked()) {
        score += 1;
        etScore.setText(String.valueOf(score));
    } else {
        score -= 1;
        etScore.setText(String.valueOf(score));
    }
    break;
case R.id.checkBoxUpperarm_3:
    if (((CheckBox) v).isChecked()) {
        score -= 1;
        etScore.setText(String.valueOf(score));
    } else {
        score += 1;
        etScore.setText(String.valueOf(score));
    }
    break;
case R.id.event_next:
    if (validate(score, etScore,
        "กรุณาประเมินตำแหน่งแขนส่วนบน(หัวไหล่
ถึงข้อศอก)") {
        if (score > Config.upperArm_Max) {

```

```

        etScore.setError("กรุณาตรวจสอบ เนื่องจาก
คะแนนที่ได้มากกว่า 6 คะแนน");
    } else {
        Intent intentNext = new
Intent(getBaseContext(),

navigation.getNext(navigation.UpperArm));

        saveData(Config.UPPER_ARM, score);
        startActivity(intentNext);
        finish();
    }
}
break;
case R.id.event_home:
    Intent intentHome = new Intent(getBaseContext(),
MainMenu.class);

    startActivity(intentHome);
    finish();
    break;
default:
    break;
}
} catch (Exception e) {
    // TODO: handle exception
}
}

private void setCheck(boolean isCheck) {
    checkBox1.setChecked(isCheck);
    checkBox2.setChecked(isCheck);
    checkBox3.setChecked(isCheck);
}

```

```
    }  
    private void setEnable(int isEnabled) {  
        layout.setVisibility(isEnabled);  
        checkBox1.setVisibility(isEnabled);  
        checkBox2.setVisibility(isEnabled);  
        checkBox3.setVisibility(isEnabled);  
    }  
    public void onBackPressed() {  
        super.onBackPressed();  
    }  
}
```

ประวัติผู้วิจัย

ชื่อ-สกุล	ว่าที่ร้อยตรีอนันต์ชัย อุ่คล้าย
ที่อยู่	243 ถ.เกษตรสิน ต.ลำพญา อ.เมืองนครปฐม จ.นครปฐม 73000
ประวัติการศึกษา	
พ.ศ. 2548	สำเร็จการศึกษาระดับมัธยมศึกษาโรงเรียนบอสโกพิทักษ์ จ.นครปฐม
พ.ศ. 2554	สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมอุตสาหกรรม) คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร พระราชวังสนามจันทร์ จนครปฐม
พ.ศ.2555	ศึกษาต่อระดับปริญญาวิศวกรรมศาสตรมหาบัณฑิต (การจัดการงานวิศวกรรม) คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร พระราชวังสนามจันทร์ จนครปฐม