



การพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว

The Report Program Development for Simulation Training

Assault Rifles

นางสาวเกศิณี อรรคชาติ

โครงงานสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย - ญี่ปุ่น

พ.ศ. 2556

การพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว
The Report Program Development for Simulation Training
Assault Rifles

นางสาวเกศินี อรรคชาติลี

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย – ญี่ปุ่น
พ.ศ. 2556

คณะกรรมการสอบ

.....ประธานกรรมการสอบ

(อาจารย์ปราณีสา อิศรเสนา)

.....กรรมการสอบและอาจารย์ที่ปรึกษา

(ผศ. ตรีรัตน์ เมตต์การุณจิต)

.....กรรมการ

(อาจารย์อภิชญา นิ่มคุ้มภัย)

.....ประธานสหกิจศึกษาคณะเทคโนโลยีสารสนเทศ

(อาจารย์ยอดศักดิ์ เสือสมิง)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย – ญี่ปุ่น

จากการที่ได้สหกิจศึกษาที่บริษัท อินโนวา ซอร์ฟแวร์ จำกัด ได้รับมอบหมายให้พัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว ในการดำเนินงานนั้น มีการปรับรูปแบบหน้าจอการแสดงผลต่อผู้ใช้งาน และ ที่ได้มีการดึงข้อมูลจากระบบฐานข้อมูลมาแสดงในรายงาน และนำข้อมูลทำได้มาสรุปผล ผลที่ได้คือโปรแกรมช่วยอำนวยความสะดวกในการวิเคราะห์ผลการฝึก ช่วยให้ผู้ใช้งานนำรายงานที่ได้มาประเมินผล วิเคราะห์การฝึกของตนเอง และนำไปพัฒนาการฝึกของตนในครั้งหน้า ประโยชน์ที่ได้รับจากการปฏิบัติสหกิจได้แก่ ได้เรียนรู้การพัฒนาโปรแกรมภาษาซีชาร์ปด้วยโปรแกรมไมโครซอฟต์วิชวลสตูดิโอ 2010



Project's name	The Report Program Development for Simulation Training
Writer	Ms. Kesinee Augkahatsee
Faculty	Faculty of Information Technology, Information Technology Program
Faculty Advisor	Mr. Trirat Metkarunchit
Job Supervisor	Mr. Parinya Dechwise
Company's name	Innova Software Co., Ltd.
Business Type	Development Software Program for Company

Summary

As to I am a Trainee at Innova Software Co., Ltd. In Department of Development Division. I have assigned to develop report program of Simulation Training for Assaults Rifles. In execute planning is going to Adjust User Interface and in this Program we get data from database to show in Report and summarize data. At last Program is helpful a user in analysis method and help a trainee get them report to evaluate their training to be a hard Train in next time. A benefit from this Project, I have learned to use another Computer Language (C Sharp) by Microsoft Visual Studio 2010 and another Program that I never use before.

กิตติกรรมประกาศ

การปฏิบัติงานสหกิจศึกษาที่บริษัทอินโนวา ซอฟต์แวร์ จำกัด สำเร็จลุล่วงได้ด้วยความกรุณาจาก
ผู้ช่วยศาสตราจารย์ ตรีรัตน์ เมตต์การุณจิต อาจารย์ที่ปรึกษา นายปริญญา เฉวิชเศษ และนายทวัฒน์
ไชยศรีที่ปรึกษาโครงการสหกิจศึกษา ที่ได้ให้คำแนะนำ แนวคิด ตลอดจนแก้ไขข้อบกพร่องต่าง ๆ
มาโดยตลอด จนโครงการเล่มนี้เสร็จสมบูรณ์ ผู้ศึกษาจึงขอกราบขอบพระคุณเป็นอย่างสูง
ขอบคุณพนักงานแผนก Development Division ทุกคน ที่ช่วยสละเวลาให้คำแนะนำดีๆ
เกี่ยวกับเทคนิคการทำงานเมื่อเกิดปัญหาขึ้น

เกศินี อรรกษาสลี

ผู้จัดทำรายงาน

TNI

สารบัญ

หน้า

บทสรุป

ก

Summary

ข

กิตติกรรมประกาศ

ค

สารบัญ

ง

สารบัญตาราง

ช

สารบัญรูปประกอบ

ซ

บทที่

1. บทนำ

1

1.1 ชื่อและที่ตั้งของสถานประกอบการ

1

1.2 ลักษณะธุรกิจของสถานประกอบการ และการให้บริการหลักขององค์กร

2

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร

3

1.4 ตำแหน่งหน้าที่และงานที่ได้รับมอบหมาย

4

1.5 พนักงานที่ปรึกษาและตำแหน่งของพนักงานที่ปรึกษา

4

1.6 ระยะเวลาที่ปฏิบัติงาน

4

1.7 วัตถุประสงค์ของการปฏิบัติงานที่ได้รับมอบหมายหรือ

โครงการที่ได้รับมอบหมายให้ปฏิบัติงานสหกิจศึกษา

5

1.8 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

6

2. ทฤษฎีและเทคโนโลยีที่ใช้ปฏิบัติงาน

6

2.1. ซอร์ฟแวร์ที่ใช้ในการปฏิบัติงาน

2.1.1 ภาษาซีชาร์ป (C Sharp)

6

2.1.1.1 ภาษาซีชาร์ปคืออะไร

6

2.1.1.2 รูปแบบไฟล์ ASPX	7
2.1.2 คอตเน็ตเฟรมเวิร์ก (.NET Framework)	7
2.1.3 วิชาเว็บสีกคอตเน็ต (VB.NET)	8
2.1.4 โพสท์เกรสคิวแอล (PostgreSQL)	9
2.1.5 ภาษาเอสคิวแอล (Standard Query Language: SQL)	10
2.1.6 คริสตัลรีพอร์ต (Crystal Report)	10
2.2 ข้อมูลจำเพาะของอาวุธปืน	13
2.2.1 ปืนเล็กยาว	13
2.2.2 คุณสมบัติ	13
2.2.3 ข้อมูลทางเทคนิค	14
3.การพัฒนาโปรแกรมรายงานผลการฝึกยิง	15
3.1 แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน	15
3.1.1 แผนงานปฏิบัติงาน	
3.1.2 รายละเอียดงานที่นักศึกษาปฏิบัติงานในงานสหกิจศึกษา หรือรายละเอียดโครงการที่ได้รับมอบหมาย	16
3.1.2.1 ส่วนติดต่อกับผู้ใช้	
3.1.2.2 ผลรายงาน	
3.1.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโครงการ	17
3.1.3.1 ศึกษาเกี่ยวกับ VB.NET และ C#	17
3.1.3.2 ศึกษาเกี่ยวกับอาวุธปืนประเภทปืนเล็กยาว	17
3.1.3.3 ออกแบบโปรโตไทป์และER-Diagram	17
3.1.3.4 ออกแบบระบบฐานข้อมูล (database interface: DLL)	17
3.1.3.5 เขียนโปรแกรม (System Develop (Coding))	18
3.1.3.5 ทดสอบระบบ (System Testing)	18

3.1.3.6 ทำคู่มือการใช้งาน (User manual preparation)	18
3.1.3.7 Installation, Training and Support	18
3.2 การพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว และรายละเอียดสหกิจศึกษา	19
3.2.1 ช่วยงานระบบ Cloud TA	19
3.2.2 แก้ไขระบบ POInstep	19
3.2.3 ตัดรูปภาพประกอบในระบบ CloudTA	20
3.2.4 ทำแบบฝึกหัดการใช้ภาษาซีชาร์ป	21
3.2.5 ขั้นตอนการวางแผนพัฒนาระบบรายงาน	22
3.2.6 ขั้นตอนการเก็บความต้องการของรายงาน	22
3.2.7 การออกแบบ	23
3.2.7.1 การออกแบบตัวต้นแบบของโปรแกรม	23
3.2.7.2 การออกแบบความสัมพันธ์ระหว่างข้อมูล (ER-Diagram)	26
3.2.8 ขั้นตอนการสร้างโปรแกรม	27
3.2.8.1 ส่วนเขียนคำสั่ง	27
3.2.8.2 ส่วนติดต่อกับผู้ใช้	29
3.2.9 การทำรายงานการฝึก	30
3.2.9.1 การวางโครงสร้างรายงาน	30
4.ผลการดำเนินงาน การวิเคราะห์	31
4.1 ขั้นตอนและผลดำเนินงาน	31
4.1.1 เมนู (Menu)	31
4.1.2 หน้าจอแสดงผลการค้นห	32
4.1.3 รายงานการฝึก	32
4.2 ผลการวิเคราะห์ข้อมูล	34

4.3 วิเคราะห์และวิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์ และ จุดมุ่งหมายในการปฏิบัติงานหรือการจัดทำโครงการ	34
--	----

5. บทสรุปและข้อเสนอแนะ	36
5.1 สรุปผลการดำเนินงาน	36
5.2 แนวทางการแก้ไขปัญหา	37
5.3 ข้อเสนอแนะจากการฝึกงาน	37

เอกสารอ้างอิง	38
---------------	----

ภาคผนวก	40
---------	----

ประวัติผู้วิจัย	80
-----------------	----

TNI

สถาบันเทคโนโลยีไทย-ญี่ปุ่น
THAI - JAPAN INSTITUTE OF TECHNOLOGY

สารบัญตาราง

ตารางที่		หน้า
2.1	ตารางแสดงชนิดของข้อมูลแบบ Character 1	1
2.2	ตารางแสดงชนิดของข้อมูลแบบ Binary 1	1
2.3	ตารางแสดงชนิดของข้อมูลแบบ Date 1	1
2.4	ตารางแสดงชนิดของข้อมูลแบบ Logical 1	2
2.5	ตารางแสดงชนิดของข้อมูลแบบ Numeric 1	2
2.6	ตารางแสดงข้อมูลทางเทคนิคของอาวุธปืนเล็กยาว 1	4
3.1	ตารางแผนการปฏิบัติงานระยะเวลา16สัปดาห์หรือประมาณ4เดือน	15

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

สารบัญรูปภาพ

รูปที่	หน้า
1.1 แผนที่สถานที่ตั้งของ บริษัท อินโนวา ซอฟต์แวร์ จำกัด	1
1.2 โครงสร้างองค์กรของ บริษัท อินโนวา ซอฟต์แวร์ จำกัด	3
2.1 ส่วนของการทำงานส่วนหนึ่งในโปรเจกของการพัฒนาโปรแกรมรายงาน ระบบจำลองการฝึกยิงอาวุธปืนเล็กยาวโดยใช้โปรแกรม Microsoft Visual Studio 2010 Professional	8
2.2 ส่วนของการทำงานส่วนหนึ่งในโปรเจกของการพัฒนาโปรแกรมรายงาน ระบบจำลองการฝึกยิงอาวุธปืนเล็กยาวโดยใช้โปรแกรม โปรแกรม PostgreSQL	9
2.3 ส่วนของการทำงานส่วนหนึ่งในโปรเจกของการพัฒนาโปรแกรมรายงาน ระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว โดยใช้โปรแกรม Microsoft Visual Studio 2010 Professional	10
2.4 ปืนเล็กยาว 13	
3.1 ตัวอย่างการนำเข้าข้อมูลระบบ POInstep19	
3.2 ส่วนของการทำงานส่วนหนึ่งในการคอปรูปในระบบ CloudTA โดยใช้โปรแกรม PhotoScape	20
3.3 หน้าจอฝั่งเซิร์ฟเวอร์โปรแกรมTicTacToe	21
3.4 หน้าจอของฝั่งผู้เล่นโปรแกรมTicTacToe	21
3.5 แผนการพัฒนาระบบ	22
3.6 ดันแบบโปรแกรม	23
3.7 ความสัมพันธ์ของตารางข้อมูล	26
3.8 การวางโครงสร้างรูปแบบรายงาน โดยใช้โปรแกรม Microsoft Visual Studio 2010	30
4.1 หน้าจอเลือกรายงานการฝึก	31

สารบัญรูปภาพ(ต่อ)

รูปที่	หน้า
4.2 หน้าจอแสดงผลการค้นหา	32
4.3 การแสดงผลรายงานการฝึก	32
4.4 การแสดงผลรายงานการฝึกส่วนกราฟ 1	33
4.5 การแสดงผลรายงานการฝึกส่วนกราฟ 2	33
4.4 การแสดงผลรายงานการฝึกส่วนกราฟ 3	34
1 ก หน้าจอการติดตั้ง Visual Studio 2010	42
2 ก ตอบรับข้อตกลงการลงโปรแกรม Visual Studio 2010	42
3 ก รูปแบบการลงโปรแกรม Visual Studio 2010	43
4 ก รอดติดตั้ง Plugin crystal report	43
5 ก เลือกภาษา	44
6 ก กำลังติดตั้ง	44
7 ก ตอบรับข้อตกลงการลงโปรแกรม crystal report	44
8 ก รอการติดตั้งโปรแกรม crystal report	45
9 ก เลือกที่ติดตั้งโปรแกรม postgresQL	45
10 ก เลือกที่เก็บข้อมูล	46
11 ก ตั้งรหัสผ่านการเข้าดาต้าเบส	46
12 ก ตั้ง port	47
13 ก เลือก Default locale	47
14 ก หน้า Connect	48
15 ก เข้ารหัสผ่าน	48
16 ก สร้างดาต้าเบสใหม่	48
17 ก ติดตั้ง ODBC	49
18 ก เลือก Data Source ใหม่	49
1 ข หน้าจอ Interface	51

สารบัญรูปภาพ(ต่อ)

รูปที่	หน้า
2 ข หน้าจอการทำรายงาน	51
3 ข ส่วนรายงานส่วน Header	52
4 ข ส่วนการทำ Formula สำหรับรายงาน	52



บทที่ 1

บทนำ

1.1 ชื่อและที่ตั้งของสถานประกอบการ

ชื่อสถานประกอบการ : บริษัท อินโนวา ซอฟต์แวร์ จำกัด (Innova Software Co., Ltd.)

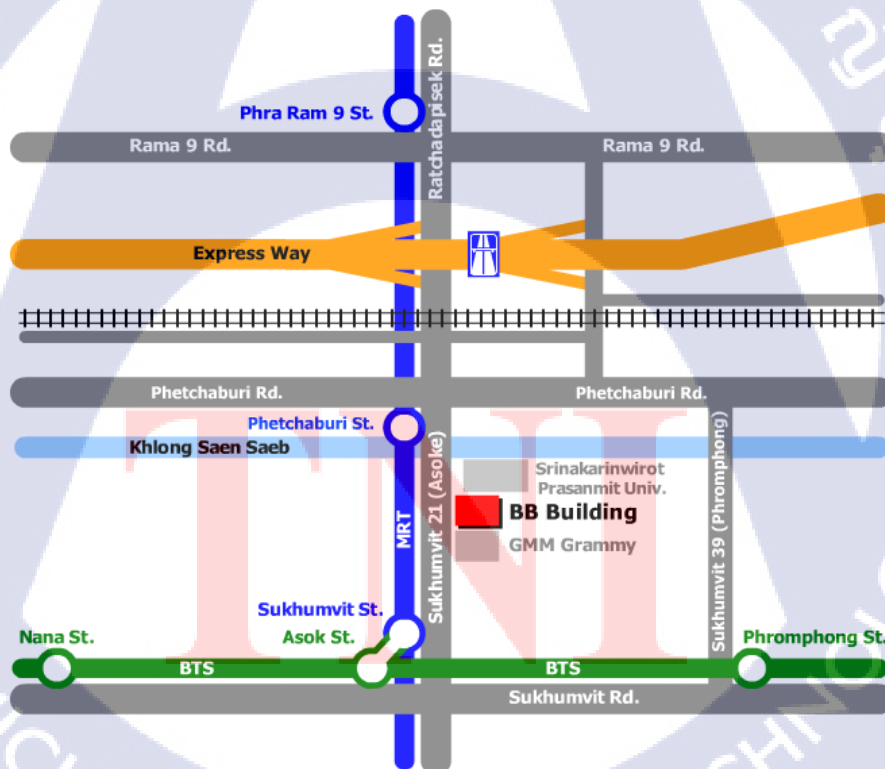
ที่ตั้งสถานประกอบการ : 54 อาคาร บีบี ชั้น 13 ห้อง 1301 ถนนสุขุมวิท 21 (อโศก)

แขวงคลองเตยเหนือ เขตวัฒนา กรุงเทพมหานคร 10110

โทรศัพท์ : 0-2664-3069

โทรสาร : 0-2264-2295

อีเมล : info@innova.co.th



รูปที่ 1.1 แผนที่สถานที่ตั้งของ บริษัท อินโนวา ซอฟต์แวร์ จำกัด

1.2 ลักษณะธุรกิจของสถานประกอบการ และการให้บริการหลักขององค์กร

บริษัท อินโนวา ซอฟต์แวร์ จำกัด เป็นบริษัทออกแบบระบบ โดยเป็นการร่วมทุนนักลงทุนไทย และนักลงทุนญี่ปุ่น ธุรกิจหลักของ บริษัท คือการพัฒนา ซอฟต์แวร์ โปรแกรมประยุกต์ทางธุรกิจ การใช้ซอฟต์แวร์ แก้ปัญหาขององค์กร และการให้บริการด้านไอทีอย่างมืออาชีพ เพื่อตอบสนองความต้องการของลูกค้า บริการของ บริษัท ครอบคลุมทั่วพื้นที่ธุรกิจหลากหลายอุตสาหกรรม เช่น การผลิต การขนส่ง การสื่อสารโทรคมนาคม เป็นต้น

นับตั้งแต่ก่อตั้ง ในปี 1999 บริษัทมีลูกค้าหลักอยู่ในภาครัฐ ภาคเอกชนในประเทศและตลาดต่างประเทศ ต่อมาในปี 2005 ได้รับการส่งเสริมจากคณะกรรมการส่งเสริมการลงทุนแห่งประเทศไทย (บีโอไอ) และกลายเป็นสมาชิกของ Oracle พันธมิตรเครือข่าย (Oracle Partner Network) บริษัทให้บริการให้คำปรึกษาและการดำเนินงานของ Oracle E-Business ซึ่งเป็นที่น่าเชื่อถือที่สุดระดับโลกโซลูชัน ERP ให้กับลูกค้าหลาย

บริษัทผ่านการรับรองและได้รับการจดทะเบียนเป็น บริษัทในการสื่อสารโทรคมนาคม และภาคคอมพิวเตอร์ที่กระทรวงการคลัง การลงทะเบียนนี้พิสูจน์ให้เห็นว่าบริษัทมีคุณสมบัติในการให้คำแนะนำ และบริการให้คำปรึกษาด้านไอทีให้กับหน่วยงานของรัฐบาลไทยที่ปราศจากเงื่อนไขใดๆ

ก่อตั้ง : เดือนมีนาคมปี 1999

ทุนชำระแล้ว : 20,000,000 บาท

ผู้ถือหุ้นหลัก : บริษัทซิสเต็มดีไซน์ จำกัด (ญี่ปุ่น) 40%

คณะกรรมการ : ดร.วิวัฒน์ วงศ์วราวิทย์ (กรรมการ)

คุณครรชิต นิงสานนท์ (กรรมการผู้จัดการ)

จำนวนพนักงาน : 30 คน (สิงหาคม 2556)

ผลิตภัณฑ์ : ระบบรักษาความปลอดภัยไบโอเมตริกซ์ (Biometric Security Solution)

ระบบบันทึกภาพ (Digital Video Recording System)

ระบบจัดการที่จอดรถรถยนต์ (Car Parking Management System)

ระบบควบคุมการเข้า-ออกและเวลา (Access Control & Time Attendance)

ระบบบริหารทรัพยากรมนุษย์และเงินเดือน (HR Management & Payroll System)

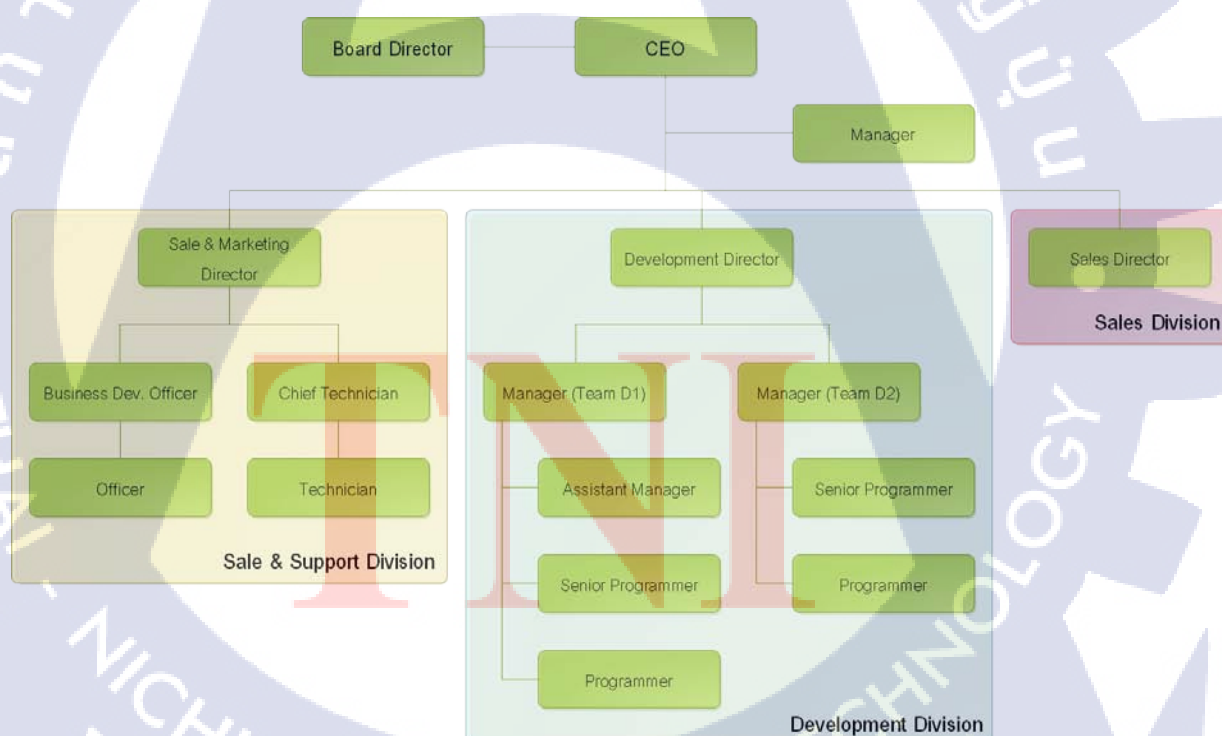
งานบริการลูกค้า : ให้คำปรึกษาด้านเทคโนโลยีสารสนเทศ (Consulting)

พัฒนาระบบ (Development)

ดำเนินการ (Implementation)

สนับสนุนและบำรุงรักษา (Support & Maintenance)

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร



รูปที่ 1.2 โครงสร้างองค์กรของ บริษัท อินโนวา ซอฟต์แวร์ จำกัด

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่งที่ได้รับมอบหมายในครั้งนี้คือนักพัฒนาซอฟต์แวร์โดยมีขอบเขตและหน้าที่ดังนี้
ออกแบบและพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว โดยเริ่มต้นจาก
ขั้นตอนพื้นฐานคือ เก็บข้อมูลความต้องการของผู้ใช้ ออกแบบฐานข้อมูล ออกแบบโครงสร้างส่วน
ต่อประสานกับผู้ใช้ (graphics user interface :GUI)และ การเขียนซอฟต์แวร์ในส่วนการรายงานผล

1.5 พนักงานที่ปรึกษา และตำแหน่งของพนักงานที่ปรึกษา

พนักงานที่ปรึกษา	: คุณปริญญา เดชวิเศษ
ตำแหน่ง	: Manager
แผนก	: Development Division
พนักงานที่ปรึกษา	: คุณวัฒน์ ไชยศรี
ตำแหน่ง	: Manager (Team D2)
แผนก	: Development Division

1.6 ระยะเวลาที่ปฏิบัติงาน

ปฏิบัติงานสหกิจศึกษาตั้งแต่วันที่ 17 มิถุนายน พ.ศ.2556 ถึงวันที่ 19 ตุลาคม พ.ศ.2556

1.7 วัตถุประสงค์หรือจุดมุ่งหมายของการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย ให้ปฏิบัติงานสหกิจศึกษา

- 1) เรียนรู้การทำงานจริงในสายงานของการเป็นนักพัฒนาซอฟต์แวร์
- 2) ทำงานตามลำดับขั้นตอน
- 3) แก้ไขระบบรายงานเดิมให้ดีขึ้น
- 4) พัฒนาหน้าจอระบบติดต่อกับผู้ใช้งาน
- 5) เชื่อมโยงความสัมพันธ์ระหว่างข้อมูล

1.8 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

- 1) นักศึกษามีความรับผิดชอบในงานที่ได้รับมอบหมายเป็นอย่างดี
- 2) นักศึกษาได้นำความรู้จากการศึกษามาใช้ประโยชน์ในการทำงานจริง
- 3) นักศึกษามีทักษะและเกิดความชำนาญในสายงานนี้มากขึ้น
- 4) นักศึกษาสามารถทำงานร่วมกับผู้อื่นได้เป็นอย่างดี
- 5) นักศึกษาสามารถทำงานภายในระยะเวลาที่กำหนดได้
- 6) ระบบรายงานที่พัฒนาสามารถนำไปใช้งานต่อได้
- 7) หน้าจอติดต่อกับผู้ใช้งานมีรูปแบบที่สมบูรณ์
- 8) ข้อมูลมีการเชื่อมโยงความสัมพันธ์กัน

บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ปฏิบัติงาน

2.1 ซอร์ฟแวร์ที่ใช้ในการปฏิบัติงาน

2.1.1 ภาษาซีชาร์ป (C Sharp)

2.1.1.1 ภาษาซีชาร์ปคืออะไร

ภาษาซีชาร์ป คือ ภาษาคอมพิวเตอร์ประเภทการเขียนโปรแกรมเชิงวัตถุ (object-oriented programming: OOP) พัฒนาโดย Microsoft โดยมีจุดมุ่งหมายในการรวมความสามารถการคำนวณของ ภาษาซีพลัสพลัส (C++) ด้วยการโปรแกรมง่ายกว่าของ visual basic โดย ภาษาซีชาร์ป มีพื้นฐานจาก C++ และเก็บส่วนการทำงานคล้ายกับ ภาษาจาวา (Java) C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของ Microsoft จุดมุ่งหมายคือ อำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศและบริการผ่านเว็บ และทำให้ผู้พัฒนาสร้าง โปรแกรมประยุกต์ในขนาดกะทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ เอกซ์เอ็มแอล (extensible markup language :XML) และ ซิมเปิลอ็อบเจกต์แอคเซสโพรโทคอล (simple object access protocol :SOAP) ซึ่งยอมให้เข้าถึงอ็อบเจกต์ของโปรแกรมหรือเมธอด โดยปราศจากความจำเป็นต้องให้ผู้เขียนโปรแกรมเขียนคำสั่งเพิ่มในแต่ละขั้นตอนเนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่ แทนที่การคัดลอกซ้ำ ภาษาซีพลัสพลัสภาษา ภาษาซีชาร์ป ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการนำข้อดีของภาษาต่าง ๆ เช่น ภาษาเดลฟี ภาษาซีพลัส พลัส มาปรับปรุงเพื่อให้มีความเป็น OOP มากขึ้น ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง ซึ่งเรียบง่ายกว่าภาษาซีพลัสพลัสและมีสิ่งที่เกินความจำเป็นน้อยลงเมื่อเทียบกับภาษาจาวา ข้อดีของภาษาซีชาร์ปมีดังนี้

- เป็นภาษาระดับกลาง คือ ได้รวมเอาข้อดีจากภาษาระดับต่ำและสูงเข้าไว้ด้วยกัน
- ใช้หน่วยความจำน้อย
- เป็นภาษาที่มีลักษณะเป็นการเขียนโปรแกรมเชิงวัตถุ
- ทำงานได้ดีบนวินโดวส์

- ฟังก์ชันแก้ไขใช้งานง่าย
- จัดการเรื่องความจำให้อัตโนมัติ

2.1.1.2 รูปแบบไฟล์ ASPX

ASPX เป็นชื่อรูปแบบไฟล์ของหน้าแบบฟอร์มเอเอสพีคอตเน็ตโดยทั่วไปแล้วในไฟล์จะมีรหัสแบบ HTML หรือ XHTML ใช้กำกับรูปแบบฟอร์ม หรือเนื้อหาในหน้าเว็บ และในส่วนของโค้ดนั้น อาจจะอยู่ในหน้าเดียวกันในแท็ก หรือบล็อก `<% -- รหัสที่ใช่ -- %>` จะคล้ายกับเทคโนโลยีที่ใช้พัฒนาเว็บ PHP และ JSP หรือแยกอยู่ในหน้าโค้ดออกมาต่างหาก (Code behind) เอเอสพีคอตเน็ตรองรับการเขียนโค้ดในบรรทัดเดียวกันทั้งหมดในไฟล์ ASPX แต่จะให้อ่านโค้ดได้ยากจึงไม่แนะนำให้ใช้

2.1.2 คอตเน็ตเฟรมเวิร์ก (.NET Framework)

คอตเน็ตเฟรมเวิร์ก คือ แพลตฟอร์มสำหรับพัฒนาซอฟต์แวร์ที่สร้างขึ้นโดยไมโครซอฟท์ รองรับภาษาที่สร้างขึ้นบนพื้นฐานของ .NET มากกว่า 40 ภาษา และมีไลบรารีเป็นจำนวนมาก สำหรับการเขียนโปรแกรม คอตเน็ตเฟรมเวิร์กทำให้การเขียนโปรแกรมมีความง่ายขึ้น โดยสามารถทำให้โปรแกรมที่เขียนขึ้นเพียงครั้งเดียวสามารถใช้งานในระบบปฏิบัติการใดก็ได้ที่รองรับคอตเน็ตเฟรมเวิร์ก โดยคอตเน็ตเฟรมเวิร์กมีส่วนประกอบภายในแบ่งออกเป็น 3 ชั้น คือ

1) ภาษาโปรแกรม (Programming Language) เป็นรูปแบบของภาษาที่ออกแบบมาเพื่อให้สามารถทำงานในรูปแบบของคอตเน็ตได้ โดยมีภาษาหลักๆ 3 ภาษาคือ

- VB.NET เป็นภาษาที่พัฒนามาจาก visual basic ในเวอร์ชัน 6.0
- C# เป็นภาษาที่พัฒนามาจาก C++ กับ JAVA
- JScript.net เป็นภาษาที่พัฒนามาจาก java script

2) คลาสไลบรารีพื้นฐาน (base class library) เปรียบเสมือนชุดคำสั่งสำเร็จรูปย่อยๆ ที่เพิ่มเข้ามาซึ่งส่วนใหญ่จะเป็นชุดคำสั่งที่ต้องใช้งานอยู่เป็นประจำ

3) เวอร์ชวลแมชีน (common language runtime: CLR) เป็นตัวที่ทำให้โปรแกรมที่เขียนด้วยภาษาต่าง ๆ กลายเป็นรูปแบบมาตรฐานเดียวกันทั้งหมด เรียกภาษานี้ว่า intermediate language หรือ

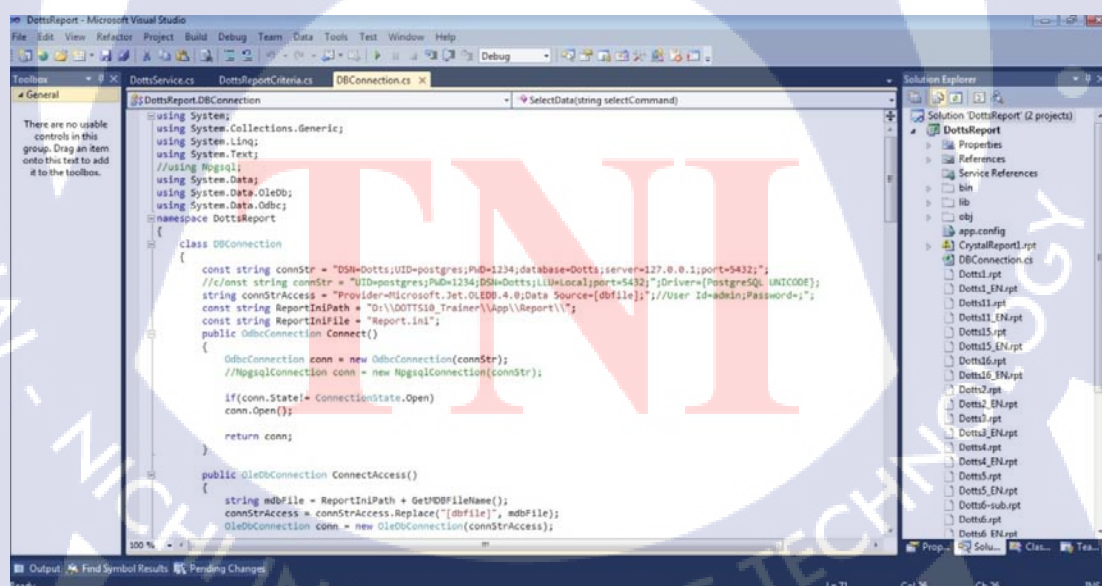
IL เมื่อรันโปรแกรม CLR จะทำการตรวจสอบสภาพของเครื่องเพื่อให้การคอมไพล์โปรแกรมที่มีประสิทธิภาพสูงสุดในแต่ละเครื่อง

2.1.3 วิวอลเบสิกคอตเน็ต (VB.NET)

วิวอลเบสิกคอตเน็ต คือ ภาษาหนึ่งในกลุ่มไมโครซอฟท์วิวอลสตูดิโอคอตเน็ต (Microsoft Visual Studio .NET) โดยมีรากฐานภาษามาจากภาษาเบสิก และทำงานบนคอตเน็ตเฟรมเวิร์ค (.NET Framework) ถูกออกแบบให้มีความสามารถในการพัฒนาโปรแกรมเชิงวัตถุ (Object-Oriented Programming) และรองรับการออกแบบด้วยยูเอ็มแอล (Unified Modeling Language)

2.1.4 ไมโครซอฟท์ วิวอลสตูดิโอ (Microsoft Visual Studio)

ไมโครซอฟท์ วิวอลสตูดิโอ คือ เครื่องมือที่มีคุณภาพอย่างหนึ่งสำหรับการพัฒนาโปรแกรม ในโปรแกรมจะมีการรวมเครื่องมือที่จำเป็นต่อการพัฒนาโปรแกรมอย่างครบถ้วน ซึ่งสามารถพัฒนาได้หลายภาษา และมีคลาสพื้นฐานให้นำมาใช้งานร่วมกันได้อย่างเป็นระบบ เช่น เอสคิวเอลเซิร์ฟเวอร์ (sql server) VB.NET วิวอลซีชาร์ป (vc#) วิวอลเจชาร์ป (vj#) วิวอลซีพลัสพลัส (Visual C++) และเอเอสพีคอตเน็ต (ASP .NET) เป็นต้น โดยทั้งหมดทำงานอยู่บน CLR ที่รองรับการประมวลผล และเข้าใช้ทรัพยากรในเครื่องได้อย่างมีประสิทธิภาพ

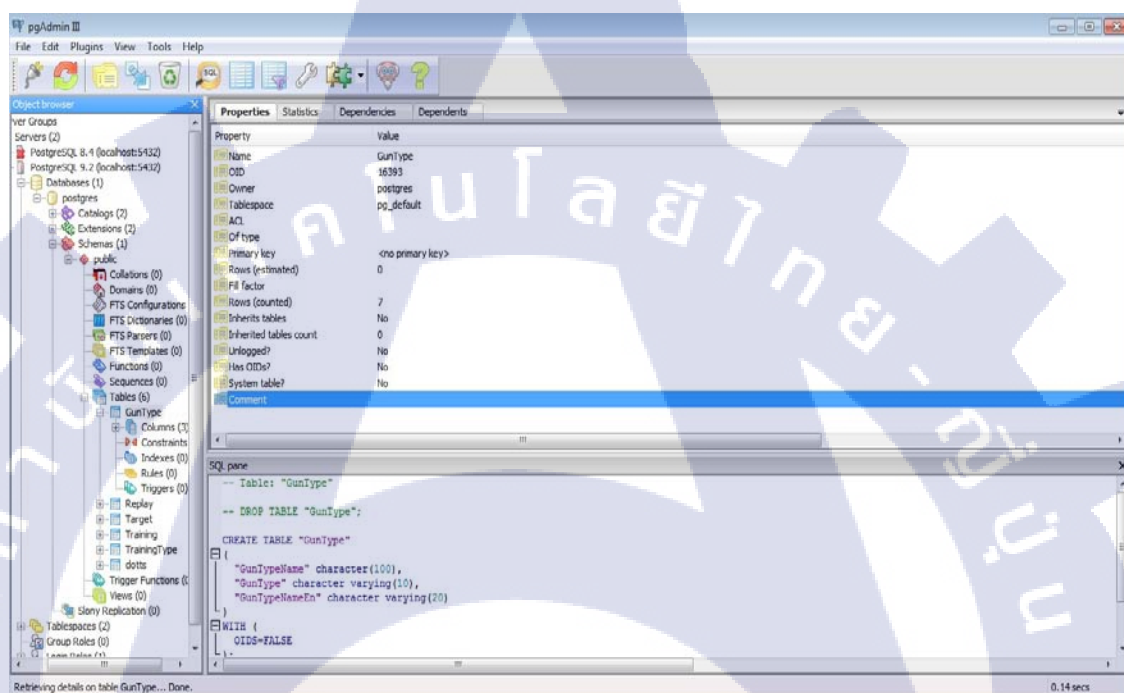


รูปที่ 2.1 ส่วนของการทำงานส่วนหนึ่งในโปรเจกของการพัฒนาโปรแกรมรายงานระบบจำลอง

การฝึกยิงอาวุธปืนเล็กยาว โดยใช้โปรแกรม Microsoft Visual Studio 2010 Professional

2.1.6 โพสต์เกรสคิวแอล (PostgreSQL)

เป็นระบบจัดการฐานข้อมูลในลักษณะของซอฟต์แวร์เสรีภายใต้สัญญาอนุญาตบีเอสดี ชื่อเดิมของซอฟต์แวร์คือ โพสต์เกรส ซึ่งต่อมาได้ถูกเปลี่ยนเป็นโพสต์เกรสคิวแอล โดยประกาศออกมาจากทีมหลักในปี 2550 ชื่อของโพสต์เกรสมาจากชื่อ post-Ingres ซึ่งหมายถึงตัวซอฟต์แวร์ที่พัฒนาต่อจากซอฟต์แวร์ชื่ออินเกรส



รูปที่ 2.2 ส่วนของการทำงานส่วนหนึ่งในโปรเจกของการพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว โดยใช้โปรแกรม โปรแกรม PostgreSQL

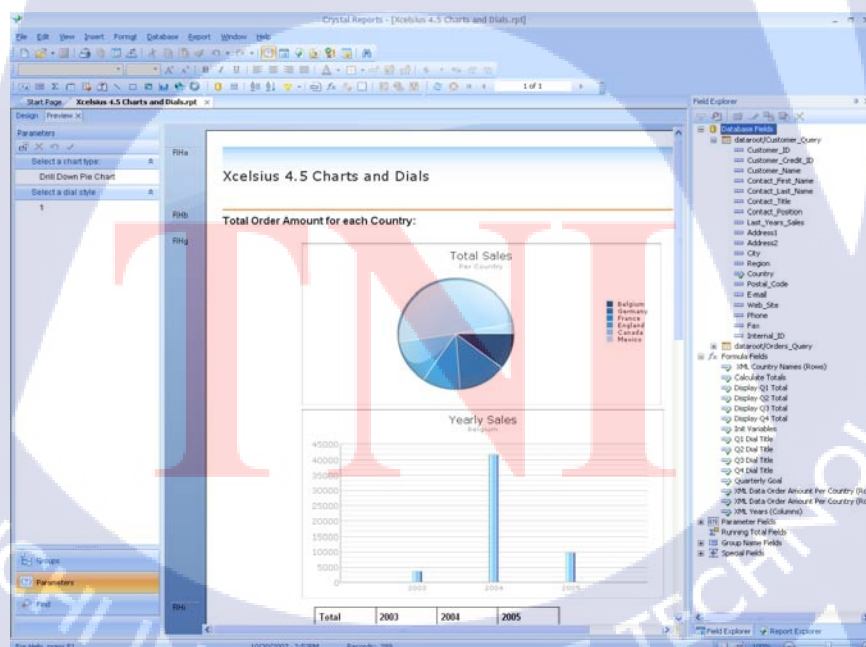
2.1.5 ภาษาเอสคิวแอล (Standard Query Language: SQL)

เอสคิวแอล เป็นภาษาที่พัฒนาขึ้นโดยบริษัทไอบีเอ็ม เพื่อใช้สำหรับการเรียกใช้ฐานข้อมูล โดยที่ SQL เป็นภาษาที่มีมาตรฐานและเป็นระบบเปิด (open system) คือสามารถใช้กับฐานข้อมูลชนิดใดก็ได้ และใช้คำสั่งเดียวกันเมื่อสั่งระบบฐานข้อมูลที่แตกต่างกันจะได้ผลลัพธ์เหมือนกัน โดยภาษาเอสคิวแอลแบ่งออกเป็น 4 รูปแบบคือ

- 1) ภาษาที่ใช้สำหรับนิยามข้อมูล (data definition language: DDL)
- 2) ภาษาสำหรับการจัดการข้อมูล (data manipulation language: DML)
- 3) ภาษาควบคุม (control language)
- 4) ภาษาในการเลือกข้อมูล (data query language)

2.1.6 คริสตัลรีพอร์ต (Crystal Report)

Crystal Report คือ เครื่องมือที่ใช้สำหรับการออกรายงานที่มีประสิทธิภาพ มีคุณสมบัติการรองรับในส่วนของการนำเสนอข้อมูล และนำเข้าข้อมูลต่าง ๆ เข้าไปเก็บไว้ในฐานข้อมูลทำได้หลากหลาย เช่น MS SQL Server Access Excel XML และ ADO.Net



รูปที่ 2.3 ส่วนของการทำงานส่วนหนึ่งในโปรเจกของการพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว โดยใช้โปรแกรม Microsoft Visual Studio 2010 Professional

ชนิดของข้อมูล (Data Type)

การใช้ชนิดข้อมูลได้อย่างถูกต้องในการสร้างฐานข้อมูลทำให้การจัดการใช้เนื้อที่หน่วยความจำได้อย่างมีประสิทธิภาพ หน่วยความจำนี้รวมถึงฮาร์ดดิสก์ด้วย ดังนั้น เราควรจะทำ ความรู้จักชนิดข้อมูลที่ใช้ในฐานข้อมูล SQL Server ก็จะมีผลคล้ายคลึงกับชนิดข้อมูลของผู้ผลิต รายอื่น ๆ เพราะใช้มาตรฐาน ANSI เป็นต้นแบบในการผลิตแอปพลิเคชันฐานข้อมูล ซึ่งแบ่งเป็น ชนิดของข้อมูล ดังนี้

ตาราง 2.1 ตารางแสดงชนิดของข้อมูลแบบ Character

ชนิดข้อมูล	ขอบเขตของชนิดข้อมูล	ขนาดหน่วยความจำ (ไบต์)	ชนิดข้อมูล
Char[(n)]	1 - 8000	n	Char[(n)]
Varchar[(n)]	1 – 8000	ความยาวข้อมูล	Varchar[(n)]

ตาราง 2.2 ตารางแสดงชนิดของข้อมูลแบบ Binary

ชนิดข้อมูล	ขอบเขตของชนิดข้อมูล	ขนาดหน่วยความจำ (ไบต์)
Binary(n)	1 - 8000	n
Varbinary(n)	1 - 8000	n + 1
Image	231 -1(2,147,483,647) ไบต์	16+Multiple of 2k
Timestamp	ใช้สำหรับเปลี่ยนการจัดการ	16

ตาราง 2.3 ตารางแสดงชนิดของข้อมูลแบบ Date

ชนิดข้อมูล	ขอบเขตของชนิดข้อมูล	ขนาดหน่วยความจำ (ไบต์)
Datetime	วันที่เริ่มต้นตั้งแต่ January 1,1753 ถึง December 31,9999 ความละเอียดถึง 1/1000 วินาที	8
Smalldatetime	วันที่เริ่มต้นตั้งแต่ January 1,1900 ถึง June 6,2079	4

ตาราง 2.4 ตารางแสดงชนิดของข้อมูลแบบ Logical

ชนิดข้อมูล	ขอบเขตของชนิดข้อมูล	ขนาดหน่วยความจำ (ไบต์)	ชนิดข้อมูล
Bit	0 หรือ 1	1	Bit

ตาราง 2.5 ตารางแสดงชนิดของข้อมูลแบบ Numeric

ชนิดข้อมูล	ขอบเขตของชนิดข้อมูล	ขนาดหน่วยความจำ (ไบต์)
Int	$\pm 2,147,483,647$	4
Smallint	$\pm 32,767$	2
Tinyint	0 255	1
Float(p) P (Precision) คือ การนับจำนวน รวมของเลขหน้าและหลัง ทศนิยม มีค่าตั้งแต่ 1 ถึง 28 S (Scale) คือ จำนวนตัวเลขหลัง ทศนิยม	$\pm 1.79E+308$	4 (Precision<16) 8 (Precision>=16)
Double p	$\pm 1.79E+308$	8
Real	$\pm 1.79E+308$	4
Numeric(p,s) Decimal(p,s)	$\pm 10^3$	2 ถึง 17
Money	$\pm \$922,337,203,685,477.5807$	8
Smallmoney	$\pm \$214,748.3647$	4

2.2 ข้อมูลจำเพาะของอาวุธปืน

2.2.1 ปืนเล็กยาว

เป็นอาวุธปืนขนาดเล็กที่มีความแม่นยำในการยิงสูง โดยการยิงลูกกระสุนจะออกมาทีละนัด และระยะหวังผลที่ 460 เมตร การบรรจุของกระสุน สามารถบรรจุได้ 20- 30 นัด เป็นอาวุธปืนที่มีน้ำหนักเบา และมีรูปแบบการยิงแบบอัตโนมัติละกึ่งอัตโนมัติ



รูปที่ 2.4 ปืนเล็กยาว

2.2.3 ข้อมูลทางเทคนิค

ตารางที่ 2.6 ตารางแสดงข้อมูลทางเทคนิคของอาวุธปืนเล็กยาว

ขนาดกว้างปากลำกล้อง	5.56 มิลลิเมตร
ความยาวของลำกล้อง (ไม่รวมปลอกกลดแสง)	508 มิลลิเมตร
ความยาวของลำกล้อง (ไม่รวมปลอกกลดแสง)	530 มิลลิเมตร
ความยาวของปืนทั้งกระบอก	986 มิลลิเมตร
น้ำหนักปืนทั้งกระบอก	2.89 กิโลกรัม
เกลียวในลำกล้อง	6 เกลียว เวียนขวาครบรอบที่ระยะ 305 มิลลิเมตร
แบบการทำงาน	ด้วยแก๊ส
ความเร็วต้น	990-1,000 เมตร/วินาที (3,248-3,280 ฟุต/วินาที)
อัตราเร็วในการยิง	650- 750 นัด/นาที
ระยะยิงไกลสุด	2,653 เมตร
ระยะยิงหวังผล	460 เมตร
เครื่องป้อนกระสุน ของกระสุนบรรจุ	20 30 นัด
กระสุนขนาด	5.56 X 45 มิลลิเมตร
ประเทศผู้ผลิต	อเมริกา

บทที่ 3

การพัฒนาโปรแกรมรายงานผลการฝึกยิง

3.1 แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1.1 แผนงานปฏิบัติงาน

ตารางที่ 3.1 แผนปฏิบัติงานสหกิจศึกษา

หัวข้อ	เดือนที่ 1	เดือนที่ 2	เดือนที่ 3	เดือนที่ 4
ศึกษาเกี่ยวกับ VB.NET และ C#				
แก้ไขระบบ POInstep				
Cloud TA: เก็บข้อมูลรูป ลายนิ้วมือ				
Planning การพัฒนาระบบ				
ศึกษา Requirement ของรายงานอาวุธปืน				
ออกแบบ Prototype และ ER- diagram				
ออกแบบ Class Diagram				
Database interface (DLL)				
Data simulator				
Report viewer				
User manual				
Test				
Deployment				
User Acceptance Test (UAT)				

3.1.2 รายละเอียดงานที่นักศึกษาปฏิบัติในงานสหกิจศึกษา หรือรายละเอียดโครงการที่ได้รับมอบหมาย

งานที่ได้รับมอบหมายในการมาสหกิจศึกษา คือ การพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว ในการพัฒนาระบบ มีดังต่อไปนี้

3.1.2.1 ส่วนติดต่อกับผู้ใช้

ในส่วนนี้จะเป็นการออกแบบหน้าต่างสำหรับผู้ใช้งานในการเลือกดูผลการฝึกที่ผ่านมา โดยการค้นหาจะกำหนดจากตัวแปรหลักต่าง ๆ เช่นวันที่ เวลาในการฝึกและประเภทการฝึก เป็นต้น

3.1.2.2 ผลรายงาน

ในส่วนนี้จะเป็นการออกแบบรูปแบบผลรายงานการฝึกย้อนหลังของผู้ฝึก ซึ่งจะแสดงรายละเอียดต่าง ๆ ของการฝึกมี 2 ส่วนดังนี้

1) ส่วนแสดงข้อมูลการฝึกทั้งหมดของผู้ฝึกผู้นั้นในส่วนนี้จะแสดง รายละเอียดของการฝึกทั้งหมด คือ ประเภทการฝึก อาวุธปืนที่ใช้ จำนวนลูกกระสุนที่บรรจุ จำนวนลูกกระสุนที่ใช้ทั้งหมด จำนวนข้าศึกที่โดนทำลาย เวลาที่เริ่มฝึก เวลาที่จบการฝึก พลังชีวิตที่เหลือ รายละเอียดการยิงแต่ละนัดกระสุน คะแนนที่ได้ในการยิงกระสุนนัดนั้น เป็นต้น

2) ส่วนแสดงกราฟการฝึก

ในส่วนนี้จะนำข้อมูลการฝึก ผลการยิงของผู้ฝึกผู้นั้นนำมาวิเคราะห์ออกมาในรูปแบบกราฟ เพื่อให้ผู้ใช้งานสามารถนำกราฟดังกล่าวนี้ไปประเมินผลการฝึกของผู้ฝึกว่า การฝึกนี้มีประสิทธิภาพมากเพียงใด

3.1.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโครงการ

3.1.3.1 ศึกษาเกี่ยวกับ VB.NET และ C#

ศึกษาเกี่ยวกับการใช้ภาษา VB.NET และ C# และการใช้โปรแกรม Microsoft Visual Studio 2010 Professional PostgreSQL และ Crystal Report ในระดับเบื้องต้นในสัปดาห์ที่ 1 และ 2 เพื่อเตรียมพร้อมในการจัดทำโครงการตามที่ได้รับมอบหมาย

3.1.3.2 ศึกษาเกี่ยวกับอาวุธปืนประเภทปืนเล็กยาว

ทำการศึกษาข้อมูลเบื้องต้นต่าง ๆ ของอาวุธปืนประเภทปืนเล็กยาว เพื่อนำมาใช้ประกอบการออกแบบรูปแบบของรายงานการฝึก

3.1.3.3 ออกแบบโปรโตไทป์และER-Diagram

ทำการออกแบบรูปแบบของรายงานการฝึกของอาวุธปืนเล็กยาว โดยนำข้อมูลที่ได้จากการศึกษาคุณสมบัติปืนดังกล่าวมาวิเคราะห์ว่า ในรูปแบบรายงานนั้นควรจะมีหัวข้ออะไรบ้าง เพื่อให้ทำให้ผู้ฝึกสามารถนำข้อมูลดังกล่าวไปใช้ประโยชน์ได้มากที่สุด และนำหัวข้อดังกล่าวนั้นมาจัดเรียงเป็นรูปแบบหน้าตาของรายงาน

การออกแบบER-Diagram ของระบบฐานข้อมูล จะกำหนดความสัมพันธ์ต่าง ๆ ของตัวแปร และเพื่อความสะดวกในการออกแบบฐานข้อมูลในขั้นตอนต่อไป

3.1.3.4 ออกแบบระบบฐานข้อมูล (database interface: DLL)

การออกแบบระบบฐานข้อมูล เพื่อแยกข้อมูลที่มีออกเป็นแต่ละตาราง และเตรียมพร้อมสำหรับการกำหนดค่าของข้อมูลให้กับตัวแปรต่าง ๆ อีกทั้งเพื่อการจัดเก็บข้อมูลที่ได้จากการฝึก การออกแบบระบบฐานข้อมูลนี้จะนำไปใช้ในการเขียนโปรแกรมเพื่อดึงค่าของข้อมูลไปแสดงผลออกมาในรูปแบบรายงาน

3.1.3.5 เขียนโปรแกรม (System Develop (Coding))

ทำการเขียนโปรแกรมจากที่ได้ออกแบบไว้ ทำการเพื่อให้ได้การทำงานที่เป็นระบบตามที่วางแผนไว้โดยใช้โปรแกรม Microsoft Visual Studio 2010 ในการเขียนโปรแกรม ภาษาที่ใช้คือ ภาษาC# และนำโปรแกรมที่ได้ไปทดสอบในขั้นตอนต่อไป

3.1.3.5 ทดสอบระบบ (System Testing)

ทำการทดสอบระบบทั้งหมด และแก้ไขจุดบกพร่องของระบบ

เมื่อเขียนเสร็จแล้วจะต้องทำการทดสอบระบบโดยจะมีเอกสารบันทึกข้อผิดพลาดที่พบเห็นในการทดสอบระบบเพื่อระบุข้อผิดพลาดที่พบเห็นในระบบเพื่อส่งต่อไปยังผู้รับผิดชอบให้ทำการแก้ไข

3.1.3.6 ทำคู่มือการใช้งาน (User manual preparation)

จัดทำคู่มือการใช้งานโปรแกรม สำหรับผู้ใช้เพื่อความสะดวกต่อการศึกษาโปรแกรม
ดังกล่าว

3.1.3.7 Installation, Training and Support

ทำการติดตั้งระบบเพื่อใช้งานจริง

3.2 การพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว และ รายละเอียดสหกิจศึกษา

3.2.1 ช่วยงานระบบ Cloud TA

ระบบ Cloud TA เป็นระบบ การจัดเวลาที่ทำงานบน Cloud ซึ่งในส่วนที่ข้าพเจ้ารับผิดชอบ นั้น คือช่วยทดสอบสคริปต์ของโปรแกรม ในช่วง สัปดาห์ 1 – 2 ต่อมาเป็นช่วงการดำเนินการติดตั้ง ระบบ และงานส่วนที่ข้าพเจ้าได้รับคือ ช่วยติดตั้งและเก็บข้อมูลของลูกค้าในพื้นที่ต่าง ๆ โดยเป็น การเก็บรูปภาพ และลายนิ้วมือของลูกค้า ต่อมาเมื่อดำเนินการเก็บข้อมูลเสร็จสิ้น ส่วนที่ได้รับ มอบหมายต่อไป คือ การตัดต่อภาพถ่ายเพื่ออัปโหลดข้อมูลลงฐานข้อมูล โดยใช้โปรแกรม Photo Scope ในช่วงสัปดาห์ที่ 4 – 5

3.2.2 แกไขระบบ POInstep

ระบบ POInstep เป็นระบบช่วยจัดการการสั่งซื้อสินค้าที่ใช้ภายในกลุ่มบริษัท Instep Group ในส่วนที่ข้าพเจ้าได้รับมอบคือ ต้องติดตั้งระบบนี้ให้สามารถใช้งานได้จริง การดำเนินการส่วน แรก คือการศึกษาระบบเก่า และติดต่อกับผู้ใช้งานเพื่อศึกษาความต้องการเพิ่มเติม และทำการนำเข้า ข้อมูลลงสู่ระบบ ในช่วงสัปดาห์ที่ 3 –

ระบบจัดซื้อ									
หน้าหลัก									
หน้าเข้าบริหารจัดการขอซื้อ									
เลือกไฟล์ ไม่ได้อัปโหลดไฟล์ใด									
Preview									
ชื่อสินค้า	จำนวน	ราคาต่อชิ้น	ประเภท	ผู้จำหน่าย	บริษัทผู้ซื้อ	งบใบขอซื้อ	วันที่ขอซื้อ	วันที่เสร็จ	หมายเหตุ
ปืนสติกเกอร์ ดรา ฟัง A1025-50 MM	22	23.36	เครื่องเขียน	POINSTEP	DESIGN GATEWAY	56003	02.01.56		
ค่าทำความสะอาด OFFICE	1	5,000.00	ค่าจ้าง	บุคคลธรรมดา	DESIGN GATEWAY	56005	03.01.56		ลูกค้าเคลมเครื่องกับ บริษัทราฟ
นามบัตรกระดาษ INKJET 240 แกรม ปลาย LININ	200	3.50	นามบัตร	PORPERN 49	DESIGN GATEWAY	56042	06.01.56		ขนาด 5.5 X 9.0 ซม. พิมพ์ Inkjet 4 สี 2หน้า 1ด้านหลัง สี
HARDDISK SEAGATE 500GB SATA 7200	2	1,795.00	HARDWARE COMPUTER	INNOVA	DESIGN GATEWAY		15.01.56		
นามบัตรกระดาษ INKJET 240 แกรม ปลาย LININ	300	3.50	นามบัตร	PORPERN 49	DESIGN GATEWAY	56049	21.01.56		
HIDEKAZU SHINOHARA 200 ใบ	300	3.50	นามบัตร	PORPERN 49	DESIGN GATEWAY	56049	21.01.56		
YASUKHIKO 100 ใบ	300	3.50	นามบัตร	PORPERN 49	DESIGN GATEWAY	56049	21.01.56		
ชาวันโก๋ หน้ต่าง กระชก ของดีสี่ อำเภอ	1	1,000.00	งานซ่อม ติดต่ออาคาร	อาคารบิณี	DESIGN GATEWAY	56058	30.01.56		
ถุงพลาสติก งานเป็ กดี	1	160.00	ถุงพลาสติก	ราชวงศ์บรรจภัณฑ์	DESIGN GATEWAY		30.01.56		
CANNON PG40+CL41 แท้	1	1,480.00	TONER	MGM	DESIGN GATEWAY	56060	30.01.56		

รูปที่ 3.1

ตัวอย่างการนำเข้าข้อมูลระบบ POInstep

3.2.3 ตัดรูปภาพประกอบในระบบ CloudTA

ส่วนนี้ได้รับมอบหมายให้ทำหน้าที่ตัดรูปภาพให้อยู่ในขนาดที่สามารถนำไปใช้กับระบบได้ คือ 4:3 หรือ 480:360 พิกเซล

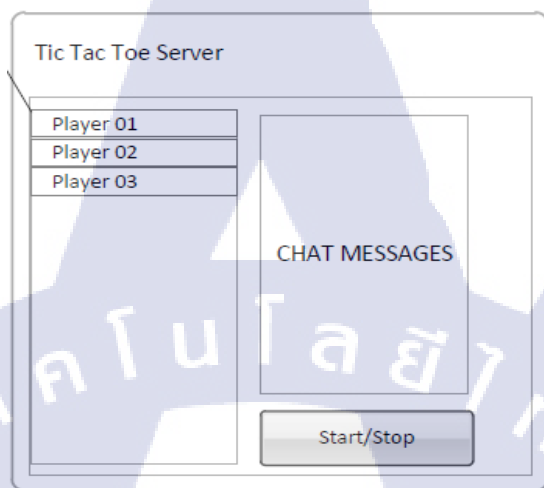


รูปที่ 3.2 ส่วนของการทำงานส่วนหนึ่งในการ cropping ในระบบ CloudTA

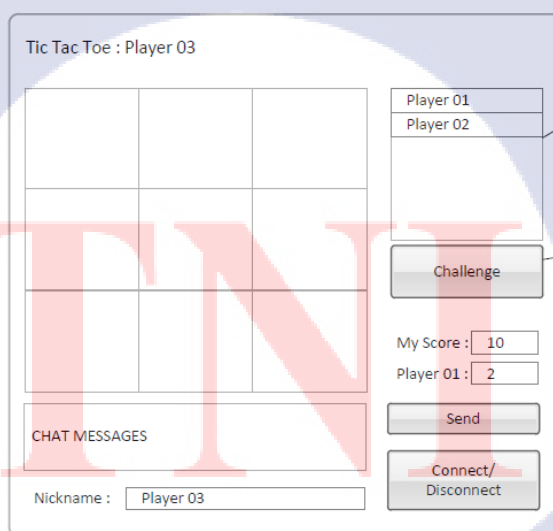
โดยใช้โปรแกรม PhotoScape

3.2.4 ทำแบบฝึกหัดการใช้ภาษาซีชาร์ป

ก่อนที่จะเริ่มต้นการทำโครงการมีการทำแบบฝึกหัดการใช้ภาษาซีชาร์ปโดยทางผู้ดูแลได้ให้โจทย์คือ การทำโปรแกรมเกมส์ Tic Tac Toe



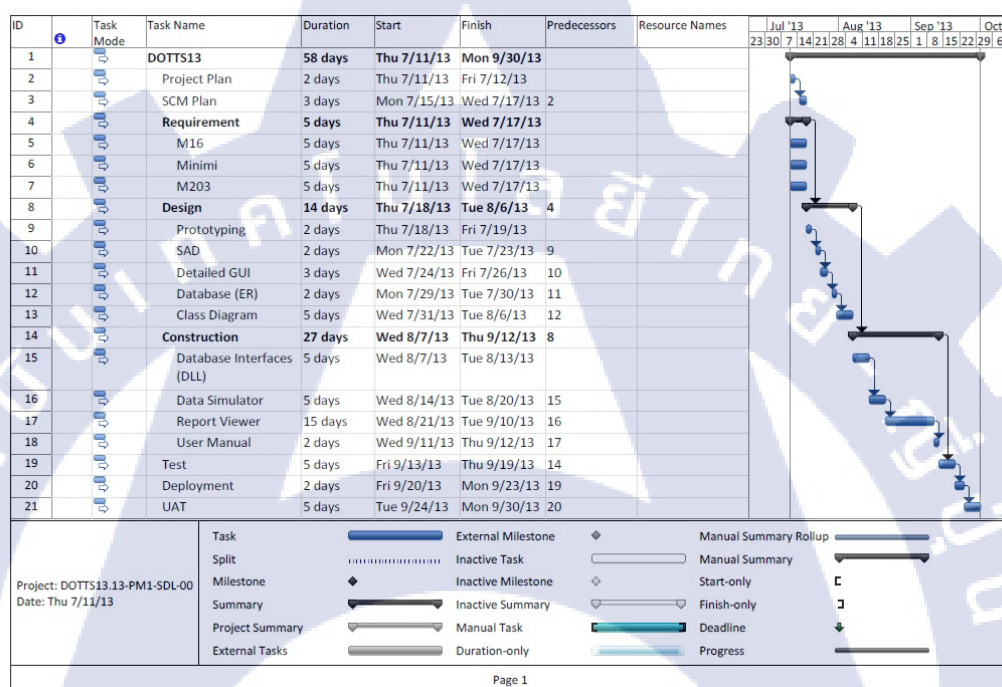
รูปที่ 3.3 หน้าจอฝั่งเซิร์ฟเวอร์โปรแกรมTicTacToe



รูปที่ 3.4 หน้าจอของฝั่งผู้เล่นโปรแกรมTicTacToe

3.2.5 ขั้นตอนการวางแผนพัฒนาระบบรายงาน

มีการประชุมเพื่อทำการวางแผนงานทั้งหมด กำหนดระยะเวลาการดำเนินงานในลำดับขั้นตอนต่าง ๆ โดยแบ่งออกเป็นส่วนอย่างชัดเจน ดังรูปภาพที่ 3.5



รูปที่ 3.5 แผนการพัฒนาแบบ

3.2.6 ขั้นตอนการเก็บความต้องการของรายงาน

ในขั้นตอนนี้จะเป็นขั้นตอนการเก็บความต้องการของรายงาน และนำมาสรุปผลเพื่อที่จะนำมาออกแบบรูปแบบรายงานได้อย่างเหมาะสม โดยการเก็บความต้องการนั้นมาจากการศึกษาข้อมูลของอาวุธปืน และ ข้อมูลของระบบเพื่อนำมาวิเคราะห์

3.2.7 การออกแบบ

3.2.7.1 การออกแบบตัวต้นแบบของโปรแกรม

ออกแบบรูปแบบหน้าต่างของโปรแกรมและการทำเอกสารประกอบรายละเอียดของต้นแบบโปรแกรม ดังรูป 3.6

รูปที่ 3.6 ต้นแบบโปรแกรม

รายละเอียดของต้นแบบโปรแกรมที่วางแผนมีดังนี้

- 1) ชนิดปิ่น เลือกชนิดอาวุธที่ต้องการดูรายงานการฝึก ประกอบด้วย
 - ปิ่นเล็กยาว
 - ปิ่นกลเบา
 - เครื่องยิงลูกระเบิด

2) ชนิดการฝึก เลือกชนิดการฝึกตามชนิดปืน ประกอบด้วย

● ปืนเล็กยาว

- ฝึกเล็ง
- ฝึกยิงปรับศูนย์ระยะ 25 เมตร
- ฝึกยิงเบื้องต้น
- ฝึกยิงเป้านิ่งบนระยะต่างกัน
- ฝึกยิงเป้าล้มลูกบนระยะต่างกัน
- ฝึกยิงเพื่อทดสอบประสาท
- ฝึกยิงเป้าเสมือนจริง

● ปืนกลเบา

- ฝึกยิงเป้าปืนกล 10 เมตร
- ฝึกยิงเป้าในสนามทราบระยะ

● เครื่องยิงลูกระเบิด

- ฝึกยิงเป้าปรับระยะ 200 เมตร
- ฝึกยิงเป้ากรอบหน้าต่างระยะ 50 เมตร
- ฝึกยิงเป้าบังเกอร์ระยะ 200 เมตร
- ฝึกยิงเป้าที่ตั้งยิงอาวุธกลระยะ 220 เมตร
- ฝึกยิงเป้าทหารในที่กำบังเปิดระยะ 300 เมตร

● ทั้งหมด

- ฝึกสถานการณ์จำลอง

3) ชื่อยิง

ระบุชื่อยิงที่ต้องการดูรายงาน

4) วันที่

ระบุวัน/เดือน/ปี ที่ต้องการเรียกดูผลรายงาน

5) เวลา

5.1) เริ่มต้น

5.2) สิ้นสุด

6) ระดับความยาก

กำหนดระดับความยากของการฝึก

7) รางวัล

กำหนดรางวัลที่ฝึกได้รับ

8) ค้นหา

ค้นหารายงานตามค่าที่ตั้งไว้

9) กร่องแสดงข้อมูลจากฐานข้อมูล

กร่องที่แสดงข้อมูลที่นำข้อมูลมาแสดงจากตารางฐานข้อมูลเมื่อกดคำสั่งค้นหา

10) กร่องแสดงรายละเอียด

แสดงรายละเอียดข้อมูล ชื่อยิง อาวุธ ระดับความยาก รางวัล และคะแนนรวม
ของข้อมูลที่ถูกเลือกจากตารางแสดงฐานข้อมูล

11) ตกลง

กดเพื่อแสดงรายละเอียดทั้งหมดที่ค้นหา โดยเข้าหน้า report

12) ยกเลิก

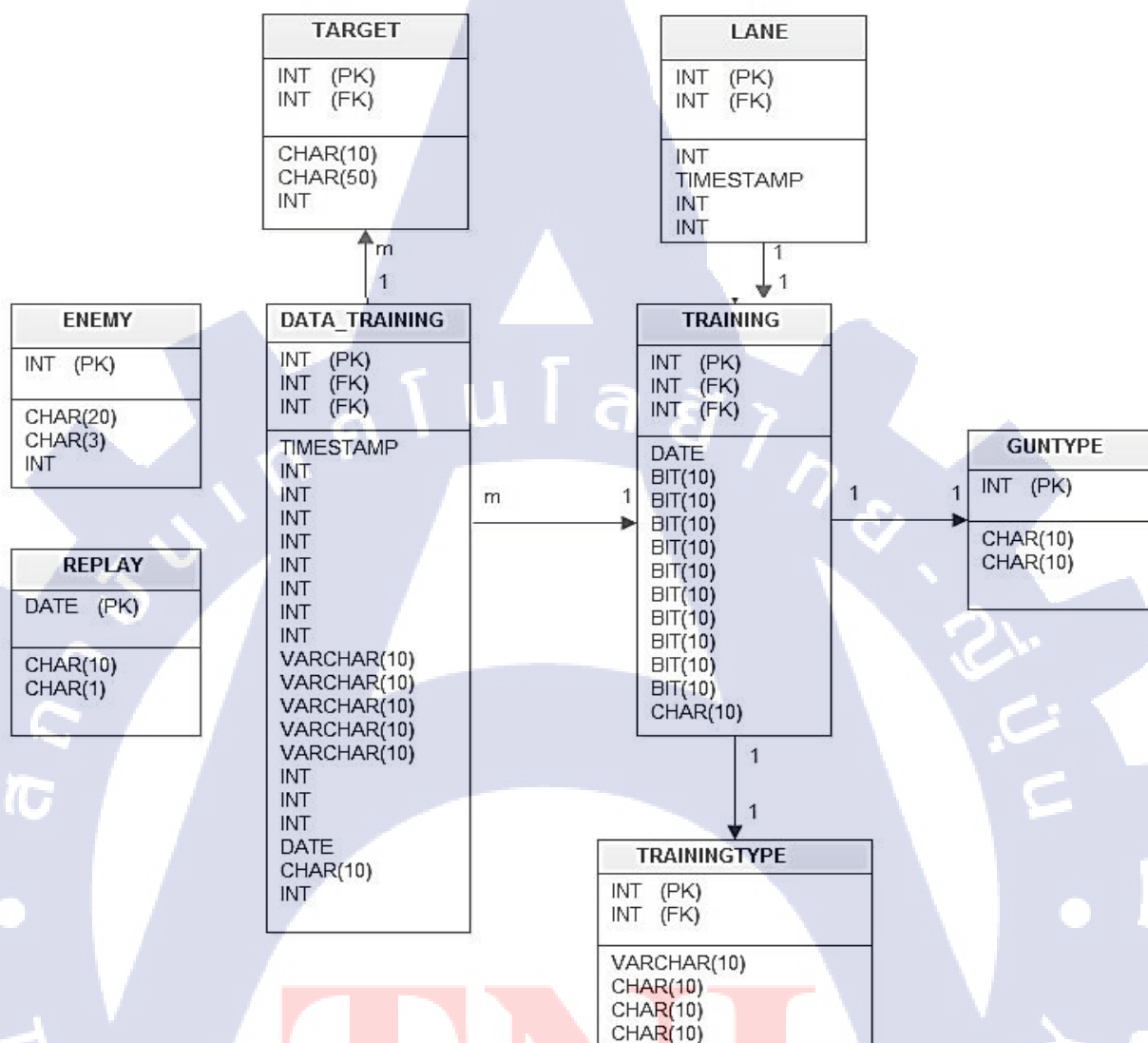
ยกเลิกทั้งหมดและกลับหน้า menu

13) ล้าง

การล้างข้อมูลที่ถูกเลือกในหน้าต่างทั้งหมด

TNI

3.2.7.2 การออกแบบความสัมพันธ์ระหว่างข้อมูล (ER-Diagram) ดังรูป 3.7



รูปที่ 3.7 ความสัมพันธ์ของตารางข้อมูล

อธิบายความหมายของตาราง

1) ตารางข้อมูลการฝึกยิง

ในตารางนี้มีการเก็บข้อมูลจากการฝึกทั้งหมด เวลาที่ฝึก ช่องยิงที่มีการฝึก

ลำดับการยิงคะแนนในแต่ละนัด

2) ตารางการฝึก

ในตารางนี้มีการเก็บข้อมูลสถานะการใช้ช่องยิมนั้น ๆ ว่ามีผู้ฝึกอยู่หรือไม่ และวันที่ในการฝึกระดับความยาก

3) ตารางอาวุธ

ในตารางนี้มีการเก็บข้อมูลชื่อของประเภทอาวุธปืน ทั้งภาษาไทยและภาษาอังกฤษ

4) ตารางข้าศึก

ตารางนี้มีการเก็บข้อมูลประเภทของข้าศึก พลังชีวิต และคะแนนของของข้าศึกแต่ละตัวเพื่อนำไปคำนวณคะแนนหลังจากจบการฝึก

5) ตารางเป้าหมาย

6) ตารางประเภทการฝึก

ในตารางนี้มีการเก็บข้อมูลประเภทของการฝึก คำอธิบายการฝึกนั้น ๆ ชื่อของประเภทการฝึกทั้งภาษาไทยและภาษาอังกฤษ

7) ตารางช่องยิง

ในตารางนี้มีการเก็บข้อมูลของการฝึกในแต่ละช่องยิมนั้น ๆ ประกอบด้วยพลังชีวิตที่เหลือ เวลาตาย กระสุนที่บรรจุทั้งหมด กระสุนที่ใช้ คะแนนรวมการฝึกทั้งหมด

8) ตารางวิดีโอการฝึก

3.2.8 ขั้นตอนการสร้างโปรแกรม

3.2.8.1 ส่วนเขียนคำสั่ง

การเขียนคำสั่งต่าง ๆ ในการเรียกใช้และแสดงข้อมูลตามที่ได้ออกแบบไว้ ประกอบด้วยฟังก์ชันดังนี้

1) ฟังก์ชันติดต่อกับระบบฐานข้อมูล

ฟังก์ชันนี้จะมีการเขียนคำสั่งให้มีการติดต่อกับฐานข้อมูลเพื่อที่สามารถเรียกใช้และบันทึกข้อมูลลงในฐานข้อมูลได้

2) ฟังก์ชันประกาศตัวแปร

ฟังก์ชันนี้จะมีการประกาศตัวแปรให้เป็นตัวแปรระดับโกลบอลซึ่งจะสามารถ

เข้าถึงได้ทั่วทั้งโปรแกรม ทั้งในส่วนของตัวแปรโลคอลที่ประกาศไว้ในฟังก์ชันและตัวแปรโกลบอลที่ประกาศไว้ในโปรแกรมหลัก

3) ฟังก์ชันอ่านคิวรี

รองรับคิวรีจากฟังก์ชันอื่น ๆ ที่เรียกใช้ และทำการอ่านค่าคิวรีนั้น เพื่อส่งการฐานข้อมูล

4) ฟังก์ชันบันทึกค่าลงฐานข้อมูล

สร้างคิวรีและรับค่าที่ผู้ใช้งานต้องการเพิ่มลงในฐานข้อมูล ส่งมาจาก User Interface และส่งไปยังฟังก์ชันอ่านคิวรีเพื่อทำการประมวลผล

5) ฟังก์ชันดึงค่าจากฐานข้อมูล

กำหนดคิวรีเพื่อดึงค่าจากฐานข้อมูล มีการสร้างตัวแปรเพื่อรับค่าที่ผู้ใช้งานต้องการ และ นำมาใส่คิวรีที่เขียนรอไว้ แล้วจึงส่งต่อไปยังฟังก์ชันอ่านคิวรี

6) ฟังก์ชันสุ่มค่าสำหรับใช้ทดสอบการแสดงผลรายงาน

สุ่มค่าและบันทึกลงฐานข้อมูล เพื่อนำมาใช้ทดสอบการแสดงผลของรายงาน

7) ฟังก์ชันการคำนวณหาเวลา

ฟังก์ชันนี้ใช้สำหรับหาค่าเวลาให้สามารถใช้คำนวณได้ ทั้งการหา ชั่วโมง นาที วัน เดือน ปี และ ช่วงเวลา เป็นต้น

8) ฟังก์ชันเชื่อมต่อ Crystal Report

สำหรับเชื่อมต่อระหว่างคลาสกับ Crystal Report ที่ใช้แสดงผล

9) ฟังก์ชันสร้างตารางฐานข้อมูลสำหรับ Crystal Report

สร้างตารางฐานข้อมูล กำหนดจำนวนและชื่อของหลักและแถวของตาราง เพื่อทำการรับค่าข้อมูล ที่ต้องการแสดงในรายงาน

10) ฟังก์ชันสร้างชุดข้อมูล

สร้างชุดข้อมูลเพื่อเก็บตารางฐานข้อมูลหลาย ๆ ตารางเข้าด้วยกัน และส่งค่านั้นต่อไปแสดงผลยัง Crystal Report

11) ฟังก์ชันค้นหาข้อมูล

ค้นหาข้อมูลโดยใช้ค่าที่ได้รับจากผู้ใช้ ค่าที่รับมาคือค่าที่ผู้ใช้ต้องการแสดงใน
รายงาน

3.2.8.2 ส่วนติดต่อกับผู้ใช้

ในส่วนนี้สามารถแบ่งได้ 2 ส่วนคือ

1) ส่วนรับค่าจากผู้ใช้

ส่วนรับค่าจากผู้ใช้ทำโดยใช้ Window Form ในการสร้าง และใช้กล่อง
อุปกรณ์ เป็นตัวช่วยสร้าง ซึ่งมีอุปกรณ์ที่ใช้ดังนี้

1. 1) Label

สำหรับเขียนข้อความลงหน้าโปรแกรม

1. 2) ปุ่ม

สำหรับให้ผู้ใช้กดเพื่อกำหนดคำสั่งต่าง ๆ

1. 3) List Box

ใช้แสดงค่าที่ต้องการให้ผู้ใช้เลือกและให้โปรแกรม ทำตามคำสั่ง

1. 4) กล่องรับข้อความ

ใช้รับข้อความจากผู้ใช้เพื่อทำตามคำสั่ง

1.5) Radio Button

สำหรับผู้ใช้เลือกทำคำสั่งใดคำสั่งหนึ่ง ซึ่งจะใช้ร่วมกับ

Label

1.6) Check Box

กล่องรับคำสั่งสำหรับผู้ใช้สามารถเลือกได้มากกว่า 1 อย่าง

1.7) DateTimePicker

อุปกรณ์สำหรับให้ผู้ใช้เลือก วัน เดือน ปี เวลา หรืออย่างใดอย่างหนึ่ง
ตามที่ ผู้ใช้ต้องการ

2) ส่วนหน้าแสดงผล

ส่วนแสดงผลต่อผู้ใช้ซึ่งมีการเชื่อมต่อกับ Window Form Crystal Report และ Database โดยมีอุปกรณ์ช่วยดังนี้

2.1) DataGridView

อุปกรณ์ใช้แสดงผลข้อมูลที่มีการเรียกมาจากฐานข้อมูล โดยมีการกำหนดโดยผู้ใช้

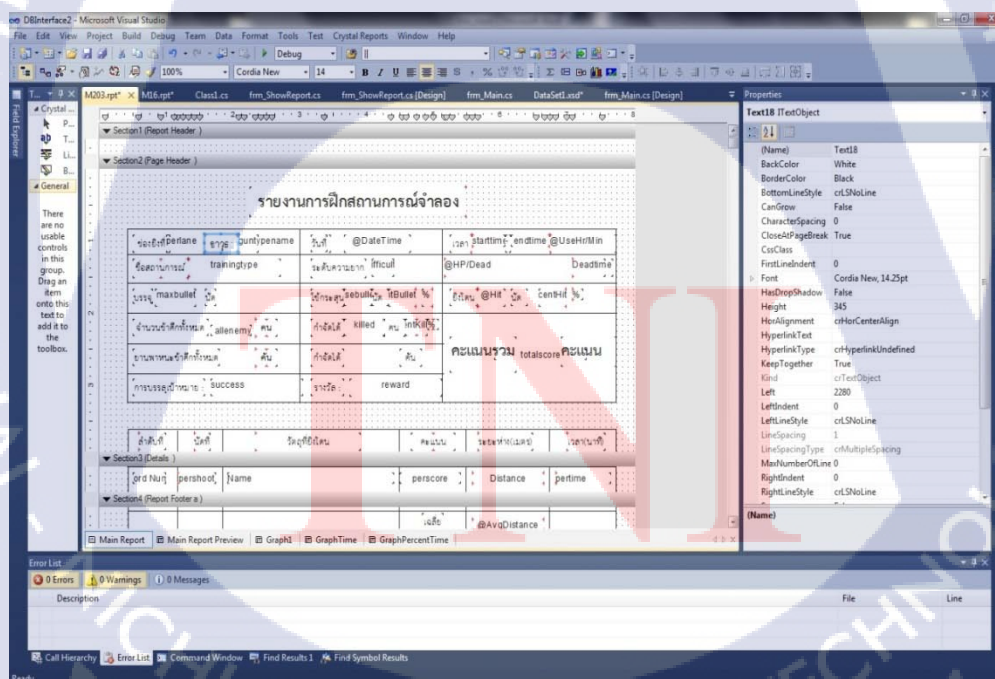
2.2) CrystalReportViewer

อุปกรณ์แสดงผลรายงานซึ่งมีการเชื่อมต่อกับโปรแกรม Crystal Report

3.2.9 การทำรายงานการฝึก

3.2.9.1 การวางโครงสร้างรายงาน

ในส่วนนี้จะมีการนำคำสั่งที่เราได้เขียนไปแล้วในขั้นตอนที่ 3.2.7 มาเชื่อมโยงกับรายงานที่เราได้ทำการออกแบบไว้โดยใช้เครื่องมือใน ขั้นตอนที่ 3.2.7.2 ทำการวางโครงสร้างของรายงานดังรูป 3.8



รูปที่ 3.8 การวางโครงสร้างรูปแบบรายงาน โดยใช้โปรแกรม Microsoft Visual Studio 2010

บทที่ 4

ผลการดำเนินงาน การวิเคราะห์

4.1 ขั้นตอนและผลดำเนินงาน

จากหัวข้อที่ 3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโครงการ (บทที่ 3) การดำเนินงานเป็นไปตามแผนการทำงานของระบบทุกอย่าง ปัญหาต่างๆที่เกิดบนระบบได้ทำการแก้ไขและทดสอบ โปรแกรมออกมาเป็นไปตามแผนงานและรูปแบบที่วางไว้ ข้อผิดพลาดต่างๆได้รับการแก้ไขภายในเวลาที่กำหนด

4.1.1 เมนู (Menu)

ผู้ใช้งานสามารถเลือกค้นหารายงานการฝึกย้อนหลัง โดยสามารถค้นหาจากข้อกำหนดต่าง ๆ ดังต่อไปนี้คือ วันที่ เดือน ปี ประเภทอาวุธ และประเภทการฝึกของการฝึก โดยผู้ใช้งานสามารถเลือกค้นหาจากข้อมูลอย่างใดอย่างหนึ่งก็ได้ มีหน้าตาการใช้งานดังรูป 4.1

Search Criteria

Gun Type:

Training Type:

Difficulty:

Lane

☒ Lane 1 ☒ Lane 2 ☒ Lane 3 ☒ Lane 4 ☒ Lane 5

☒ Lane 6 ☒ Lane 7 ☒ Lane 8 ☒ Lane 9 ☒ Lane 10

Select Training

Date	Time	Gun type	Training type
------	------	----------	---------------

จำนวนการฝึกทั้งหมด : 0 ครั้ง

Date and Time

☒ Date:

☐ Month:

☐ Year:

☐ Period: Begin: End:

รูปที่ 4.1 หน้าจอเลือกรายงานการฝึก

4.1.2 หน้าจอแสดงผลการค้นหา

หน้าจอแสดงผลการค้นหานำรายการของรายงานให้ผู้ใช้งานได้เลือก มีหน้าตาการใช้งานดังรูป 4.2

No.	Date	Time	Gun Type	Training	Difficulty	Lane	Training Number
91	23-Sep-13	11:12:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7
92	23-Sep-13	11:30:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7
93	23-Sep-13	11:30:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7
94	23-Sep-13	11:42:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7
95	23-Sep-13	11:42:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7
96	23-Sep-13	11:42:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7
97	23-Sep-13	12:02:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7
98	23-Sep-13	12:02:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7
99	23-Sep-13	12:12:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7
100	23-Sep-13	12:32:00	เอ็ม16	การยิงเป้าสัมฤ...	A	1	7

รูปที่ 4.2 หน้าจอแสดงผลการค้นหา

4.1.3 รายงานการฝึก

หลังจากที่ผู้ใช้งานได้เลือกค้นหาการฝึกแล้ว มีรูปแบบการแสดงผลดังรูป

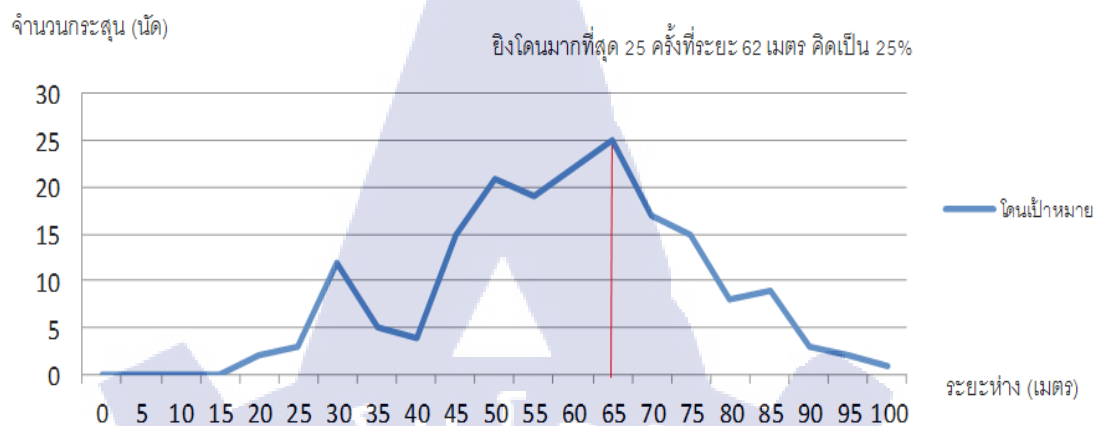
รายงานการฝึกสถานการณ์จำลอง

ชื่องานที่ 1 อาวุธ : ปืนเอ็ม 16 เอ 1	วันที่ 17/07/2556	เวลา 10.30 – 12.00 น. (1 ชั่วโมง 30 นาที)
ชื่อสถานการณ์ AAA	ระดับความยาก B	เวลาที่ HP เหลือ 0 (ตาย) 5 นาที
บรรจุ 5 ของกระสุน ของละ 10 นัด	ใช้กระสุน 10 นัด (20%)	ยิงโดน 5 นัด (10%)
จำนวนข้าศึกทั้งหมด 100 คน	กำจัดได้ 5 คน (5%)	คะแนน รวม 77 คะแนน
ยานพาหนะข้าศึกทั้งหมด 10 คัน	กำจัดได้ 0 คัน (0%)	
การบรรจุเป้าหมาย : สำเร็จ/ล้มเหลว	รางวัล : DEAD FIRST	

ลำดับที่	นัดที่	วัตถุที่ยิงโดน	คะแนน	ระยะทาง(เมตร)	เวลา(นาที)
1	5	แขนซ้ายบน (ข้าศึก A1)	13	2.70	2.10
2	8	ต้นขาซ้าย (ข้าศึก A1)	12	2.55	3.00
3	12	เชิงกราน (ข้าศึก A1)	15	1.50	5.00
4	17	ต้นขาซ้าย (ข้าศึก A8)	12	2.70	6.00
5	20	ศีรษะ (ข้าศึก B12)	4	4.20	8.00
6	25	ส่วนล่างขวา (ข้าศึก B5)	21	1.20	12.00
		เฉลี่ย		2.48	1.60

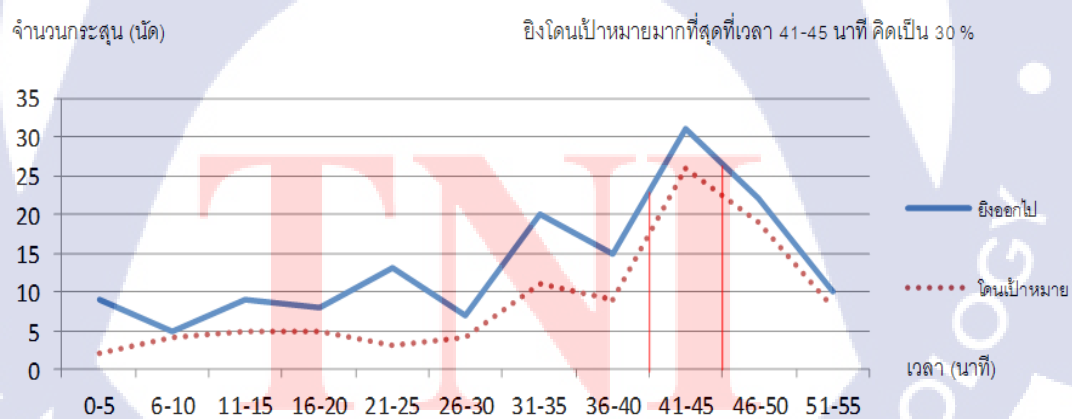
รูปที่ 4.3 การแสดงผลรายงานการฝึก

กราฟผลการยิงตามระยะเป้าหมาย



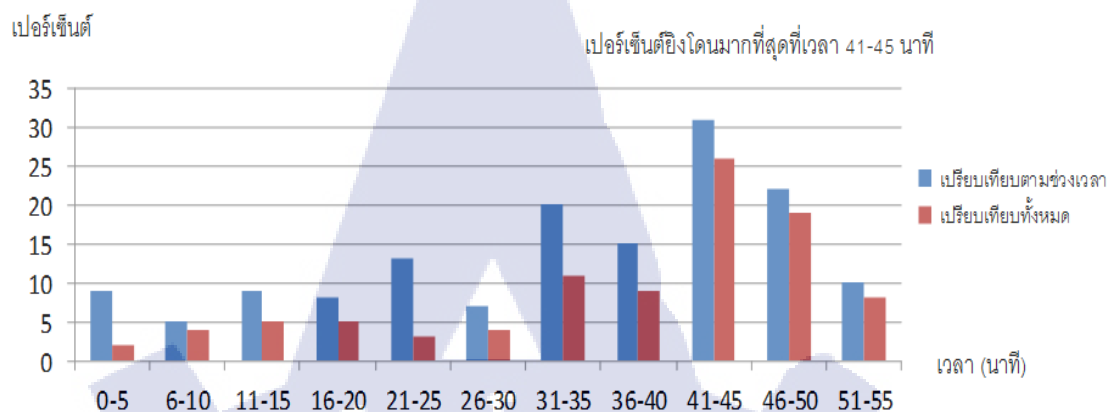
รูปที่ 4.4 การแสดงผลรายงานการฝึกส่วนกราฟ 1

กราฟผลการยิงตามช่วงเวลา



รูปที่ 4.5 การแสดงผลรายงานการฝึกส่วนกราฟ 2

เปอร์เซ็นต์การยิงโดนเป้าหมายตามช่วงเวลา



รูปที่ 4.4 การแสดงผลรายงานการฝึกส่วนกราฟ 3

4.2 ผลการวิเคราะห์ข้อมูล

การพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว มีการวิเคราะห์ข้อมูลจากที่ผู้ดำเนินงานได้ศึกษาข้อมูลทางทั่วไปและข้อมูลทางเทคนิคของอาวุธปืนนั้น ๆ เพื่อนำมาพัฒนา การออกแบบรูปแบบรายงานให้มีความถูกต้องและตรงกับความต้องการของผู้ใช้อย่างสมบูรณ์

4.3 วิเคราะห์และวิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์ และจุดมุ่งหมายในการปฏิบัติงานหรือการจัดทำโครงการ

ในการสหกิจศึกษาครั้งนี้ การพัฒนาโปรแกรมรายงานระบบจำลองการฝึกยิงอาวุธปืนเล็กยาว ที่ได้รับมอบหมายได้มีการปฏิบัติงานตามแผนที่กำหนดครบถ้วนทุกขั้นตอน ผลงานเมื่อทำเสร็จแล้วได้วิเคราะห์เปรียบเทียบผลที่ได้รับ กับวัตถุประสงค์ที่ตั้งไว้ ซึ่งผลที่ได้คือผลงานตรงกับวัตถุประสงค์ที่ได้กำหนดไว้ในตอนแรก โดยการปฏิบัติในแต่ละขั้นตอนจะมีการรายงานผลให้กับผู้ดูแลตลอดระยะการทำงาน เพื่อตรวจสอบความถูกต้องของงานอย่างสม่ำเสมอ และในการพัฒนาระบบรายงานในครั้งนี้นับว่าบริษัทสามารถที่จะนำระบบไปพัฒนาต่อได้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการดำเนินงานมานั้น ทำให้ได้รู้ถึงรายละเอียด และกระบวนการทำงานของการพัฒนาเว็บแอปพลิเคชัน หรือการทำโปรแกรมต่างๆ ว่ามีกระบวนการอย่างไร กล่าวคือ ได้เรียนรู้ในเรื่องของวงจรชีวิตการพัฒนาโปรแกรม และเพิ่มทักษะการเขียนโปรแกรมภาษา C# ซึ่งนอกจากจะได้รับความรู้เกี่ยวกับการพัฒนาระบบการทำงานของเว็บแอปพลิเคชันในรูปแบบของการนำไปใช้งานจริง

นอกเหนือจากที่ได้รับความรู้จากการปฏิบัติสหกิจศึกษาตลอดสี่เดือนที่ผ่านมา ยังได้รับประสบการณ์การทำงานจริง ซึ่งได้รับความเอาใจใส่จากพนักงานภายในแผนกตลอดระยะเวลาปฏิบัติสหกิจ อีกทั้งยังได้รับความรู้นอกเหนือจากความรู้ที่ใช้ในโครงการ และคำแนะนำต่างๆ มากสำหรับนำไปประกอบการเลือกสาขาอาชีพในอนาคต รวมถึงการมีปฏิสัมพันธ์กับเพื่อนร่วมงาน ยังไปถึงการวางตัวในที่ทำงานให้เหมาะสม เพื่อเสริมบุคลิกภาพให้น่าเชื่อถือ จากการปฏิบัติสหกิจศึกษานี้ ทำให้ได้ทราบถึงเรื่องของความสำคัญของการสื่อสาร การทำงานเป็นทีม และความรับผิดชอบในการทำงานที่เป็นหน้าที่ ที่ต้องรับผิดชอบโดยตรง

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

5.2 แนวทางการแก้ไขปัญหา

จากการจัดทำโครงการระบบจัดซื้อระบบใหม่เมื่อเกิดปัญหาข้าพเจ้าจะค้นหาวิธีแก้ไขทางอินเทอร์เน็ต หนังสือที่เกี่ยวกับปัญหาที่เกิดขึ้น ขอความช่วยเหลือจากพี่ๆในบริษัทและเพื่อนร่วมงาน ซึ่งได้ทำให้ได้รับคำแนะนำและสามารถแก้ไขปัญหาได้เป็นอย่างดี โดยส่วนใหญ่แล้วข้าพเจ้าจะลองแก้ปัญหาด้วยตนเองก่อนเสมอเพื่อเป็นการฝึกตัวเองไปในตัวแต่หากไม่ได้จริงๆจึงจะทำวิธีการอย่างที่ได้กล่าวมา

5.3 ข้อเสนอแนะจากการดำเนินงาน

- การเริ่มต้นการทำงานต่าง ๆ เราควรที่จะศึกษาข้อมูลนั้นให้เข้าอย่างแนชัด เพื่อที่จะพัฒนางานที่เรากำลังทำอยู่นั้นได้อย่างถูกต้อง
- ควรวางแผนการทำงานให้ดี
- การทำงานควรที่จะทำงานให้ตรงตามแผนการทำงานที่กำหนดไว้แล้ว
- เมื่อมีปัญหาควรปรึกษาพี่ที่ดูแลทันที

เอกสารอ้างอิง

1. วิกิพีเดีย สารานุกรมเสรี, 2555, **เอเอสพีดอตเน็ต** [ออนไลน์], เข้าถึงได้จาก:
<http://th.wikipedia.org/wiki/ASP.NET> [2555, 19 มิถุนายน 2556].
2. เว็บบมอนสเตอร์, 2555, **ASP.NET คืออะไร มีที่มาและความสำคัญอย่างไร** [ออนไลน์],
เข้าถึงได้จาก : <http://www.sapaan.net/forum/programming-community/asp-net-xiidaa-aooeaoaadcoaeoniaeoaa/> [2555, 20 มิถุนายน 2556]
3. ศุภชัย สมพานิช, 2553, **Basic ASP.NET 4.0**, พิมพ์ครั้งที่ 1, ไอดีซี อินโฟ ดิสทริบิวเตอร์
เซ็นเตอร์, ธงศ์ ปิยอนงค์ บุ๊คส์, บจก., หน้า xxx-xxx
4. วิกิพีเดีย สารานุกรมเสรี, 2554, **Entity-Relationship model** [ออนไลน์], เข้าถึงได้จาก:
http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model [2555, 20 มิถุนายน 2556]
5. รัชชัย เอี่ยมไพโรจน์, 2552, **วิธีใช้ postgresql** [ออนไลน์], เข้าถึงได้จาก:
<http://www.scribd.com/doc/16810741/PostgreSQL-Manual-> [2552, 30 มิถุนายน 2556]
6. Astah, **Astah Tutorials Tips** [ออนไลน์], เข้าถึงได้จาก: <http://astah.net/videos/class-diagram> [2555, 1 กรกฎาคม 2556]
7. ไมโครซอฟท์, 2551, **How to connect to database C#** [ออนไลน์], เข้าถึงได้จาก :
[http://msdn.microsoft.com/en-us/library/bb655884\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/bb655884(v=vs.90).aspx) [2551, 10 กรกฎาคม 2556]

8. ไมโครซอฟต์, 2549 , **Array Tutorials** [ออนไลน์], เข้าถึงได้จาก :
[http://msdn.microsoft.com/en-us/library/aa288453\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa288453(v=vs.71).aspx) [2549, 12 กรกฎาคม 2556]
9. ไมโครซอฟต์, 2552 , **String. Join Method(String, String[])** [ออนไลน์], เข้าถึงได้จาก :
<http://msdn.microsoft.com/en-us/library/57a79xd0.aspx> [2552, 14 กรกฎาคม 2556]
10. ไมโครซอฟต์, 2553 , **Threading Tutorial** [ออนไลน์], เข้าถึงได้จาก :
[http://msdn.microsoft.com/en-us/library/aa645740\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa645740(v=vs.71).aspx) [2553, 16 กรกฎาคม 2556]
11. ไมโครซอฟต์, 2553 , **Network Stream.Write Method** [ออนไลน์], เข้าถึงได้จาก :
<http://msdn.microsoft.com/en-us/library/system.net.sockets.networkstream.write.aspx> [2553, 17 กรกฎาคม 2556]
12. ไมโครซอฟต์, 2553 , **System.net NameSpace** [ออนไลน์], เข้าถึงได้จาก :
<http://msdn.microsoft.com/en-us/library/gg145039.aspx> [2553, 22 กรกฎาคม 2556]
13. ไมโครซอฟต์, 2553 , **System.net Namespace** [ออนไลน์], เข้าถึงได้จาก :
<http://msdn.microsoft.com/en-us/library/gg145039.aspx> [2553, 22 กรกฎาคม 2556]
14. ไมโครซอฟต์, 2553 , **System.net.Socket Namespace** [ออนไลน์], เข้าถึงได้จาก :
<http://msdn.microsoft.com/en-us/library/system.net.sockets.aspx> [2553, 24 กรกฎาคม 2556]
15. dotnetdev, 2552 , **Join two list together** [ออนไลน์], เข้าถึงได้จาก :
<http://stackoverflow.com/questions/1528171/joining-two-lists-together> [2553, 26 กรกฎาคม 2556]

16. dotnetdev, 2552 , **Concatenate a constant string to each item in a List<string> using LINQ** [ออนไลน์], เข้าถึงได้จาก :
<http://stackoverflow.com/questions/13006411/concatenate-a-constant-string-to-each-item-in-a-liststring-using-linq> [2553,30 กรกฎาคม 2556]
17. Sachin, 2552 , **Convert String to Date time** [ออนไลน์], เข้าถึงได้จาก :
<http://social.msdn.microsoft.com/Forums/en-US/af4f5a1e-f81d-47fe-981d-818e785b8847/convert-string-to-datetime-object> [2553,1 สิงหาคม 2556]
18. Ryan 2556 , **2 line chart in Crystal Report** [ออนไลน์], เข้าถึงได้จาก :
<http://stackoverflow.com/questions/17953409/how-do-i-achieve-a-multi-series-line-chart-in-crystal-reports> [2553,15 สิงหาคม 2556]
19. ThaiCreate 2553 , **Crystal Report บน Visual Studio 2010**[ออนไลน์], เข้าถึงได้จาก :
<http://www.thaicreate.com/dotnet/crystal-report-vb-net-cs.html> [2553,20 สิงหาคม 2556]

ภาคผนวก

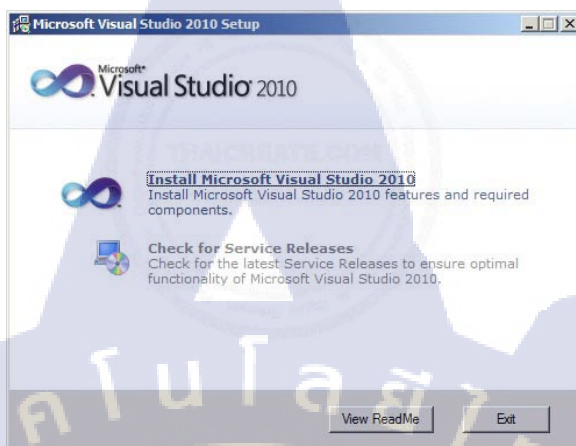
TNI

ภาคผนวก ก.

TNI

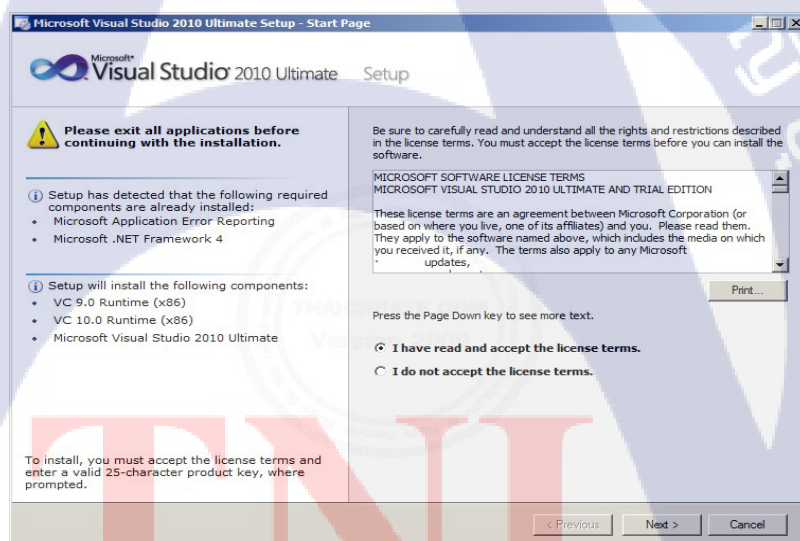
วิธีติดตั้งโปรแกรม Microsoft Visual Studio 2010

1. หน้าจอของการ Setup ให้คลิกที่ Install Microsoft Visual Studio 2010



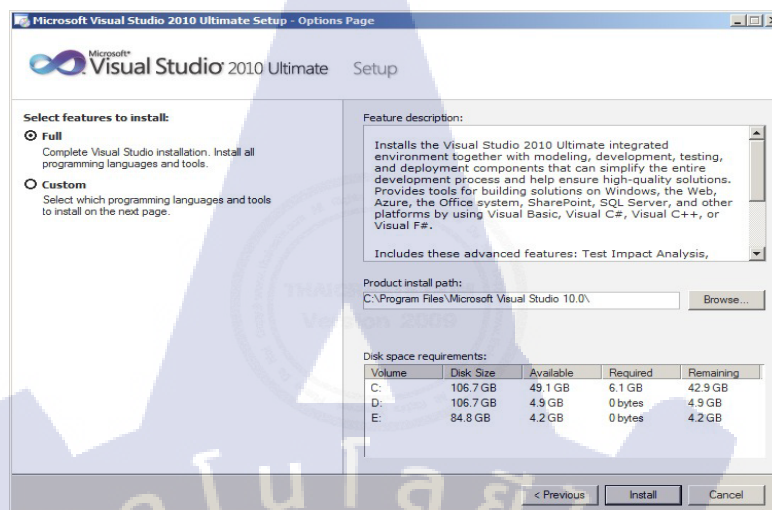
รูปที่ 1 ก หน้าจอการติดตั้ง Visual Studio 2010

2. Click Next



รูปที่ 2 ก ตอบรับข้อตกลงการลงโปรแกรม Visual Studio 2010

3. Click Next เลือกฟีเจอร์ในการติดตั้งในที่นี้ให้เลือก Full

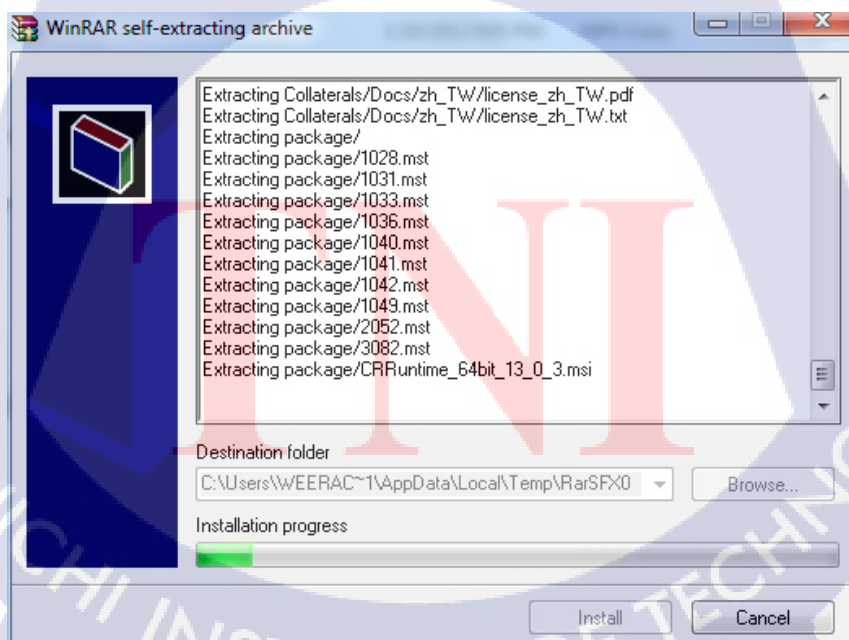


รูปที่ 3 ก รูปแบบการลงโปรแกรม Visual Studio 2010

4. ติดตั้งสมบูร์น

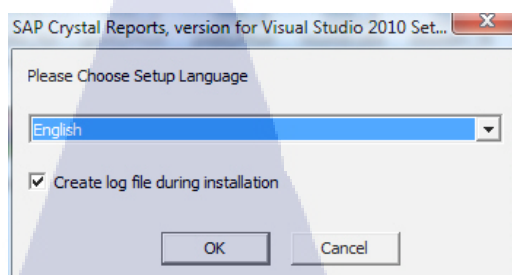
วิธีการติดตั้ง Crystal Report Plugin in Microsoft Visual Studio 2010

1. หลังจากที่ได้ไฟล์แล้วให้ดับเบิลคลิกที่ Package เพื่อ Install
2. รอสักครู่



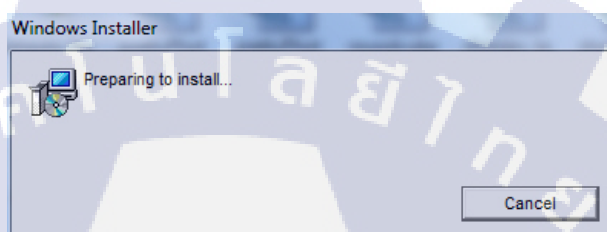
รูปที่ 4 ก รอติดตั้ง Plugin crystal report

3. เลือกภาษา



รูปที่ 5 ก เลือกภาษา

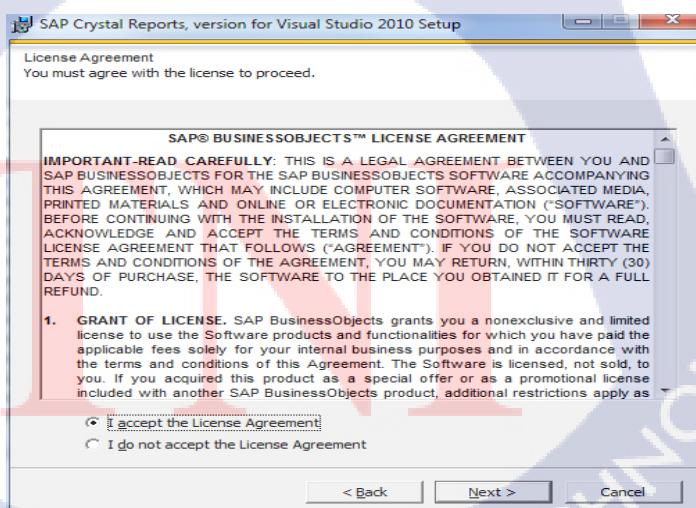
4. กำลัง Install



รูปที่ 6 ก กำลังติดตั้ง

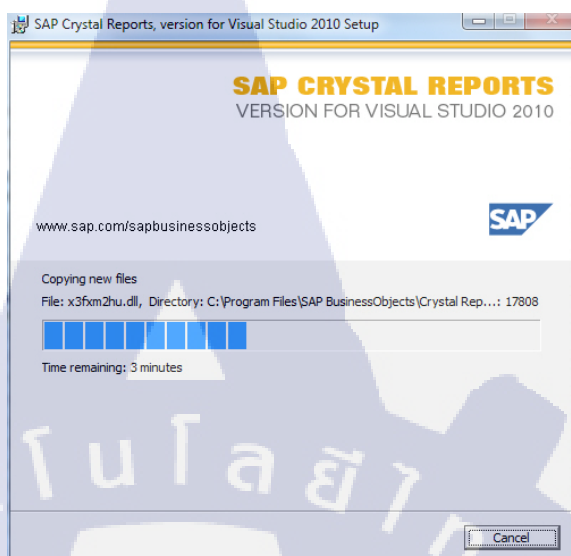
5. คลิกที่ Next เพื่อไปยังขั้นตอนถัดไป

6. คลิกเลือกที่ I accept the License Agreement และ Next เพื่อไปยังขั้นตอนถัดไป



รูปที่ 7 ก ตอบรับข้อตกลงการลงโปรแกรม crystal report

7. กำลังติดตั้งซึ่งจะใช้เวลาประมาณ 15-30 นาที

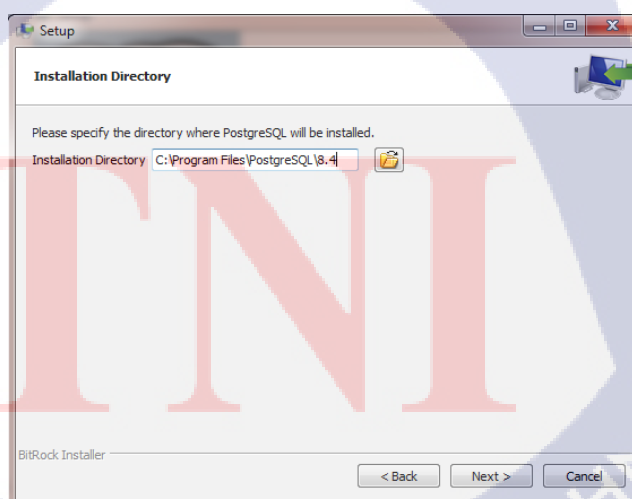


รูปที่ 8 กรอการติดตั้งโปรแกรม crystal report

8. การติดตั้งสมบูรณ์

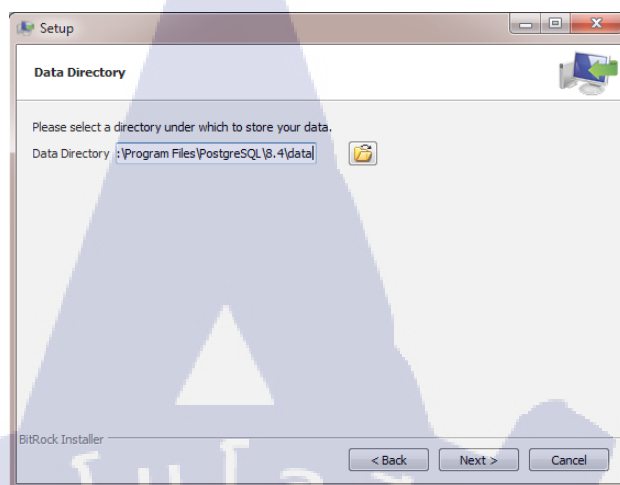
วิธีการติดตั้ง PostgreSQL

1. ลงโปรแกรม postgresql-8.4.16-2-windows.exe
2. เลือก Installation Directory เป็นไปตามที่ default แล้วกด Next



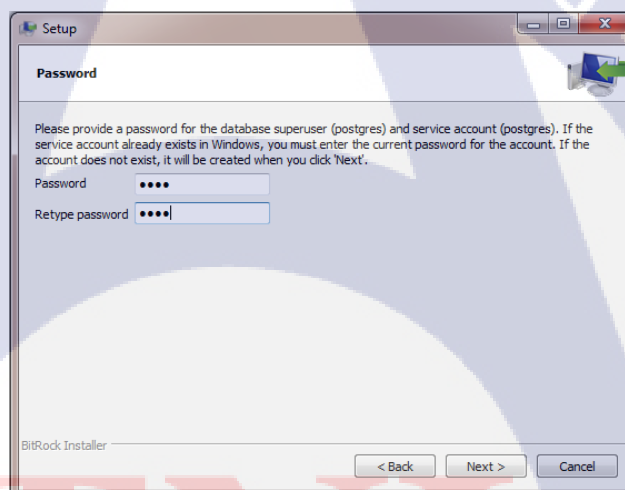
รูปที่ 9 ก เลือกที่ติดตั้งโปรแกรม postgresQL

3. เลือก Data Directory เป็นไปตามที่ default ไว้แล้วกด Next



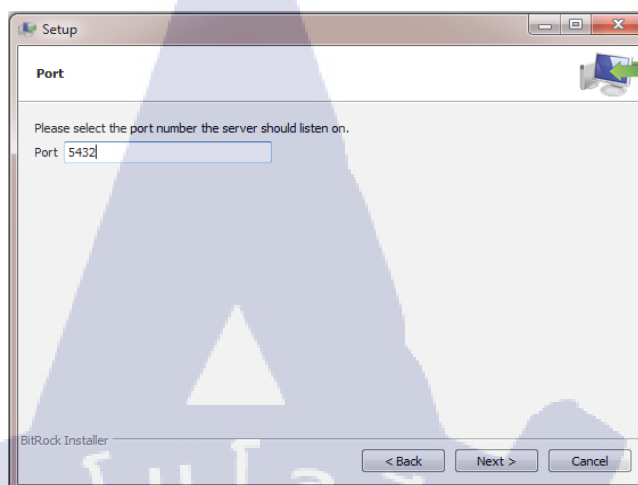
รูปที่ 10 ก เลือกที่เก็บข้อมูล

4. กำหนด password เป็น 1234



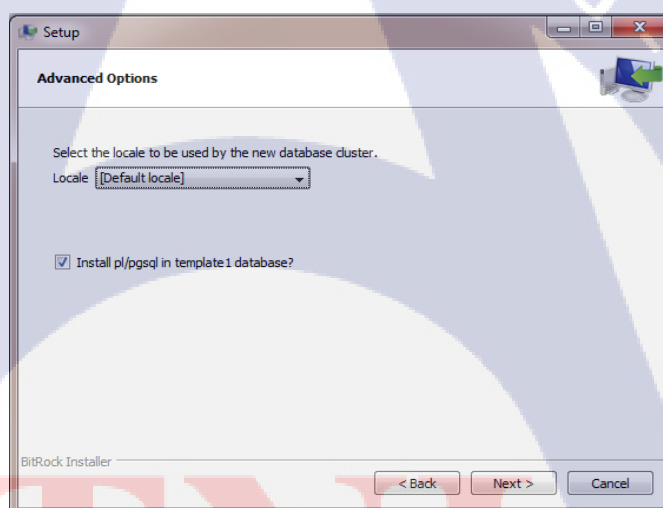
รูปที่ 11 ก ตั้งรหัสผ่านการเข้าดาต้าเบส

5. ค่า Port ตั้งต้นเป็น 5432



รูปที่ 12 ก ตั้ง port

6. เลือก Default locale แล้วกด Next

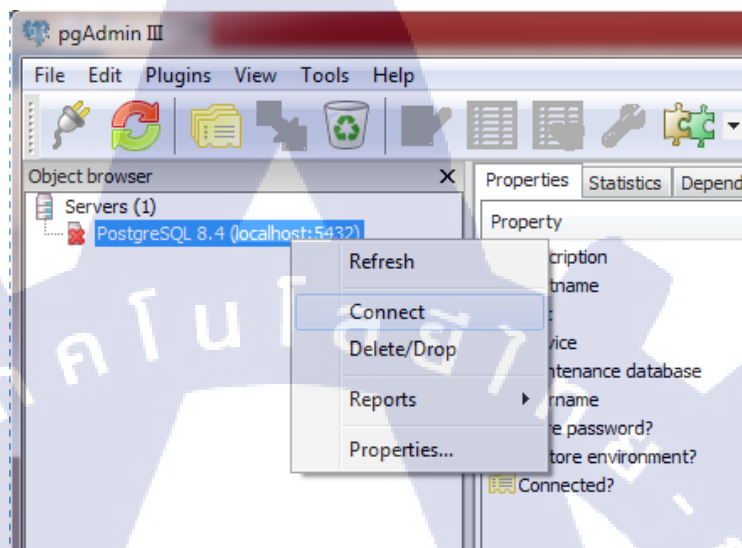


รูปที่ 13 ก เลือก Default locale

7. รอให้โปรแกรมทำงานสักครู่ ก็เสร็จสิ้นการติดตั้ง

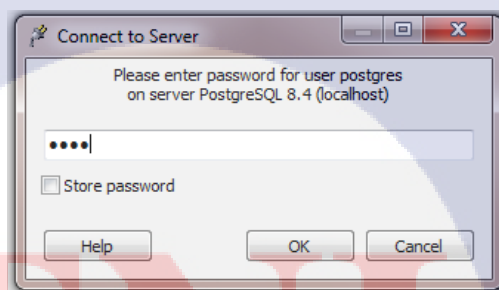
สร้าง database

1. เปิดโปรแกรม pgAdmin III
2. คลิกขวาที่ PostgreSQL 8.4 (localhost:5432) ภายใต้ Servers แล้วเลือก Connect



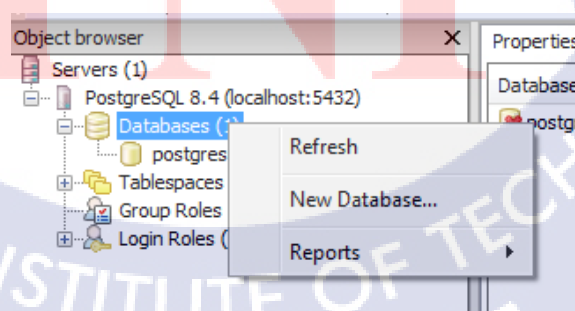
รูปที่ 14 กด หน้า Connect

3. ใส่ password = 1234 แล้วกด OK



รูปที่ 15 กด เข้าสู่ผ่าน

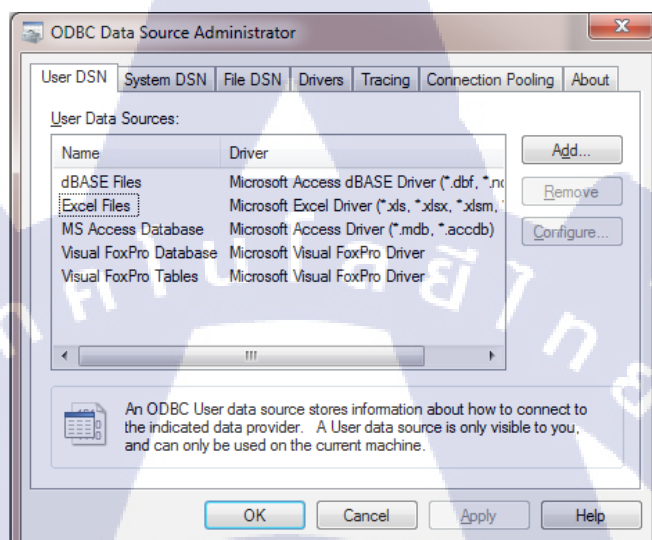
4. คลิกขวาที่ Databases แล้วเลือก New Database และตั้งชื่อ



รูปที่ 16 กด สร้างดาต้าเบสใหม่

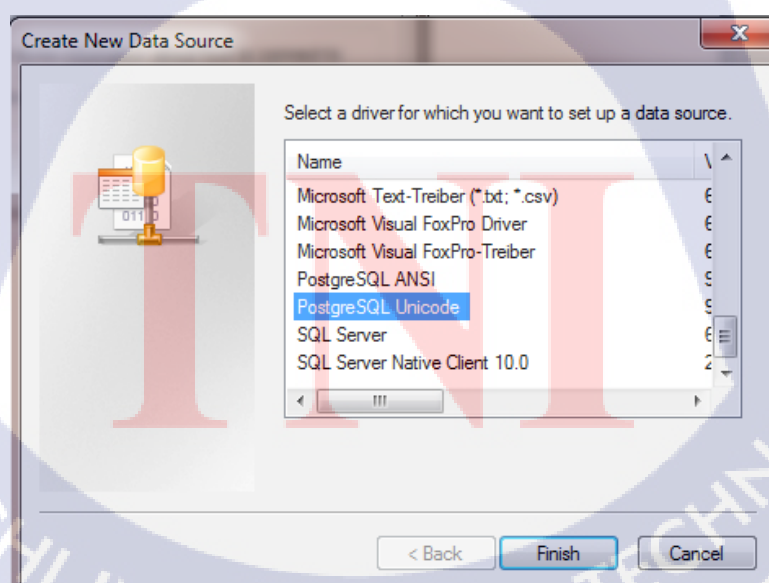
ติดตั้ง ODBC for PostgreSQL

1. ติดตั้ง psqlodbc.exe
2. กด Next ในทุกๆหน้าก็เสร็จสิ้น
3. ไปที่ Control Panel → Administrative Tools → ODBC



รูปที่ 17 ก ติดตั้ง ODBC

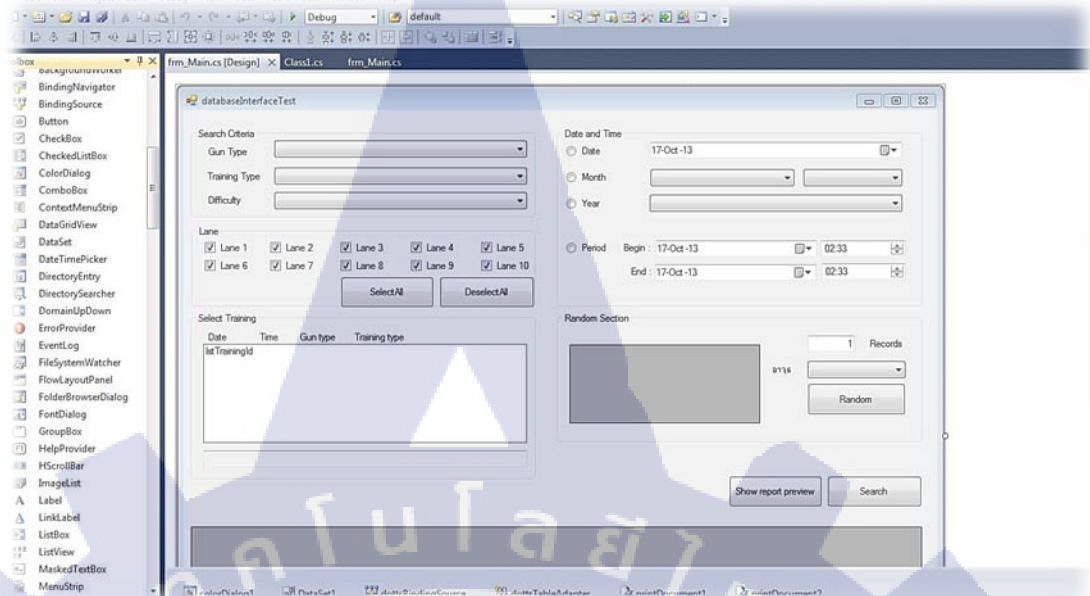
4. ในหน้า Create new Data Source เลือก PostgreSQL Unicode แล้วกดปุ่ม Finish



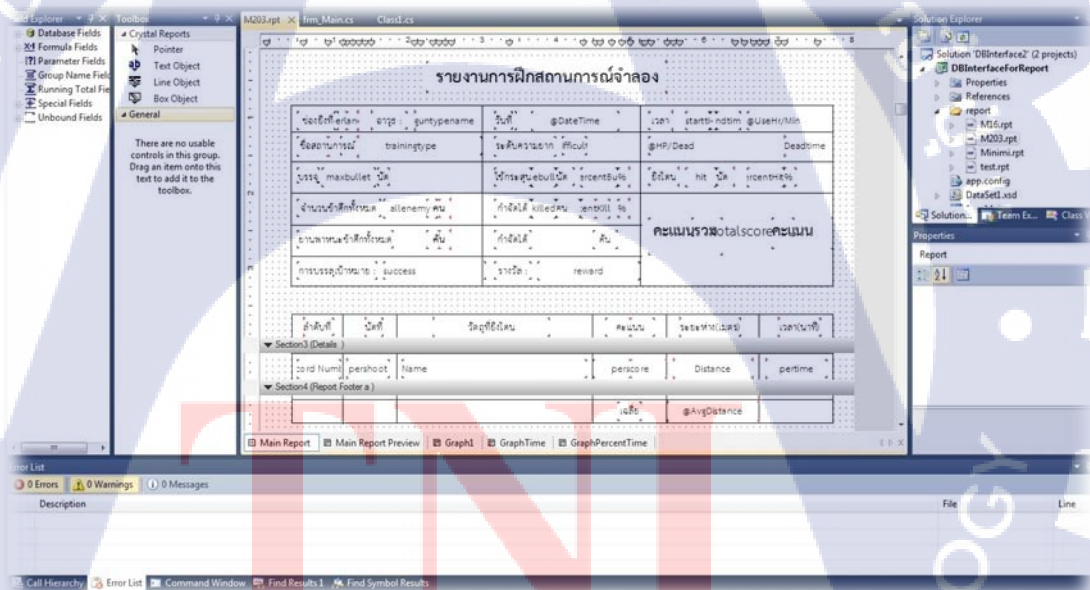
รูปที่ 18 ก เลือก Data Source ใหม่

ภาคผนวก ข.

TNI



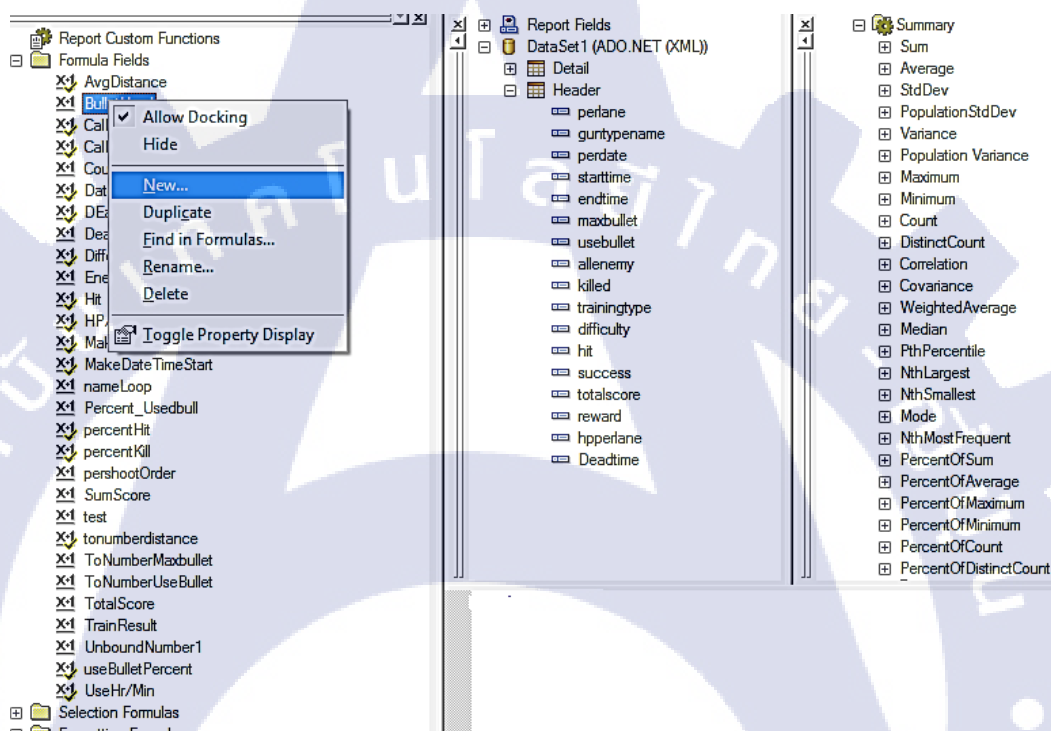
รูปที่ 1 ข หน้าจอ Interface



รูปที่ 2 ข หน้าจอการทำงาน

บรรจุ	นัด	ใช้กระสุน	นัด	@useBulletPercent %	ยิงโดน	นัด	%
จำนวนข้าศึกทั้งหมด	คน	กำจัดได้	คน		คะแนนรวม		
ยานพาหนะข้าศึกทั้งหมด	คัน	กำจัดได้	คัน				
การบรรจุเป้าหมาย :		รางวัล :					

รูปที่ 3 ข ส่วนรายงานส่วน Header



รูปที่ 4 ข ส่วนการทำ Formula สำหรับรายงาน

ภาคผนวก ก.

TNI

1. ส่วนที่ใช้เชื่อมต่อกับดาต้าเบส

```
const string connStr =
```

```
"DSN=xx;UID=postgres;PWD=abc*123;database=xx;server=127.0.0.1;port=5432;";
```

```
public OdbcConnection Connect()
```

```
{
```

```
    OdbcConnection conn = new OdbcConnection(connStr);
```

```
    if (conn.State != ConnectionState.Open)
```

```
        conn.Open();
```

```
    return conn;
```

```
}
```

```
Criteria crit = new Criteria();
```

2. ส่วนรับค่าคิวรีสำหรับทำคำสั่ง

```
public void ExecuteQuery(string sqlCommand)
```

```
{
```

```
    OdbcConnection conn = this.Connect();
```

```
    OdbcCommand executeQuery = new OdbcCommand(sqlCommand, conn);
```

```
    try
```

```
    {
```

```
        executeQuery.ExecuteNonQuery();
```

```
    }
```

```
    catch (Exception ex)
```

```
    {
```

```
        throw ex;
```

```
    }
```

```
    finally
```

```
    {
```

```
        conn.Close();
```

```
    }
```

```
}
```

3. ส่วนเพิ่มค่าตัวเทเบิล

```
public DataTable AddNewDataTable(string selectCommand)
{
    DataTable dt = new DataTable();
    OdbcConnection conn = this.Connect();
    OdbcDataAdapter da = new OdbcDataAdapter(selectCommand, conn);
    try
    {
        da.Fill(dt);
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        conn.Close();
    }
    return dt;
}
```

4. ส่วนการเพิ่มคำสั่งลงดาต้าเบสสำหรับตารางฝึก

```
public int SaveNewTrainInfo(TTrainInfo saveNewTrain)
{
    string training = "select max(" + TRAINING_TRAININGID + ") from" +
    TABLENAME_TRAINING;
    OdbcConnection conn = this.Connect();
    OdbcCommand cmdTrainingId = new OdbcCommand(training, conn);
    int trainingID = 0;
    try
```

```

{
    trainingID = (int)cmdTrainingId.ExecuteScalar();
    trainingID += 1;
    saveNewTrain.trainingId = trainingID;
}
catch
{
    trainingID = 1;
}
finally
{
    conn.Close();
}

string strSql = "INSERT INTO " + TABLENAME_TRAINING + " (" +
TRAINING_TRAININGID + "," + TRAINING_TRAININGTYPE + "," +
TRAINING_TRAININGDATE + "," + TRAINING_DIFFICULTY + "," +
TRAINING_GUNTYPEID + "," + TRAINING_BEGINTRAINING + "," +
TRAINING_ENDTRAINING + "," + TRAINING_ALLENEMY + ")";

strSql += " Values ( " + trainingID + "," + saveNewTrain.trainTypeID + "," +
saveNewTrain.dateTime.ToString("yyyy-MM-dd") + "," + saveNewTrain.level + "," +
saveNewTrain.gunTypeID + "," + saveNewTrain.dateTime.ToString("HH:mm:ss") + "," +
saveNewTrain.enddateTime.ToString("HH:mm:ss") + "," + saveNewTrain.allEnemy + ")";

ExecuteQuery(strSql);
return trainingID;
}

```

5. ส่วนการเพิ่มคำสั่งลงดาต้าเบสสำหรับตารางช่องยิง

```

public int SaveLane(TLaneInfo SaveLane, int trainingId)
{
    string Sql = "select max(" + LANE_LANENO + ") from " + TABLE_LANE;
    OdbcConnection conn = this.Connect();

```

```

OdbcCommand cmdSaveRecord = new OdbcCommand(Sql, conn);
int LaneNo = 0;
SaveLane.TrainingId = trainingId;
try
{
    LaneNo = (int)cmdSaveRecord.ExecuteScalar();
    LaneNo += 1;
    SaveLane.LaneNo = LaneNo;
}
catch
{
    LaneNo = 1;
}
finally
{
    conn.Close();
}
string strSql = "INSERT INTO " + TABLE_LANE + "(" + LANE_LANENO + "," +
LANE_HPPERLANE + "," + LANE_MAXBULLET + "," + LANE_USEBULLET + "," +
LANE_TOTALSCORE + "," + LANE_TRAININGID + ")";
strSql += "Values (" + SaveLane.LaneNo + "," + SaveLane.HPperLane + "," +
SaveLane.MaxBullet + "," + SaveLane.UseBullet + "," + SaveLane.TotalScore + "," +
SaveLane.TrainingId + ")";
ExecuteQuery(strSql);
return LaneNo;
}

```

6. ส่วนการเพิ่มค่าลงดาต้าเบสสำหรับตารางข้อมูลการยิง

```

public int SaveRecord(TRecordInfo SaveRecord)
{

```

```

string record = "select max(" + RECORDNUMBER + ") from " +
TABLENAME_DATA;

OdbcConnection conn = this.Connect();

OdbcCommand cmdSaveRecord = new OdbcCommand(record, conn);

int RecordNumber;

try
{
    RecordNumber = (int)cmdSaveRecord.ExecuteScalar();
    RecordNumber += 1;
}
catch
{
    RecordNumber = 1;
}

string strSql = "INSERT INTO " + TABLENAME_DATA + " (" +
RECORDNUMBER + "," + TRAININGID + "," + PERORDER + "," + PERLANE + "," +
DISTANCE + "," + PERDATE + "," + PERTIME + "," + PERSCORE + "," + PERSHOOT +
"," + PERTYPE + "," + TARGETID + "," + ENEMYID + ") ";

strSql += "Values ( " + RecordNumber + "," + SaveRecord.trainId + "," +
SaveRecord.perorder + "," + SaveRecord.perlane + "," + SaveRecord.Distance + "," +
SaveRecord.perdate + "," + SaveRecord.perTime + "," + SaveRecord.perscore + "," +
SaveRecord.pershoot + "," + SaveRecord.pertype + "," + SaveRecord.TargetId + "," +
SaveRecord.EnemyId + ")";

ExecuteQuery(strSql);
return RecordNumber;
}

```

7. การสร้างคำเ่เปิดส่วนครีที่เรีย

```

public DataTable QueryByCriteria(DottsCriteria crits)
{
    string sql = string.Empty;
    string whereclause = string.Empty;
    if (crits.GunTypeId != 0)
    {
        whereclause += " and t." + GUNTYPE_GUNTYPEID + " = " + crits.GunTypeId;
    }
    if (crits.TrainingTypeId != 0)
    {
        whereclause += " and t." + TRAINING_TRAININGTYPE + " = " +
crits.TrainingTypeId;
    }
    if (crits.Difficulty == 'A' || crits.Difficulty == 'B' || crits.Difficulty == 'C')
    {
        whereclause += " and t." + TRAINING_DIFFICULTY + " = '" + crits.Difficulty + "'";
    }
    switch (crits.TimeChoice)
    {
        case 1:
            whereclause += " AND " + PERDATE + " = '" +
DateTime.Parse(crits.Date).ToString("yyyy-MM-dd", CultureInfo.CreateSpecificCulture("en-
EN")) + "'";
            break;
        case 2:
            whereclause += " AND extract(month from " + PERDATE + ") = " + crits.Month;
            whereclause += " AND extract(year from " + PERDATE + ") = " +
DateTime.Now.Year;
            break;
    }
}

```

case 3:

```
whereclause += " AND extract(year from " + PERDATE + ")=" + crits.Year;
```

```
break;
```

case 4:

```
whereclause += " AND " + PERDATE + " || ' ' || " + PERTIME + " >=" +
```

```
DateTime.Parse(crits.DateStart).ToString("yyyy-MM-dd HH:mm:00") + "";
```

```
whereclause += " AND " + PERDATE + " || ' ' || " + PERTIME + " <=" +
```

```
DateTime.Parse(crits.DateEnd).ToString("yyyy-MM-dd HH:mm:59") + "";
```

```
break;
```

```
}
```

```
if (crits.ListLaneNo != null && crits.ListLaneNo.Count > 0)
```

```
{
```

```
string allLane = string.Empty;
```

```
foreach (int i in crits.ListLaneNo)
```

```
{
```

```
allLane += "," + i.ToString() + "";
```

```
}
```

```
whereclause += " AND " + PERLANE + " IN (" + allLane.Substring(1) + ")";
```

```
}
```

```
whereclause += " AND t." + TRAINING_TRAININGID + " = " + crits.TrainingId;
```

```
sql += "select " + ECORDNUMBER + " as \"No.\" , " + PERDATE + " as \"Date\" , " +
```

```
PERTIME + " as \"Time\" , " + GUNTYPE_GUNTYPENAME + " as \"Gun Type\" , " +
```

```
TRAININGTYPE_TRAININGNAME + " as \"Training\" , " + TRAINING_DIFFICULTY +
```

```
\" , " + PERLANE + " as \"Lane\" , d." + TRAININGID + " as \"Training Number\"";
```

```
sql += "from " + TABLENAME_DATA + " d ";
```

```
sql += "inner join " + TABLENAME_TRAINING + " t on d." + TRAININGID + " = t." +
```

```
+ TRAINING_TRAININGID + " ";
```

```
sql += "inner join " + TABLENAME_TRAININGTYPE + " tt on t." +
```

```
TRAINING_TRAININGTYPE + " = tt." + TRAININGTYPE_TRAININGTYPEID + " ";
```

```

        sql += "inner join " + TABLENAME_GUNTYPE + " g on t." +
TRAINING_GUNTYPEID + " = g." + GUNTYPE_GUNTYPEID + " ";
        sql += "where 1=1 " + whereclause + " order by \"No.\" ";
        DataTable dt = AddNewDataTable(sql);
        whereclause = string.Empty;
        return dt;
    }

    public string GetCriteriaTimeChoice(DottsCriteria criteria)
    {
        string timewhClause = " ";
        switch (criteria.TimeChoice)
        {
            case 0:
                timewhClause += " AND " + TRAINING_TRAININGDATE + "=" +
DateTime.Parse(criteria.Date).ToString("yyyy-MM-dd",
CultureInfo.CreateSpecificCulture("en-EN")) + """;
                break;
            case 1:
                timewhClause += " AND " + TRAINING_TRAININGDATE + "=" +
DateTime.Parse(criteria.Date).ToString("yyyy-MM-dd",
CultureInfo.CreateSpecificCulture("en-EN")) + """;
                break;
            case 2:
                timewhClause += " AND extract(month from " +
TRAINING_TRAININGDATE + " )=" + criteria.Month+ " AND extract(year from " +
TRAINING_TRAININGDATE + " )=" + criteria.Year;
                break;
            case 3:
                timewhClause += " AND extract(year from " + TRAINING_TRAININGDATE
+ " )=" + criteria.Year;

```

```

        break;
    case 4:
        timewhClause += " AND " + TRAINING_TRAININGDATE + ">=" +
DateTime.Parse(criteria.DateStart).ToString("yyyy-MM-dd HH:mm:00") + "";
        timewhClause += " AND " + TRAINING_TRAININGDATE + "<=" +
DateTime.Parse(criteria.DateEnd).ToString("yyyy-MM-dd HH:mm:59") + "";
        break;
    }
    return timewhClause;
}

public DataSet QueryDottsReportPerLane(int trainingId, string laneNo, string GunType)
{
    string sql = "select * from " + TABLENAME_DATA + " where " + PERLANE + "=" +
laneNo + " AND " + TRAININGID + " = " + trainingId + " order by " + RECORDNUMBER;
    OdbcConnection conn = this.Connect();
    DataSet ds = new DataSet();
    DataTable dt = new DataTable();
    DataTable dtnew = new DataTable();
    dt = AddNewDataTable(sql);
    dtnew = CreateReportDetailTable(dt, GunType);
    ds.Tables.Add(dtnew);
    dtnew = CreateReportHeaderTable(dt);
    ds.Tables.Add(dtnew);
    dtnew = GraphDistanceAndShootTable(dt);
    ds.Tables.Add(dtnew);
    dtnew = GraphShootAndTimeTable(dt);
    ds.Tables.Add(dtnew);
    dtnew = GraphTimePercentTable(dt);
    ds.Tables.Add(dtnew);
    return ds;
}

```

8. ส่วนสร้างดาต้าเทเบิลสำหรับส่งไปยังรายงาน

```
public DataTable CreateReportHeaderTable(DataTable dt)
{
    DataTable dtHeader = new DataTable();
    DataColumn dcperlane = new DataColumn("perlane", typeof(string));
    DataColumn dcguntypename = new DataColumn("guntypename", typeof(string));
    DataColumn dcperdate = new DataColumn("perdate", typeof(DateTime));
    DataColumn dcstarttime = new DataColumn("starttime", typeof(string));
    DataColumn dcendtime = new DataColumn("endtime", typeof(string));
    DataColumn dcmaxbullet = new DataColumn("maxbullet", typeof(int));
    DataColumn dcusebullet = new DataColumn("usebullet", typeof(int));
    DataColumn dcallenemy = new DataColumn("allenemy", typeof(int));
    DataColumn dckilled = new DataColumn("killed", typeof(int));
    DataColumn dctrainingtype = new DataColumn("trainingtype", typeof(string));
    DataColumn dcdifficulty = new DataColumn("difficulty", typeof(char));
    DataColumn dchit = new DataColumn("hit", typeof(int));
    DataColumn dctotalscore = new DataColumn("totalscore", typeof(int));
    DataColumn dsuccess = new DataColumn("success", typeof(string));
    DataColumn dcreward = new DataColumn("reward", typeof(string));
    DataColumn dchpperlane = new DataColumn("hpperlane", typeof(int));
    DataColumn dcdeadtime = new DataColumn("Deadtime", typeof(int));
    dtHeader.Columns.Add(dcperlane);
    dtHeader.Columns.Add(dcguntypename);
    dtHeader.Columns.Add(dcperdate);
    dtHeader.Columns.Add(dcstarttime);
    dtHeader.Columns.Add(dcendtime);
    dtHeader.Columns.Add(dcmaxbullet);
    dtHeader.Columns.Add(dcusebullet);
    dtHeader.Columns.Add(dcallenemy);
    dtHeader.Columns.Add(dckilled);
```

```

dtHeader.Columns.Add(dctrainingtype);
dtHeader.Columns.Add(dcdifficulty);
dtHeader.Columns.Add(dchit);
dtHeader.Columns.Add(dctotalscore);
dtHeader.Columns.Add(dcsuccess);
dtHeader.Columns.Add(dcreward);
dtHeader.Columns.Add(dchpperlane);
dtHeader.Columns.Add(dcdeadtime);
DataRow drHeader;
drHeader = dtHeader.NewRow();
drHeader[0] = dt.Rows[0]["perlane"];
drHeader[1] = queryForReportGunTypeName((int)dt.Rows[0]["TrainingId"]);
drHeader[2] = dt.Rows[0]["perdate"];
drHeader[3] =
queryForReportTrainingTime(BEGINTRAINING,(int)dt.Rows[0]["TrainingId"]);
drHeader[4] = queryForReportTrainingTime(TRAINING_ENDTRAINING,
(int)dt.Rows[0]["TrainingId"]);
drHeader[5] =
queryForReportGetBullet(LANE_MAXBULLET,(string)dt.Rows[0]["perlane"],(int)dt.Rows
[0]["TrainingId"]);
drHeader[6] =
queryForReportGetBullet(LANE_USEBULLET,(string)dt.Rows[0]["perlane"],
(int)dt.Rows[0]["TrainingId"]);
drHeader[7] = queryForReportAllEnemy((int)dt.Rows[0]["TrainingId"]);
drHeader[8] = countKill(dt);
drHeader[9] = queryForReportTrainingTypeName((int)dt.Rows[0]["TrainingId"]);
drHeader[10] = queryForReportDifficulty((int)dt.Rows[0]["TrainingId"]);
drHeader[11] = countHit(dt);
drHeader[12] = totalScore(dt);
drHeader[13] = queryForReportTrainingResult((int)dt.Rows[0]["TrainingId"]);

```

```

        drHeader[15] =
queryForReportGetBullet(LANE_HPPERLANE,(string)dt.Rows[0]["perlane"],
(int)dt.Rows[0]["TrainingId"]);

        drHeader[16] = Deadtme((int)dt.Rows[0]["TrainingId"],
(string)dt.Rows[0]["perlane"]);

        dtHeader.Rows.Add(drHeader);
        dtHeader.TableName = "Header";
        return dtHeader;
    }

    public DataTable CreateReportDetailTable(DataTable dt,string GunType)
    {
        DataTable dtDetail = new DataTable();
        DataColumn dcperlane = new DataColumn("perlane", typeof(string));
        DataColumn dcTrainingId = new DataColumn("TrainingId", typeof(int));
        DataColumn dcperdate = new DataColumn("perdate", typeof(DateTime));
        DataColumn dcpershoot = new DataColumn("pershoot", typeof(string));
        DataColumn dcDistance = new DataColumn("Distance", typeof(string));
        DataColumn dcvertime = new DataColumn("vertime", typeof(string));
        DataColumn dcperscore = new DataColumn("perscore", typeof(int));
        DataColumn dcName = new DataColumn("Name", typeof(string));
        DataColumn dcDistanceCal = new DataColumn("DistanceCal", typeof(string));
        dtDetail.Columns.Add(dcperlane);
        dtDetail.Columns.Add(dcTrainingId);
        dtDetail.Columns.Add(dcperdate);
        dtDetail.Columns.Add(dcpershoot);
        dtDetail.Columns.Add(dcDistance);
        dtDetail.Columns.Add(dcpertime);
        dtDetail.Columns.Add(dcperscore);
        dtDetail.Columns.Add(dcName);
        dtDetail.Columns.Add(dcDistanceCal);
    }

```

```

if (GunType == "ปืนเล็กยาว ")
{
    Gun(dt, dtDetail);
}
else if (GunType == "กลปาว ")
{
    Machine(dt, dtDetail);
}
else if (GunType == "เครื่องยิงลูกระเบิด ")
{
    Boom(dt, dtDetail);
}
dtDetail.TableName = "Detail";
return dtDetail;
}

```

9. ส่วนการคำนวณที่เรียกใช้จากดาต้าเบส

```

public string calculateShotMinutes(int DetailTime)
{
    string sql = "select \"BeginTraining\" from \"Training\" where \"TrainingId\" = \" +
    KeepTrainId;
    DataTable dt = AddNewDataTable(sql);
    DateTime moment = Convert.ToDateTime(dt.Rows[0][0].ToString());
    int hr = moment.Hour;
    int min = moment.Minute;
    int sec = moment.Second;
    int timeCal = (((hr * 60) + min) * 60) + sec;
    string timeReturn = "";
    DetailTime -= timeCal;
    TimeSpan t = TimeSpan.FromSeconds(DetailTime);
}

```

```

        timeReturn = t.ToString(@"hh\mm\ss");
        return timeReturn;
    }

    public string queryForName(int targetid, string pershoot)
    {
        string sql = "select " + TARGET_NAME + " from " + TABLENAME_DATA + " d inner
join " + TABLENAME_TARGET + " t on d." + TARGETID + " = t." + TARGET_TARGETID;
        sql += "where t." + TARGET_TARGETID + " = " + targetid + " AND " + PERSHOOT +
"=" + pershoot.ToString() + """";
        DataTable dt = AddNewDataTable(sql);
        return dt.Rows[0][0].ToString();
    }

    public string queryForReportGunTypeName(int TrainingId)
    {
        string sql = "select" + GUNTYPE_GUNTYPENAME + " from " +
TABLENAME_TRAINING + " t inner join " + TABLENAME_GUNTYPE + " a on t." +
TRAINING_GUNTYPEID + " = a." + GUNTYPE_GUNTYPEID + " where " +
TRAINING_TRAININGID + " = " + TrainingId;
        DataTable dt = AddNewDataTable(sql);
        return dt.Rows[0][0].ToString();
    }

    public string queryForReportTrainingTime(string selectField,int TrainingId)
    {
        string sql = "select" + selectField + " from " + TABLENAME_TRAINING + " where " +
TRAININGID + " = " + TrainingId;
        DataTable dt = AddNewDataTable(sql);
        DateTime returnTime = Convert.ToDateTime(dt.Rows[0][0].ToString());
        return returnTime.Hour + ":" + returnTime.Minute;
    }
}

```

```

public int queryForReportGetBullet(string selectField,string LaneNo,int TrainingId)
{
    string sql = "select " + selectField + " from " + TABLE_LANE + " where " +
LANE_LANENO + " = " + LaneNo + " and " + TRAINING_TRAININGID + " = " +
TrainingId;

    DataTable dt = AddNewDataTable(sql);
    return Convert.ToInt32(dt.Rows[0][0]);
}

public int queryForReportAllEnemy(int TrainingId)
{
    string sql = "select" + TRAINING_ALLENEMY + " from " +
TABLENAME_TRAINING + " where " + TRAININGID + " = " + TrainingId;

    DataTable dt = AddNewDataTable(sql);
    return Convert.ToInt32(dt.Rows[0][0]);
}

public int countKill(DataTable dt)
{
    int kill = 0;
    for (int i = 0; i < dt.Rows.Count; i++)
    {
        if (dt.Rows[i]["perscore"] != "0")
        {
            kill++;
        }
    }
    return kill;
}

public string queryForReportTrainingTypeName(int TrainingId)
{

```

```

        string sql = "select" + TRAININGTYPE_TRAININGNAME + " from " +
        TABLENAME_TRAINING + " t inner join " + TABLENAME_TRAININGTYPE + " a on
        t." + TRAINING_TRAININGTYPE + " = a." + TRAININGTYPE_TRAININGTYPEID + " where " +
        TRAININGID + " = " + TrainingId;

        DataTable dt = AddNewDataTable(sql);
        return dt.Rows[0][0].ToString();
    }

    public string queryForReportDifficulty(int TrainingId)
    {
        string sql = "select" + TRAINING_DIFFICULTY + " from " +
        TABLENAME_TRAINING + " where " + TRAININGID + " = " + TrainingId;

        DataTable dt = AddNewDataTable(sql);
        return dt.Rows[0][0].ToString();
    }

    public int countHit(DataTable dt)
    {
        int hit = 0;
        string check = "0";
        for (int i = 0; i < dt.Rows.Count; i++)
        {
            if ((string)dt.Rows[i]["pershoot"] != check.ToString())
            {
                if ((int)dt.Rows[i]["perscore"] != 0)
                {
                    hit++;
                }
                check = (string)dt.Rows[i]["pershoot"];
            }
        }
        return hit;
    }

```

```

}

public int totalScore(DataTable dt)
{
    int sum = 0;
    for (int i = 0; i < dt.Rows.Count; i++)
    {
        sum += (int)dt.Rows[i]["perscore"];
    }
    return sum;
}

public int Deadtime(int TrainingId,string Lane)
{
    string sql = "select " + LANE_DEADTIME + "from " + TABLE_LANE + " where "
    "+LANE_TRAININGID"+"="+TrainingId+" AND "+ LANE_LANENO+" = " + Lane +"";
    string sqlBegin = "select \"BeginTraining\" from \"Training\" where \"TrainingId\" = " +
    TrainingId;

    int hr = 0, hrBegin = 0, min = 0, minBegin = 0;
    DataTable dt = AddNewDataTable(sql);
    if (dt.Rows[0][0].ToString() != "")
    {
        DateTime deadTime = Convert.ToDateTime(dt.Rows[0][0].ToString());
        dt = AddNewDataTable(sqlBegin);
        DateTime BeginTrain = Convert.ToDateTime(dt.Rows[0][0].ToString());
        hr = deadTime.Hour;
        min = deadTime.Minute;
        hrBegin = BeginTrain.Hour;
        minBegin = BeginTrain.Minute;
        return ((hr * 60) + min) - ((hrBegin * 60) + minBegin);
    }
    else

```

```

        return 0;
    }

    public string queryForReportTrainingResult(int TrainingId)
    {
        string sql = "select" + TRAINING_RESULT + " from " + TABLENAME_TRAINING +
        " where " + TRAININGID + " = " + TrainingId;
        string Result = "ล้มเหลว";
        DataTable dt = AddNewDataTable(sql);
        if (dt.Rows[0][0].ToString() == "1")
        {
            Result = "สำเร็จ";
        }
        return Result;
    }

```

10. ส่วนการคำนวณเป็นกราฟ

```

public DataTable GraphDistanceAndShootTable(DataTable dt)
{
    DataTable dtGraph = new DataTable();
    DataColumn dcDistance = new DataColumn("Distance", typeof(string));
    DataColumn dcDistanceCount = new DataColumn("DistanceCount", typeof(int));
    dtGraph.Columns.Add(dcDistance);
    dtGraph.Columns.Add(dcDistanceCount);
    int[] count={0,0,0,0,0};
    for (int i = 0; i < dt.Rows.Count;i++ )
    {
        if (Convert.ToInt32(dt.Rows[i]["perscore"]) != 0)
        {
            if (calDistance(Convert.ToInt32(dt.Rows[i]["Distance"])) == "0 - 300")
            {

```

```

        count[0]++;
    }
    else if (calDistance(Convert.ToInt32(dt.Rows[i]["Distance"])) == "301 - 600")
    {
        count[1]++;
    }
    else if (calDistance(Convert.ToInt32(dt.Rows[i]["Distance"])) == "601 - 900")
    {
        count[2]++;
    }
    else if (calDistance(Convert.ToInt32(dt.Rows[i]["Distance"])) == "901 - 1200")
    {
        count[3]++;
    }
    else if (calDistance(Convert.ToInt32(dt.Rows[i]["Distance"])) == "1201 - 1500")
    {
        count[4]++;
    }
}
DataRow drGraph;
int rangeDistance = 150;
for (int i = 0; i < 5; i++)
{
    drGraph = dtGraph.NewRow();
    drGraph[0] = calDistance(rangeDistance);
    drGraph[1] = count[i];
    dtGraph.Rows.Add(drGraph);
    rangeDistance += 300;
}

```

```

        dtGraph.TableName = "GraphDistanceAndShoot";
        return dtGraph;
    }

    public DataTable GraphShootAndTimeTable(DataTable dt)
    {
        string timeCheck = "";
        int countHit = 0, countShoot = 0;
        DataTable dtGraph = new DataTable();
        DataColumn dcTime = new DataColumn("ptime", typeof(string));
        DataColumn dcHit = new DataColumn("hit", typeof(int));
        DataColumn dcShoot = new DataColumn("shoot", typeof(int));
        dtGraph.Columns.Add(dcTime);
        dtGraph.Columns.Add(dcHit);
        dtGraph.Columns.Add(dcShoot);
        DataRow drGraph;
        for (int j = 0; j < dt.Rows.Count; j++)
        {
            if (calTime(dt.Rows[j]["ptime"].ToString()) != timeCheck)
            {
                timeCheck = calTime(dt.Rows[j]["ptime"].ToString());
                for (int i = 0; i < dt.Rows.Count; i++)
                {
                    if (calTime(dt.Rows[i]["ptime"].ToString()) == timeCheck)
                    {
                        countShoot++;
                        if (dt.Rows[i]["perscore"].ToString() != "0")
                        {
                            countHit++;
                        }
                    }
                }
            }
        }
    }

```

```

    }
    drGraph = dtGraph.NewRow();
    drGraph[0] = timeCheck;
    drGraph[1] = countHit;
    drGraph[2] = countShoot;
    dtGraph.Rows.Add(drGraph);
}
countShoot = 0;
countHit = 0;
}
dtGraph.TableName = "GraphTimeOnTarget";
return dtGraph;
}
public DataTable GraphTimePercentTable(DataTable dt)
{
    int usedbullet = callUseBullet(dt);
    string timeCheck = "";
    string shotCheck = "";
    int countShot = 0;
    int countHit = 0;
    int countHitAll = 0;
    DataTable dtGraphPercent = new DataTable();
    DataColumn dcTime = new DataColumn("pertime", typeof(string));
    DataColumn perPeriod = new DataColumn("period", typeof(int));
    DataColumn perAll = new DataColumn("all", typeof(int));
    dtGraphPercent.Columns.Add(dcTime);
    dtGraphPercent.Columns.Add(perPeriod);
    dtGraphPercent.Columns.Add(perAll);
    DataRow drGraph;
    string a="";

```

```

for (int b = 0; b < dt.Rows.Count;b++ )
{
    if (a != dt.Rows[b]["pershoot"].ToString())
    {
        a = dt.Rows[b]["pershoot"].ToString();
        if (dt.Rows[b]["pershoot"].ToString() == a)
        {
            if (Convert.ToInt32(dt.Rows[b]["perscore"]) != 0)
            {
                countHitAll++;
            }
        }
    }
}

for (int j = 0; j < dt.Rows.Count; j++)
{
    if (timeCheck != calTime(dt.Rows[j]["pertime"].ToString()))
    {
        timeCheck = calTime(dt.Rows[j]["pertime"].ToString());
        for (int i = 0; i < dt.Rows.Count; i++)
        {
            if (timeCheck == calTime(dt.Rows[i]["pertime"].ToString()))
            {
                if (shotCheck != dt.Rows[i]["pershoot"].ToString())
                {
                    shotCheck = dt.Rows[i]["pershoot"].ToString();
                    countShot++;
                    if (Convert.ToInt32(dt.Rows[i]["perscore"]) != 0)
                    {
                        countHit++;
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}

float hh = countHit * 100 / countHitAll;
drGraph = dtGraphPercent.NewRow();
drGraph[0] = timeCheck;
drGraph[1] = countHit*100/countShot;
drGraph[2] = hh;
dtGraphPercent.Rows.Add(drGraph);
countShot = 0;
countHit = 0;
}
}

dtGraphPercent.TableName = "GraphTimePercent";
return dtGraphPercent;
}

```

11. ส่วนเซตค่าของ Interface

```

private void defaultSet()
{
    testbtn.Enabled = false;
    gunTypeLoad();
    loadGunTypeForRandom();
    rdoDaily.Checked = true;
    dtpDate.Value = DateTime.Now;
    dtpStartDate.Value = DateTime.Now;
    dtpEndDate.Value = DateTime.Now;
    dtpStartTime.Value = DateTime.Now;
    dtpEndTime.Value = DateTime.Now;
    DataTable dtMonth = new DataTable();
}

```

```

dtMonth.Columns.Add("MonthNo", typeof(int));
dtMonth.Columns.Add("MonthName", typeof(string));
int countMonth = 0;
for (int i = DateTime.Now.Month; i >= 1; i--)
{
    dtMonth.Rows.Add(new object[] { i,
DateTime.Now.AddMonths(countMonth).ToString("MMMM") });
    countMonth--;
}
cbMonth.DisplayMember = "MonthName";
cbMonth.ValueMember = "MonthNo";
cbMonth.DataSource = dtMonth;
DataTable dtYear = new DataTable();
dtYear.Columns.Add("Year", typeof(int));
for (int i = DateTime.Now.Year; i > 2010; i--)
    dtYear.Rows.Add(new object[] { i });
cbYear.DataSource = dtYear;
cbYear.DisplayMember = "Year";
cbYear.ValueMember = "Year";
cmbMonthYear.DataSource = dtYear;
cmbMonthYear.DisplayMember = "Year";
cmbMonthYear.ValueMember = "Year";
GetTraingId();
}
private Criteria BuildCriteria(bool includeLanes)
{
    crit.GunTypeId = SetGunType(int.Parse(cmb_GunType.SelectedValue.ToString()));
    crit.Difficulty = SetDifficulty(cmb_Level.SelectedIndex);
    crit.TrainingTypeId = int.Parse(cmb_trainingtype.SelectedValue.ToString());
    CheckDateTimeChoice(crit);
}

```

```
List<int> listLane = new List<int>();  
if (chkLane1.Checked) listLane.Add(1);  
if (chkLane2.Checked) listLane.Add(2);  
if (chkLane3.Checked) listLane.Add(3);  
if (chkLane4.Checked) listLane.Add(4);  
if (chkLane5.Checked) listLane.Add(5);  
if (chkLane6.Checked) listLane.Add(6);  
if (chkLane7.Checked) listLane.Add(7);  
if (chkLane8.Checked) listLane.Add(8);  
if (chkLane9.Checked) listLane.Add(9);  
if (chkLane10.Checked) listLane.Add(10);  
crit.ListLaneNo = listLane;  
dbInterface.GetListofTrainingFromTrainingDate(crit);  
crit.TrainingId = int.Parse(lstTrainingId.SelectedValue.ToString());  
return crit;  
}
```

ประวัติผู้จัดทำโครงการ

ชื่อ – สกุล

นางสาวเกศินี อรรถชาติสี

วัน เดือน ปีเกิด

11 พฤษภาคม พ.ศ.2535

ประวัติการศึกษา

ระดับประถมศึกษา ประถมศึกษาตอนปลาย พ.ศ. 2547

โรงเรียนพระหฤทัยคอนแวนต์

ระดับมัธยมศึกษา มัธยมศึกษาตอนปลาย พ.ศ. 2552

โรงเรียนพระหฤทัยคอนแวนต์

ระดับอุดมศึกษา คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีสารสนเทศ พ.ศ. 2556

สถาบันเทคโนโลยีไทย – ญี่ปุ่น

ประวัติการอบรม

1. อบรม Google Apps
2. อบรมเรื่องการสื่อสาร ของแคทเทเลคอม



TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY