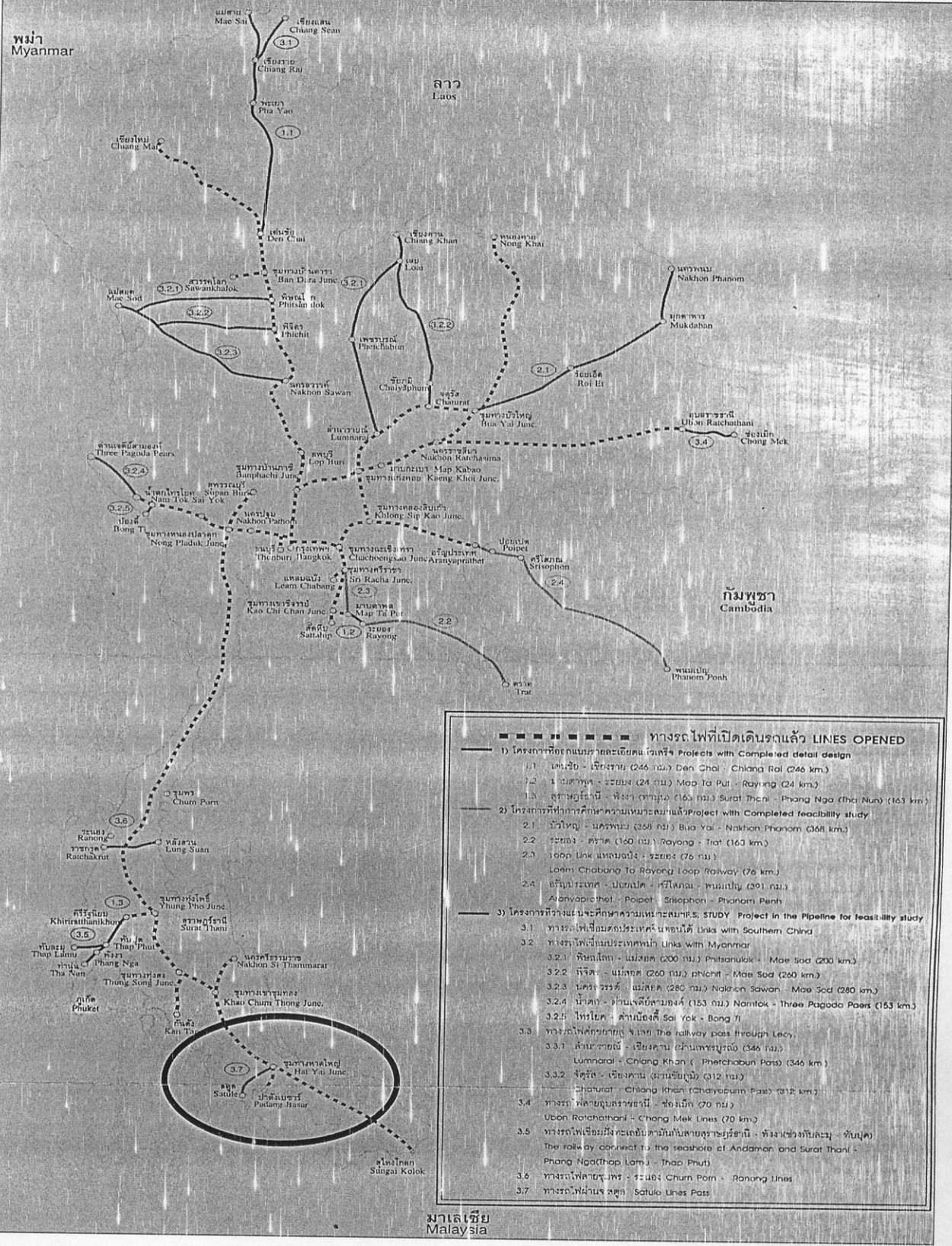


พม่า
Myanmar

ลาว
Laos

กัมพูชา
Cambodia

มาเลเซีย
Malaysia



รูปที่ ก.1 รูปแสดงโครงการขยายเส้นทางต่างๆของการรถไฟแห่งประเทศไทย (รายงานประจำปี. 2541.)

ภาคผนวก ข.

การเปรียบเทียบโปรแกรม Delphi กับ Microsoft Visual Basic

ตารางที่ ข.1 การเปรียบเทียบโปรแกรม Delphi กับ Microsoft Visual Basic สรุปรูปจากเว็บไซต์

โปรแกรม Delphi

ข้อดี	ข้อเสีย
1. เขียนโปรแกรมง่ายใช้คำสั่งตรงกว่า VB	1. โหลดช้าตัว IDE
2. ตัว IDE ก่อนข้างเสถียร	2. ประเทศไทยใช้กันน้อยแต่ต่างประเทศใช้กันมาก
3. Editor มีความสามารถสูงมากๆ ใช้ง่าย	
4. ติดต่อกับ Database ง่าย	
5. ลง Component เพิ่มได้ง่ายกว่า และมี Help ในตัวเอง	
6. เขียนระดับ System ของ Window ได้ดี	
7. โค้ดโปรแกรมมีขนาดปานกลางเขียนแล้วโหลดเร็วกว่า	

โปรแกรม Microsoft Visual Basic

ข้อดี	ข้อเสีย
1. เขียนโปรแกรมง่าย	1. ภาษา Basic ไม่ค่อยเป็น Structure ถ้าใช้ใหม่ๆ อาจจะเขียน bug ได้
2. ก่อนข้างเสถียร กิน resource น้อย load เร็วตัว(IDE)	2. Code ของโปรแกรมมีขนาดใหญ่ และต้องมีไฟล์ runtime ติดไปด้วยถึงจะใช้โปรแกรมได้
3. Editor มีความสามารถสูงมากๆ ใช้งานง่าย	3. ไม่เป็น OOP
4. ติดต่อกับ Database ง่าย	4. ต้องลง Help ของ Msdn
5. ลง Component เพิ่มได้ง่ายกว่า	5. โปรแกรมที่เขียนโหลดช้าเนื่องจากมีการจองหน่วยความจำในการ create Form ไว้ทุก Form

ตารางที่ ข.2 ตารางสรุปการใช้โปรแกรม Delphi กับงานวิจัยชิ้นนี้

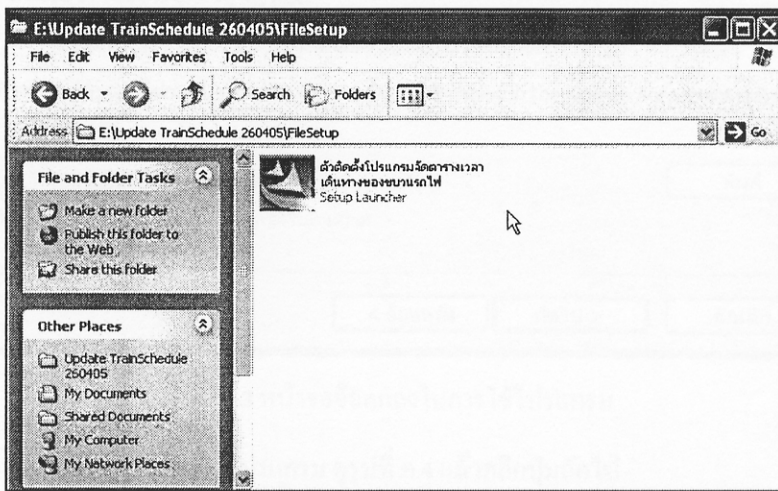
โปรแกรม Delphi กับ งานวิจัยชิ้นนี้
1. การใช้งาน Event ของ Component มีความยืดหยุ่นกว่า ใช้งานได้ง่ายกว่าสะดวก
2. มี Component ให้เลือกใช้เยอะมากกว่าก่อนข้างสะดวกซึ่งตัว Component ที่เลือกใช้ในครั้งนี้เหมาะสมกับงานมาก สามารถใช้ได้ง่าย เขียนCode ควบคุมง่ายกว่า
3. การทำงานของ Component ที่ใช้กับฐานข้อมูล ยืดหยุ่นกว่า มีอิสระ แยกหน้าที่การทำงานจากกันดี ทำให้ควบคุมการทำงาน ได้ดีขึ้น และสามารถติดต่อกับ Database ได้ง่าย
4. Delphi มีความเป็น OOP มากกว่า VB
5. Delphi สามารถเขียนให้มีการ create from ขึ้นมาใช้งานได้เมื่อไรก็ เรียกขึ้นมา ซึ่งต่างกับ VB ที่ ต้องมีการ create from ทั้งหมดใน โปรแกรมก่อนทำให้เปลืองเนื้อที่ความจำมากกว่า
6. Delphi มี Help ที่มาพร้อมกับโปรแกรม ซึ่งสามารถเรียกดูได้ กด F1 ซึ่งเขียนอ่านง่ายดี แต่ VB ต้องติดตั้ง MSDN ก่อนถึงจะเรียก Help ได้
7. หน้าต่างโปรแกรม ถูกออกแบบให้เลือกใช้งาน Component ได้ง่าย ตัว Component ก็มีมากด้วย ทั้งยังมีประสิทธิภาพมาก
8. Delphi ในการรัน ตัวApplication (.exe) ไม่จำเป็นต้องมีตัว runtime เหมือนกับ VB ซึ่งต้องมีตัว runtime ติดไปด้วย
9. การเขียน Code ของทั้งสองก็พอๆกัน ไม่ง่ายไม่ยากไปกว่ากัน
10. การควบคุมการทำงานของ Event ของตัว Component ทำได้ดีกว่าซึ่ง Component แต่ละ Component จะมี Event ให้ใช้งานเป็นของตัวเองไปทำให้ทำงานง่ายขึ้นมาก

ภาคผนวก ก.

คู่มือการใช้งานโปรแกรมการจัดตารางเวลารถไฟ

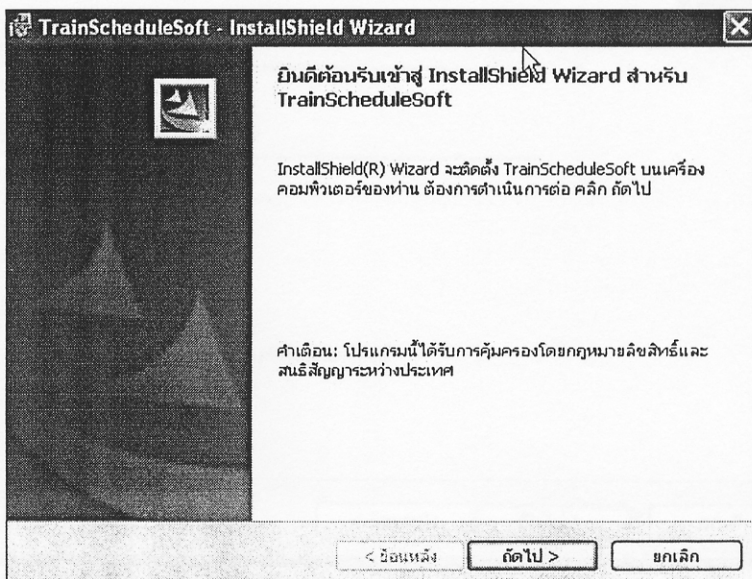
ขั้นตอนการติดตั้งโปรแกรม Train Schedule Programming มีดังนี้

- ใส่แผ่น CD-ROM ที่ใช้สำหรับติดตั้งลงใน CD-ROM Drive รอให้โปรแกรมรันแผ่นเอง คับเบิลคลิกที่ตัวติดตั้งโปรแกรมจัดตารางเวลาเดินทางของขบวนรถไฟ จากรูปที่ ก.1



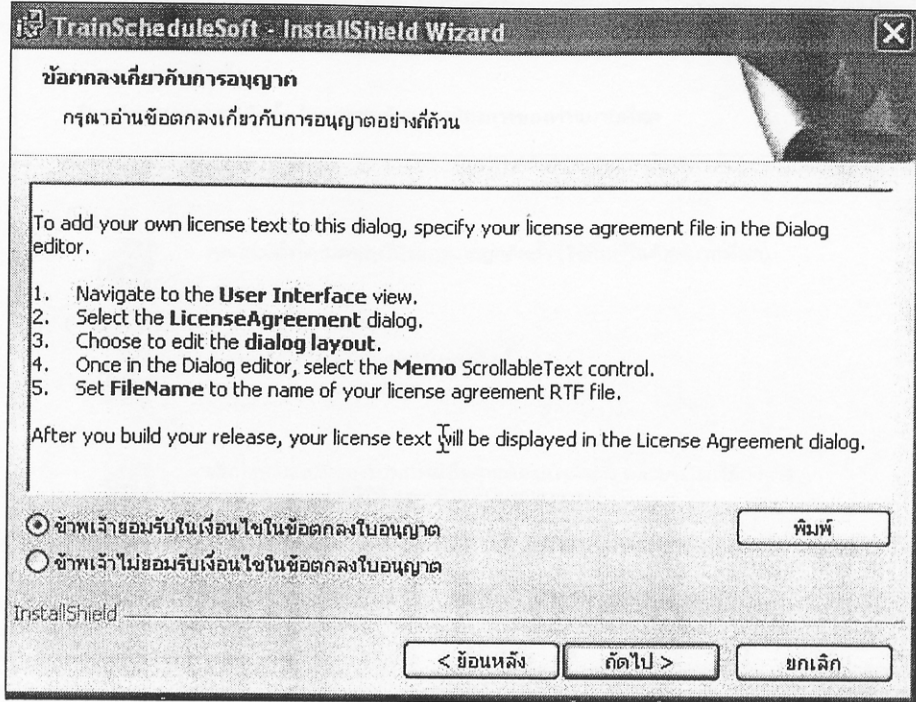
รูปที่ ก.1 การติดตั้งโปรแกรมจากตัวติดตั้งโปรแกรมจากตัวติดตั้งใน CD-ROM

- เมื่อเข้าสู่หน้าจอ ยินดีต้อนรับเข้าสู่การติดตั้งโปรแกรม คลิกปุ่มถัดไปเพื่อติดตั้งต่อไป จากรูปที่ ก.2



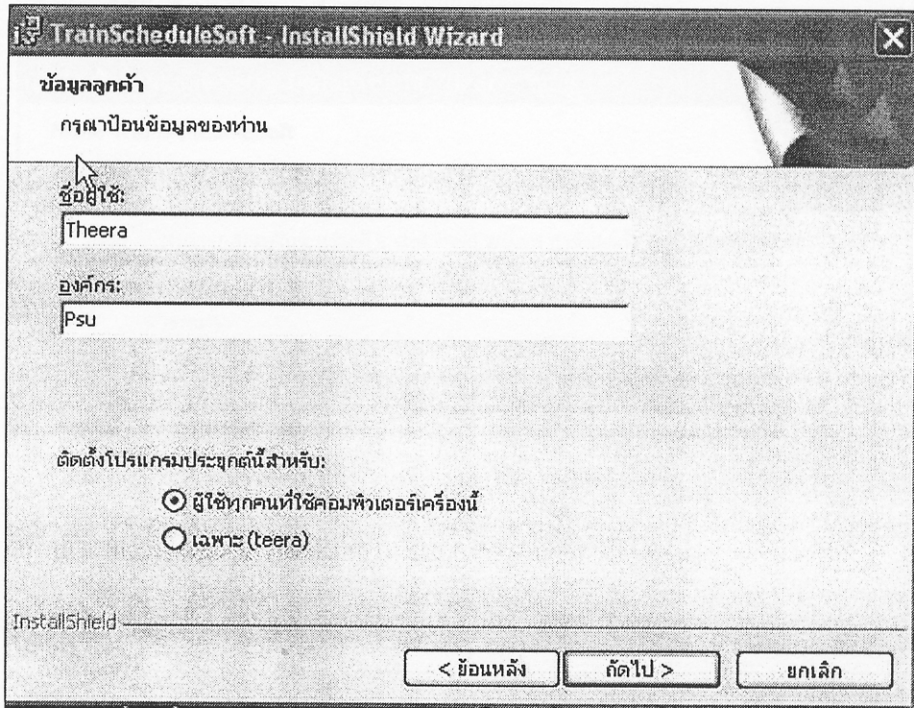
รูปที่ ก.2 หน้าจอต้อนรับเข้าสู่การติดตั้งโปรแกรม

- เมื่อเข้าสู่หน้าจอตกลงในการใช้โปรแกรม จากรูปที่ ค.3 คลิกปุ่มยอมรับแล้วกดปุ่มถัดไป



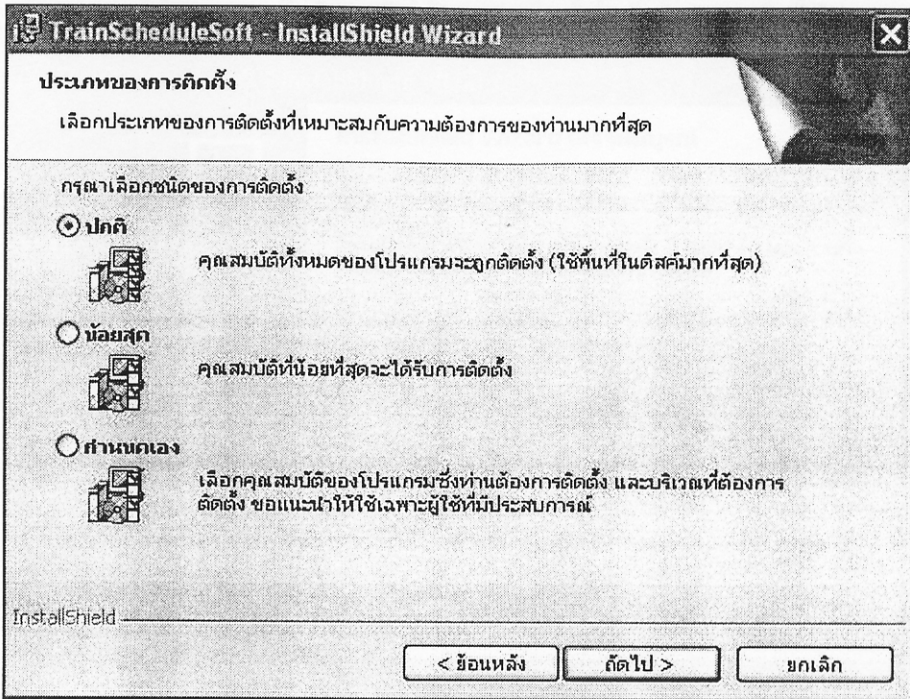
รูปที่ ค.3 หน้าจอข้อตกลงในการใช้โปรแกรม

- กรอกรายละเอียดผู้ใช้งานในโปรแกรม จากรูปที่ ค.4 แล้วคลิกปุ่มถัดไป



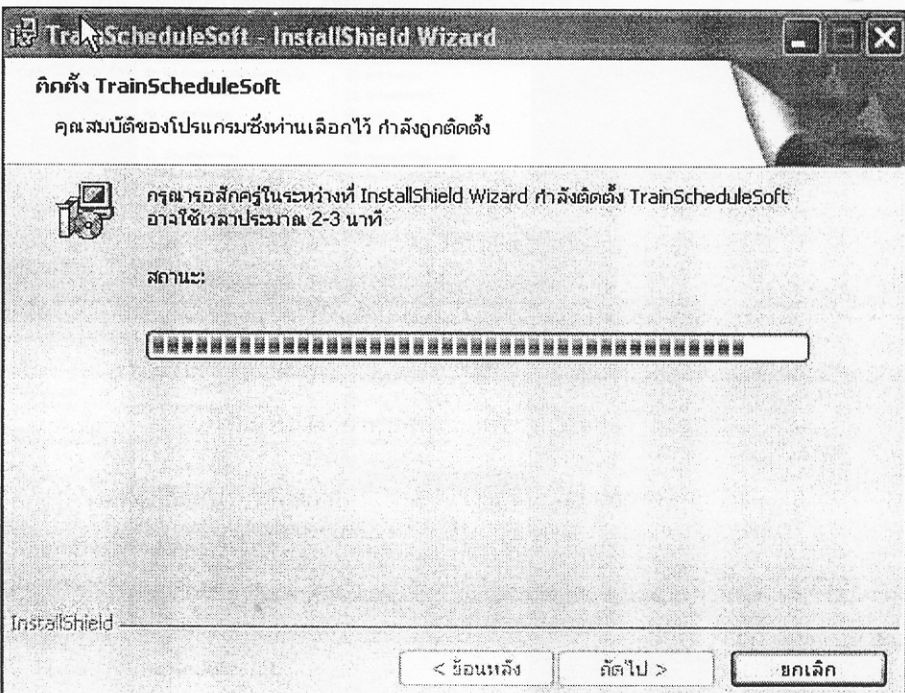
รูปที่ ค.4 หน้าจอรายละเอียดของผู้ใช้โปรแกรม

- เข้าสู่หน้าจอกำหนดประเภทของการติดตั้ง แล้วคลิกปุ่มถัดไป ดูรูปที่ ค.5 ประกอบ



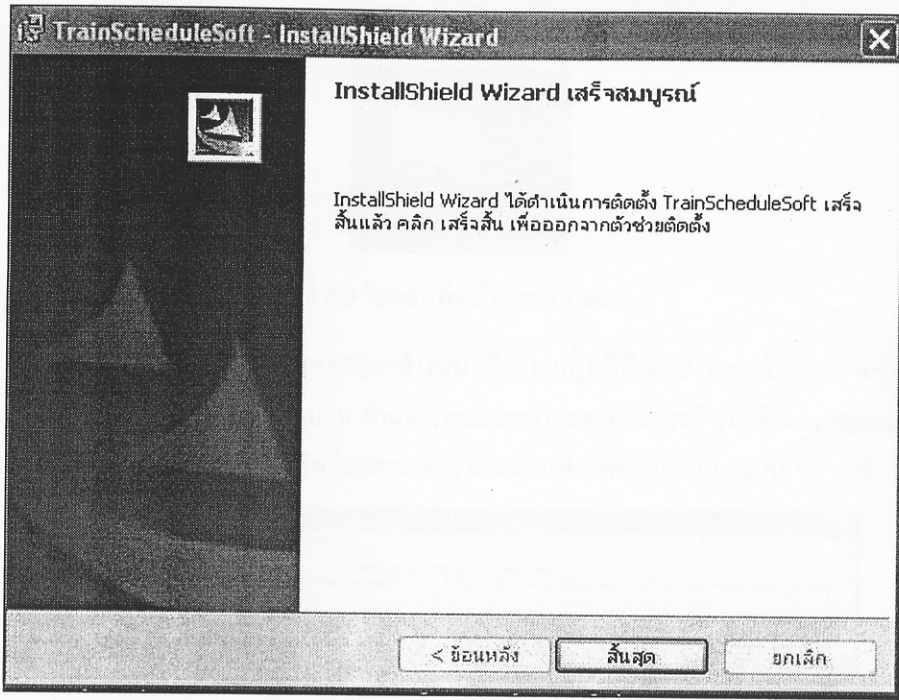
รูปที่ ค.5 หน้าจอกำหนดประเภทของการติดตั้งโปรแกรม

- คลิกปุ่มติดตั้ง เมื่อพร้อมที่จะติดตั้งโปรแกรมจากนั้นรอรการติดตั้งไฟล์ต่างๆ ลงในโฟลเดอร์ ที่เราต้องการให้ติดตั้ง ดูรูปที่ ค.6 ประกอบ



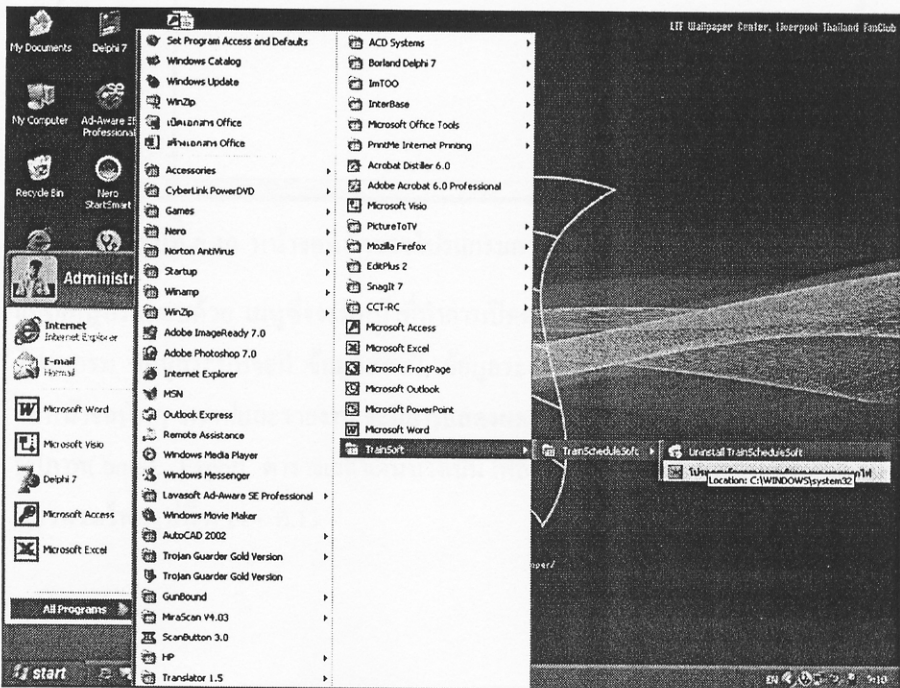
รูปที่ ค.6 หน้าจอโปรแกรมกำลังติดตั้งโปรแกรมลงในเครื่อง

- เมื่อติดตั้งไฟล์เสร็จเรียบร้อยแล้ว จะปรากฏหน้าจอเสร็จสิ้นการติดตั้งโปรแกรมขึ้นมา คลิกปุ่มสิ้นสุด เพื่อออกจากโปรแกรมการติดตั้งจากนั้นให้Restartเครื่องก็เป็นอันสิ้นสุดการติดตั้งโปรแกรมดูรูปที่ ค.7



รูปที่ ค.7 หน้าจอสิ้นสุดการติดตั้งโปรแกรม

- การลบโปรแกรมออกจากระบบ นั้นทำได้โดยการคลิกที่ Start > All Program > TrainSoft > TrainscheduleSoft > Uninstall TrainscheduleSoft ดูรูปที่ ค.8



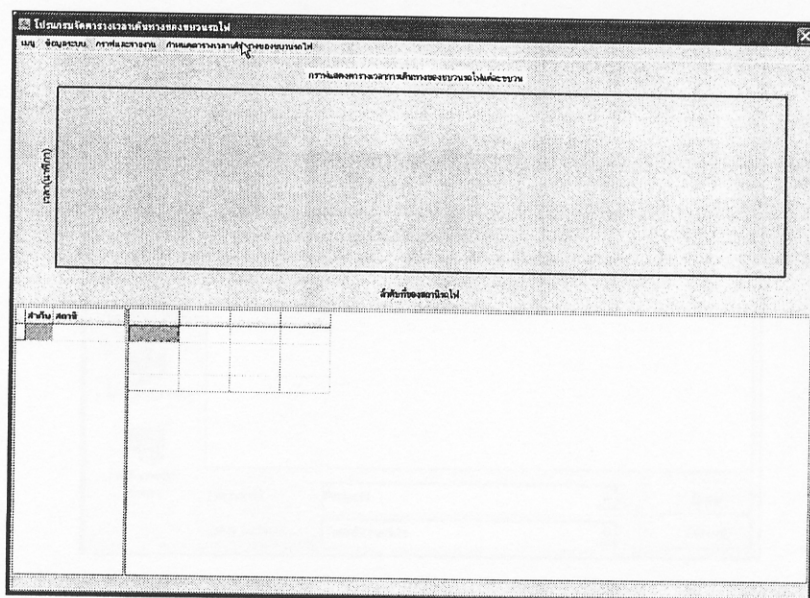
รูปที่ ค.8 หน้าจอลบโปรแกรมออกจากระบบ

- เริ่มต้นการใช้งานครั้งแรกนั้นสามารถใช้งานได้จากการคลิกที่ Start > All Program > TrainSoft > TrainscheduleSoft โปรแกรมการจัดการตารางเวลาเดินรถไฟหรือดับเบิลคลิกที่ ไอคอน บนหน้า Desktop ดังรูปที่ ค.9



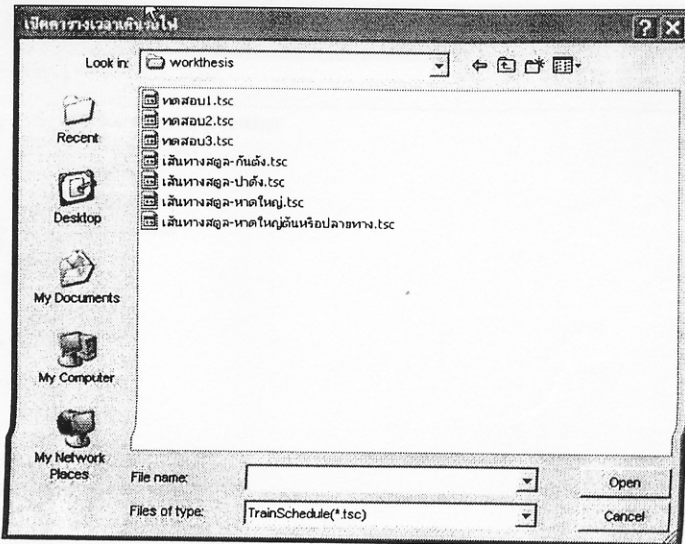
รูปที่ ค.9 ไอคอนที่หน้าจอบน Desktop

- เมื่อเข้าสู่โปรแกรมแล้วจะมีแถบเมนูอยู่ข้างบน มีแถบเมนูให้เลือกใช้งานอยู่ข้างบน, หน้าจอแสดงผลกราฟของขบวนรถไฟแต่ละขบวน ด้านล่างจะเป็นหน้าจอตารางเวลาเดินรถไฟซึ่งจะแสดงผลชื่อสถานีชื่อเส้นทาง, ประเภทขบวนรถไฟพร้อมตารางเวลาออกและถึงแต่ละสถานีมีค่าเป็น(ชม.นาที.) ดังรูปที่ ค.10

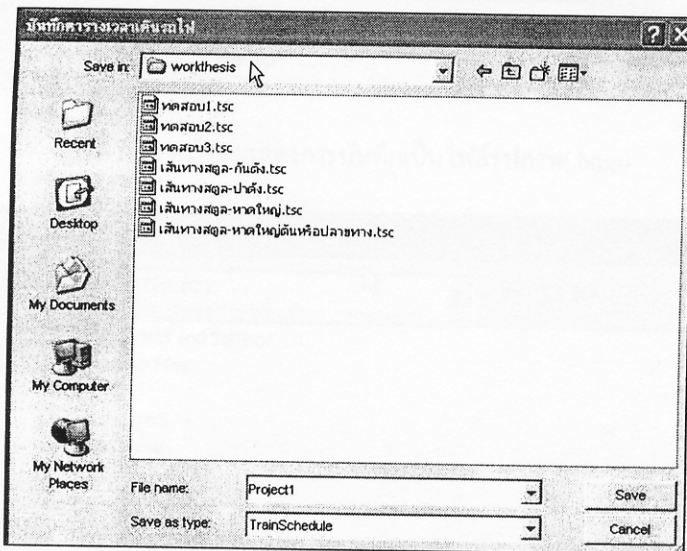


รูปที่ ค.10 หน้าจอหลักของโปรแกรมการจัดการตารางเวลาเดินรถไฟ

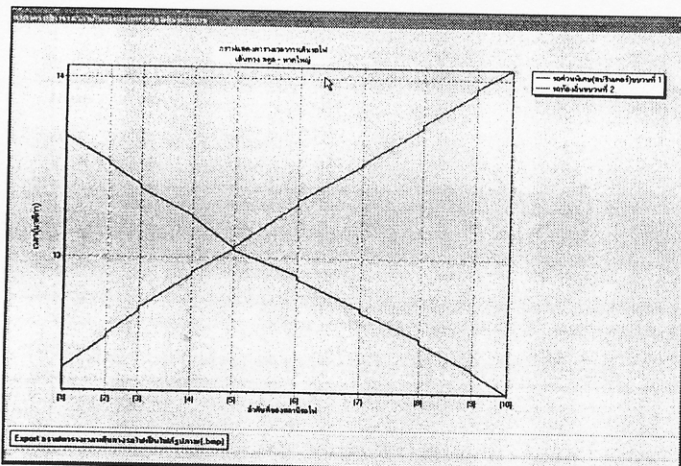
- แถบเมนูประกอบด้วย เมนูซึ่งจะมีหน้าที่ทำการเปิดตารางเวลา, บันทึกตารางเวลาเดินรถไฟ, ออกจากโปรแกรม ข้อมูลระบบจะมี ข้อมูลสถานี, ข้อมูลระยะทางระหว่างสถานีซึ่งสามารถแก้ไขปรับปรุงได้ในหน้าจออื่นๆ กราฟและรายงาน จะมีการแสดงผลกราฟตารางเวลาเดินรถไฟ, Export กราฟเป็นไฟล์รูปภาพ(.bmp), Export ตารางเวลาเดินทางเป็นไฟล์ Excel เมนูการกำหนดตารางเวลาเดินทางของขบวนรถไฟดังรูปที่ ค.11 – ค.13



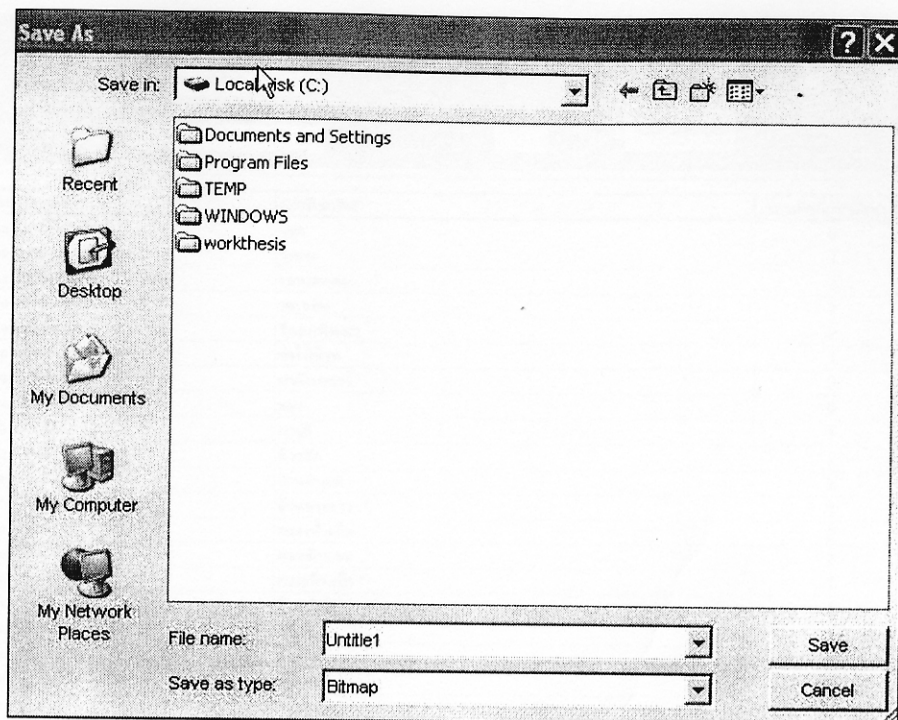
รูปที่ ค.11 หน้าจอการเปิดตารางเวลาใช้งานในโปรแกรม



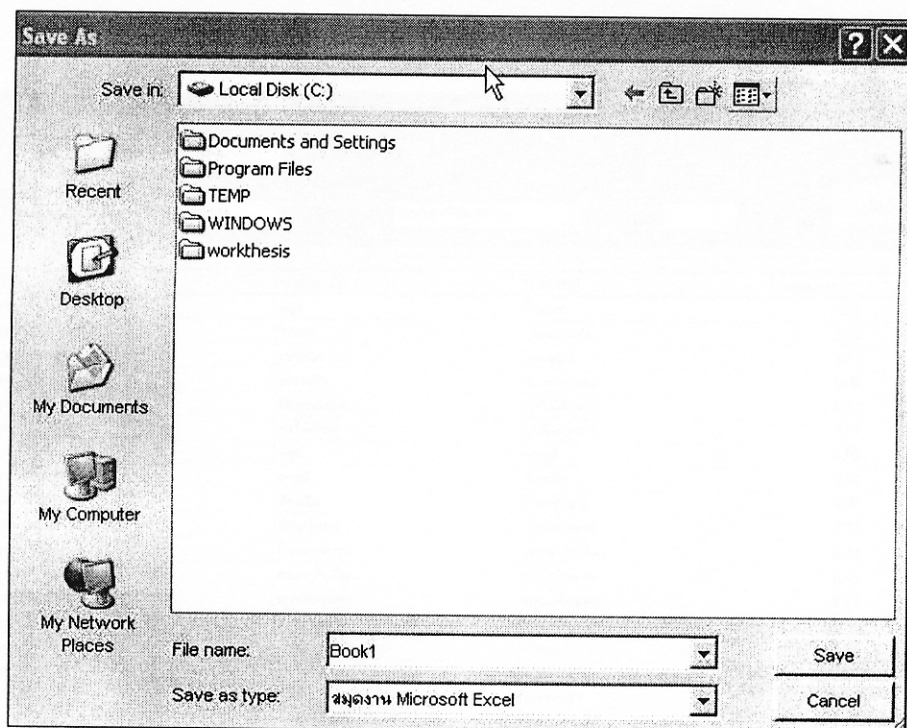
รูปที่ ค.12 หน้าจอการบันทึกงานการจัดการตารางเวลาเดินรถไฟ



รูปที่ ค.13 หน้าจอแสดงผลกราฟเพิ่มเติมหน้าจอแทนหน้าจอหลักซึ่งแสดงไม่พอ



รูปที่ ค.14 หน้าจอแสดงการบันทึกเป็นไฟล์รูปภาพ(.bmp)



รูปที่ ค.15 หน้าจอแสดงการบันทึกเป็นไฟล์ตาราง(Excel)

ข้อมูลสถานีรถไฟ

สถานีรถไฟ

เส้นทางรถไฟ

สถานีรถไฟ ใช้เวลาที่สถานี นาที

เส้นทางรถไฟ	สถานีรถไฟ	เวลาที่สถานี(นาที)
สจล - ปาดังเบซาร์	สจล	0
สจล - ปาดังเบซาร์	วังประ	1
สจล - ปาดังเบซาร์	เขมดแดง	2
สจล - ปาดังเบซาร์	ทะเลบัน	2
สจล - ปาดังเบซาร์	สีแยกพัฒนา	2
สจล - ปาดังเบซาร์	ทุ่งไม้ดวน	1
สจล - ปาดังเบซาร์	ปาดังเบซาร์	0
สจล - กันตัง	สจล	0
สจล - กันตัง	กาบูลี	1
สจล - กันตัง	ห้วยลึก	2
สจล - กันตัง	บ้านท่าแพ	2
สจล - กันตัง	บ้านทางยาว	1
สจล - กันตัง	คลองน้ำเค็ม	2
สจล - กันตัง	ควนฟ้าแลบ	2
สจล - กันตัง	คลองห้วยป้า	1
สจล - กันตัง	ห้วยน้ำซุ่น	1
สจล - กันตัง	ท่าเขา	1
สจล - กันตัง	คลองแร่	1
สจล - กันตัง	ทลนงปรีอ	1
สจล - กันตัง	บ้านสามแยก	2
สจล - กันตัง	บ้านทุ่งควาย	2
สจล - กันตัง	กันตัง	0

รูปที่ ค.16 หน้าจอข้อมูลสถานีรถไฟซึ่งเป็นการแสดงเวลาหยุดที่สถานีต่างๆ

ข้อมูลระะทางระหว่างสถานี

เส้นทางรถไฟ

ระะทางจากสถานี ถึง สถานี เก็บกับ กิโลเมตร

เส้นทางรถไฟ	จากสถานี	ถึงสถานี	ระยะทาง
สจล - ปาดังเบซาร์	สจล	วังประ	4.71
สจล - ปาดังเบซาร์	วังประ	เขมดแดง	3.02
สจล - ปาดังเบซาร์	เขมดแดง	ทะเลบัน	3.92
สจล - ปาดังเบซาร์	ทะเลบัน	สีแยกพัฒนา	8.36
สจล - ปาดังเบซาร์	สีแยกพัฒนา	ทุ่งไม้ดวน	8.42
สจล - ปาดังเบซาร์	ทุ่งไม้ดวน	ปาดังเบซาร์	7.93
สจล - กันตัง	สจล	กาบูลี	6.90
สจล - กันตัง	กาบูลี	ห้วยลึก	8.38
สจล - กันตัง	ห้วยลึก	บ้านท่าแพ	8.06
สจล - กันตัง	บ้านท่าแพ	บ้านทางยาว	7.79
สจล - กันตัง	บ้านทางยาว	คลองน้ำเค็ม	6.78
สจล - กันตัง	คลองน้ำเค็ม	ควนฟ้าแลบ	8.16
สจล - กันตัง	ควนฟ้าแลบ	คลองห้วยป้า	8.28
สจล - กันตัง	คลองห้วยป้า	ห้วยน้ำซุ่น	10.52
สจล - กันตัง	ห้วยน้ำซุ่น	ท่าเขา	6.44
สจล - กันตัง	ท่าเขา	คลองแร่	7.53
สจล - กันตัง	คลองแร่	ทลนงปรีอ	12.31
สจล - กันตัง	ทลนงปรีอ	บ้านสามแยก	6.83
สจล - กันตัง	บ้านสามแยก	บ้านทุ่งควาย	7.45
สจล - กันตัง	บ้านทุ่งควาย	กันตัง	13.14
สจล - ทาดโหล	สจล	โต๊ะไสม	7.97
สจล - ทาดโหล	โต๊ะไสม	ทุ่งพัฒนา	5.97

รูปที่ ค.17 หน้าจอข้อมูลระะทางระหว่างสถานี

- เริ่มกำหนดตารางเวลาเดินทางของขบวนรถไฟ ในหน้าเมนูกำหนดตารางเวลา เลือกเส้นทางรถไฟจาก 3 เส้นทางกดที่ Combo box กำหนดสถานีต้นทาง, เลือกประเภทของรถไฟที่ต้องการจัด, กำหนดเวลาที่สถานีต้นทางหรือปลายทางใส่ค่าเวลา (ชม.นาท.) กดปุ่มเพิ่มขบวนรถไฟเข้าไปในหน้าตารางเวลา สามารถลบข้อมูลหรือ Double - Click เพื่อแก้ไขข้อมูลหรือกดปุ่มเคลียร์ตารางเวลาทั้งหมดหรือลบข้อมูลที่ค้างไว้ยู่อุ่เดิมได้ กดปุ่มจัดตารางเวลาจะไปออกที่หน้าจอหลักต่อไป (การเลือกจัดตารางเวลานั้นสามารถทำได้ 1 เส้นทางเท่านั้น) รูปที่ ค.18

กำหนดตารางเวลาเดินทางขบวนรถไฟ

เส้นทางรถไฟ

จัดการเวลา

สถานีต้นทาง ประเภทรถไฟ กำหนด เวลาที่สถานีต้นทาง น.

ตารางเวลา

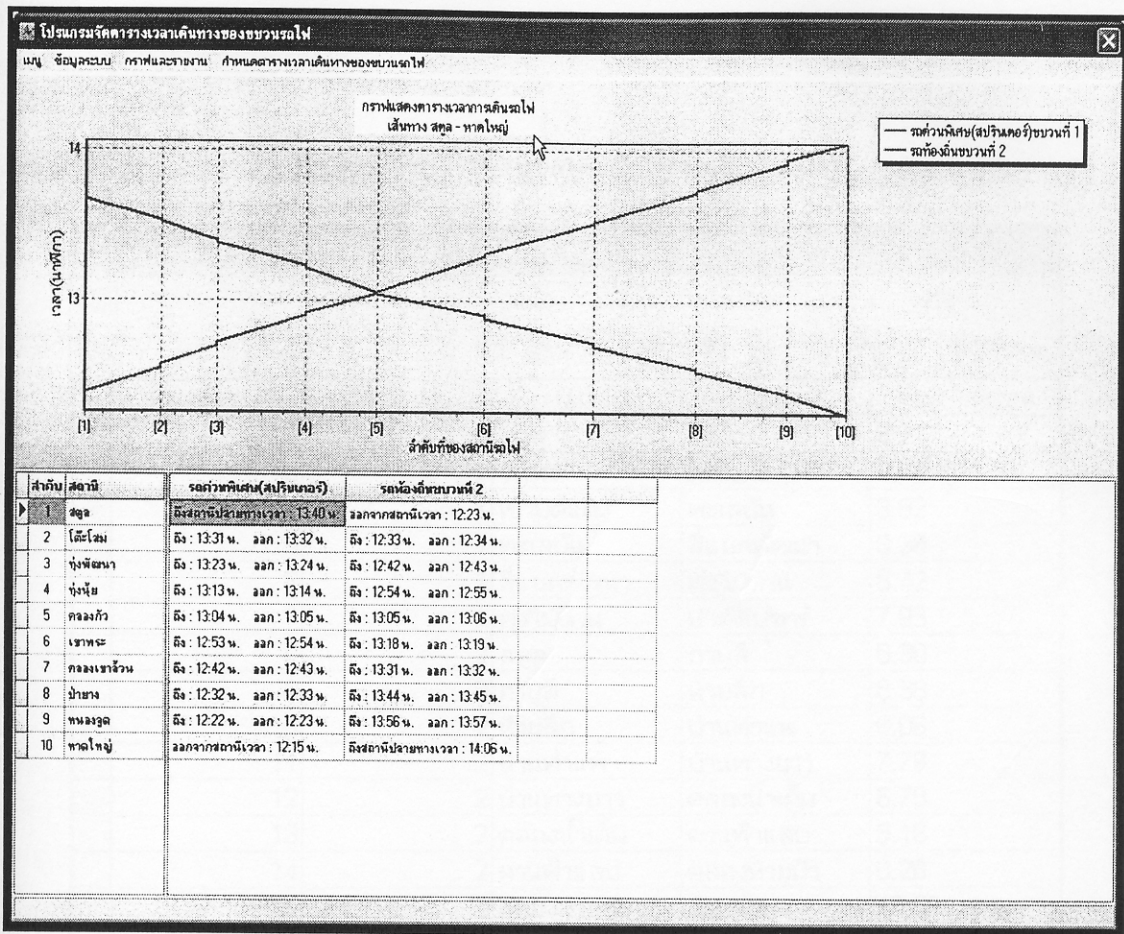
ลำดับขบวนรถไฟ	สถานีต้นทาง	ประเภทรถไฟ	เวลาออกจากสถานีต้นทาง	เวลาถึงสถานีปลายทาง
1	สุว	รถห้องเกิน	12:23	
2	ทดใหญ่	รถด่วนพิเศษ(สปรีนเตอร์)	12:15	

Double Click = แก้ไข Del = ลบ

เคลียร์ตารางเวลา

รูปที่ ค.18 หน้าจอกำหนดตารางเวลาเดินทางของขบวนรถไฟ

- หน้าจอแสดงผลการทำงานของการจัดตารางเวลาจะแสดงผล ในรูปกราฟเส้นตรงระหว่างเวลากับระยะทางแทนขบวนรถไฟ แต่ละขบวนที่ต้องการจัดด้านล่างฝั่งซ้ายแสดงลำดับชื่อสถานีตามเส้นทางที่เราเลือกได้, ด้านล่างฝั่งขวาแสดงชื่อประเภทรถต่างๆมีค่าเวลาออกหรือถึงแต่ละสถานี เฉพาะต้นทางหรือปลายทางเท่านั้นที่แสดงค่าเวลาออกหรือถึงอย่างใดอย่างหนึ่งรูปที่ ค.19
- ถ้าเสร็จสิ้นการใช้งานโปรแกรม สามารถออกจากโปรแกรมได้โดยกดที่ เมนู > ออกจากโปรแกรมหรือกดปิดที่เครื่องหมายปิดที่มุมบนทางขวาได้



รูปที่ ค.19 หน้าจอแสดงผลการทำงานของกรจัดการตารางเวลาเดินรถไฟ

ภาคผนวก ง.

ข้อมูลที่เกิดขึ้นไว้ในฐานข้อมูล (Microsoft Access)

Microsoft Access - [TBL_DISTANCE : ตาราง]

เพิ่ม แก้ไข มุมมอง แทรก รูปแบบ ระเบียบ เครื่องมือ หน้าต่าง วิธืใช้

distance_id	root_id	station1	station2	st_distance
1	1	สตูล	วังประ	4.71
3	1	วังประ	เขามตแดง	3.02
4	1	เขามตแดง	ทะเลบัน	3.92
5	1	ทะเลบัน	สี่แยกพัฒนา	8.36
6	1	สี่แยกพัฒนา	หุ้งไม้ด้วน	8.42
7	1	หุ้งไม้ด้วน	ปาดังเบซาร์	7.93
8	2	สตูล	กาบูลี	6.90
9	2	กาบูลี	ห้วยลึก	8.38
10	2	ห้วยลึก	บ้านท่าแพ	8.06
11	2	บ้านท่าแพ	บ้านทางยาว	7.79
12	2	บ้านทางยาว	คลองน้ำเค็ม	6.78
13	2	คลองน้ำเค็ม	ควนฟ้าแลบ	8.16
14	2	ควนฟ้าแลบ	คลองห้วยป่า	8.28
15	2	คลองห้วยป่า	ห้วยน้ำซุ่น	10.52
16	2	ห้วยน้ำซุ่น	ท่าเขา	6.44
17	2	ท่าเขา	คลองแร่	7.53
18	2	คลองแร่	หนองปรือ	12.31
19	2	หนองปรือ	บ้านสามแยก	6.83
20	2	บ้านสามแยก	บ้านหุ้งคาย	7.45
21	2	บ้านหุ้งคาย	กันตัง	13.14
22	3	สตูล	โต๊ะโลม	7.97
23	3	โต๊ะโลม	หุ้งพัฒนา	5.97
24	3	หุ้งพัฒนา	หุ้งนัย	9.36
25	3	หุ้งนัย	คลองแก้ว	7.67
26	3	คลองแก้ว	เขาพระ	11.51
27	3	เขาพระ	คลองเขาล้าน	11.69
28	3	คลองเขาล้าน	ป่ายาง	10.90
29	3	ป่ายาง	หนองจูด	9.71
30	3	หนองจูด	หาดใหญ่	6.27
* (AutoNumber)	0			

รูปที่ ง.1 ตารางข้อมูลระยะทางระหว่างสถานีในเส้นทาง 3 เส้นทาง

Microsoft Access - [TBL_ROOT : ตาราง]

root_id	root_name
1	สตูล - ปาดังเบซาร์
2	สตูล - กันตัง
3	สตูล - หาดใหญ่
*	(AutoNumber)

รูปที่ ง.2 ตารางข้อมูลกำหนดเส้นทางที่ต้องการเชื่อมต่อ

Microsoft Access - [TBL_SCHEDULE : ตาราง]

sch_id	head_station	train_type	time_type	train_time
1	0	3	I	0 11:00
2	1	5		0 11:10
*	0	0	0	

รูปที่ ง.3 ตารางข้อมูลเก็บค่าที่เรากำหนดตารางเวลาในโปรแกรมหลัก

Microsoft Access - [TBL_TRAIN : ตาราง]

train_id	train_name	acc_km	acc_time	V	deacc_km	deacc time
1	รถเร็ว	1.5	4	90	1	2
2	รถท้องถิ่น	1.5	4	90	1	2
3	รถธรรมดา	1.5	4	90	1	2
4	รถรวม	1.5	4	90	1	2
5	รถด่วน	1.6	3.5	90	1	2
6	รถด่วนพิเศษ	1.6	3.5	90	1	2
7	รถด่วนพิเศษ(สปรินเตอร์)	1.3	3	110	0.85	1.5
*	(AutoNumber)					

รูปที่ ง.4 ตารางข้อมูลประเภทรถต่างๆที่เราใช้ในโปรแกรม

Microsoft Access - [TBL_STATION : ตาราง]

เพิ่ม แก้ไข มุมมอง แทรก รูปแบบ ระเบียบ เครื่องมือ หน้าตา

ค้นหา บันทึก พิมพ์ ค้นหา ABC ล้าง ก๊อปปี้ วาง เลื่อน เลื่อนขึ้น เลื่อนลง

station_id	root_id	station_name	rest_time
1	1	สตูล	0
2	1	วังประ	1
3	1	เขมตแดง	2
4	1	ทะเลบัน	2
5	1	สี่แยกพัฒนา	2
6	1	ทุ่งไม้ดำน	1
7	1	ปาดังเบซาร์	0
8	2	สตูล	0
9	2	กาบูลี	1
10	2	ห้วยลึก	2
11	2	บ้านท่าแพ	2
12	2	บ้านทางยาว	1
13	2	คลองน้ำเค็ม	2
14	2	ควนฟ้าแลบ	2
15	2	คลองห้วยป้า	1
16	2	ห้วยน้ำขุน	1
17	2	ท่าเขา	1
18	2	คลองแร	1
19	2	หนองปรือ	1
20	2	บ้านสามแยก	2
21	2	บ้านทุ่งควาย	2
22	2	กันตัง	0
23	3	สตูล	0
24	3	โต๊ะโสม	1
25	3	ทุ่งพัฒนา	1
26	3	ทุ่งนุ้ย	1
27	3	คลองแก้ว	1
28	3	เขาพระ	1
29	3	คลองเขาลิ้น	1
30	3	ปายาง	1
31	3	หนองจูด	1
32	3	หาดใหญ่	0
* (AutoNumber)	1		

รูปที่ ง.5 ตารางข้อมูลเวลาหยุดที่สถานีในเส้นทาง 3 เส้นทาง

ภาคผนวก จ.

รายละเอียดในโปรแกรมการจั้ตตารางเวลา (Source Code)

```
unit DistanceU;
```

```
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;
type
TfrmDistance = class(TForm)
GroupBox1: TGroupBox;
GroupBox2: TGroupBox;
Label1: TLabel;
cbxroot: TComboBox;
Label2: TLabel;
cbxst1: TComboBox;
Label3: TLabel;
cbxst2: TComboBox;
Label4: TLabel;
edtdis: TEdit;
btnsave: TButton;
Label5: TLabel;
tbgdistance: TDBGrid;
qdistance: TADOQuery;
dsdistance: TDataSource;
qdistancedistance_id: TAutoIncField;
qdistanceroot_id: TIntegerField;
qdistancestation1: TWideStringField;
qdistancestation2: TWideStringField;
qdistancest_distance: TWideStringField;
qdistanceTroot: TStringField;
procedure FormCreate(Sender: TObject);
procedure cbxrootClick(Sender: TObject);
procedure edtdisKeyPress(Sender: TObject; var Key: Char);
procedure btnsaveClick(Sender: TObject);
procedure tbgdistanceDbClick(Sender: TObject);
procedure qdistanceCalcFields(DataSet: TDataSet);
procedure cbxrootKeyPress(Sender: TObject; var Key: Char);
private
{ Private declarations }
procedure LoadStationAtRoot;
procedure RefreshDistance;
public
{ Public declarations }
end;
var
frmDistance: TfrmDistance;
dis_id : Integer;
const
sql_ins = 'insert into TBL_DISTANCE(root_id,station1,station2,st_distance) values(%d,"%s","%s","%s");';
sql_upd = 'update TBL_DISTANCE set root_id=%d,station1="%s",station2="%s",st_distance="%s"
where
distance_id=%d ';
```

```
sql_del = 'delete from TBL_DISTANCE where distance_id=%d';
```

```
implementation
```

```
uses dmTrainSchedule;
```

```
{$R *.dfm}
```

```
procedure TfrmDistance.FormCreate(Sender: TObject);
```

```
begin
```

```
    dbAddNew:=True;
```

```
    LoadRoot(cbxroot.Items);
```

```
end;
```

```
    procedure TfrmDistance.LoadStationAtRoot;
```

```
const
```

```
    s = 'select station_id,station_name from TBL_STATION where root_id=%d';
```

```
var
```

```
    xsql: String;
```

```
begin
```

```
    xsql:=Format(s,[Integer(TObject(cbxroot.Items.Objects[cbxroot.ItemIndex]))]);
```

```
    dmTrain.GetToTString(xsql,cbxst1.Items);
```

```
    dmTrain.GetToTString(xsql,cbxst2.Items);
```

```
end;
```

```
    procedure TfrmDistance.cbxrootClick(Sender: TObject);
```

```
begin
```

```
    cbxroot.Style:=csDropDownList;
```

```
    LoadStationAtRoot;
```

```
end;
```

```
    procedure TfrmDistance.edtdisKeyPress(Sender: TObject; var Key: Char);
```

```
const
```

```
    ch = '0123456789.';
```

```
begin
```

```
    if (Pos(Key,ch)=0) and (Key >= chr(32)) then
```

```
        Key:=#0;
```

```
    if Key = #13 then
```

```
        btnsave.SetFocus;
```

```
end;
```

```
    procedure TfrmDistance.btnsaveClick(Sender: TObject);
```

```
var
```

```
    s : String;
```

```
begin
```

```
    if (cbxroot.ItemIndex<>-1) and (cbxst1.Text<>") and (cbxst2.Text<>") and (edtdis.Text<>") then
```

```
begin
```

```
    if dbAddNew then //insert
```

```
        begin
```

```
s:=Format(sql_ins,[Integer(TObject(cbxroot.Items.Objects[cbxroot.ItemIndex])),cbxst1.Text,cbxst2.Text,edtdis.Text]);
```

```
        dmTrain.RunCmd(s);
```

```
        end
```

```
        else //edit
```

```
            begin
```

```
s:=Format(sql_upd,[Integer(TObject(cbxroot.Items.Objects[cbxroot.ItemIndex])),cbxst1.Text,cbxst2.Text,edtdis.Text,dis_id]);
```

```
            dmTrain.RunCmd(s);
```

```
            end;
```

```
            RefreshDistance;
```

```
            dbAddNew:=True;
```

```
            cbxst1.SetFocus;
```

```
        end
```

```
    else
```

```
        MessageDlg('การลบสถานีการระบุสถานี',mtError,[mbOK],0);
```

```

end;
procedure TfrmDistance.tbqdistanceDbClick(Sender: TObject);
begin
  dbAddNew:=False;
  with qdistance do
  begin
    dis_id:=FieldByName('distance_id').AsInteger;
    cbxroot.ItemIndex:=cbxroot.Items.IndexOfObject(TObject(FieldByName('root_id').AsInteger));
    cbxrootClick(nil);
    cbxst1.Text:=FieldByName('station1').Text;
    cbxst2.Text:=FieldByName('station2').Text;
    edtdis.Text:=FieldByName('st_distance').Text;
  end;
end;
procedure TfrmDistance.RefreshDistance;
begin
  with qdistance do
  begin
    Active:=False;
    Active:=True;
  end;
end;
procedure TfrmDistance.qdistanceCalcFields(DataSet: TDataSet);
begin
  DataSet['Troot_name']:=TRoot[DataSet.FieldByName('root_id').AsInteger];
end;
procedure TfrmDistance.cbxrootKeyPress(Sender: TObject; var Key: Char);
begin
  if Key = #13 then
    SendMessage(Self.Handle,WM_NEXTDLGCTL,0,0);
end;
end.

```

unit dmTrainSchedule;

```

interface
uses
  SysUtils, Classes, DB, ADODB,IniFiles;
type
  TTimes = record
    hour,minutes : Integer;
  end;
  type
  TClock = record
    arrive,leave : String;
  end;
  type
  TTrain = record
    headstation : Integer;
    tr_type : Integer;
    tm_type : Integer;
    tr_time : String;
  end;
  type
  TLines = record
    slope,c_const : Double;
  end;
  type

```

```

TPoint = record
    x1V,y1V,x2V,y2V : Double;
end;

type
TdmTrain = class(TDataModule)
    conn: TADOConnection;
    qDataset: TADOQuery;
    qRun: TADOQuery;
private
    { Private declarations }
public
    { Public declarations }
    function GetDataSet(fsq: String): TDataSet;
    procedure RunCmd(fsq: String);
    procedure GetToTString(fsq: String; fitem: TStrings);
end;

var
dmTrain: TdmTrain;
openfile : Boolean;
dbAddNew : Boolean;
Tdistance : array of Double;
Tresttime : array of Double;
Theadresttime : array[0..1] of Double;
StationName : array of String;
Train_No : array of TTrain;
Train_Clock : array of array of TClock;
Train_Line : array of array of TLines;
Train_Point : array of array of TPoint;
Temp_Point : array of array of TPoint;
FirstClock : array of String; //เวลาที่ออกจากสถานีต้นทาง
LastClock : array of String; //เวลาที่ถึงสถานีปลายทาง
//
KM_ACC, KM_DEACC, TIME_ACC, TIME_DEACC, Vv: Double;
station_count: Integer;
Train_Count : Integer;
Select_Root : Integer;

const
TRoot : array[1..3] of String = ('สตูล - ปาดังเบซาร์', 'สตูล - กันตัง', 'สตูล - หาดใหญ่');
TTail : array[1..3] of String = ('ปาดัง', 'กันตัง', 'หาดใหญ่');
procedure LoadRoot(cbxItem: TStrings);
function ConvertToTTime(time: String): TTimes;
function ConvertToXYLine(time: String): Double;
//
function GetStationNum(rt_id: Integer): Integer;
procedure GetDistance(rt_id: Integer);
procedure GetRestTime(rt_id: Integer);
procedure GetDataTrain(train: Integer);

implementation
{$R *.dfm}
procedure TdmTrain.RunCmd(fsq: String);
begin
    with qRun do
    begin
        SQL.Text:=fsq;
        ExecSQL;
    end;
end;

```

```

end;
function TdmTrain.GetDataSet(fsq:String):TDataSet ;
begin
  with qDataset do
  begin
    Active:=False;
    SQL.Text:=fsq;
    Active:=True;
  end;
  Result:=qDataset;
end;
procedure TdmTrain.GetToTString(fsq:String;fitem:TStrings);
begin
  with qDataset do
  begin
    Active:=False;
    SQL.Text:=fsq;
    Active:=True;
    fitem.Clear;
    First;
    while not EOF do
    begin
      fitem.AddObject(Fields[1].Text,TObject(Fields[0].AsInteger));
      Next;
    end;
  end;
end;
procedure LoadRoot(cbxItem:TStrings);
const
  s = 'select root_id,root_name from TBL_ROOT';
begin
  dmTrain.GetToTString(s,cbxItem);
end;
procedure GetDistance(rt_id:Integer);
var
  i : Integer;
  xsq:String;
  qdis : TDataSet;
begin
  xsq:='select st_distance from TBL_DISTANCE where root_id=' + IntToStr(rt_id);
  qdis:=dmTrain.GetDataSet(xsq);
  SetLength(Tdistance,qdis.RecordCount); //ระยะทางระหว่างสถานีเท่ากับ station-1
  //
  with qdis do
  begin
    First;
    for i:=0 to RecordCount-1 do
    begin
      Tdistance[i]:=Fields[0].AsFloat;
      Next;
    end;
  end;
end;
procedure GetRestTime(rt_id:Integer);
var
  i : Integer;
  xsq:String;

```



```

qtime : TDataSet;
begin
  xsql:='select rest_time from TBL_STATION where root_id=' + IntToStr(rt_id);
  qtime:=dmTrain.GetDataSet(xsql);
  SetLength(Tresttime,qtime.RecordCount-2); //เวลาพักเท่ากับ station-2
  //
  with qtime do
  begin
    First;
    Theadresttime[0]:=Fields[0].AsFloat;
    Last;
    Theadresttime[1]:=Fields[0].AsFloat;
    First;
    Next;
    for i:=0 to RecordCount-3 do
    begin
      Tresttime[i]:=Fields[0].AsFloat;
      Next;
    end;
  end;
end;
procedure GetDataTrain(train:Integer);
var
  xsql : String;
  qtrain : TDataSet;
begin
  xsql:='select * from TBL_TRAIN where train_id=' + IntToStr(train);
  qtrain:=dmTrain.GetDataSet(xsql);
  //
  with qtrain do
  begin
    KM_ACC:=FieldByName('acc_km').AsFloat;
    TIME_ACC:=FieldByName('acc_time').AsFloat;
    Vv:=FieldByName('V').AsFloat;
    KM_DEACC:=FieldByName('deacc_km').AsFloat;
    TIME_DEACC:=FieldByName('deacc_time').AsFloat;
  end;
end;
function ConvertToTTime(time:String):TTimes;
var
  ts : TTimes;
begin
  //Copy(S; Index, Count: Integer)
  ts.hour:=StrToInt(Copy(time,0,2));
  ts.minutes:=StrToInt(Copy(time,Pos(':',time)+1,Length(time)));
  Result:=ts;
end;
function ConvertToXYLine(time:String):Double;
var
  a,b:Double;
begin
  a:=StrToFloat(Copy(time,0,2));
  b:=StrToFloat(Copy(time,Pos(':',time)+1,Length(time)))/100;
  Result:=a+b ;
end;
function GetStationNum(rt_id:Integer):Integer;
var

```

```

    xsql : String;
    qnum : TDataSet;
begin
    xsql := 'select station_id from TBL_STATION where root_id=' + IntToStr(rt_id);
    qnum := dmTrain.GetDataSet(xsql);
    Result := qnum.RecordCount;
end;
end.

```

unit InsertTrainU;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, ExtCtrls, Mask, DB, ADODB, Grids, BaseGrid, AdvGrid,
 DBAdvGrd, dmTrainSchedule, Buttons, IniFiles;

type

```

    TfrmInsertTrain = class(TForm)
    GroupBox1: TGroupBox;
    Label1: TLabel;
    cbxroot: TComboBox;
    Panel1: TPanel;
    Label2: TLabel;
    cbxhead: TComboBox;
    Label4: TLabel;
    cbxtrain: TComboBox;
    Label6: TLabel;
    edtime: TMaskEdit;
    Label7: TLabel;
    btnadd: TButton;
    GroupBox2: TGroupBox;
    tbgTrain: TDBAdvStringGrid;
    qTrain: TADOQuery;
    dsTrain: TDataSource;
    qTrainsch_id: TIntegerField;
    qTrainhead_station: TIntegerField;
    qTraintrain_name: TWideStringField;
    qTraintrain_time: TWideStringField;
    qTraintrain_type: TIntegerField;
    btnProcess: TBitBtn;
    btnclear: TBitBtn;
    qTrainTid: TIntegerField;
    qTrainThead: TStringField;
    cbxtime: TComboBox;
    qTrainTout: TStringField;
    qTrainTto: TStringField;
    qTraintime_type: TIntegerField;
    Label3: TLabel;
    Panel2: TPanel;
    procedure FormShow(Sender: TObject);
    procedure btnaddClick(Sender: TObject);
    procedure tbgTrainKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure tbgTrainDbClick(Sender: TObject);
    procedure qTrainCalcFields(DataSet: TDataSet);
    procedure btnProcessClick(Sender: TObject);
    procedure edtimeExit(Sender: TObject);
    procedure btnclearClick(Sender: TObject);

```

```

    procedure cbxrootExit(Sender: TObject);
private
    { Private declarations }
    procedure LoadHeadStation;
    procedure LoadTrain;
    procedure ClearText;
    procedure RefreshTrain;
    function GetMAXid:Integer;
public
    { Public declarations }
    function LoadSelectRoot:Integer;
    procedure WriteSelectRoot;
end;
var
    frmInsertTrain: TfrmInsertTrain;
    hs : array[0..1] of String;
    id : Integer;
const
    sql_ins_sch = 'insert into TBL_SCHEDULE values(%d,%d,%d,%d,%s)';
    sql_upd_sch = 'update TBL_SCHEDULE set head_station=%d,train_type=%d,time_type=%d,train_time=%s
where sch_id=%d';
    sql_del_sch = 'delete from TBL_SCHEDULE where sch_id=%d';
implementation
{$R *.dfm}
procedure TfrmInsertTrain.FormShow(Sender: TObject);
begin
    dbAddNew:=True;
    LoadRoot(cbxroot.Items);
    qTrain.Close;
    qTrain.Open;
    if qTrain.RecordCount > 0 then
        begin
            if not openfile then
                Select_Root:=LoadSelectRoot;
            //
            cbxroot.ItemIndex:=cbxroot.Items.IndexOfObject(TObject(Select_Root));
            cbxrootExit(nil);
        end;
    LoadTrain;
    RefreshTrain;
end;
procedure TfrmInsertTrain.LoadHeadStation;
const
    s = 'select station_name from TBL_STATION where root_id=%d';
var
    fsql : String;
    qhead : TDataSet;
begin
    fsql:=Format(s,[Integer(TObject(cbxroot.Items.Objects[cbxroot.ItemIndex]))]);
    qhead:=dmTrain.GetDataSet(fsql);
    //
    with qhead do
    begin
        cbxhead.Clear;
        First;
        cbxhead.Items.Add(Fields[0].Text);
        hs[0]:=Fields[0].Text;
    end;
end;

```

```

Last;
cbxhead.Items.Add(Fields[0].Text);
hs[1]:=Fields[0].Text;
cbxhead.Text:='ระบุสถานีต้นทาง';
end;
end;
procedure TfrmInsertTrain.LoadTrain;
const
s = 'select train_id,train_name from TBL_TRAIN';
begin
dmTrain.GetToTString(s,cbxtrain.Items);
end;
procedure TfrmInsertTrain.btnAddClick(Sender: TObject);
var
s : String;
head,train,typetime: Integer;
begin
if (cbxroot.ItemIndex<>-1) and (cbxhead.ItemIndex <>-1) and (cbxtrain.ItemIndex <> -1) then
begin
head:=cbxhead.ItemIndex;
train:=Integer(TObject(cbxtrain.Items.Objects[cbxtrain.ItemIndex]));
typetime:=cbxtime.ItemIndex;
if dbAddNew then //add
begin
s:=Format(sql_ins_sch,[GetMAXid+1, head, train,typetime,"" + edtime.Text + ""]);
dmTrain.RunCmd(s);
end
else //edit
begin
s:=Format(sql_upd_sch,[head,train,typetime,"" + edtime.Text + " ",id]);
dmTrain.RunCmd(s);
dbAddNew:=True;
end;
ClearText;
RefreshTrain;
end
else
MessageDlg('ตรวจสอบการระบุข้อมูล',mtError,[mbOK],0);
end;
function TfrmInsertTrain.GetMAXid:Integer;
var
qmax : TDataSet;
begin
qmax:=dmTrain.GetDataSet('select max(sch_id) from TBL_SCHEDULE');
Result:=qmax.Fields[0].AsInteger;
end;
procedure TfrmInsertTrain.tbqTrainKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
var
s : String;
begin
if Key in [VK_DELETE] then
begin
if MessageDlg('ต้องการลบข้อมูลขบวนรถไฟหรือไม่',mtConfirmation,[mbYes,mbNo],0)=mrYes then
begin
id:=qTrain.FieldByName('sch_id').AsInteger;
s:=Format(sql_del_sch,[id]);

```

```

dmTrain.RunCmd(s);
ClearText;
RefreshTrain;
if qTrain.RecordCount=0 then
  begin
    cbxroot.ItemIndex:=-1;
    cbxroot.Text:='ระบุเส้นทางรถไฟ';
  end;
end
else
  Abort;
end;
end;
procedure TfrmInsertTrain.tbqTrainDbClick(Sender: TObject);
begin
  dbAddNew:=False;
  id:=qTrain.FieldByName('sch_id').AsInteger;
  cbxhead.ItemIndex:=qTrain.FieldByName('head_station').AsInteger;
  cbxtrain.ItemIndex:=cbxtrain.Items.IndexOfObject(TObject(qTrain.FieldByName('train_type').AsInteger));
  cbxtime.ItemIndex:=qTrain.FieldByName('time_type').AsInteger;
  edtime.Text:=qTrain.FieldByName('train_time').Text;
end;
procedure TfrmInsertTrain.ClearText;
begin
  cbxhead.ItemIndex:=-1;
  cbxhead.Text:='ระบุสถานีต้นทาง';
  cbxtrain.ItemIndex:=-1;
  cbxtrain.Text:='ระบุประเภทรถไฟ';
  cbxtime.ItemIndex:=0;;
  edtime.Text:='00:00';
  cbxhead.SetFocus;
end;
procedure TfrmInsertTrain.RefreshTrain;
begin
  with qTrain do
  begin
    Active:=False;
    Active:=True;
  end;
end;
procedure TfrmInsertTrain.qTrainCalcFields(DataSet: TDataSet);
begin
  with DataSet do
  begin
    DataSet['Thead']:=hs[FieldByName('head_station').AsInteger];
    DataSet['Tid']:=RecNo;
    if FieldByName('time_type').AsInteger = 0 then
      DataSet['Tout']:=FieldByName('train_time').Text
    else
      DataSet['Tto']:=FieldByName('train_time').Text;
  end;
end;
end;
procedure TfrmInsertTrain.btnProcessClick(Sender: TObject);
var
  i : Integer;
begin
  with qTrain do

```

```

begin
if RecordCount > 0 then
begin
SetLength(Train_No,RecordCount);
Train_Count := RecordCount;
First;
while not EOF do
begin
i:=RecNo - 1;
Train_No[i].headstation:=FieldByName('head_station').AsInteger;
Train_No[i].tr_type:=FieldByName('train_type').AsInteger;
Train_No[i].tm_type:=FieldByName('time_type').AsInteger;
Train_No[i].tr_time:=FieldByName('train_time').Text;
Next;
end;
Self.Close;
Self.ModalResult:=mrOK;
end
else
begin
MessageDlg('ยังไม่มีข้อมูลขบวนรถไฟ',mtError,[mbOK],0);
cbxroot.SetFocus;
end;
end;
end;
procedure TfrmInsertTrain.edttimeExit(Sender: TObject);
var
tt : TTimes;
begin
tt:=ConvertToTTime(edttime.Text);
if not ((tt.hour < 24) and (tt.minutes < 60)) then
begin
MessageDlg('ระบุหน่วยเวลาไม่ถูกต้อง',mtError,[mbOK],0);
edttime.SetFocus;
end;
end;
end;
procedure TfrmInsertTrain.btnclearClick(Sender: TObject);
begin
dmTrain.RunCmd('delete from TBL_SCHEDULE');
ClearText;
cbxroot.ItemIndex:=-1;
cbxroot.Text:='ระบุเส้นทางรถไฟ';
cbxhead.Clear;
cbxhead.Text:='ระบุสถานีต้นทาง';
RefreshTrain;
end;
function TfrmInsertTrain.LoadSelectRoot:Integer;
var
root_ini :TIniFile;
fpath : String;
begin
fpath:=ExtractFilePath(Application.ExeName) + 'Config.ini';
root_ini:=TIniFile.Create(fpath);
Result:=root_ini.ReadInteger('InsertTrain','root',1);
root_ini.Destroy;
end;
end;

```

```

procedure TfrmInsertTrain.WriteSelectRoot;
var
  root_ini : TIniFile;
  fpath : String;
begin
  fpath:=ExtractFilePath(Application.ExeName) + 'Config.ini';
  root_ini:=TIniFile.Create(fpath);
  root_ini.WriteInteger('InsertTrain','root',Select_Root);
  root_ini.Destroy;
end;
procedure TfrmInsertTrain.cbxrootExit(Sender: TObject);
begin
  if (qTrain.RecordCount > 0) and (Integer(cbxroot.Items.Objects[cbxroot.ItemIndex])<>Select_Root) then
    begin
      MessageDlg('โปรแกรมไม่สามารถกำหนดตารางเดินรถไฟได้พร้อมกันหลายเส้นทาง',mtError,[mbOK],0);
      cbxroot.ItemIndex:=cbxroot.Items.IndexOfObject(TObject(Select_Root));
      cbxroot.SetFocus;
    end
  else
    begin
      Select_Root:=cbxroot.ItemIndex+1;
      if cbxroot.ItemIndex <> -1 then
        LoadHeadStation;
        WriteSelectRoot;
      end;
    end;
end;
end.

```

unit MainU;

```

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ExtCtrls, Grids, DBGrids, StdCtrls, DB, ADODB, BaseGrid,
  AdvGrid, DBAdvGrd, Mask, dmTrainSchedule, TeeProcs, TeEngine, Chart,
  DbChart, Series, Buttons, ComCtrls, Teeprevi;
type
  TfrmMain = class(TForm)
    mmnuMain: TMainMenu;
    FileMenu: TMenuItem;
    ExitProgram: TMenuItem;
    Panel2: TPanel;
    qstation: TADOQuery;
    dsstation: TDataSource;
    tbgtimeschedule: TAdvStringGrid;
    tbgstation: TDBAdvStringGrid;
    InsTrain: TMenuItem;
    TrainChart: TChart;
    qstationstation_name: TWideStringField;
    qstationTorder: TStringField;
    qSave: TADOQuery;
    tblSave: TADOTable;
    adoSave: TADOConnection;
    OpenTCH: TMenuItem;
    SaveTCH: TMenuItem;
    Panel1: TPanel;
    Panel3: TPanel;
    SaveDlgTCH: TSaveDialog;

```

```

ProgressFile: TProgressBar;
OpenDialog1: TOpenDialog;
N1: TMenuItem;
SysData: TMenuItem;
StationData: TMenuItem;
DistanceData: TMenuItem;
ReportData: TMenuItem;
ShowTrGraph: TMenuItem;
ExportTrGraphToBMP: TMenuItem;
ExportExcel: TMenuItem;
adoOpen: TADOConnection;
tblOpen: TADOTable;
qOpen: TADOQuery;
SaveDlgExcel: TSaveDialog;
SaveDlgBMP: TSaveDialog;
procedure StationDataClick(Sender: TObject);
procedure DistanceDataClick(Sender: TObject);
procedure ExitProgramClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure edtamountKeyPress(Sender: TObject; var Key: Char);
procedure FormShow(Sender: TObject);
procedure InsTrainClick(Sender: TObject);
procedure qstationCalcFields(DataSet: TDataSet);
procedure SaveTCHClick(Sender: TObject);
procedure OpenTCHClick(Sender: TObject);
procedure StationDClick(Sender: TObject);
procedure DistanceDClick(Sender: TObject);
procedure ShowTrGraphClick(Sender: TObject);
procedure ExportTrGraphToBMPClick(Sender: TObject);
procedure ExportExcelClick(Sender: TObject);
private
{ Private declarations }
procedure LoadStation;
procedure LoadDistance;
procedure InsertRowStation;
procedure InsertColumnStation;
procedure DeleteRowColumnStation;
procedure Delete_st_row;
procedure CalcTime(calc: TTimes);
procedure DecreaseTime(calc: TTimes);
//
procedure ProcessSchedule;
procedure SortPriority;
procedure SwapPriority(source1,source2:Integer);
procedure CreateLineEquation;
procedure CheckCrashTrain;
procedure GenNewTime_Line_Point_Head0(train1,train2,j_intersect:Integer);
procedure GenNewTime_Line_Point_Head1(train1,train2,j_intersect:Integer);
function XYspot(m1,c1,m2,c2:Double):Double;
function IntersectLine(m1,m2:Double):Boolean;
function GenNew_Y_Value(y_axis:Double):Double;
procedure PlotSchedule;
procedure ShowStation(rt_id:Integer);
procedure PlotLinesGraph;
//
function WalkTime(KM,U:Double):Double;
function PriorTrain(tr_type:Integer):Integer;

```



```

function GenLines(x1,y1,x2,y2:Double):TLines;
function GetTotalDistance:Double;
function GetPoint(x1,y1,x2,y2:Double):TPoint;
function GetFirstTime(train_id:Integer):String;
//
procedure Processing;
procedure SaveTrainSchedule(filesave:String);
procedure OpenTrainSchedule(fileopen:String);
procedure ProcessingFile;
public
{ Public declarations }
procedure GetTempPoint;
function GetTrainTypeName(train_type:Integer):String;
end;
var
frmMain: TfrmMain;
TotalTime : TTimes;
del : Boolean;
stationD : array of Double;
const
  THead : array[0..1] of String = ('สถานีต้นทาง:', 'ถึงสถานีปลายทาง:');
  Train_Color : array[1..7] of TColor = (clBlack, clOlive, clAqua, clFuchsia, clGreen, clBlue, clRed);
implementation
uses StationU, DistanceU, ProcessingU, InsertTrainU, StrUtils, Math,
  ShowTrainGraphU, WelcomeU;
{$R *.dfm}
procedure TfrmMain.FormCreate(Sender: TObject);
begin
  del:=False;
end;
procedure TfrmMain.FormShow(Sender: TObject);
begin
  Application.CreateForm(TfrmImage, frmImage);
  if frmImage.ShowModal=mrOK then
    frmImage.Destroy;
  openfile:=False;
end;
procedure TfrmMain.LoadStation;
begin
  Application.CreateForm(TfrmStation, frmStation);
  frmStation.ShowModal;
  frmStation.Destroy;
end;
procedure TfrmMain.LoadDistance;
begin
  Application.CreateForm(TfrmDistance, frmDistance);
  frmDistance.ShowModal;
  frmDistance.Destroy;
end;
procedure TfrmMain.StationDataClick(Sender: TObject);
begin
  LoadStation;
end;
procedure TfrmMain.DistanceDataClick(Sender: TObject);
begin
  LoadDistance;
end;

```

```

procedure TfrmMain.ExitProgramClick(Sender: TObject);
begin
  Self.Close;
end;
procedure TfrmMain.ShowStation(rt_id:Integer);
const
  s = 'select station_name from TBL_STATION where root_id=%d';
var
  fsql : String;
  i:Integer;
begin
  fsql:=Format(s,[rt_id]);
  with qstation do
  begin
    Active:=False;
    SQL.Text:=fsql;
    Active:=True;
  end;
end;
procedure TfrmMain.edtamountKeyPress(Sender: TObject; var Key: Char);
const
  ch = '0123456789';
begin
  if (Pos(Key,ch)=0) and (Key>=chr(32)) then
    Key:=#0;
end;
procedure TfrmMain.CalcTime(calc:TTimes);
begin
  TotalTime.hour:=(TotalTime.hour + calc.hour) + ((TotalTime.minutes + calc.minutes) div 60);
  TotalTime.hour:=TotalTime.hour mod 24;
  TotalTime.minutes:=(TotalTime.minutes + calc.minutes) mod 60;
end;
procedure TfrmMain.DecreaseTime(calc:TTimes);
begin
  TotalTime.hour:=(TotalTime.hour - calc.hour);
  if TotalTime.minutes - calc.minutes < 0 then
    TotalTime.hour:=TotalTime.hour-1;
  //
  if TotalTime.hour < 0 then
    TotalTime.hour:=24+TotalTime.hour;
  //
  TotalTime.minutes:=((60 - calc.minutes)+TotalTime.minutes) mod 60;
end;
function TfrmMain.WalkTime(KM,U:Double):Double ;
var
  t_walk : Double;
begin
  //formular : T = S(km) / U(km/hr -> km/min)
  t_walk:=(KM*60)/U;
  t_walk := StrToFloat(FormatFloat('0.00',t_walk));
  Result:=t_walk;
end;
procedure TfrmMain.Delete_st_row;
var
  i : Integer;
begin
  i:=tbgtimeschedule.RowCount;

```

```

while i > 5 do
begin
  tbgtimeschedule.RemoveRows(i-1,1);
  i:=i-1;
end;
end;
procedure TfrmMain.InsertRowStation;
var
  rows : Integer;
begin
  for rows:=4 to station_count-1 do
  begin
    tbgtimeschedule.InsertRows(rows,1);
  end;
end;
procedure TfrmMain.InsertColumnStation;
var
  columns : Integer;
begin
  for columns:=4 to Train_Count-1 do
  begin
    tbgtimeschedule.InsertCols(columns,1);
  end;
end;
procedure TfrmMain.DeleteRowColumnStation;
var
  i : Integer;
  a,b : Integer;
begin
  i:=tbgtimeschedule.RowCount;
  while i > 5 do
  begin
    tbgtimeschedule.RemoveRows(i-1,1);
    i:=i-1;
  end;
  //
  i:=tbgtimeschedule.ColCount;
  while i > 5 do
  begin
    tbgtimeschedule.RemoveCols(i-1,1);
    i:=i-1;
  end;
  for a :=0 to 4 do
  begin
    for b := 0 to 4 do
    begin
      tbgtimeschedule.Cells[a,b]:= "";
    end;
  end;
end;
end;
procedure TfrmMain.InsTrainClick(Sender: TObject);
begin
  Application.CreateForm(TfrmInsertTrain,frmInsertTrain);
  with frmInsertTrain do
  begin
    if ShowModal = mrOK then
    begin

```

```

    ProcessSchedule;
    SaveTCH.Enabled:=True;
    ExportExcel.Enabled:=True;
    Self.Caption:='โปรแกรมจัดการเวลาเดินทางของขบวนรถไฟ';
end;
Destroy;
end;
end;
procedure TfrmMain.ProcessSchedule; //*****
begin
    Processing;
    SortPriority;
    CreateLineEquation;
    CheckCrashTrain;
    PlotSchedule;
    PlotLinesGraph;
    del:=True;
end;
procedure TfrmMain.Processing;
var
    frm : TForm;
    i : Integer;
begin
    try
        Application.CreateForm(TfrmProcessing, frmProcessing);
        frm := Screen.ActiveForm ;
        frm.Enabled := false;
        frmProcessing.Parent := frm;
        frmProcessing.Show ;
        Application.ProcessMessages ;
        //
        with frmProcessing.ProgressBar1 do
            begin
                i:=0;
                while i <= Max do
                    begin
                        Position:=i;
                        frmProcessing.IPersen.Caption := formatfloat('#,##0.00',((Position * 100) / Max )) + ' %';
                        inc(i);
                        Application.ProcessMessages ;
                    end;
            end;
        finally
            frm.Enabled := true;
            frm.SetFocus ;
            frmProcessing.close;
            frmProcessing.Destroy ;
        end;
        //
    end;
procedure TfrmMain.SortPriority;
var
    round,j : Integer;
    t1,t2 : Double;
begin
    //sort ประเภทรถไฟก่อน จากนั้นจึงทำการ SORT เวลาจากออกก่อนไปหาออกทีหลัง
    //sort type of train FIRST

```

```

for round := 0 to Train_Count-2 do
begin
  for j := 0 to Train_Count-2 do
  begin
    if PriorTrain(Train_No[j].tr_type) > PriorTrain(Train_No[j+1].tr_type) then
      SwapPriority(j,j+1);
    end; //j
  end;//round
  //sort time of train NEXT
  for round := 0 to Train_Count-2 do
  begin
    for j := 0 to Train_Count-2 do
    begin
      if PriorTrain(Train_No[j].tr_type) = PriorTrain(Train_No[j+1].tr_type) then
        begin
          if Train_No[j].tm_type= 0 then
            t1:=ConvertToTTime(Train_No[j].tr_time).hour + ((ConvertToTTime(Train_No[j].tr_time).minutes)/100)
          else
            begin
              t1:=ConvertToTTime(GetFirstTime(j)).hour + ((ConvertToTTime(GetFirstTime(j)).minutes)/100);
            end;
          //
          if Train_No[j+1].tm_type= 0 then
            t2:=ConvertToTTime(Train_No[j+1].tr_time).hour +
            ((ConvertToTTime(Train_No[j+1].tr_time).minutes)/100)
          else
            begin
              t2:=ConvertToTTime(GetFirstTime(j+1)).hour + ((ConvertToTTime(GetFirstTime(j+1)).minutes)/100);
            end;
          //t1:=ConvertToTTime(Train_No[j].tr_time).hour + ((ConvertToTTime(Train_No[j].tr_time).minutes)/100);
          //t2:=ConvertToTTime(Train_No[j+1].tr_time).hour +
          ((ConvertToTTime(Train_No[j+1].tr_time).minutes)/100);
          //ShowMessage('t1='+FloatToStr(t1));
          //ShowMessage('t2='+FloatToStr(t2));
          //
          if t2 < t1 then
            SwapPriority(j,j+1);
          end;
        end; //j
      end;//round
    end;
  end;
end;
procedure TfrmMain.SwapPriority(source1,source2:Integer);
var
  temp:TTrain;
begin
  //Train_No[source1] to temp
  temp.headstation:=Train_No[source1].headstation;
  temp.tr_type:=Train_No[source1].tr_type;
  temp.tr_time:=Train_No[source1].tr_time;
  temp.tm_type :=Train_No[source1].tm_type;
  //Train_No[source2] to Train_No[source1]
  Train_No[source1].headstation:=Train_No[source2].headstation;
  Train_No[source1].tr_type:=Train_No[source2].tr_type;
  Train_No[source1].tr_time:=Train_No[source2].tr_time;
  Train_No[source1].tm_type:=Train_No[source2].tm_type;
  //temp to Train_No[source2]
  Train_No[source2].headstation:=temp.headstation;

```

```

Train_No[source2].tr_type:=temp.tr_type;
Train_No[source2].tr_time:=temp.tr_time;
Train_No[source2].tm_type:=temp.tm_type;
end;
function TfrmMain.PriorTrain(tr_type:Integer):Integer ;
var
  prior:Integer;
begin
  case tr_type of
    4 : prior:=7; //รถรวม
    3 : prior:=6; //รถธรรมดา
    2 : prior:=5; //รถห้องดิน
    1 : prior:=4; //รถเร็ว
    5 : prior:=3; //รถด่วน
    6 : prior:=2; //รถด่วนพิเศษ
    7 : prior:=1 ;//รถด่วนพิเศษ(สปรินเตอร์)
  end;
  Result:=prior;
end;
procedure TfrmMain.CreateLineEquation;
var
  i,st_rows,dot,tt : Integer;
  tr,Tleave,Tarrive:String;
  each_train_time,chktime,station_distance : Double;
  Twork,Trest : TTimes;
  x1,x2,y1,y2 : Double;
begin
  station_count:=GetStationNum(Select_Root);
  GetDistance(Select_Root); //n-1 value Tdistance : array of Double;
  GetRestTime(Select_Root); //n-2 value Tresttime : array of Double;
  SetLength(FirstClock,Train_Count);
  SetLength(LastClock,Train_Count);
  SetLength(Train_Clock,Train_Count,station_count-2); //station -2
  SetLength(Train_Line,Train_Count,station_count-1);
  SetLength(Train_Point,Train_Count,station_count-1);
  //
  for i := 0 to Train_Count-1 do
  begin
    GetDataTrain(Train_No[i].tr_type);
    TotalTime:=ConvertToTTime(Train_No[i].tr_time);
    Tleave:=Train_No[i].tr_time;
    //
    if Train_No[i].headstation=0 then //สถานีเริ่มต้นที่ สตูล
    begin
      if Train_No[i].tm_type=0 then
      begin
        FirstClock[i]:=Tleave;
        station_distance:=0;
        for st_rows := 0 to station_count-2 do
        begin
          x1:=station_distance;
          y1:=ConvertToXYLine(Tleave);
          each_train_time:=0;
          each_train_time:=each_train_time + TIME_ACC + WalkTime((Tdistance[st_rows]-KM_ACC-
KM_DEACC),Vv) + TIME_DEACC;
          tr:=FloatToStr(each_train_time);

```

```

dot:=Pos('.',tr);
chctime:=StrToFloat(Copy(tr,dot+1,Length(tr)-dot))/10;
each_train_time:=StrToFloat(FormatFloat('0',each_train_time));
if (chctime >= 4) and (chctime <5) then
  each_train_time:=each_train_time + 1;
//
tt:=StrToInt(FloatToStr(each_train_time));
Twork.hour:=tt div 60;
Twork.minutes:=tt mod 60;
CalcTime(Twork);
Tarrive:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
station_distance:=station_distance + Tdistance[st_rows];
x2:=station_distance;
y2:=ConvertToXYLine(Tarrive);
Train_Point[i][st_rows]:=GetPoint(x1,y1,x2,y2);
Train_Line[i][st_rows]:=GenLines(x1,y1,x2,y2);
//ShowMessage('m=' + FloatToStr(Train_Line[i][st_rows].slope));
//ShowMessage('c=' + FloatToStr(Train_Line[i][st_rows].c_const));
//
if st_rows < station_count-2 then
  Train_Clock[i][st_rows].arrive:=Tarrive; //*****
//
Trest.hour:=0;
if st_rows<>station_count-2 then
  Trest.minutes:=StrToInt(FloatToStr(Tresttime[st_rows]))
else
  Trest.minutes:=StrToInt(FloatToStr(Theadresttime[1]));
//
CalcTime(Trest);
Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
//
if st_rows = station_count-2 then
  LastClock[i]:=Tleave
else
  if st_rows < station_count-2 then
    Train_Clock[i][st_rows].leave:=Tleave; //*****
    {ShowMessage('x1=' + FloatToStr(Train_Point[i][st_rows].x1V));
    ShowMessage('y1=' + FloatToStr(Train_Point[i][st_rows].y1V));
    ShowMessage('x2=' + FloatToStr(Train_Point[i][st_rows].x2V));
    ShowMessage('y2=' + FloatToStr(Train_Point[i][st_rows].y2V)); }
    // ShowMessage('m=' + FloatToStr(Train_Line[i][st_rows].slope) + '
c'+FloatToStr(Train_Line[i][st_rows].c_const));
  end;//station_count
end//if time_type=0
else //1
begin
  LastClock[i]:=Tleave;
  station_distance:=GetTotalDistance;
  st_rows:=station_count - 2;
  while st_rows >= 0 do
  begin
    x2:=station_distance;
    y2:=ConvertToXYLine(Tleave);
    //ShowMessage('x2=' + FloatToStr(x2));
    //ShowMessage('y2=' + FloatToStr(y2));

```

```

each_train_time:=0;
each_train_time:=each_train_time + TIME_ACC + WalkTime((Tdistance[st_rows]-KM_ACC-
KM_DEACC),Vv) + TIME_DEACC;
tr:=FloatToStr(each_train_time);
dot:=Pos('.',tr);
chctime:=StrToFloat(Copy(tr,dot+1,Length(tr)-dot)) /10;
each_train_time:=StrToFloat(FormatFloat('0',each_train_time));
if (chctime >= 4) and (chctime <5) then
  each_train_time:=each_train_time + 1;
//
tt:=StrToInt(FloatToStr(each_train_time));
Twork.hour:=tt div 60;
Twork.minutes:=tt mod 60;
DecreaseTime(Twork);
Tarrive:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes)) ;
//
station_distance:=StrToFloat(FormatFloat('0.00',station_distance));
Tdistance[st_rows]:=StrToFloat(FormatFloat('0.00',Tdistance[st_rows]));
station_distance:=station_distance-Tdistance[st_rows];
//
x1:=station_distance;
y1:=ConvertToXYLine(Tarrive);
//ShowMessage('x1='+FloatToStr(x1));
//ShowMessage('y1='+FloatToStr(y1));
Train_Point[i][st_rows]:=GetPoint(x1,y1,x2,y2);
Train_Line[i][st_rows]:=GenLines(x1,y1,x2,y2);
//
if st_rows > 0 then //*****
  Train_Clock[i][st_rows-1].leave:=Tarrive; //*****
//
Trest.hour:=0;
if (st_rows >=1) then
  Trest.minutes:=StrToInt(FloatToStr(Tresttime[st_rows-1]))
else
  Trest.minutes:=StrToInt(FloatToStr(Theadresttime[0]));
//
DecreaseTime(Trest);
Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes)) ;
if st_rows=0 then //*****
  FirstClock[i]:=Tleave
else
  Train_Clock[i][st_rows-1].arrive:=Tleave;
// ShowMessage('(x1,y1)='+FloatToStr(x1)+'+',FloatToStr(y1));
// ShowMessage('(x2,y2)='+FloatToStr(x2)+'+',FloatToStr(y2));
st_rows:=st_rows-1;
end;//while st_rows >= 0 do
end;//if time_type=1
end//headstation=0 //สถานีต้นทางเริ่มที่ขบวน
else //สถานีต้นทางอยู่ที่ กั้นคัง หรือ หวากใหญ่ หรือ ปาดัง
begin
if Train_No[i].tm_type=0 then
begin
LastClock[i]:=Tleave;
station_distance:=GetTotalDistance;
st_rows:=station_count - 2;

```



```

while st_rows >= 0 do
begin
  x2:=station_distance;
  y2:=ConvertToXYLine(Tleave);
  each_train_time:=0;
  each_train_time:=each_train_time + TIME_ACC + WalkTime((Tdistance[st_rows]-KM_ACC-
KM_DEACC),Vv) + TIME_DEACC;
  tr:=FloatToStr(each_train_time);
  dot:=Pos('.',tr);
  chkttime:=StrToFloat(Copy(tr,dot+1,Length(tr)-dot)) /10;
  each_train_time:=StrToFloat(FormatFloat('0',each_train_time));
  if (chkttime >= 4) and (chkttime <5) then
    each_train_time:=each_train_time + 1;
  //
  tt:=StrToInt(FloatToStr(each_train_time));
  Twork.hour:=tt div 60;
  Twork.minutes:=tt mod 60;
  CalcTime(Twork);
  Tarrive:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
  //
  station_distance:=StrToFloat(FormatFloat('0.00',station_distance));
  Tdistance[st_rows]:=StrToFloat(FormatFloat('0.00',Tdistance[st_rows]));
  station_distance:=station_distance-Tdistance[st_rows];
  //
  x1:=station_distance;
  y1:=ConvertToXYLine(Tarrive);
  Train_Point[i][st_rows]:=GetPoint(x1,y1,x2,y2);
  Train_Line[i][st_rows]:=GenLines(x1,y1,x2,y2);
  //
  //ShowMessage('m=' + FloatToStr(Train_Line[i][st_rows].slope));
  //ShowMessage('c=' + FloatToStr(Train_Line[i][st_rows].c_const));
  if st_rows > 0 then
    Train_Clock[i][st_rows-1].arrive:=Tarrive; //*****
  //
  Trest.hour:=0;
  if (st_rows >=1) then
    Trest.minutes:=StrToInt(FloatToStr(Tresttime[st_rows-1]))
  else
    Trest.minutes:=StrToInt(FloatToStr(Theadresttime[0]));
  //
  CalcTime(Trest);
  Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
  if st_rows=0 then
    FirstClock[i]:=Tleave
  else
    Train_Clock[i][st_rows-1].leave:=Tleave;

//ShowMessage('tr1(x1,y1)='+FloatToStr(Train_Point[i][st_rows].x1V)+' '+FloatToStr(Train_Point[i][st_rows].y1V));

//ShowMessage('tr1(x2,y2)='+FloatToStr(Train_Point[i][st_rows].x2V)+' '+FloatToStr(Train_Point[i][st_rows].y2V));
  //ShowMessage('(x1,y1)='+FloatToStr(x1)+' '+FloatToStr(y1));
  //ShowMessage('(x2,y2)='+FloatToStr(x2)+' '+FloatToStr(y2));
  st_rows:=st_rows-1;
end;//while st_rows >= 0 do
end //if Train_No[i].tm_type=0 then

```

```

else //tm_type=1
begin
  FirstClock[i]:=Tleave;
  station_distance:=0;
  for st_rows := 0 to station_count-2 do
  begin
    x1:=station_distance;
    y1:=ConvertToXYLine(Tleave);
    each_train_time:=0;
    each_train_time:=each_train_time + TIME_ACC + WalkTime((Tdistance[st_rows]-KM_ACC-
KM_DEACC),Vv) + TIME_DEACC;
    tr:=FloatToStr(each_train_time);
    dot:=Pos('.',tr);
    chktme:=StrToFloat(Copy(tr,dot+1,Length(tr)-dot)) /10;
    each_train_time:=StrToFloat(FormatFloat('0',each_train_time));
    if (chktme >= 4) and (chktme <5) then
      each_train_time:=each_train_time + 1;
    //
    tt:=StrToInt(FloatToStr(each_train_time));
    Twork.hour:=tt div 60;
    Twork.minutes:=tt mod 60;
    DecreaseTime(Twork); //*****
    Tarrive:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
    station_distance:=station_distance + Tdistance[st_rows];
    x2:=station_distance;
    y2:=ConvertToXYLine(Tarrive);
    Train_Point[i][st_rows]:=GetPoint(x1,y1,x2,y2);
    Train_Line[i][st_rows]:=GenLines(x1,y1,x2,y2);
    Trest.hour:=0;
    if st_rows<>station_count-2 then
      Trest.minutes:=StrToInt(FloatToStr(Tresttime[st_rows]))
    else
      Trest.minutes:=StrToInt(FloatToStr(Theadresttime[1]));
    //
    DecreaseTime(Trest);
    Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
    //
    //
    if st_rows < station_count-2 then
      Train_Clock[i][st_rows].arrive:=Tleave; //*****
    //
      if st_rows = station_count-2 then
        LastClock[i]:=Tleave
      else
        if st_rows < station_count-2 then
          Train_Clock[i][st_rows].leave:=Tarrive; //*****
        { ShowMessage('x1='+ FloatToStr(Train_Point[i][st_rows].x1V));
          ShowMessage('y1='+ FloatToStr(Train_Point[i][st_rows].y1V));
          ShowMessage('x2='+ FloatToStr(Train_Point[i][st_rows].x2V));
          ShowMessage('y2='+ FloatToStr(Train_Point[i][st_rows].y2V)); }
          end;//for st_rows := 0 to station_count-2 do
    end;
  end;
end; // end สถานีต้นทางอยู่ที่ กั้นคัง หรือ หาดใหญ่ หรือ ปาดัง
end; //Train_Count
end;

```

```

function TfrmMain.GenLines(x1,y1,x2,y2:Double):TLines;
var
  m,c : Double;
  L : TLines;
begin
  m:=(y2-y1)/(x2-x1);
  m:=StrToFloat(FormatFloat('0.000',m));
  c:=y1-(m*x1);
  c:=StrToFloat(FormatFloat('0.000',c));
  L.slope:=m;
  L.c_const:=c;
  Result:=L;
end;
function TfrmMain.GetPoint(x1,y1,x2,y2:Double):TPoint;
var
  P : TPoint;
begin
  P.x1V:=x1;
  P.y1V:=y1;
  P.x2V:=x2;
  P.y2V:=y2;
  Result:=P;
end;
function TfrmMain.GetTotalDistance:Double ;
var
  i : Integer;
  tt:Double;
begin
  tt:=0;
  for i :=0 to station_count-2 do
  begin
    tt:=tt + Tdistance[i];
  end;
  Result:=tt;
end;
function TfrmMain.GetTrainTypeName(train_type:Integer):String ;
var
  qTName : TDataSet;
begin
  qTName:=dmTrain.GetDataSet('select train_name from TBL_TRAIN where train_id='+IntToStr(train_type));
  Result:=qTName.Fields[0].Text;
end;
procedure TfrmMain.PlotSchedule;
var
  iplot,colz,rowz:Integer;
  itime:Integer;
begin
  if del then
    DeleteRowColumnStation;
  //
  ShowStation(Select_Root);
  InsertRowStation;
  InsertColumnStation;
  for iplot := 0 to Train_Count-1 do
  begin
    with tbgtimeschedule do
    begin

```

```

//head
//col,row
colz:=iplot+1;
Alignments[colz,0]:=taCenter;
ColWidths[colz]:=160;
Cells[colz,0]:=GetTrainTypeName(Train_No[iplot].tr_type)+' '+IntToStr(colz);
//head
for itime := 1 to station_count do
begin
  Alignments[colz,itime]:=taLeftJustify;
  if itime = 1 then
    Cells[colz,itime]:=IfThen(Train_No[iplot].headstation=0,THead[0],THead[1]) + FirstClock[iplot] + ' u.'
  else if itime=station_count then
    Cells[colz,itime]:=IfThen(Train_No[iplot].headstation=0,THead[1],THead[0]) + LastClock[iplot] + ' u.'
  else
    Cells[colz,itime]:=' ':+ Train_Clock[iplot][itime-2].arrive + ' u.' + ' ':+ Train_Clock[iplot][itime-
2].leave + ' u.';
  end;
end;
end;
end;
procedure TfrmMain.CheckCrashTrain;
var
  tr1,tr2,j,st : Integer;
  x_intersect : Double;
  m1,m2,c1,c2 : Double;
  //stationD:array of Double;
begin
  SetLength(stationD,station_count);
  for st := 0 to station_count-2 do
  begin
    stationD[st]:=Train_Point[0][st].x1V;
  end;
  stationD[st]:=Train_Point[0][st-1].x2V;
  //
  //
  for tr1:=0 to Train_Count-1 do
  begin
    for tr2 :=tr1+1 to Train_Count-1 do
    begin
      //ShowMessage('tr1=' + IntToStr(tr1) + 'tr2=' + IntToStr(tr2));
      for j := 0 to station_count-2 do
      begin
        m1:=Train_Line[tr1][j].slope;
        c1:=Train_Line[tr1][j].c_const;
        //
        // ShowMessage('tr2(x1,y1)='+FloatToStr(Train_Point[tr2][j].x1V)+';'+FloatToStr(Train_Point[tr2][j].y1V));
        // ShowMessage('tr2(x2,y2)='+FloatToStr(Train_Point[tr2][j].x2V)+';'+FloatToStr(Train_Point[tr2][j].y2V));
        m2:=Train_Line[tr2][j].slope;
        c2:=Train_Line[tr2][j].c_const;
        //
        if IntersectLine(m1,m2) then
        begin
          x_intersect:=XYspot(m1,c1,m2,c2);
          //
          if (x_intersect >= stationD[j]) and (x_intersect <= stationD[j+1]) then

```

```

begin
  //ShowMessage(IntToStr(j));
  {if (j=0) or (j=station_count-2) then
    begin
      MessageDlg('กำหนดตารางเวลาผิดพลาด ทำให้เกิดการชนกันของขบวนรถไฟตั้งแต่เริ่มออกจากสถานี',mtError,[mbOK],0);
      Exit;
    end;}
  //
  if Train_No[tr2].headstation=0 then
    begin
      GenNewTime_Line_Point_Head0(tr1,tr2,j);
      //วิ่งจากสถานีปลายทาง
    end
  else
    begin
      GenNewTime_Line_Point_Head1(tr1,tr2,j);
      //วิ่งจากปลายทางไปหาสถานี
    end;
  end; //end if (x_intersect >= stationD[j]) and (x_intersect <= stationD[j+1]) then
end;
end;
end;
end;
end;
//วิ่งจากสถานีปลายทาง
procedure TfrmMain.GenNewTime_Line_Point_Head0(train1,train2,j_intersect:Integer);
var
  i,st_rows,dot,tt : Integer;
  tr,Tleave,Tarrive:String;
  each_train_time,chktime,station_distance : Double;
  Twork,Trest,Ttemp : TTimes;
  x1,x2,y1,y2 : Double;
begin
  { ShowMessage('x1=' + FloatToStr(Train_Point[train2][j_intersect].x1V));
  ShowMessage('y1=' + FloatToStr(Train_Point[train2][j_intersect].y1V));
  ShowMessage('x2=' + FloatToStr(Train_Point[train2][j_intersect].x2V));
  ShowMessage('y2=' + FloatToStr(Train_Point[train2][j_intersect].y2V));
  ShowMessage('ถึง'+ Train_Clock[train2][j_intersect-1].arrive);
  ShowMessage('ออก'+ Train_Clock[train2][j_intersect-1].leave);
  ShowMessage(FloatToStr(stationD[j_intersect]));
  ShowMessage('1ออก'+ Train_Clock[train1][j_intersect-1].leave); }
  i:=train2;
  GetDataTrain(Train_No[i].tr_type);
  Ttemp.hour:=0;
  Ttemp.minutes:=3;
  if j_intersect <> 0 then
    begin
      TotalTime:=ConvertToTTime(Train_Clock[train1][j_intersect-1].leave);
      CalcTime(Ttemp); //บวกรออีก 3 นาที;
      Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
      IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
      //ShowMessage(Tleave);
      station_distance:=stationD[j_intersect];
      Train_Clock[i][j_intersect-1].leave:=Tleave; //gen เวลาออกจากสถานีใหม่
    end
  else
end

```

```

begin
  TotalTime:=ConvertToTTime(FirstClock[train1]);
  CalcTime(Ttemp); //ບັນທຶກ 3 ນາທີ;
  Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
  FirstClock[i]:=Tleave;
  station_distance:=0;
  //ShowMessage(FirstClock[train1]);
end;
//
//
//ບັນທຶກ gen line ໂທກ ໂທກ check tm_type
for st_rows := j_intersect to station_count-2 do
begin
  x1:=station_distance;
  y1:=ConvertToXYLine(Tleave);
  //ShowMessage('x1=' + FloatToStr(x1));
  //ShowMessage('y1=' + FloatToStr(y1));
  each_train_time:=0;
  each_train_time:=each_train_time + TIME_ACC + WalkTime((Tdistance[st_rows]-KM_ACC-KM_DEACC),Vv)
+ TIME_DEACC;
  // ShowMessage('each=' + FloatToStr(each_train_time));
  tr:=FloatToStr(each_train_time);
  dot:=Pos('.',tr);
  chktme:=StrToFloat(Copy(tr,dot+1,Length(tr)-dot)) /10;
  each_train_time:=StrToFloat(FormatFloat('0',each_train_time));
  if (chktme >= 4) and (chktme <5) then
    each_train_time:=each_train_time + 1;
  //
  tt:=StrToInt(FloatToStr(each_train_time));
  Twork.hour:=tt div 60;
  Twork.minutes:=tt mod 60;
  CalcTime(Twork);
  //ShowMessage('hh:'+IntToStr(TotalTime.hour));
  //ShowMessage('mm:'+IntToStr(TotalTime.minutes));
  Tarrive:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
  station_distance:=station_distance + Tdistance[st_rows];
  x2:=station_distance;
  y2:=ConvertToXYLine(Tarrive);
  //ShowMessage('x2=' + FloatToStr(x2));
  //ShowMessage('y2=' + FloatToStr(y2));
  Train_Point[i][st_rows]:=GetPoint(x1,y1,x2,y2);
  Train_Line[i][st_rows]:=GenLines(x1,y1,x2,y2);
  //
  if st_rows < station_count-2 then
    Train_Clock[i][st_rows].arrive:=Tarrive; //*****
  //
  Trest.hour:=0;
  if st_rows<>station_count-2 then
    Trest.minutes:=StrToInt(FloatToStr(Tresttime[st_rows]))
  else
    Trest.minutes:=StrToInt(FloatToStr(Theadrestime{1}));
  //
  CalcTime(Trest);
  Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));

```

```

//
if st_rows = station_count-2 then
  LastClock[j]:=Tleave
else
  if st_rows < station_count-2 then
    Train_Clock[j][st_rows].leave:=Tleave; //*****
    //ShowMessage(Train_Clock[j][st_rows].arrive);
    //ShowMessage(Train_Clock[j][st_rows].leave);
  end; //station_count
end;
//วิธีจัดการเวลาขบวนรถไฟ
procedure TfrmMain.GenNewTime_Line_Point_Head1(train1,train2,j_intersect:Integer);
var
  i,st_rows,dot,tt : Integer;
  tr,Tleave,Tarrive:String;
  each_train_time,chktime,station_distance : Double;
  Twork,Trest,Ttemp : TTimes;
  x1,x2,y1,y2 : Double;
begin
  {ShowMessage('x1=' + FloatToStr(Train_Point[train2][j_intersect].x1V));
  ShowMessage('y1=' + FloatToStr(Train_Point[train2][j_intersect].y1V));
  ShowMessage('x2=' + FloatToStr(Train_Point[train2][j_intersect].x2V));
  ShowMessage('y2=' + FloatToStr(Train_Point[train2][j_intersect].y2V));
  ShowMessage('ถึง'+ Train_Clock[train2][j_intersect-1].arrive);
  ShowMessage('ออก'+ Train_Clock[train2][j_intersect-1].leave);
  ShowMessage('1ออก'+ Train_Clock[train1][j_intersect-1].leave); }
  //ShowMessage(FloatToStr(stationD[j_intersect]));
  i:=train2;
  GetDataTrain(Train_No[i].tr_type);
  Ttemp.hour:=0;
  Ttemp.minutes:=3;
  //
  if j_intersect <> (station_count-2) then
    begin
      TotalTime:=ConvertToTTime(Train_Clock[train1][j_intersect].leave);
      CalcTime(Ttemp); //บวกเวลาอีก 3 นาที;
      Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
      IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
      station_distance:=stationD[j_intersect+1];
      st_rows:=j_intersect;
      Train_Clock[j][j_intersect].leave:=Tleave; //gen เวลาขบวนรถไฟขบวนใหม่
    end
  else
    begin
      //ShowMessage(LastClock[train1]);
      TotalTime:=ConvertToTTime(LastClock[train1]);
      CalcTime(Ttemp); //บวกเวลาอีก 3 นาที
      Tleave:=ifThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
      IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
      LastClock[j]:=Tleave;
      //ShowMessage(LastClock[j]);
      station_distance:=stationD[j_intersect+1];
      //ShowMessage(FloatToStr(station_distance));
      st_rows:=station_count-2;
    end;
  //ShowMessage(Train_Clock[train2][j_intersect].leave);

```

```

//
//ShowMessage(IntToStr(st_rows));
while st_rows >= 0 do
begin
  x2:=station_distance;
  y2:=ConvertToXYLine(Tleave);
  //ShowMessage('x2=' + FloatToStr(x2));
  //ShowMessage('y2=' + FloatToStr(y2));
  each_train_time:=0;
  each_train_time:=each_train_time + TIME_ACC + WalkTime((Tdistance[st_rows]-KM_ACC-KM_DEACC),Vv)
+ TIME_DEACC;
  tr:=FloatToStr(each_train_time);
  dot:=Pos('.',tr);
  chktme:=StrToFloat(Copy(tr,dot+1,Length(tr)-dot)) /10;
  each_train_time:=StrToFloat(FormatFloat('0',each_train_time));
  if (chktme >= 4) and (chktme <5) then
    each_train_time:=each_train_time + 1;
  //
  tt:=StrToInt(FloatToStr(each_train_time));
  Twork.hour:=tt div 60;
  Twork.minutes:=tt mod 60;
  CalcTime(Twork);
  Tarrive:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
  //
  station_distance:=StrToFloat(FormatFloat('0.00',station_distance));
  Tdistance[st_rows]:=StrToFloat(FormatFloat('0.00',Tdistance[st_rows]));
  station_distance:=station_distance-Tdistance[st_rows];
  //
  x1:=station_distance;
  y1:=ConvertToXYLine(Tarrive);
  //ShowMessage('x1=' + FloatToStr(x1));
  //ShowMessage('y1=' + FloatToStr(y1));
  Train_Point[i][st_rows]:=GetPoint(x1,y1,x2,y2);
  Train_Line[i][st_rows]:=GenLines(x1,y1,x2,y2);
  //
  if st_rows > 0 then
    Train_Clock[i][st_rows-1].arrive:=Tarrive; //*****
  //
  Trest.hour:=0;
  if (st_rows >=1) then
    Trest.minutes:=StrToInt(FloatToStr(Tresttime[st_rows-1]))
  else
    Trest.minutes:=StrToInt(FloatToStr(Theadresttime[0]));
  //
  CalcTime(Trest);
  Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
  if st_rows=0 then
    FirstClock[i]:=Tleave
  else
    Train_Clock[i][st_rows-1].leave:=Tleave;
  // ShowMessage('(x1,y1)='+(FloatToStr(x1)+','+FloatToStr(y1));
  // ShowMessage('(x2,y2)='+(FloatToStr(x2)+','+FloatToStr(y2));
  st_rows:=st_rows-1;
end;//while st_rows >= 0 do
end;

```



```

function TfrmMain.IntersectLine(m1,m2:Double):Boolean;
begin
  Result:=(m1 <> m2);
end;
function TfrmMain.XYspot(m1,c1,m2,c2:Double):Double;
var
  x_ans,y_ans:Double;
begin
  x_ans:=(c2-c1)/(m1-m2);
  y_ans:=((-m2*c1)+(m1*c2))/(m1-m2);
  //
  x_ans:=StrToFloat(FormatFloat('0.00',x_ans));
  Result:=x_ans;
end;
procedure TfrmMain.PlotLinesGraph;
var
  iTrain,iStation,i : Integer;
  tmpline:TLineSeries;
begin
  with TrainChart do
  begin
    Title.Text.Clear;
    Title.Text.Add('กราฟแสดงตารางเวลาการเดินรถไฟ') ;
    Title.Text.Add('เส้นทาง '+ TRoot[Select_Root]);
    if SeriesCount > 0 then
      begin
        i:=SeriesCount-1;
        while i>=0 do
          begin
            Series[i].Destroy;
            i:=i-1;
          end;
        end;
      //
      GetTempPoint; //
      //
      for iTrain := 0 to Train_Count -1 do
        begin
          tmpline:=TLineSeries.Create(Self);
          tmpline.LinePen.Width:=2;
          AddSeries(tmpline);
          Series[iTrain].Title:=GetTrainTypeName(Train_No[iTrain].tr_type)+'ขบวนที่ '+IntToStr(iTrain+1);
          Series[iTrain].SeriesColor :=Train_Color[Train_No[iTrain].tr_type];
          //
          for iStation :=0 to station_count-2 do
            begin
              //Series[iTrain].AddXY(Train_Point[iTrain][iStation].x1V,Train_Point[iTrain][iStation].y1V,FloatToStr(Train_Point[iTrain][iStation].x1V),Series[iTrain].SeriesColor);

              //Series[iTrain].AddXY(Train_Point[iTrain][iStation].x2V,Train_Point[iTrain][iStation].y2V,FloatToStr(Train_Point[iTrain][iStation].x2V),Series[iTrain].SeriesColor);
              //สำหรับสถานี

              //Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x1V,Temp_Point[iTrain][iStation].y1V,['+IntToStr(iStation+1)'+],Series[iTrain].SeriesColor);

```

```

//Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x2V,Temp_Point[iTrain][iStation].y2V,['+IntToStr(iStation+2)+'],
,Series[iTrain].SeriesColor);

Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x1V,Temp_Point[iTrain][iStation].y1V,['+IntToStr(iStation+1)+'],
Train_Color[Train_No[iTrain].tr_type]);

Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x2V,Temp_Point[iTrain][iStation].y2V,['+IntToStr(iStation+2)+'],
Train_Color[Train_No[iTrain].tr_type]);
//
{if iTrain=1 then
begin
ShowMessage(IntToStr(iStation));
ShowMessage('x1=' + FloatToStr(Temp_Point[1][iStation].x1V));
ShowMessage('y1=' + FloatToStr(Temp_Point[1][iStation].y1V));
end;

Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x1V,Temp_Point[iTrain][iStation].y1V,FloatToStr(Temp_Point[iT
rain][iStation].x1V),Series[iTrain].SeriesColor);
if iTrain=1 then
begin
ShowMessage('x2=' + FloatToStr(Temp_Point[1][iStation].x2V));
ShowMessage('y2=' + FloatToStr(Temp_Point[1][iStation].y2V));
end;

Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x2V,Temp_Point[iTrain][iStation].y2V,FloatToStr(Temp_Point[iT
rain][iStation].x2V),Series[iTrain].SeriesColor);
}
end;
end;
//Series[0].XLabel[0]:= Series[0].XLabel[0] + '[n00]';
//Series[0].XLabel[Series[0].Count-1]:=Series[0].XLabel[Series[0].Count-1] + '[' + TTail[Select_Root] + ']';
end;
end;
procedure TfrmMain.GetTempPoint;
var
i,j:Integer;
begin
SetLength(Temp_Point,Train_Count,station_count-1);
for i := 0 to Train_Count-1 do
begin
for j := 0 to station_count-2 do
begin
Temp_Point[i][j].x1V := Train_Point[i][j].x1V;
Temp_Point[i][j].y1V := Train_Point[i][j].y1V;
Temp_Point[i][j].x2V := Train_Point[i][j].x2V;
Temp_Point[i][j].y2V := Train_Point[i][j].y2V;
//
Temp_Point[i][j].y1V := StrToFloat(formatfloat('0.00',GenNew_Y_Value(Temp_Point[i][j].y1V)));
Temp_Point[i][j].y2V := StrToFloat(formatfloat('0.00',GenNew_Y_Value(Temp_Point[i][j].y2V)));
end;
end;
end;
function TfrmMain.GenNew_Y_Value(y_axis:Double):Double;
var
d1,d2:Double;
t:Double;

```

```

y_str:String;
begin
y_str:=FloatToStr(y_axis);
//
if Pos('.',y_str) <> 0 then
begin
d1:=StrToInt(Copy(y_str,0,Pos('.',y_str)-1));
if Copy(y_str,Pos('.',y_str)+1,1)<>'0' then
begin
d2:=StrToInt(Copy(y_str,Pos('.',y_str)+1,Length(y_str)-Pos('.',y_str)));
d2:=(d2*100) / 60;
end
else
begin
d2:=StrToFloat(Copy(y_str,Pos('.',y_str)+2,1))/10;
end;
end
else
begin
d1:=y_axis;
d2:=0;
end;
//
t:=d1 + StrToFloat(IfThen(d2>=10,FloatToStr(d2/100),FloatToStr(d2/10)));
Result:=t;
//ShowMessage('t'+FloatToStr(t));
end;
procedure TfrmMain.qstationCalcFields(DataSet: TDataSet);
begin
with DataSet do
begin
DataSet['Torder']:=IntToStr(RecNo);
end;
end;
procedure TfrmMain.SaveTrainSchedule(filesave:String);
var
fcp,fconn : String;
i : integer;
begin
filesave:=filesave + '.tsc';
//
if FileExists(filesave) then
begin
if MessageDlg('ต้องการบันทึกทับข้อมูลเดิมหรือไม่',mtConfirmation,[mbYes,mbNo],0)=mrYes then
DeleteFile(filesave)
else
Exit;
end;
//
fcp := ExtractFilePath(application.ExeName) + 'FileSave.fs';
copyfile(pchar(fcp),pchar(filesave),True);
fconn := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' + filesave +';Persist Security Info=False';
adosave.Close ;
adosave.ConnectionString := fconn;
adosave.Open ;
adosave.Execute('delete from TBL_SCHEDULE');
//

```

```

adoSave.Execute('delete from TBL_ROOT');
adoSave.Execute('insert into TBL_ROOT values(' + IntToStr(Select_Root) + ')');
//
tblSave.Active := true;
qSave.Active:=False;
qSave.Active := True;
//
if not qSave.Eof then
with qSave do
begin
First ;
while not eof do
begin
tblSave.Append ;
for i:=0 to Fields.Count -1 do
tblSave.FieldByName(Fields[i].FieldName).Value := Fields[i].Value;
//
tblSave.Post ;
next;
end;
end;
qSave.Active := false;
tblSave.Active := false;
adosave.Connected := false;
//
ProcessingFile;//บันทึกข้อมูล
end;
procedure TfrmMain.ProcessingFile;
var
i : Integer;
begin
with ProgressFile do
begin
i:=0;
while i < Max do
begin
Position:=i;
inc(i);
end;
Position:=0;
end;
end;
procedure TfrmMain.SaveTCHClick(Sender: TObject);
begin
with SaveDlgTCH do
begin
if Execute then
SaveTrainSchedule(FileName);
end;
end;
procedure TfrmMain.OpenTrainSchedule(fileopen:String);
var
fconn : String;
fexe:String;
i : integer;
begin
if FileExists(fileopen) then

```

```

begin
  fconn := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' + fileopen +';Persist Security Info=False';
  adoOpen.Close;
  adoOpen.ConnectionString:=fconn;
  adoOpen.Open;
  //
  dmTrain.conn.Close;
  fexe:=ExtractFilePath(Application.ExeName)+'TrainSchedule.mdb';
  dmTrain.conn.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' + fexe + ';Persist Security
Info=False';
  dmTrain.conn.Open;
  dmTrain.conn.Execute('delete from TBL_SCHEDULE');
  //
  tblOpen.Active:=True;
  qOpen.Active:=False;
  qOpen.SQL.Text:='select * from TBL_SCHEDULE';
  qOpen.Active:=True;
  //
  if not qOpen.Eof then
    begin
      with qOpen do
        begin
          First;
          while not Eof do
            begin
              tblOpen.Append ;
              for i:=0 to Fields.Count -1 do
                tblOpen.FieldName(Fields[i].FieldName).Value := Fields[i].Value;
              //
              tblOpen.Post ;
              Next;
            end;
          qOpen.Active := false;
          tblOpen.Active := false;
          adoOpen.Connected := false;
        end;
      end;
    end;
  end;
end;
procedure TfrmMain.OpenTCHClick(Sender: TObject);
var
  qOpen_SCH:TDataSet;
  i : Integer;
begin
  with OpenFileDialog1 do
    begin
      if Execute then
        begin
          OpenTrainSchedule(FileName);
          qOpen.Active:=False;
          qOpen.SQL.Text:='select root from TBL_ROOT';
          qOpen.Active:=True;
          Select_Root:=qOpen.Fields[0].AsInteger;
          qOpen_SCH:=dmTrain.GetDataSet('select * from TBL_SCHEDULE');
          with qOpen_SCH do
            begin
              if RecordCount > 0 then

```

```

begin
  SetLength(Train_No,RecordCount);
  Train_Count := RecordCount;
  First;
  while not EOF do
  begin
    i:=RecNo - 1;
    Train_No[i].headstation:=FieldByName('head_station').AsInteger;
    Train_No[i].tr_type:=FieldByName('train_type').AsInteger;
    Train_No[i].tm_type:=FieldByName('time_type').AsInteger;
    Train_No[i].tr_time:=FieldByName('train_time').Text;
    Next;
  end;
end;
end;
end; //with
ProcessSchedule;
SaveTCH.Enabled:=True;
ExportExcel.Enabled:=True;
Self.Caption:='โปรแกรมจัดการเวลากาเดินทางของขบวนรถไฟ';
Self.Caption:=Self.Caption + ' [' + ExtractFileName(FileName) + '];
qOpen.Active:=False;
openfile:=True;
end; //if Execute then
end;
end;
procedure TfrmMain.StationDClick(Sender: TObject);
begin
  LoadStation;
end;
procedure TfrmMain.DistanceDClick(Sender: TObject);
begin
  LoadDistance;
end;
procedure TfrmMain.ShowTrGraphClick(Sender: TObject);
begin
  if TrainChart.SeriesCount <> 0 then
  begin
    Application.CreateForm(TfrmShowTrainGraph,frmShowTrainGraph);
    with frmShowTrainGraph do
    begin
      ShowModal;
      Destroy;
    end;
  end
  else
    MessageDlg('กราฟจัดการเวลากาเดินทางรถไฟก่อน',mtError,[mbOK],0);
end;
procedure TfrmMain.ExportTrGraphToBMPClick(Sender: TObject);
var
  fname:String;
begin
  with SaveDlgBMP do
  begin
    if Execute then
    begin
      fname:=FileName + '.bmp';
      if FileExists(fname) then

```

```

begin
  if MessageDlg('ต้องการบันทึกข้อมูลเพิ่มหรือไม่',mtConfirmation,[mbYes,mbNo],0)=mrYes then
    DeleteFile(fname)
  else
    Exit;
  end;
//
  TrainChart.SaveToBitmapFile(fname);
  ProcessingFile;
end;
end;
end;

```

```

procedure TfrmMain.ExportExcelClick(Sender: TObject);
var
  i:Integer;
begin
  with SaveDlgExcel do
    begin
      if Execute then
        begin
          if FileExists(FileName) then
            DeleteFile(FileName);
          //
          qstation.First;
          //tbgtimeschedule.Alignments[0,1]:=taLeftJustify;
          //tbgtimeschedule.Cells[0,1]:='สถานีขบวนรถไฟ';
          for i := 1 to station_count do
            begin
              with tbgtimeschedule do
                begin
                  Alignments[0,i+1]:=taLeftJustify;
                  Cells[0,i]:=qstation.FieldByName('station_name').Text;
                end;
              qstation.Next;
            end;
          //
          tbgtimeschedule.SaveToXLS(FileName);
          ProcessingFile;//เดบรู๊จ
        end;
      end;
    end;
end;

```

```

function TfrmMain.GetFirstTime(train_id:Integer):String;
var
  station_distance,each_train_time,chktime:Double;
  st_rows,dot,tt:Integer;
  tr,Tleave:String;
  Twork,Trest:TTimes;
begin
  GetDistance(Select_Root); //n-1 value Tdistance : array of Double;
  GetRestTime(Select_Root); //n-2 value Tresttime : array of Double;
  station_count:=GetStationNum(Select_Root);
  GetDataTrain(Train_No[train_id].tr_type);
  TotalTime:=ConvertToTTime(Train_No[train_id].tr_time);
  Tleave:=Train_No[train_id].tr_time;
//

```

```

if Train_No[train_id].headstation=0 then //สถานีเริ่มคันที่ ๓๓๓
begin
station_distance:=GetTotalDistance;
st_rows:=station_count - 2;
while st_rows >= 0 do
begin
each_train_time:=0;
each_train_time:=each_train_time + TIME_ACC + WalkTime((Tdistance[st_rows]-KM_ACC-
KM_DEACC),Vv) + TIME_DEACC;
tr:=FloatToStr(each_train_time);
dot:=Pos('.',tr);
chktime:=StrToFloat(Copy(tr,dot+1,Length(tr)-dot)) /10;
each_train_time:=StrToFloat(FormatFloat('0',each_train_time));
if (chktime >= 4) and (chktime <5) then
each_train_time:=each_train_time + 1;
//
tt:=StrToInt(FloatToStr(each_train_time));
Twork.hour:=tt div 60;
Twork.minutes:=tt mod 60;
DecreaseTime(Twork);
//
station_distance:=StrToFloat(FormatFloat('0.00',station_distance));
Tdistance[st_rows]:=StrToFloat(FormatFloat('0.00',Tdistance[st_rows]));
station_distance:=station_distance-Tdistance[st_rows];
//
Trest.hour:=0;
if (st_rows >=1) then
Trest.minutes:=StrToInt(FloatToStr(Tresttime[st_rows-1]))
else
Trest.minutes:=StrToInt(FloatToStr(Theadresttime(0)));
//
DecreaseTime(Trest);
Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour))+ '.' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes)) ;
//
st_rows:=st_rows-1;
end;//while st_rows >= 0 do
Result:=Tleave;
end
else //1
begin
station_distance:=0;
for st_rows := 0 to station_count-2 do
begin
each_train_time:=0;
each_train_time:=each_train_time + TIME_ACC + WalkTime((Tdistance[st_rows]-KM_ACC-
KM_DEACC),Vv) + TIME_DEACC;
tr:=FloatToStr(each_train_time);
dot:=Pos('.',tr);
chktime:=StrToFloat(Copy(tr,dot+1,Length(tr)-dot)) /10;
each_train_time:=StrToFloat(FormatFloat('0',each_train_time));
if (chktime >= 4) and (chktime <5) then
each_train_time:=each_train_time + 1;
//
tt:=StrToInt(FloatToStr(each_train_time));
Twork.hour:=tt div 60;
Twork.minutes:=tt mod 60;

```



```

DecreaseTime(Twork); //****
station_distance:=station_distance + Tdistance[st_rows];
  Trest.hour:=0;
if st_rows<>station_count-2 then
  Trest.minutes:=StrToInt(FloatToStr(Tresttime[st_rows]))
else
  Trest.minutes:=StrToInt(FloatToStr(Theadresttime[1]));
//
DecreaseTime(Trest);
Tleave:=IfThen(TotalTime.hour < 10,'0'+FloatToStr(TotalTime.hour),FloatToStr(TotalTime.hour)) + ':' +
IfThen(TotalTime.minutes < 10,'0'+FloatToStr(TotalTime.minutes),FloatToStr(TotalTime.minutes));
end;//for st_rows := 0 to station_count-2 do
  Result:=Tleave;
end;
end;
end.

```

unit ProcessingU;

```

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, ExtCtrls;
type
  TfrmProcessing = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    ProgressBar1: TProgressBar;
    IPersen: TLabel;
    Label2: TLabel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmProcessing: TfrmProcessing;
implementation
{$R *.dfm}
end.

```

unit ShowTrainGraphU;

```

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, TeeProcs, TeEngine, Chart, Series, StdCtrls, ComCtrls;
type
  TfrmShowTrainGraph = class(TForm)
    TrainChart: TChart;
    Panel1: TPanel;
    Panel2: TPanel;
    btnExportBMP: TButton;
    SaveDlgBMP: TSaveDialog;
    ProgressFile: TProgressBar;
    procedure FormShow(Sender: TObject);
    procedure btnExportBMPClick(Sender: TObject);
  private
    { Private declarations }
    procedure ProcessingFile;
  public

```

```

{ Public declarations }
procedure Plot;
end;
var
frmShowTrainGraph: TfrmShowTrainGraph;
implementation
uses dmTrainSchedule, MainU;
{$R *.dfm}
procedure TfrmShowTrainGraph.FormShow(Sender: TObject);
begin
//
Plot;
end;
procedure TfrmShowTrainGraph.Plot;
var
iTrain,iStation,i : Integer;
tmpline:TLineSeries;
begin
with TrainChart do
begin
Title.Text.Clear;
Title.Text.Add('กราฟแสดงตารางเวลาการเดินรถไฟ') ;
Title.Text.Add('เส้นทาง '+ TRoot[Select_Root]);
if SeriesCount > 0 then
begin
i:=SeriesCount-1;
while i>=0 do
begin
Series[i].Destroy;
i:=i-1;
end;
end;
//
frmMain.GetTempPoint; //
//
for iTrain := 0 to Train_Count -1 do
begin
tmpline:=TLineSeries.Create(Self);
tmpline.LinePen.Width:=2;
AddSeries(tmpline);
Series[iTrain].Title:=frmMain.GetTrainTypeName(Train_No[iTrain].tr_type)+'ขบวนที่ '+IntToStr(iTrain+1);
Series[iTrain].SeriesColor :=Train_Color[Train_No[iTrain].tr_type];
//
for iStation :=0 to station_count-2 do
begin
//Series[iTrain].AddXY(Train_Point[iTrain][iStation].x1V,Train_Point[iTrain][iStation].y1V,FloatToStr(Train_Point[iTrain][iStation].x1V),Series[iTrain].SeriesColor);
//Series[iTrain].AddXY(Train_Point[iTrain][iStation].x2V,Train_Point[iTrain][iStation].y2V,FloatToStr(Train_Point[iTrain][iStation].x2V),Series[iTrain].SeriesColor);
//
//สำหรับสถานี
//Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x1V,Temp_Point[iTrain][iStation].y1V,['+IntToStr(iStation+1)'+']',Series[iTrain].SeriesColor);

```

```

//Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x2V,Temp_Point[iTrain][iStation].y2V,[''+IntToStr(iStation+2)+'',
,Series[iTrain].SeriesColor);

Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x1V,Temp_Point[iTrain][iStation].y1V,[''+IntToStr(iStation+1)+'',
Train_Color[Train_No[iTrain].tr_type]);

Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x2V,Temp_Point[iTrain][iStation].y2V,[''+IntToStr(iStation+2)+'',
Train_Color[Train_No[iTrain].tr_type]);

//Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x1V,Temp_Point[iTrain][iStation].y1V,FloatToStr(Temp_Point[i
Train][iStation].x1V),Series[iTrain].SeriesColor);

//Series[iTrain].AddXY(Temp_Point[iTrain][iStation].x2V,Temp_Point[iTrain][iStation].y2V,FloatToStr(Temp_Point[i
Train][iStation].x2V),Series[iTrain].SeriesColor);
    end;
end;
//Series[0].XLabel[0]:= Series[0].XLabel[0] + '[สถานี]';
//Series[0].XLabel[Series[0].Count-1]:=Series[0].XLabel[Series[0].Count-1] + '[' + TTail[Select_Root] + ']';
end;
end;
procedure TfrmShowTrainGraph.btnExportBMPClick(Sender: TObject);
var
    fname:String;
begin
    with SaveDlgBMP do
        begin
            if Execute then
                begin
                    fname:=FileName + '.bmp';
                    if FileExists(fname) then
                        begin
                            if MessageDlg('ต้องการบันทึกทับข้อมูลเดิมหรือไม่',mtConfirmation,[mbYes,mbNo],0)=mrYes then
                                DeleteFile(fname)
                            else
                                Exit;
                        end;
                    //
                    TrainChart.SaveToBitmapFile(fname);
                    ProcessingFile;
                end;
            end;
        end;
end;
procedure TfrmShowTrainGraph.ProcessingFile;
var
    i : Integer;
begin
    with ProgressFile do
        begin
            i:=0;
            while i < Max do
                begin
                    Position:=i;
                    inc(i);
                end;
            Position:=0;
        end;
end;

```

```

end;
end.

```

```

unit StationU;

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;
type
  TfrmStation = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    Label1: TLabel;
    cbxroot: TComboBox;
    Label2: TLabel;
    edtstation: TEdit;
    btnsave: TButton;
    tbgStation: TDBGrid;
    dsStation: TDataSource;
    qStation: TADOQuery;
    qStationstation_name: TWideStringField;
    qStationTroot_name: TStringField;
    qStationstation_id: TAutoIncField;
    Label3: TLabel;
    edtrest: TEdit;
    Label4: TLabel;
    Label5: TLabel;
    qStationrest_time: TWideStringField;
    qStationroot_id: TIntegerField;
    procedure FormCreate(Sender: TObject);
    procedure cbxrootClick(Sender: TObject);
    procedure qStationCalcFields(DataSet: TDataSet);
    procedure tbgStationKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure btnsaveClick(Sender: TObject);
    procedure tbgStationDbClick(Sender: TObject);
    procedure edtstationKeyPress(Sender: TObject; var Key: Char);
    procedure edtrestKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
    procedure RefreshStation;
  public
    { Public declarations }
  end;
var
  frmStation: TfrmStation;
  stid : Integer;
  const
    sql_ins = 'insert into TBL_STATION(root_id,station_name,rest_time) values(%d,"%s","%s")';
    sql_upd = 'update TBL_STATION set root_id=%d,station_name="%s",rest_time="%s" where station_id=%d ';
    sql_del = 'delete from TBL_STATION where station_id=%d ';
    //
    TRoot : array[1..3] of String = ('สตูล - ปาดังเบซาร์', 'สตูล - กันตัง', 'สตูล - หาดใหญ่');
implementation
uses dmTrainSchedule;
{$R *.dfm}
procedure TfrmStation.FormCreate(Sender: TObject);
begin

```

```

dbAddNew:=True;
LoadRoot(cbxroot.Items);
end;
procedure TfrmStation.cbxrootClick(Sender: TObject);
begin
  cbxroot.Style:=csDropDownList;
end;
procedure TfrmStation.qStationCalcFields(DataSet: TDataSet);
begin
  with DataSet do
  begin
    DataSet['Troot_name']:=TRoot[FieldByName('root_id').AsInteger];
  end;
end;
procedure TfrmStation.RefreshStation;
begin
  with qStation do
  begin
    Active:=False;
    Active:=True;
    Last;
  end;
end;
procedure TfrmStation.tbqStationKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
var
  xsql : String;
begin
  case Key of
    VK_DELETE :begin
      if MessageDlg('ต้องการลบสถานี (y/n)?',mtConfirmation,[mbYes,mbNo],0)=mrYes then
        begin
          stid:=qStation.FieldByName('station_id').AsInteger;
          xsql := Format(sql_del,[stid]);
          dmTrain.RunCmd(xsql);
          RefreshStation;
        end;
      end;
end;
end;
procedure TfrmStation.btnsaveClick(Sender: TObject);
var
  s : String;
begin
  if dbAddNew then
  begin
    s:=Format(sql_ins,[Integer(TObject(cbxroot.Items.Objects[cbxroot.ItemIndex])),edtstation.Text,edttrest.Text]);
    dmTrain.RunCmd(s);
  end
  else
  begin
    s:=Format(sql_upd,[Integer(TObject(cbxroot.Items.Objects[cbxroot.ItemIndex])),edtstation.Text,edttrest.Text,stid]);
    dmTrain.RunCmd(s);
  end;
  dbAddNew:=True;
  RefreshStation;
  edtstation.Text:= "";

```

```

    edtstation.SetFocus;
end;
procedure TfrmStation.tbqStationDbfClick(Sender: TObject);
begin
    dbAddNew:=False;
    stid:=qStation.FieldByName('station_id').AsInteger;
    cbxroot.ItemIndex:=cbxroot.Items.IndexOfObject(TObject(qStation.FieldByName('root_id').AsInteger));
    edtstation.Text:=qStation.FieldByName('station_name').Text;
    edtrest.Text:=qStation.FieldByName('rest_time').Text;
end;
procedure TfrmStation.edtstationKeyPress(Sender: TObject; var Key: Char);
begin
    if Key = #13 then
        btmsave.SetFocus;
end;
procedure TfrmStation.edtrestKeyPress(Sender: TObject; var Key: Char);
const
    ch = '0123456789.';
begin
    if (Pos(Key,ch)=0) and (Key >= chr(32)) then
        Key:=#0;
end;
end.

```

unit WelcomeU;

```

interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, jpeg;
type
    TfrmImage = class(TForm)
        imgWelcome: TImage;
        Timer1: TTimer;
        procedure Timer1Timer(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    frmImage: TfrmImage;
implementation
{$R *.dfm}
procedure TfrmImage.Timer1Timer(Sender: TObject);
begin
    Self.Close;
    Self.ModalResult:=mrOK;
end;
end.

```

ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์

ธีระ รุ่งโรจน์สุวรรณ และ จักรกริสน์ กนกกันตพงษ์, “การขยายเส้นทางรถไฟภาคใต้ ตอนล่าง - กรณีเชื่อมจังหวัดสตูล”, การประชุมวิชาการวิศวกรรมโยธาแห่งชาติ ครั้งที่ 9 สาขาวิชา วิศวกรรมจราจรและขนส่ง, เพชรบุรี, ประเทศไทย, 19 – 21 พฤษภาคม 2547.

การขยายเส้นทางรถไฟภาคใต้ตอนล่าง – กรณีเชื่อมจังหวัดสตูล

Expansion of Lower Southern Railway Line – Linkage to Satun Province

ธีระ รุ่งโรจน์สุวรรณ (Theera Rungrojsuwan)

นักศึกษาระดับปริญญาโท (การขนส่ง) ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

e-mail: s4512028@maliwan.psu.ac.th

จักรกริช กนกกันตพงษ์ (Chakkrit Kanok-kantapong)

รองศาสตราจารย์ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

e-mail: kchakkri@ratree.psu.ac.th

บทคัดย่อ : การขยายเส้นทางรถไฟเชื่อมจังหวัดสตูลในภาคใต้ตอนล่างของประเทศไทย มีเหตุผลสนับสนุนจากหลายด้าน ด้านที่หนึ่ง หน่วยงานการรถไฟแห่งประเทศไทยที่ดูแลเรื่องนี้เห็นความสำคัญและบรรจุในแผนงาน การคมนาคมจังหวัดสตูลซึ่งเสียเปรียบในระบบขนส่งเชื่อมโยงที่มีอยู่โดยเฉพาะทางอากาศและทางรถไฟที่จะช่วยในการพัฒนาจังหวัด นอกจากนี้จากการสำรวจความคิดเห็นของผู้ที่อาศัยในต่างจังหวัดที่สนใจการเชื่อมด้วยรถไฟมานานหลายสิบปี บทความนี้ได้กล่าวถึงการศึกษาเส้นทางเชื่อมที่เป็นไปได้ โดยใช้สมมุติฐานที่ว่าไม่มีการเพิ่มขบวนรถไฟ เพื่อหลีกเลี่ยงการจัดการขบวนเดินรถใหม่ทั้งหมด เส้นทางที่เชื่อมได้จากสถานีหาดใหญ่ คือ ขบวนรถเร็ว รถท้องถิ่น 6 ขบวน, จากสถานีป่าดงเบขาร์ คือขบวนมาต่างรถจักรไทยที่ต้องเปลี่ยนใช้รถจักรของมณฑลเขียนขบวนรถด่วนพิเศษ 2 ขบวน จากครึ่งหรือกึ่งคันมีขบวนรถด่วน 2 ขบวนและขบวนรถเร็ว 2 ขบวน สำหรับความต้องการเดินทางหากมีมากกว่าจำนวนรถไฟที่จะนำมาขยายเส้นทางได้ และจะต้องมีการขยายเส้นทางและจัดการขบวนเดินรถใหม่ เพื่อให้รองรับเส้นทางรถไฟที่จะเชื่อมไปจังหวัดสตูลใหม่ในอนาคต

ABSTRACT : The expansion of railway line linking Satun province in the Lower South of Thailand has many supportive reasons.

Firstly, The State Railway of Thailand includes this linkage as a part of its master plan. Secondly, Satun lacks of both air and rail transportation, to assist the province development. finally, from the survey on the prominent province people, they all agree on the need of railway linkage which have been discussed for many decades. This paper has placed limited conditions for an expansion of exist routes and service frequency with no new routes or new trains. So there is no need for new scheduling. With these conditions, the feasible linkages will be Hatyai station with 2 rapid and 4 local trains, Padang Besar station with 2 international express trains by using the idle engine car for the linkage, Trang or Kantang stations with 2 express and 2 rapid trains. However, if the demand for train travel is higher than the expansion of existing routes with no new trains, there should be a detailed study on the routes, trains and the scheduling to carry this extra demand.

KEYWORDS: RAIL LINKAGE, SCHEDULE, SATUN PROVINCE, ROUTE EXPANSION

2. ความสำคัญของการเชื่อมจังหวัดสกลนครโดยรถไฟ

2.1 ปัญหา, ความสำคัญและที่มาของโครงการ

ศักยภาพได้ค่อนข้างประกอบด้วย 5 จังหวัดคือ สงขลา สตูล ปัตตานี ยะลาและนราธิวาส จึงมีความสำคัญต่อเศรษฐกิจของประเทศไทยและเป็นพื้นที่เชื่อมกับประเทศมาเลเซีย การเดินทางขนส่งสินค้าภายในพื้นที่มีระบบขนส่งสำคัญให้ความสะดวก เช่น ที่จังหวัดสงขลามีถนนหลายสายขนาด 2-4 ช่องจราจรประกอบด้วยเส้นทางหมายเลข 42, เส้นทางหมายเลข 4 เชื่อมต่อจังหวัดปัตตานี ยะลา นราธิวาส ส่วนทางรถไฟมีชุมทางรถไฟที่อำเภอหาดใหญ่แยกเป็น 2 สายคือ สายหาดใหญ่-ป่าตองบารายและสายหาดใหญ่-ยะลา-นราธิวาส ทางน้ำมีท่าเทียบเรือสากลที่จังหวัดสงขลา และทางอากาศก็มีท่าอากาศยานที่อำเภอหาดใหญ่ โดยอยู่ห่างจากตัวเมืองประมาณ 11 กม. เท่านั้น ส่วนจังหวัดครั้งก็มีถนนหลายสายเข้าสู่จังหวัด เช่น ถนนเพชรเกษมผ่านจังหวัดอื่นเป็นทางหลักในการคมนาคมจากกรุงเทพมหานครมาครั้ง ทางรถไฟมีทั้งขบวนรถโดยสารและรถสินค้าไปถึงสถานีครั้งและกันคัง ทางน้ำมีทั้งเรือที่อำเภอกันคังและปะเหลียน ทางอากาศมีท่าอากาศยานครั้งอีกด้วย ส่วนจังหวัดอื่นๆสรุปเป็นตารางที่ 1 เปรียบเทียบการบริการขนส่งทุกประเภทของ 5 จังหวัดชายแดนภาคใต้

ตารางที่ 1 สถานะภาพของระบบขนส่งของ 5 จังหวัดชายแดนภาคใต้

จังหวัด	ทางน้ำ	ทางอากาศ	ทางถนน	ทางรถไฟ
สงขลา	มี	มี	มี	มี
ปัตตานี	มี	ไม่มี	มี	มี
นราธิวาส	มี	มี	มี	มี
ยะลา	ไม่มี	มี	มี	มี
สตูล	มี	ไม่มี	มี	ไม่มี

* ปัจจุบันไม่มีการบินขนส่ง, ปิดปรับปรุงชั่วคราวเพื่อการขยายทางวิ่ง

เมื่อพิจารณาจากตารางที่ 1 มีเพียง 2 จังหวัดเท่านั้น ที่ควรเพิ่มระบบการขนส่งให้สมบูรณ์คือ ยะลากับสตูล โดยจังหวัดยะลาไม่มีพื้นที่คิดทางน้ำ จึงไม่สามารถพัฒนาการขนส่งทางน้ำได้ การขนส่งทางอากาศสามารถใช้ที่จังหวัดนราธิวาสได้ แต่จังหวัดสตูลระบบขนส่งทางอากาศ มีสนามบินในตัวเมืองจังหวัดจึงอยู่ในความดูแลของกองทัพอากาศ แต่ไม่ได้ใช้ขนานขนส่งหากต้องการเดินทาง โดยทางอากาศก็ใช้ที่สนามบินครั้งได้หรือที่สนามบินหาดใหญ่ ดังนั้นทางรถไฟน่าจะเป็นทางเลือกที่ดีที่สุด

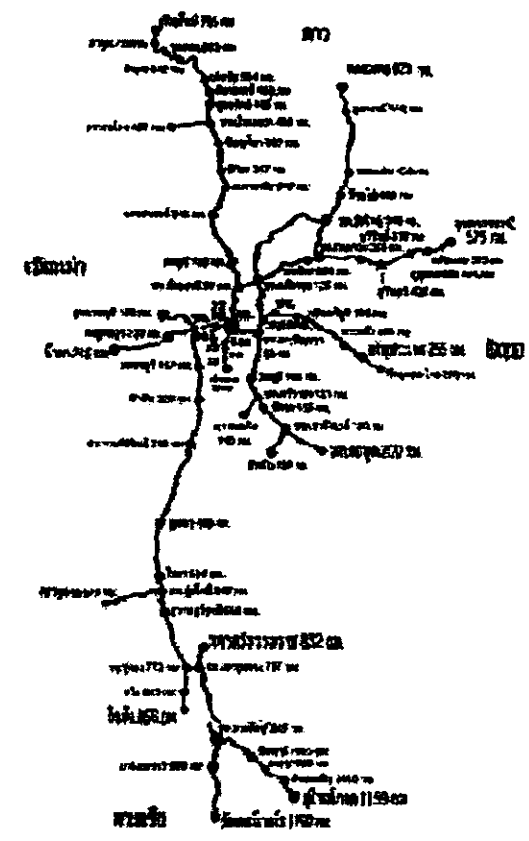
ในการพัฒนาระบบขนส่งเพิ่มเติม เพราะหากเชื่อมทางรถไฟจากสถานีหาดใหญ่มาสตูล หรือจากสถานีป่าตองบารายมาสตูลและทำต่อเชื่อมจังหวัดครั้งก็อันเนื่องกันก็จะทำให้ประชาชน 3 จังหวัดได้เกิดการติดต่อเดินทางมากขึ้น จึงเป็นการเพิ่มระบบขนส่งที่ดี เพราะอาจคุ้มค่าทางเศรษฐกิจมากกว่าการสร้างถนนเพิ่มหรือสร้างสนามบิน หากพิจารณาในเรื่องบริการขนส่งคนหรือสินค้าเชื่อมไปมาเลเซียเพิ่มอีกด้วย ก็จะทำให้เกิดการขยายตัวทางเศรษฐกิจภายในประเทศเพิ่มขึ้น โดยมีเหตุผลสนับสนุนการเชื่อมโยงโดยรถไฟหลายประเด็นดังนี้

- ก. จังหวัดสตูลต้องการเพิ่มระบบการขนส่งอีก 2 แบบ คือ ทางอากาศและทางรถไฟ (ทางอากาศใช้พื้นที่ทุนสูง)
- ข. ทางรถไฟน่าจะมีความเป็นไปได้สูง เพราะอยู่ในแผนของการรถไฟฯ (รายงานประจำปี 2541) [1] ที่จะสร้างเพิ่มในอนาคตและมีสถานีรองรับอยู่แล้ว คือที่หาดใหญ่และที่กันคัง
- ค. การพัฒนาของจังหวัดสตูลนั้นต้องการมีศักยภาพในการขนส่งสองฟากฝั่งทะเล (อันดามัน - อ่าวไทย) ซึ่งในอนาคตจะมีการก่อสร้างท่าเรือน้ำลึกที่ปากบารา เพื่อใช้ในการขนส่งทางน้ำจึงต้องการตั้งทางรถไฟ ในการขนส่งสินค้าต่อไปยังจังหวัดอื่นๆและกรุงเทพมหานคร
- ง. แนวคิดอันนี้อยู่ในใจของประชาชนจังหวัดสตูลมานานแล้วแต่ไม่มีการศึกษาอย่างจริงจัง
- จ. ได้มีการเก็บข้อมูลเบื้องต้นจากผู้นำท้องถิ่นต่างๆ เช่น สมาชิกสภาผู้แทนราษฎรจังหวัดสตูล นายกองค์การบริหารส่วนจังหวัดสตูล สมาชิกองค์การบริหารส่วนตำบลต่างๆ เทศบาลและหอการค้าจังหวัดจึงทุกฝ่ายเห็นพ้องต้องกันในการที่จะต้องมีการศึกษาเพื่อประโยชน์ของจังหวัดสตูลในอนาคตอันใกล้นี้

อย่างไรก็ตาม การพัฒนาระบบขนส่งต้องพิจารณาเส้นทางในการเชื่อมต่างๆให้เหมาะสมทั้งในด้านระยะทาง สภาพภูมิประเทศ/สังคมเศรษฐกิจ และค่าใช้จ่ายลงทุนด้วย ในเบื้องต้นควรพิจารณาจากเส้นทางเดิมที่มีอยู่และต่อเชื่อมให้เป็นวงรอบจึงจะเกิดประโยชน์ต่อจังหวัดที่เชื่อมต่อได้ง่ายและลงทุนน้อยสุด

1. เส้นทางรถไฟปัจจุบันและโครงการขยายในอนาคต

ปัจจุบันมีเส้นทางเดินรถทุกขบวนระยะทางรวมกันถึง 4,044 กิโลเมตร(รูปที่1) จากแผนที่สากล(2)เส้นทางส่วนมากมีวิศกรจะจ่ายออกไปจากกรุงเทพฯในทุกทิศทางและยังมีบางสายที่เชื่อมต่อกับการรถไฟของมาเลเซียและการรถไฟของกัมพูชา ความกว้างของรางรถไฟมีขนาด1.00 เมตรโดยรางส่วนมากขนาด 70ปอนด์ต่อทอนและมีรางขนาด80ปอนด์ต่อทอนอยู่เป็นระยะทางประมาณ1,000 กิโลเมตรเท่านั้น โดยรางรถไฟของเส้นทางสายหลักในอนาคตได้วางแผนจะใช้รางขนาด 100 ปอนด์ต่อทอน, ประมวลร้อยละ 80 ของรางใช้หมอนรองรถไฟทำด้วยไม้ที่มีหรือเป็นหมอนคอนกรีต ส่วนแผนงานในอนาคตนั้น จะได้นำเอาหมอนคอนกรีตอัดแรง (mono-block pre-stressed concrete) มาใช้ตลอดเส้นทาง เส้นทางรถไฟมีสะพานจำนวน 2,629 แห่ง หรือคิดเป็นระยะทางความยาวรวมกันประมาณ 63.5 กิโลเมตรตลอดทุกเส้นทาง



รูปที่ 1 เส้นทางรถไฟทั่วประเทศในปัจจุบันของโครงการรถไฟแห่งประเทศไทย

1.2 โครงการก่อสร้างทางรถไฟสายใหม่เพื่อการขยายเครือข่ายเส้นทางรถไฟสู่ภูมิภาค

รายละเอียดแบ่งได้ดังนี้

ก. โครงการที่ออกแบบรายละเอียดแล้วเสร็จ

- นครชัย - เชียงราย ระยะทาง 246 กิโลเมตร
- มาบตาพุด - ระยอง ระยะทาง 24 กิโลเมตร
- สุราษฎร์ธานี - พังงา (ท่าขุน) ระยะทาง 163 กิโลเมตร

ข. โครงการที่ทำการศึกษาค่าความเหมาะสมแล้ว

- บัวใหญ่ - นครพนม ระยะทาง 368 กิโลเมตร
- ระยอง - ตราด ระยะทาง 160 กิโลเมตร
- แพร่ลงบัง - ระยอง ระยะทาง 76 กิโลเมตร
- อรัญประเทศ - ปอแต - ศรีโสภณ - พนมเปญ ระยะทาง 391 กิโลเมตร

ค. โครงการที่วางแผนจะศึกษาค่าความเหมาะสม

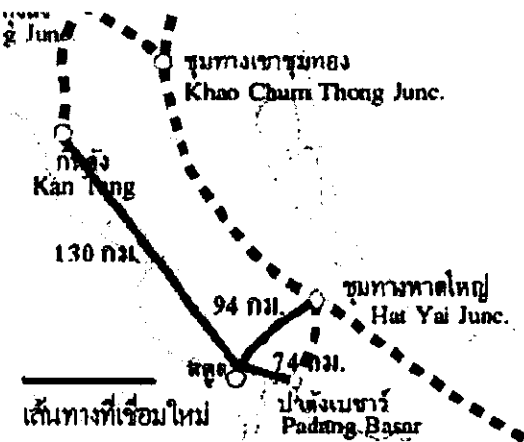
- ทางรถไฟเชื่อมต่อกับประเทศจีนตอนใต้
- ทางรถไฟเชื่อมประเทศพม่า
 - พิษณุโลก - แม่สอด ระยะทาง 200 กิโลเมตร
 - พิจิตร - แม่สอด ระยะทาง 260 กิโลเมตร
 - นครสวรรค์ - แม่สอด ระยะทาง 280 กิโลเมตร
 - น้ำตก - ค่านเจดีย์สามองค์ระยะทาง 153 กิโลเมตร
 - ไทรโยค - ค่านบองตี้
- ทางรถไฟต่อขยายสู่จังหวัดเลย
 - ลำปาง - เชียงคาน (ผ่านเพชรบูรณ์) ระยะทาง 346 กิโลเมตร
 - จักรีส - เชียงคาน (ผ่านชัยภูมิ) ระยะทาง 312 กิโลเมตร
- ทางรถไฟสายอุบลราชธานี - ชื่องม็ก ระยะทาง 70 กิโลเมตร
- ทางรถไฟเชื่อมฝั่งทะเลอันดามันกับสายสุราษฎร์ธานี - พังงา (ช่วงทับละมุ - ทับปุด)
- ทางรถไฟผ่าน จังหวัดสตูล

จะเห็นได้ว่ารถไฟมีโครงการต่างๆที่เป็นประโยชน์ต่อประเทศชาติมากมาย ผู้ศึกษาจึงขอเสนอแนวคิดในการเชื่อมโยงภาคใต้ตอนล่าง (จ.สตูล โคตรดอท) อันเป็นที่มาการศึกษาครั้งนี้

3. แนวคิดของเส้นทางเชื่อมและขบวนรถไฟที่เดินใน

ปัจจุบัน

ในการเชื่อมต่อเส้นทางรถไฟครั้งนี้มีระยะเอ็ดซึ่งผู้ศึกษาได้วางแนวเส้นทางในการเชื่อมครั้งนี้ 3 เส้นทาง (รูปที่ 2) คือหนึ่งจากอำเภอหาดใหญ่ - สตูล, สองจากตำบลปาดังเบซาร์ - สตูล, และจากอำเภอกันตัง - สตูลโดยระยะเอ็ดของพื้นที่ผ่านใน 3 เส้นทางได้ระบุในตารางที่ 2



รูปที่ 2 แนวคิดการเชื่อมเส้นทางเดินรถไฟสตูล - ตรัง - สงขลา

ตารางที่ 2 อำเภอ, จังหวัดที่ผ่านจากการเชื่อมเส้นทางรถไฟไปยังจังหวัดสตูล

ทางเชื่อม	ระยะทาง (กม.)	อำเภอ / จังหวัดที่ผ่าน
สตูล - กันตัง	130	อ.กันตัง อ.อานดาขาว อ.ปะเหลียน อ.หาดสำราญ อ.ตรัง อ.ทุ่งหว้า อ.สะงู อ.ท่าแพ อ.ควนโดน อ.เมือง อ.สตูล
สตูล - ปาดังเบซาร์	74	อ.สะเตาะ อ.ตำบลปาดังเบซาร์ อ.สงขลา อ.ควนโดน อ.ควนกาหลง อ.เมือง อ.สตูล
สตูล - หาดใหญ่	94	อ.สะเตาะ อ.หาดใหญ่ อ.ฉลุง อ.คลองหวัด อ.สงขลา อ.ควนโดน อ.ควนกาหลง อ.เมือง อ.สตูล

4. ประเภทของขบวนรถโดยสาร

การแบ่งประเภทของขบวนรถนั้นขึ้นอยู่กับประเภทของรถที่ใช้หัว ชั้นที่นั่ง การบริการและสถานที่จอด โดยประเภทของขบวนรถที่สามารถใช้ในการเดินทางประจำวันแบ่งได้เป็น 2 ประเภทคือ

4.1 ประเภทของขบวนรถที่การรถไฟแห่งประเทศไทยกำหนดไว้ให้เป็นขบวนรถชานเมืองหรือรถไฟระยะสั้นได้แก่

ขบวนรถพิเศษ มีบริการชั้น 3 ทั้งหมด การรถไฟ ฯ ได้จัดรถพิเศษวิ่งเป็นขบวนรถชานเมืองและใช้วิ่งในระยะสั้น และในปัจจุบันได้มีการจัดขบวนรถพิเศษวิ่งปรับอากาศให้บริการที่เมืองชั้น 2 ปรับอากาศ และชั้น 3 ธรรมดาใช้วิ่งระหว่างกรุงเทพ - คอนเมือง - กรุงเทพ

ขบวนรถธรรมดา มีบริการทุกสายไม่เก็บค่าธรรมเนียม มีรถโดยสารชั้น 2 และชั้น 3 แต่โดยส่วนใหญ่แล้วจะเป็นชั้น 3 ซึ่งร้อยละ 90 ของขบวนรถ ธรรมดาจะจอดรับส่งผู้โดยสารทุกสถานีและป้ายหยุดรถ การรถไฟ ฯ ได้จัดรถธรรมดาบางขบวนให้เป็นรถชานเมือง เพื่อรับส่งผู้โดยสารเดินทางระหว่างเมือง และเดินทางในเมืองเพื่ออำนวยความสะดวกแก่ผู้ที่เดินทางเพื่อเข้ามาทำงาน เรือน และปฏิบัติการกิจประจำวัน ในแหล่งงานและแหล่งชุมชนธุรกิจต่าง ๆ

4.2 ประเภทของขบวนรถที่ให้บริการทางไกลระหว่างจังหวัดในเส้นทางแต่ละสายของประเทศ

ผู้โดยสารที่ถือการเดินทางข้ามพื้นที่ปฏิบัติการกิจประจำวันสามารถใช้ขบวนรถเหล่านี้ได้ แต่ต้องเสียค่าโดยสารตามระยะทางรวมค่าธรรมเนียมและค่าบริการต่าง ๆ ตามที่การรถไฟ ฯ กำหนดไว้ ประเภทของขบวนรถชนิดนี้ได้แก่

ขบวนรถด่วนพิเศษ จอดรับส่งผู้โดยสารเฉพาะสถานีในเขตจังหวัด และสถานีใหญ่ ๆ ที่มีความสำคัญและมีชุมชนหนาแน่นเท่านั้น ความเร็วเฉลี่ย 100 กิโลเมตรต่อชั่วโมง รถด่วนพิเศษนี้จะไม่มียอดชั้น 3 ท่าง ในขบวนรถจะมีรถนอนชั้น 1 ปรับอากาศหรือโต๊ะ 10 ชั้น 2 นอนปรับอากาศหรือโต๊ะ 60 และรถนอนธรรมดาหรือโต๊ะ 30 ของขบวนรถ

ขบวนรถด่วน มีบริการคือกรุงเทพ - อุดรราชธานี, กรุงเทพ - หอนงคาย, กรุงเทพ - เชียงใหม่ และกรุงเทพ - นครศรีธรรมราชเป็นที่จอดรับส่งในอำเภอ จังหวัด และสถานีที่เป็นชุมชน ความเร็วเฉลี่ย 90 กิโลเมตรต่อชั่วโมง ตามเป้าหมายของการรถไฟ ฯ ไม่มีชั้น 3 ท่างในขบวนรถด่วนนี้ คือจะให้บริการรถชั้น 2 นอนปรับอากาศหรือโต๊ะ 50 ชั้น 2 ไม่มีเตียงทอดความต้องการของผู้โดยสาร บางขบวนจึงท่างชั้น 3 เข้าไปด้วย

รถโดยสารร่วม มีบริการในสายใต้ระหว่าง กรุงเทพฯ - สุราษฎร์ธานี และในสายอีสาน ระหว่าง กรุงเทพฯ - อุตรดิตถ์ พ่วงรถดีเซลรางชั้น 3 และชั้น 2 ปรับอากาศ

ขบวนรถเร็ว มีคืนประจำทุกสัปดาห์ เป็นขบวนรถที่จัดบริการรถนั่งชั้น 2 และชั้น 3 เป็นส่วนใหญ่ รถเร็วนี้จะจอดตามสถานีที่ป็นชุมชน เป็นอำเภอ และจังหวัดเท่านั้น

5. การเลือกขบวนเชื่อมไปจังหวัดสตูล

5.1 จำนวนขบวนรถไฟที่ผ่านอำเภอและจังหวัดภาคใต้ตอนล่าง

ในเส้นทางสายใต้มีรถไฟหลายขบวนเดินทางไป-กลับ โดยสรุปได้ดังนี้ขบวนรถด่วนพิเศษมีจำนวน 10 ขบวน ขบวนรถด่วน 4 ขบวน ขบวนรถเร็ว 10 ขบวน ขบวนรถธรรมดา 10 ขบวน ขบวนรถชานเมือง 6 ขบวน ขบวนรถท้องถิ่น 20 ขบวน ขบวนรถสินค้า 34 ขบวน ขบวนรถพิเศษระหว่างประเทศ 2 ขบวนรวมทั้งหมด 96 ขบวน และสรุปขบวนรถไฟที่กินรถผ่านอำเภอและจังหวัดที่สำคัญต่างๆออกมาได้ในตารางที่ 3

ตารางที่ 3 จำนวนขบวนรถไฟโดยสารในปัจจุบันที่กินรถผ่านจังหวัดและอำเภอต่างๆ ในภาคใต้ตอนล่าง

จังหวัด, อำเภอ	จำนวนรถไฟที่เข้า - ออก (ขบวน)
นครศรีธรรมราช	10
รุมท่างู้งง	22
กรัง	4
คันฉั่ง	2
ชุมทางหาดใหญ่	24
ปาดังเบซาร์	2
ปิลลารมี	22
ตะตา	22
สุโหงโกลก	18

5.2 รายละเอียดของเส้นทางจากสถานีชุมทางหาดใหญ่ กันฉั่งและปาดังเบซาร์ นำมาเป็นแนวคิดในการเลือกขบวนเชื่อมไปสตูล

ในการเลือกขบวนที่จะเชื่อมกันได้ สรุปรายละเอียดดังตารางที่ 4 และ 5 (สรุปกำหนดเวลาเดินทางรถสายใต้) (3) โดยมิใช่สัญลักษณ์กำกับในตารางดังนี้

- A เชื่อมสตูลได้จากปาดังเบซาร์เป็นเส้นทางหรือปลายทาง
- B เชื่อมสตูลไม่ได้เพราะ ำหนดินรถต่อไปจังหวัดอื่นๆ

C เชื่อมสตูลได้จากปาดังเบซาร์เป็นเส้นทางไปสตูลใช้พ่วงจักรที่จอดที่คับทางรถไฟ

D เชื่อมสตูลได้โดยขบวนรถไปหยุดที่สถานีและเปลี่ยนเวลาเดินรถใหม่

ตารางที่ 4 ขบวนรถโดยสารที่หยุดที่จุดเชื่อมต่างๆ (เพื่อไป)

ประเภทรถเลขที่	เส้นทาง	ปลายทาง	หยุดที่จุดเชื่อม	สรุปการเชื่อมสตูล
รถเร็ว 175	หาดใหญ่ 06.30 น.	สุโหงโกลก 09.50น.	หาดใหญ่เป็นต้นทาง	A
รถท้องถิ่น 449	หาดใหญ่ 06.55น.	สุโหงโกลก 11.10น.	หาดใหญ่เป็นต้นทาง	A
รถเร็ว 171	คท 13.00.น.	สุโหงโกลก 10.20น. วันใหม่	หาดใหญ่จอด 06.42น.	B
รถด่วนพิเศษ 35	คท 14.45.น.	มีถดอร์วอร์ 11.55น. วันใหม่	หาดใหญ่จอด 06.27น. ปาดังเบซาร์จอด 07.10น.	C
รถด่วนพิเศษ 37	คท 15.15. น.	สุโหงโกลก 10.40น. วันใหม่	หาดใหญ่จอด 07.10น.	B
รถเร็ว 169	คท 15.50. น.	ตะตา 10.30น. วันใหม่	หาดใหญ่จอด 06.43น.	B
รถท้องถิ่น 463	พิทลุง 06.10 น.	สุโหงโกลก 12.40 น.	หาดใหญ่จอด 08.10 น.	B
รถท้องถิ่น 451	นครศรีธรรมราช 06.10 น.	สุโหงโกลก 15.20 น.	หาดใหญ่จอด 10.32 น.	B
รถด่วนพิเศษ 41	คท 22.50 น.	ตะตา 13.55 น. วันใหม่	หาดใหญ่จอด 12.10 น.	B
รถท้องถิ่น 455	นครศรีธรรมราช 10.20 น.	ตะตา 17.25 น.	หาดใหญ่จอด 14.25 น.	B
รถท้องถิ่น 447	สุราษฎร์ธานี 06.20 น.	สุโหงโกลก 17.30 น.	หาดใหญ่จอด 12.50 น.	B
รถท้องถิ่น 445	ชุมพร 06.35 น.	หาดใหญ่ 16.00 น.	หาดใหญ่เป็นปลายทาง	A
รถด่วน 83	คท. 17.05 น.	ศรีง 07.35 น. วันใหม่	ศรีงเป็นปลายทาง	D
รถเร็ว 167	คท. 18.20 น.	กันฉั่ง 10.50 น. วันใหม่	ศรีงจอด 10.16 น. กันฉั่งเป็นปลายทาง	D

ซึ่งจากเนื้อหาในตารางที่ 4 และ 5 สามารถหาเส้นทางเชื่อมไปสตูลแยกเป็นกรณีดังนี้

5.2.1 การเชื่อมต่อจากหาดใหญ่ - สตูลสามารถเชื่อมได้โดยการให้หาดใหญ่เป็นต้นทาง หรือปลายทางแล้วเดินรถต่อไปยังสตูลได้ มีขบวนรถเร็ว 175, 176 รถท้องถิ่น 449, 450 และ 445, 446 รวมได้ 6 ขบวน ที่สามารถจัดการเดินรถต่อไปได้

5.2.2 การเชื่อมต่อจากปาดังเบซาร์ - สตูลสามารถเชื่อมได้โดยการให้ปาดังเบซาร์เป็นต้นทางเมื่อรถจักรจาก ขบวนรถ ข.35 มา

ถึงปาดังบขาร์แล้วจะคิดตู้โดยสารและ ไร่รถจักรของมาเลเซีย
ลากตู้โดยสารเข้าไปในมาเลเซียไทย สามารถเอารถจักรของ
ไทยลากรอดทางต่อไปยังสตูดิโอ (ข.35 ถึงปาดังเวลา 07.10
น.แต่ ข.36ออกจากปาดังเวลา 17.15 น.มีเวลาเหลือพอที่จะวิ่ง
รถไป - กลับสตูดิโอ) มีขบวนรถ 35, 36 รวม 2 ขบวน

5.2.3 การเชื่อมต่อจากครึ่งหรือกึ่งคัน - สตูดสามารถเชื่อม
ตู้โดยสารครึ่งหรือกึ่งคันจากสถานีปลายทางเดิม นั้นเป็นการ
ขยับรถไปหยุดที่สตูดแทนเป็นการต่อเส้นทางออกไป เท่านั้น
โดยไม่เพิ่มขบวนรถใหม่เพียงแต่เราจะจัดเวลาเดินรถใหม่
มีขบวนรถด่วน 83, 84 และรถเร็ว 167, 168 รวม 4 ขบวน

5.2.4 ส่วนนอกนั้นไม่สามารถจัดเดินรถไปสตูดเนื่องจาก
ต้องเดินรถต่อไปยังจังหวัดต่างๆแทน

ตารางที่ 5 ขบวนรถโดยสารที่หยุดที่จุดต่อเชื่อมต่างๆ (เที่ยวกลับ)

ประเภทรถ / สถานี	เส้นทาง	ปลายทาง	หยุดวิ่ง ที่จุด ต่อเชื่อม	สรุปการ เชื่อมสตูด
รถเร็ว 176	สุโหง โกลก 14.55 น.	หาดใหญ่ 18.00 น.	หาดใหญ่เป็น ปลายทาง	A
รถท้องถิ่น 450	สุโหง โกลก 13.25 น.	หาดใหญ่ 17.45 น.	หาดใหญ่เป็น ปลายทาง	A
รถเร็ว 172	สุโหง โกลก 11.50 น.	กท. 07.55 น. วันใหม่	หาดใหญ่จอด 17.27 น.	B
รถด่วน พิเศษ 36	บันดอร์เวอรอร์ 13.10 น.	กท. 09.55 น. วันใหม่	หาดใหญ่จอด 18.20 น. ปาดังบขาร์ จอด 17.15 น.	C
รถด่วน พิเศษ 38	สุโหง โกลก 14.05 น.	กท. 09.35 น. วันใหม่	หาดใหญ่จอด 17.40 น.	B
รถเร็ว 170	ธะลา 14.15 น.	กท. 09.05 น. วันใหม่	หาดใหญ่จอด 16.08 น.	B
รถท้องถิ่น 464	สุโหง โกลก 12.10 น.	พัทลุง 18.45 น.	หาดใหญ่จอด 17.00 น.	B
รถท้องถิ่น 452	สุโหง โกลก 08.45 น.	นครศรีฯ 18.00 น.	หาดใหญ่จอด 17.10 น.	B
รถด่วน พิเศษ 42	ธะลา 15.00 น.	กท. 05.35 น. วันใหม่	หาดใหญ่จอด 16.30 น.	B
รถท้องถิ่น 456	ธะลา 06.32 น.	นครศรีฯ 13.50 น.	หาดใหญ่จอด 09.20 น.	B
รถท้องถิ่น 448	สุโหง โกลก 06.30 น.	สุราษฎร์ธานี 18.05 น.	หาดใหญ่จอด 11.10 น.	B
รถท้องถิ่น 446	หาดใหญ่ 06.30 น.	ชุมพร 16.25 น.	หาดใหญ่เป็น ต้นทาง	A
รถด่วน 84	ครึ่ง 17.30 น.	กท. 08.35 น. วันใหม่	ครึ่งเป็น ต้นทาง	D
รถเร็ว 168	กึ่งคัน 13.10 น.	กท. 06.05 น. วันใหม่	ครึ่งจอด 13.45 น. กึ่งคัน ต้นทาง	D

ดังนั้นการขยายเส้นทาง ไปจังหวัดสตูดอาจไม่ต้องเพิ่ม
ขบวนรถใหม่ เราเพียงแต่กำหนดเวลาเดินรถใหม่เท่านั้นเพื่อให้
เชื่อมต่อ ไปยังจังหวัดสตูดได้ซึ่งเราขอคิดได้ 2 กรณีคือ

* ถ้าไม่ต้องการเพิ่มขบวนรถใหม่ เราเพียงแต่จัดเวลาเดิน
รถไปให้ขบวนรถที่มีอยู่ต่อออกไปยังจังหวัดสตูด โดยเริ่มจาก
ครึ่งหรือกึ่งคันก็ได้มีทั้งรถด่วนและรถเร็ว

** แต่ถ้าต้องการเพิ่มขบวนรถใหม่นั้นก็จะต้องจัดเวลาเดินรถ
ใหม่ทั้งหมดเป็นการอุ้งยากและไม่คุ้มต่อการลงทุนของการรถไฟ
เองด้วย จึงน โอบายของรฟท. เองก็ไม่มีนโยบายที่ให้เกิดการเดิน
รถใหม่ซึ่งในปัจจุบันที่มีอยู่ก็ เริ่มปิดการเดินรถลงไปหลายสาย
แล้วเนื่องจากการขาดทุน

เพราะฉะนั้นผู้ศึกษาจึงขอเสนอให้ไม่เพิ่มขบวนรถใหม่หรือ
การต่อขบวนรถออกไปนั้นเองในคนต่อไปจะมีการศึกษาเชิงการ
พัฒนาและความต้องการ ที่เกิดขึ้นว่ามีความจำเป็นที่ต้องเพิ่มเส้น
ทางหรือขบวนรถหรือไม่ต่อไป

สรุป

บทความนี้ได้กล่าวถึงความสำคัญของการเชื่อมจังหวัด
สตูดโดยรถไฟและเหตุผลที่สนับสนุน การเลือกเส้นทาง
เชื่อมที่เป็นไปได้ ใช้สมมุติฐานที่ไม่มีมีการเพิ่มขบวนรถไฟเพื่อ
หลีกเลี่ยงการจัดการจาดตารางเวลาเดินรถใหม่ทั้งหมด เส้นทางที่เชื่อมได้
จากสถานีหาดใหญ่ คือขบวนรถเร็ว รถท้องถิ่น 6 ขบวน, จาก
สถานีปาดังบขาร์ ต้องจัดหาคู่มาท่วงรถจักรไทยที่ต้องเปลี่ยนใช้
รถจักรของมาเลเซีย อาจเข้าไปในประเทศมาเลเซียแทนในขบวน
รถด่วนพิเศษ 2 ขบวน จากครึ่งหรือกึ่งคันมีขบวนรถด่วน 2
ขบวนและขบวนรถเร็ว 2 ขบวน สำหรับความต้องการเดินทาง
หากมีมากกว่าจำนวนรถไฟก็จะนำมาขยายเส้นทางได้ ยกตัวอย่าง
เช่นการเกิดท่าเรือน้ำลึกขึ้นที่สตูด ก็จะต้องมีการขนส่งสินค้าและ
คนเพิ่มมากขึ้นกว่าปกติ ดังนั้นจะต้องมีการขยายเส้นทางและ
จัดทำตารางเวลาเดินรถใหม่ เพื่อให้รองรับเส้นทางรถไฟที่จะ
เชื่อม ไปจังหวัดสตูดใหม่ในอนาคต

เอกสารอ้างอิง

- [1] การรถไฟแห่งประเทศไทย, รายงานประจำปี 2541, กระทรวงคมนาคม, 2541
- [2] สำนักนโยบายและแผน, แผนวิสาหกิจ พ.ศ.2540-2544, การรถไฟแห่งประเทศไทย, 2540
- [3] กองการเดินรถ ฝ่ายการเดินรถ, สมุดกำหนดเวลาเดินรถ สายใต้, การรถไฟแห่งประเทศไทย, 2545