

บทที่ 3

ระเบียบวิธีวิจัย

3.1 แนวทางการวิจัยและพัฒนา

แนวทางการวิจัยและพัฒนาของวิทยานิพนธ์ฉบับนี้ มีดังนี้

1. ศึกษาและรวบรวมข้อมูลการทำงานของระบบ

- 1.1 ศึกษาข้อมูลการใช้งานของโปรแกรม Asterisk เพื่อใช้เป็น Server VoIP ต้นแบบ
- 1.2 ศึกษาหลักการของการ Complier Card 4FXO เพื่อใช้ร่วมกับ โปรแกรม Asterisk
- 1.3 ศึกษาเพิ่มเติมการใช้ภาษา PHP ใช้เป็นตัวติดต่อเพื่อเป็นตัวแทนฐานข้อมูลในการเลือก

การช่องทางในการโทรออก

- 1.4 ศึกษาการทำงานของ PHP.AGI ซึ่งเป็นตัวเชื่อมต่อการทำงานของระบบ Asterisk มายัง

Source ภายนอก

- 1.5 ศึกษาโครงสร้างของระบบโทรศัพท์ IP-PBX
- 1.6 ศึกษาข้อมูลส่งเสริมการขายของแต่ละผู้ให้บริการ
- 1.7 ศึกษาการทำงานของระบบปฏิบัติการ Linux ซึ่งเป็นระบบปฏิบัติการหลัก

2. เครื่องมือที่ใช้ในงานวิจัย

2.1 ฮาร์ดแวร์

2.1.1 คอมพิวเตอร์ทำหน้าที่เป็น Sever ของระบบทำหน้าที่เป็นตัวเลือกเส้นทางในการโทรออกโดยอัตโนมัติ

2.1.2 คอมพิวเตอร์แบบพกพาทำหน้าที่เป็น Client สำหรับผู้ให้บริการ

2.1.3 หูฟัง Headset ใช้สำหรับส่งสัญญาณเสียงและรับสัญญาณเสียง

2.1.3 การ์ด 4FXO ทำหน้าที่เป็นช่องทางในการเลือกเส้นทางในการโทรออก

2.1.4 GSM Gateway ทำหน้าที่เป็นตัวติดต่อไปยังปลายทางที่ต้องการ

2.1.5 Router ใช้เป็นอุปกรณ์เชื่อมต่อระหว่าง Server และ Client

2.2 ซอฟต์แวร์

2.2.1 ระบบที่พัฒนาขึ้นเลือกใช้โปรแกรม Asterisk 10.8.13 โดยประกอบด้วย

2.2.2 PHP เวอร์ชัน 5.2.6 เป็นภาษาสำหรับการเขียนโปรแกรมช่วยในการตัดสินใจในการเลือกเส้นทางโทรออกของระบบ

2.2.3 MySQL เวอร์ชัน 5.0.51a ทำหน้าที่เป็นระบบจัดการฐานข้อมูลที่ใช้บน Server

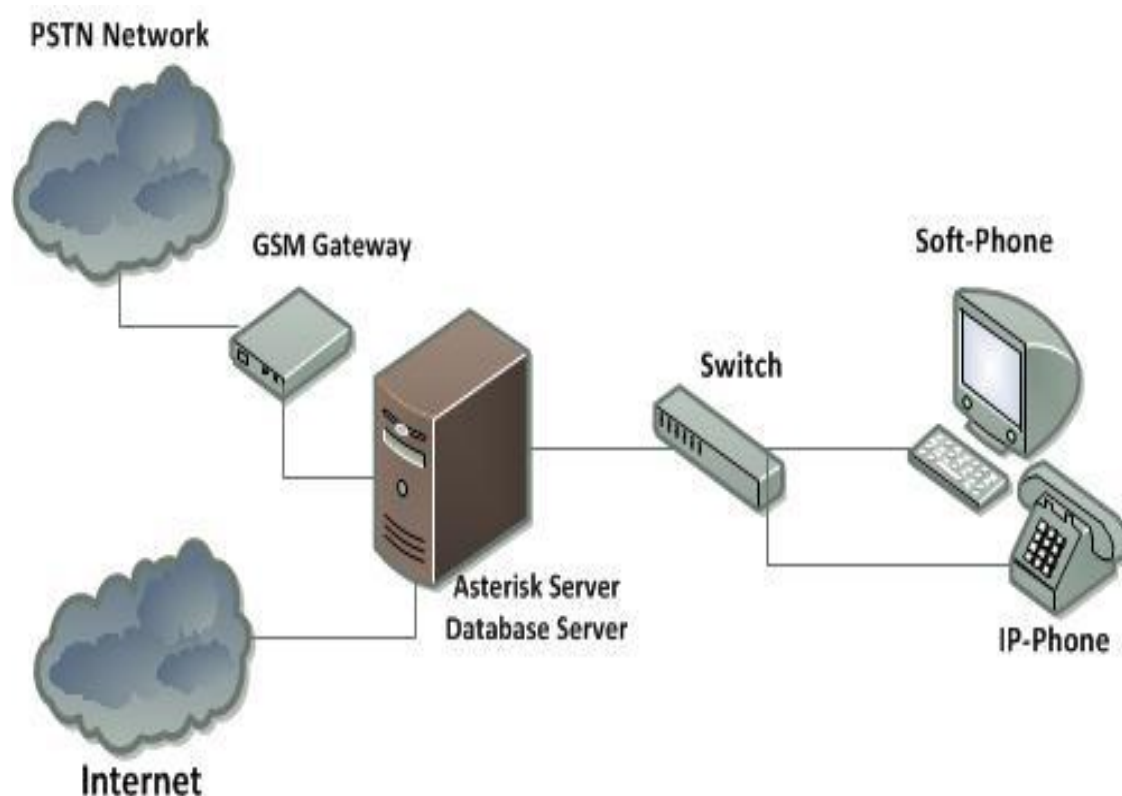
2.2.4 PHPMyadmin เวอร์ชัน 2.10.3 ทำหน้าที่เป็นตัวจัดการฐานข้อมูล

2.2.5 Linux Ubuntu เวอร์ชัน 11 ทำหน้าที่เป็นระบบปฏิบัติการหลักของ Server IP-PBX

2.2.6 Xlite เวอร์ชัน 4 ใช้เป็นโทรศัพท์แบบ Softphone

3.2 รูปแบบการเชื่อมต่อของระบบที่พัฒนา

งานวิจัยนี้ออกแบบและพัฒนา application บน VoIP Sever ต้นแบบที่สามารถเลือกเส้นทางในการโทรออกที่มีต้นทุนต่ำที่สุดโดยอัตโนมัติ เพื่อลดค่าใช้จ่ายขององค์กรที่มีความต้องการติดต่อสื่อสารทางโทรศัพท์ลักษณะของการต่ออุปกรณ์จะแสดงดังภาพที่ 3.1



ภาพที่ 3.1 การต่ออุปกรณ์ของระบบต้นแบบระบบการหาเส้นทางโทรศัพท์ที่มีต้นทุนต่ำที่สุด

จากภาพที่ 3.1 แสดงลักษณะการต่อของอุปกรณ์นั้นมีรายละเอียดแต่ละอุปกรณ์ดังนี้

3.1.1 IP-Phone, Soft-Phone ทั้งสองอุปกรณ์ทำงานในลักษณะเดียวกันที่โดยเป็น User ส่ง Client โดยอุปกรณ์ดังกล่าวนี้ใช้เลือกเบอร์โทรศัพท์สำหรับโทรออกไปยังปลายทาง

3.1.2 Switch, Router เป็นอุปกรณ์สำหรับรับส่งข้อมูลและเชื่อมต่อระหว่าง Client และ Server

3.1.3 Asterisk Server, Database Server เป็นอุปกรณ์หลักของระบบ IP-PBX โดยจะเป็นอุปกรณ์หลักในการบริหารจัดการระบบโทรศัพท์สร้างหมายเลขหรือการโทรศัพท์ติดต่อกันเนื่องจาก Asterisk Server จะทำงานเป็นชุมสายระบบ IP และยังเชื่อมต่อการโทรของระบบโทรศัพท์ไม่ว่าภายในโครงข่ายเดียวกันหรือนอกโครงข่ายในกรณีที่ต้องการโทรออกไปยังนอกโครงข่ายมี 2 กรณี ดังนี้

กรณีที่ 1 โทรออกไปยังโครงข่ายอื่นๆ หรือ PSTN (Public Switch Telephone Network) ผ่านการ์ด FXO (Foreign Exchange Office) เนื่องจากปกติแล้ว Asterisk Server จะไม่สามารถแปลงสัญญาณส่งไปยังเครือข่าย PSTN ได้จึงจำเป็นต้องใช้การ์ด FXO มาช่วยแปลงสัญญาณ

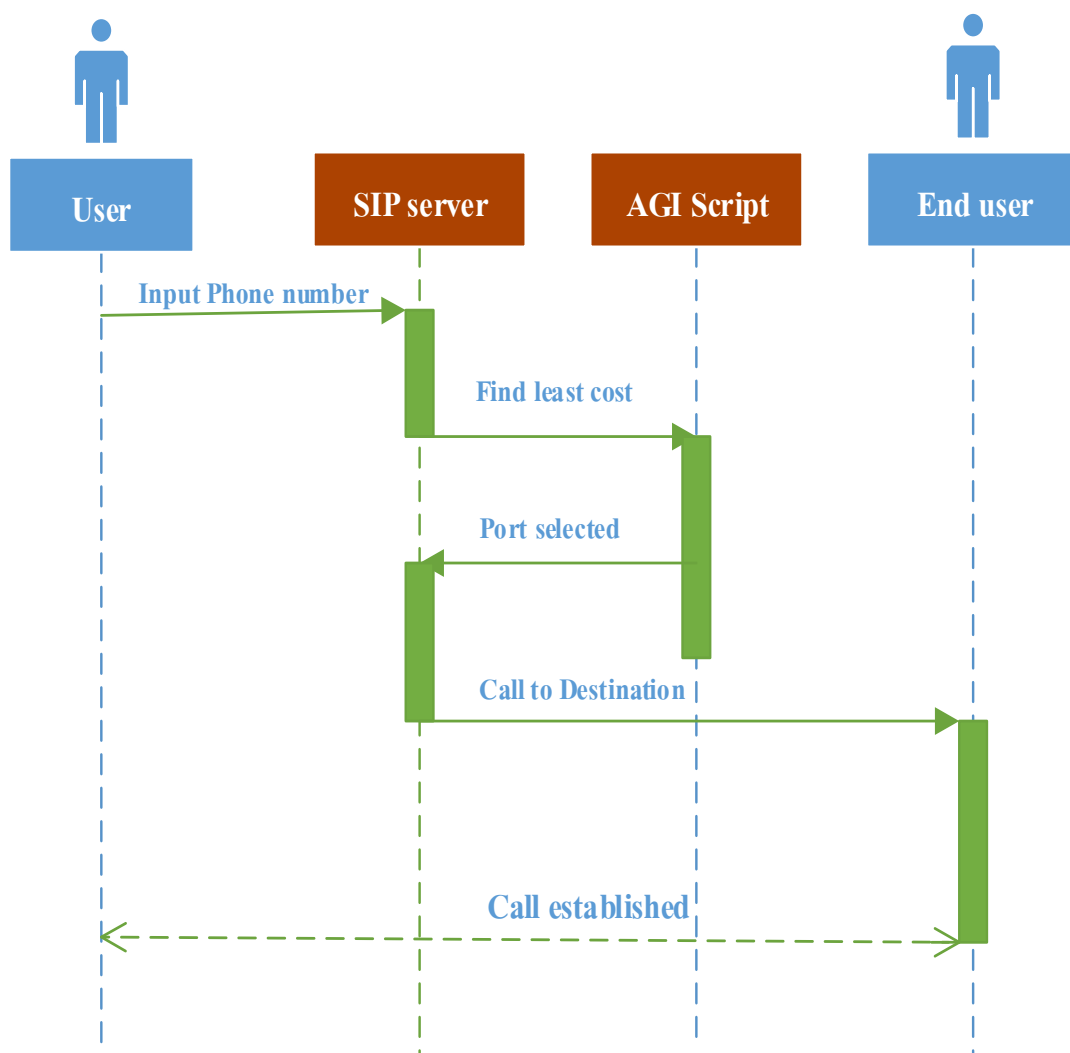
กรณีที่ 2 โทรออกไปยังโครงข่ายอื่นๆ หรือ PSTN ผ่านการ์ด LAN (Local Area Network) โดยจำเป็นต้องต่อผ่านอินเทอร์เน็ตเนื่องจากผู้ให้บริการจะมี Server อีกตัวที่เป็น SIP Trunk เพื่อโทรออกไปยังปลายทางที่เป็น PSTNให้อีกที

3.1.4 อุปกรณ์ GSM Gateway ใช้เป็นอุปกรณ์ที่ต่อเข้ากับเครือข่ายที่เป็นโทรศัพท์มือถือหลังจากการ์ด FXO ส่งสัญญาณมาให้ GSM Gateway เพื่อใช้เครือข่ายที่เป็นโทรศัพท์เคลื่อนที่โทรออกไปยังปลายทาง

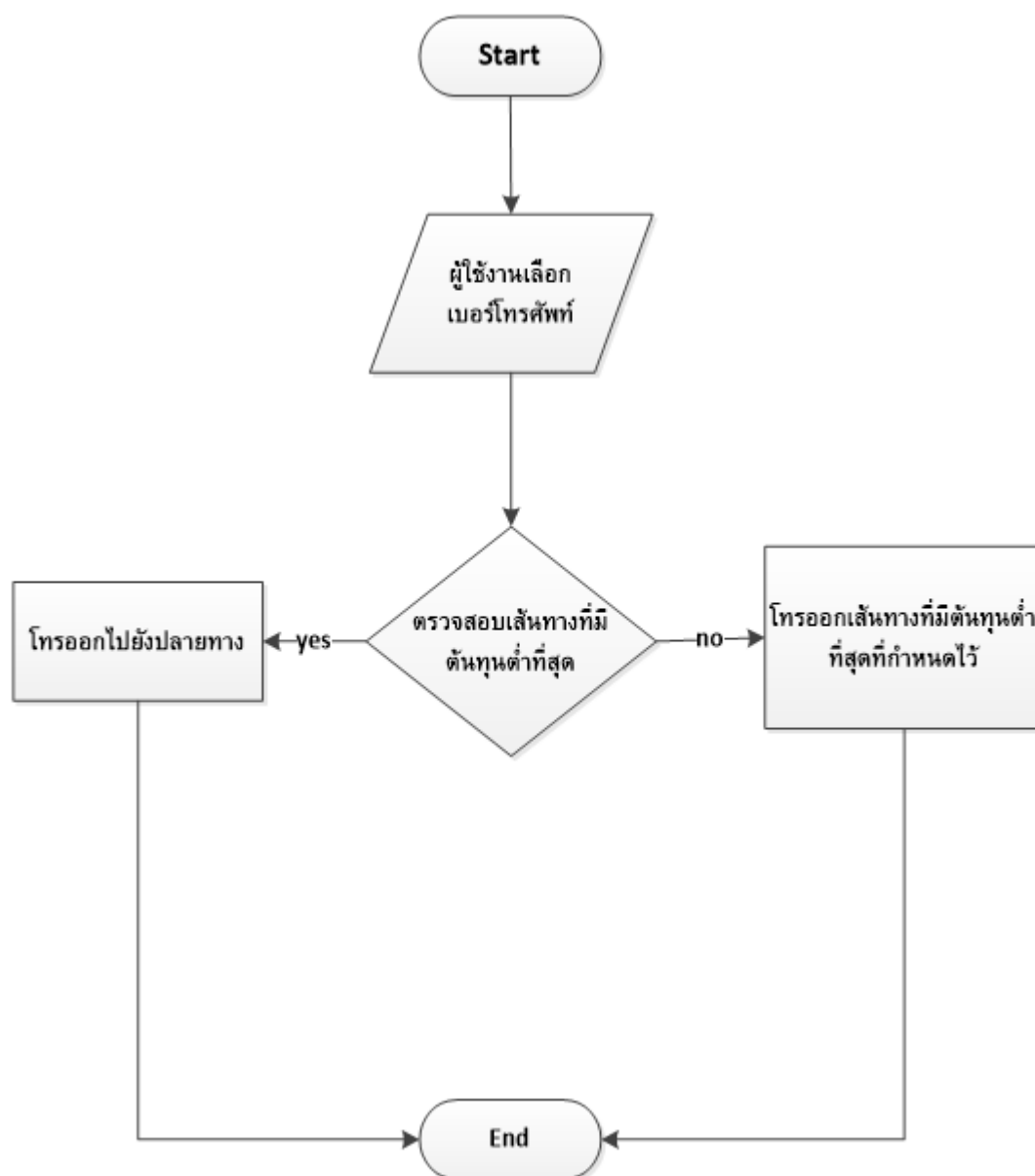
3.1.5 สายสัญญาณของโทรศัพท์พื้นฐานเพื่อต่อเข้ากับการ์ด FXO สำหรับติดต่อไปยังระบบ PSTN

3.3 หลักการทำงานของระบบที่พัฒนา

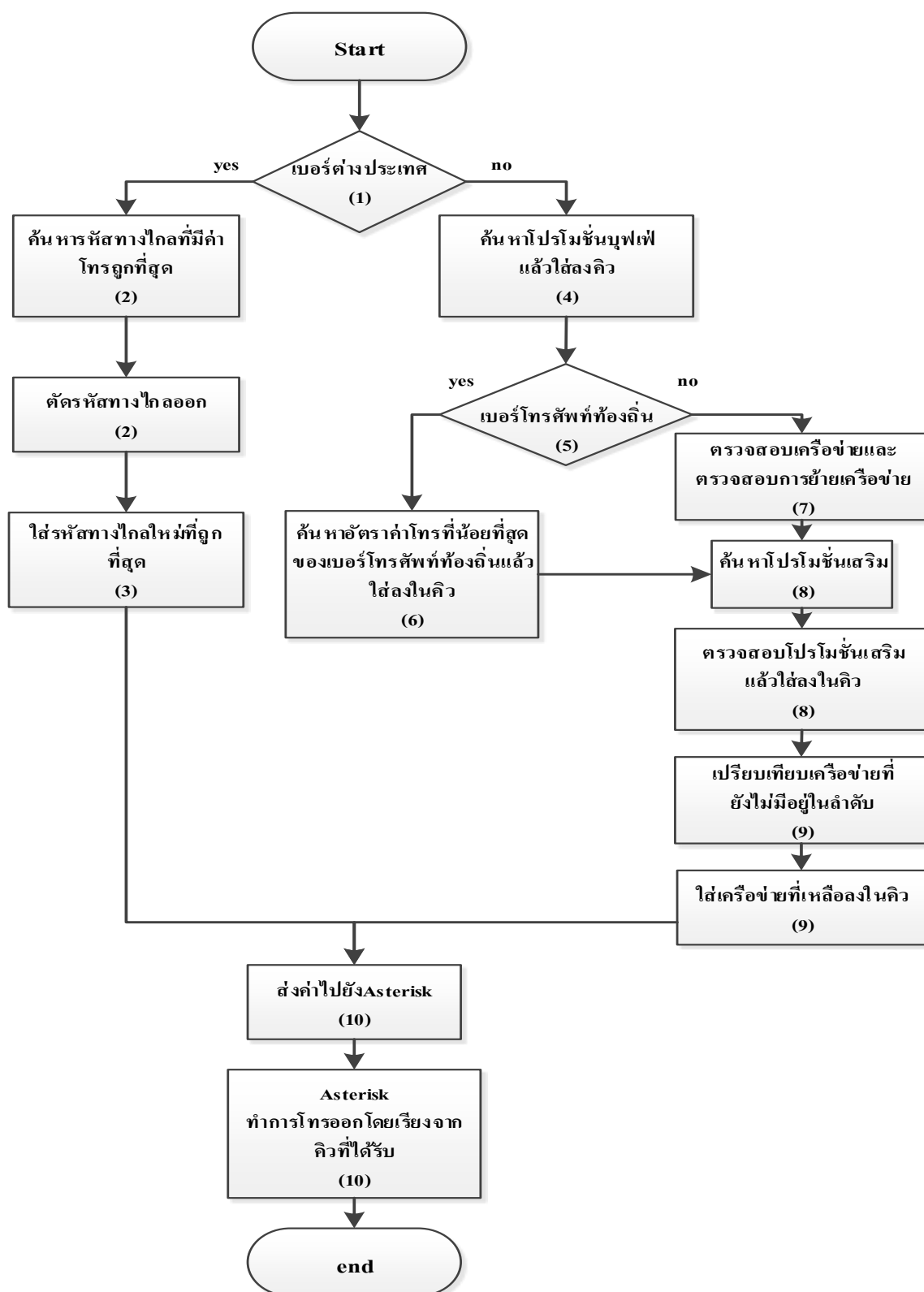
การออกแบบหลักการทำงานของระบบเริ่มจากตรวจสอบเบอร์โทรศัพท์ที่ได้รับมาจากผู้ใช้งานเพื่อให้ทราบเครือข่ายปลายทางโดยตรวจสอบจากฐานข้อมูลที่มีอยู่ หลังจากตรวจสอบเบอร์โทรศัพท์ปลายทางได้แล้วว่าเป็นเครือข่ายใดระบบจะตรวจสอบอัตราค่าโทรศัพท์ต่อนาทีว่าเบอร์โทรศัพท์ที่ได้รับมานั้นอัตราค่าโทรศัพท์เส้นทางไหนมีอัตราค่าโทรศัพท์ต่ำที่สุดแต่ในกรณีที่ไม่มีพบเบอร์โทรศัพท์ในฐานข้อมูลนั้นระบบจะตรวจสอบอัตราค่าโทรที่มีต้นทุนต่ำที่สุดแล้วส่งโทรออกไปยังปลายทางหรือเรียกอีกอย่างว่า Default gateway ตามลักษณะของ Flowchart สรุปแบบการทำงานแสดงดังภาพที่ 3.2 ภาพที่ 3.3 และการทำงานของระบบในการเลือกเส้นทางอย่างละเอียดอย่างละเอียดในภาพที่ 3.4



ภาพที่ 3.2 รูปแบบการทำงานของระบบ



ภาพที่ 3.3 Flowchart รูปแบบการทำงานของระบบ



ภาพที่ 3.4 การทำงานของระบบในการเลือกเส้นทางอย่างละเอียด

ขั้นตอนที่ 1 กลไกการแยกเบอร์โทรศัพท์ทางไกลระหว่างประเทศออกจากเบอร์โทรศัพท์อื่นๆ ลำดับแรกระบบจะรับเบอร์โทรศัพท์มาจาก Asterisk ซึ่งเป็นตัวรับค่าเบอร์โทรศัพท์จากผู้ใช้งาน หลังจากนั้นระบบจะแยกเบอร์โทรศัพท์โดยการดูตัวเลข 2 ลำดับแรกและตรวจสอบจำนวนของเบอร์โทรศัพท์ว่ามีมากกว่า 10 ตัวเลขขึ้นไป เช่น ถ้าเป็นเบอร์โทรศัพท์ทางไกลระหว่างประเทศจะขึ้นต้นด้วย 00, 10 และมีจำนวนมากกว่า 10 ตัวเลขขึ้นไป นอกจากนี้เป็นเบอร์โทรศัพท์พื้นฐานหรือโทรศัพท์เคลื่อนที่ และระบบจะตรวจสอบรหัสประเทศปลายทางโดยจะเปรียบเทียบกับรหัสประเทศที่ได้ใส่ข้อมูลไว้ในระบบเพื่อที่จะหาอัตราค่าโทรที่มีค่าบริการถูกที่สุดในแต่ละประเทศนั้นในส่วนการทำงานของ Code นั้นเริ่มจากระบบรับหมายเลขโทรศัพท์มาจาก asterisk ตัดเอาเฉพาะเลข 2 ตัวหน้าหลังจากนั้นตรวจสอบว่าหมายเลขที่ตัดมามีค่าเท่ากับเลข“08” และมียาวของหมายเลขเกิน 10 หลักหรือไม่ถ้าเกิน 10 หลักและไม่ใช่ 08 แสดงว่าเป็นหมายเลขต่างประเทศ จะตัดเลขออกมา 2 ตัวใหม่ ซึ่งก็คือรหัสประเทศโดยนำรหัสประเทศไปค้นหาในฐานข้อมูล ถ้าไม่พบระบบจะตัดตัวเลขเพิ่มเป็น 3 ตัว แล้วนำไปเปรียบเทียบในฐานข้อมูลดังภาพที่ 3.5

```

if(($preNumber2digit!='08')&&($len>10)){
    $numberingSplit2 = substr($numbering,3,2);
    $sqlCountry = mysql_query('SELECT * FROM prefixcountry WHERE countryCode
='.$numberingSplit2.' ORDER BY countryCode ASC',$objConnect) or die(mysql_error());
    $rowCountry = mysql_num_rows($sqlCountry);
    if($rowCountry==0){
        $numberingSplit2 = substr($numbering,3,3);
        $sqlCountry = mysql_query('SELECT * FROM prefixcountry WHERE countryCode
='.$numberingSplit2.' ORDER BY countryCode ASC',$objConnect) or die(mysql_error());
        $rowCountry = mysql_num_rows($sqlCountry);
        if($rowCountry==0){
            $portOrder = 'SELECT * FROM prefixcountry GROUP BY sp ORDER BY
countryCost ASC';
            $objOrder = mysql_query($portOrder,$objConnect);

```

ภาพที่ 3.5 Code ที่ใช้ตรวจสอบว่าเบอร์โทรศัพท์ที่ได้รับมานั้นเป็นเบอร์โทรศัพท์ที่โทรออกไปยังต่างประเทศ

```

$resultPrefix = mysql_fetch_array($objOrder);
$prefixNew = $resultPrefix['prefixDistance'];

$cutPrefix = substr($numbering,3);

//***** New Country Number *****

$newCountryNumber= $prefixNew.$cutPrefix;
//echo $newCountryNumber;

$sqlLeastCost = mysql_query('SELECT `port` FROM `normalrate` order by
perMinasc',$objConnect);

while($resultLeastCost=mysql_fetch_array($sqlLeastCost)){
    if($sportSP=="")
        $sportSP = $resultLeastCost['port'];
    else
        $sportSP = $sportSP.'-'.$resultLeastCost['port'];
}

echo $newCountryNumber.'-'.$sportSP;

exit();

```

ภาพที่ 3.5 (ต่อ)

ขั้นตอนที่ 2 กลไกการหารหัสโทรทางไกลระหว่างประเทศที่มีค่าใช้จ่ายถูกที่สุดวิธีการโทรทางไกลระหว่างประเทศนั้นระบบจะพิจารณาจากรหัสโทรทางไกล 3 ตัวแรกแล้วโทรออกไปยังผู้ให้บริการโทรศัพท์ทางไกลระหว่างประเทศ ซึ่งค่าใช้จ่ายนั้นจะเป็นไปตามเครือข่ายต่าง ๆ ตามรหัสขึ้นต้นและ ตรวจสอบรหัสประเทศปลายทางที่ต้องการจะโทรออก เพื่อให้ได้อัตราค่าโทรศัพท์ที่มีค่าบริการถูกที่สุดตามประเทศที่ต้องการจะโทรออก โดยระบบจะคำนวณเลือกผู้ให้บริการที่มีต้นทุนต่ำที่สุดจากข้อมูลราคาของผู้ดูแลระบบกรอกเข้าในระบบ และสามารถเพิ่มลดได้ตามต้องการ โดยระบบจะนำข้อมูลดังกล่าวมาจัดอันดับตามค่าใช้จ่ายเรียงลำดับจากต้นทุนต่ำที่สุดไปหาต้นทุนสูงสุดลักษณะการทำงานของ Code การตรวจสอบรหัสทางไกลทั้ง 3 จะเริ่มจากเมื่อได้รับรหัสโทรทางไกลต่างประเทศ 3 หลักมาจากรฐานข้อมูล ระบบจะตัดรหัสโทรทางไกลระหว่างประเทศอันเดิมออก หลังจากนั้นใส่รหัสโทรทางไกลใหม่เข้าไปแทนที่รหัสที่ตัดออก กลายเป็นหมายเลขใหม่แล้วเก็บค่าหมายเลข port แรกที่ได้มาไว้ในตัวแปรหลังจากนั้นเรียงลำดับ port ที่เหลือตามอัตราค่าโทร

ที่ถูกตามลำดับ จึงนำมาเก็บไว้ในตัวแปรแล้วส่งหมายเลขใหม่พร้อมหมายเลข port ให้ asterisk โทรออกไปยังปลายทางดังภาพที่ 3.6

```
$resultPrefix = mysql_fetch_array($sqlCountry);

    $prefixNew = $resultPrefix['prefixDistance'];
    $cutPrefix = substr($numbering,3);

    /******* New Country Number *****/

    $newCountryNumber= $prefixNew.$cutPrefix;
    echo $newCountryNumber.';';

    $portSP =$resultPrefix['sp'];
    $portSP = spToPortNo($portSP);
    $sqlLeastCost = mysql_query('SELECT `port` FROM `normalrate`
where `sp`!="'.$resultPrefix['sp'].'" order by perMinasc',$objConnect);
    while($resultLeastCost=mysql_fetch_array($sqlLeastCost)){
        $portSP = $portSP.'-'.$resultLeastCost['port'];
    }

    echo $portSP;
    exit();
```

ภาพที่ 3.6 Code คัดรหัสโทรทางไกล 3 หลักหน้าออกแล้วใส่รหัสโทรทางไกลระหว่างประเทศที่มีอัตราที่ต่ำที่สุดในขณะนั้นลงไปใหม่

ขั้นตอนที่ 3 กลไกการเปลี่ยนรหัสโทรทางไกลระหว่างประเทศและนำรหัสโทรทางไกลระหว่างประเทศที่ถูกที่สุดใส่ลงไปใหม่กลไกนี้จะคัดรหัสโทรทางไกลระหว่างประเทศ 3 หลักแรกออก หลังจากนั้นระบบจะเปลี่ยนรหัสโทรทางไกลระหว่างประเทศ โดยนำรหัสที่จัดอันดับค่าโทรศัพท์ระหว่างประเทศที่มีต้นทุนต่ำที่สุดใส่ลงไปแทนค่าเก่า ซึ่งวิธีนี้จำเป็นต้องเรียงลำดับเครือข่ายที่ใช้ในการโทรออกเหมือนโทรศัพท์ประเภทอื่น แต่จะแตกต่างตรงที่วิธีการคิด

ค่าบริการจะตรวจสอบจากรหัสโทรทางไกลระหว่างประเทศ ไม่ได้ดูจากโครงข่ายที่ใช้โทรออก ลักษณะการตัดรหัสทางไกล และหน้าตาสำหรับใส่รหัสทางไกลต่างประเทศและตรวจรหัสทางไกลที่มีอยู่ในระบบดังภาพที่ 3.7

เพิ่ม ลบ รหัสทางไกลต่างประเทศ

เครือข่าย :

รหัสโทรข้ามประเทศ :

รหัสประเทศ :

อัตราค่าโทร(บาท/นาที) :

เครือข่าย	รหัสโทร	รหัสประเทศ	อัตราค่าโทร	ลบ
ais	003	65	3	
ais	003	60	4	

ภาพที่ 3.7 หน้าตาการใส่ข้อมูลรหัสทางไกลระหว่างประเทศ

ขั้นตอนที่ 4 กลไกการตรวจสอบว่ามีโปรโมชั่นบุฟเฟต์หรือไม่กรณีที่ไม่ใช่การโทรทางไกลระหว่างประเทศ ระบบจะตรวจสอบข้อมูลในระบบว่ามีโปรโมชั่นบุฟเฟต์หรือไม่ โดยปกติแล้วโปรโมชั่นบุฟเฟต์จะจำกัดเป็นช่วงเวลาสามารถโทรออกได้ไม่จำกัดเครือข่ายและระยะเวลา เช่น ถ้าโปรโมชั่นตั้งแต่ 08.00-17.00 น. ของเครือข่ายดีแทค ระบบจะเลือกเครือข่ายดีแทคใส่เข้าไปในคิวการจัดอันดับ ถ้าไม่มีโปรโมชั่นบุฟเฟต์ ระบบก็จะไม่ได้เลือกเครือข่ายใดลงไปในคิวโดยลักษณะการใส่ข้อมูลโปรโมชั่นนั้นแสดงตามภาพที่ 3.8

โปรโมชั่นเสริม

1 ชื่อรายการ :

2 เครือข่าย :

3 วัน : ☒ จ. ☒ อ. ☒ พ. ☒ พท. ☒ ศ. ☒ ส. ☒ อา.

4 วัน-เวลาเริ่มต้น :

5 วัน-เวลาสิ้นสุด :

6 อัตราค่าโทร : บาท/นาที 7

8

ภาพที่ 3.8 หน้าต่างการใส่รูปแบบโปรโมชั่น

การใส่ข้อมูลโปรโมชั่นเริ่มจาก

1. ใส่ชื่อของโปรโมชั่น
2. เครือข่ายที่ใช้สำหรับโทรในโปรโมชั่นนี้
3. วันสำหรับระบุว่าโปรโมชั่นนี้ใช้ได้ในวันอะไรบ้างเนื่องจากบางโปรโมชั่นจะมีการระบุวันสำหรับการใช้งานไว้ เช่น วันจันทร์ถึงวันศุกร์ โทรฟรี เป็นต้น
4. วันเวลาเริ่มต้น ที่สามารถใช้งานโปรโมชั่นนี้ได้
5. วันเวลาสิ้นสุด ที่เราสามารถใช้งานโปรโมชั่นนี้ได้
6. อัตราค่าโทร
7. หน่วยของค่าโทรศัพท์
8. ปุ่มสำหรับบันทึกการเพิ่มโปรโมชั่น

ขั้นตอนที่ 5 กลไกการตรวจสอบหมายเลขโทรศัพท์ท้องถิ่นระบบจะตรวจสอบหมายเลขโทรศัพท์ท้องถิ่นจากการดูรหัส 3 หลักแรกจากข้อมูลที่อยู่และระบบระบุซึ่งจะสามารถปรับเปลี่ยนได้ที่หน้าเว็บเพจสาเหตุที่ต้องระบุเบอร์โทรศัพท์ท้องถิ่นลงไปในระบบใช้สำหรับกรณีที่ต้องการย้าย server ไปใช้งานในต่างจังหวัดระบบก็จะสามารถใช้งานเบอร์โทรศัพท์ที่เป็นเบอร์ท้องถิ่นนั้นๆ ได้อย่างสะดวกและไม่ได้ต้องแก้ไขที่ Code อีก ลักษณะหน้าตาสำหรับใส่รหัสโทรศัพท์ท้องถิ่นแสดงดังภาพที่ 3.9 และการทำงานของ Code เริ่มจากค้นหาหมายเลขขึ้นต้นของเบอร์โทรศัพท์ที่ได้บันทึกไว้ในฐานข้อมูล เป็นหมายเลขใดและมีกี่หลัก หลังจากนั้นตัดหมายเลขโทรศัพท์ที่ได้รับมาจาก asterisk ตามจำนวนหลักที่ได้จากฐานข้อมูลแล้วเปรียบเทียบกับหมายเลขที่ตัดได้และหมายเลขที่ได้จากฐานข้อมูลว่าตรงกันหรือไม่ถ้าตรงกันจะกำหนดหมายเลข port แรกให้เป็น port local และหาหมายเลข port ที่เหลือเรียงตามอัตราค่าโทรจึงส่งหมายเลข port ที่เรียงลำดับแล้วให้ asterisk โทรออกไปยังปลายทาง รูปแบบ Code ดังภาพที่ 3.10

ภาพที่ 3.9 หน้าตาการสำหรับใส่รหัสโทรศัพท์ท้องถิ่น

```

$resultPrefix = mysql_fetch_array($sqlCountry);

    $prefixNew = $resultPrefix['prefixDistance'];
    $cutPrefix = substr($numbering,3);
    $portSP = $resultPrefix['sp'];

    //***** New Country Number *****

    $newCountryNumber= $prefixNew.$cutPrefix;
    echo $newCountryNumber.';';

    $sqlLeastCost = mysql_query('SELECT `port` FROM `normalrate`
where `sp`!="'.$resultPrefix['sp'].'"' order by perMinasc',$objConnect);

    while($resultLeastCost=mysql_fetch_array($sqlLeastCost)){
        $portSP = $portSP.'-'.$resultLeastCost['port'];
    }

    echo $portSP;

```

ภาพที่ 3.10 Code ตรวจสอบหาเบอร์โทรศัพท์ท้องถิ่น

ขั้นตอนที่ 6 กลไกการหาอัตราค่าโทรที่ถูกที่สุดของเบอร์โทรศัพท์ท้องถิ่น ระบบจะตรวจสอบค่าโทร โดยดูจากข้อมูลจากผู้ดูแลระบบเป็นผู้ระบุ สาเหตุที่ต้องแยกเบอร์โทรศัพท์ท้องถิ่น ออกนั้นเนื่องจากการคิดค่าบริการไม่เหมือนกับโทรศัพท์แบบอื่น เนื่องจากถ้าเป็นโทรศัพท์ท้องถิ่น นั้นจะมีบางเครือข่ายที่คิดค่าบริการเป็น บาท/ครั้ง ไม่ใช่เป็น บาท/นาที หลังจากได้ค่าโทรที่ถูกที่สุด แล้วจะใส่เครือข่ายลงไปในคิวการจัดอันดับ

ขั้นตอนที่ 7 กลไกการตรวจสอบเบอร์โทรศัพท์ว่าเป็นของเครือข่ายใดและตรวจสอบการย้ายค่ายโดยใช้เบอร์เดิม (Mobile number portability) ระบบจะตรวจสอบจากเบอร์โทรศัพท์ที่ต้องการโทรออกว่าเป็นเครือข่ายใด โดยตรวจสอบจากฐานข้อมูลที่มีอยู่และสามารถเพิ่ม ลด หรือ แก้ไขเครือข่ายในกรณีที่มีการย้ายเครือข่าย โดยใช้เบอร์โทรศัพท์เดิมได้จากผู้ดูแลระบบ ลักษณะการใส่เบอร์โทรศัพท์และเบอร์โทรศัพท์เคลื่อนที่ ที่ย้ายค่ายดังภาพที่ 3.11 และภาพที่ 3.12 โดยการทำงานของ Code จะเริ่มจากตรวจสอบหมายเลขโทรศัพท์ที่ตัดเฉพาะ 2 ตัวหน้าออกว่ามีค่าเท่ากับ “08” หรือ “09” หรือไม่ถ้าใช่ นำหมายเลขที่ได้รับมา ไปเปรียบเทียบกับฐานข้อมูลที่เก็บหมายเลขย้ายเครือข่ายถ้ามีหมายเลขในฐานข้อมูล จะดึงค่าชื่อเครือข่ายใหม่ที่เปลี่ยนไปมาเก็บไว้เป็น

หมายเลข port แรกในลำดับภาพที่ 3.13 และ Code ตรวจสอบเครือข่ายเบอร์โทรศัพท์เคลื่อนที่เริ่มจากเรียกดูข้อมูลของช่วงหมายเลขเริ่มต้นและหมายเลขสิ้นสุดของแต่ละแถวจากตารางที่ chk_number ที่เก็บไว้ในฐานข้อมูลแล้วจึงเปรียบเทียบหมายเลขที่ได้รับมาจาก asterisk กับช่วงของหมายเลขจากฐานข้อมูลถ้าพบว่าหมายเลขที่ได้รับมาอยู่ในช่วงที่กำหนด ให้เก็บค่าชื่อผู้ให้บริการของช่วงนั้นมาไว้ในตัวแปรตรวจสอบเครือข่ายของโทรศัพท์เคลื่อนที่ดังภาพที่ 3.14

การเพิ่มเบอร์ใหม่

หมายเลขที่ต้องการเพิ่ม :

ผู้ให้บริการ : Dtac ▼

เพิ่มเบอร์ใหม่

*สามารถเพิ่มทีละหมายเลข หรือ เป็นช่วงได้ เช่น 0887405000-0887405099

ภาพที่ 3.11 หน้าต่างสำหรับใส่เบอร์โทรศัพท์

การปรับเปลี่ยนเครือข่ายผู้ให้บริการ

หมายเลขที่ต้องการเปลี่ยน :

จากผู้ให้บริการรายเดิม : Dtac ▼

ผู้ให้บริการรายใหม่ : AIS ▼

บันทึกเปลี่ยนแปลง

ภาพที่ 3.12 หน้าต่างสำหรับเปลี่ยนแปลงข้อมูลเบอร์โทรศัพท์เคลื่อนที่

```

if($preNumber2digit=='08' && $len==10){

    $sqlMoveSp = mysql_query('SELECT * FROM movesp WHERE
numbering="'. $numbering ."', $objConnect) or die(mysql_error());

    $rowMove = mysql_num_rows($sqlMoveSp);

    if($rowMove!=0){

        $resultSP = mysql_fetch_array($sqlMoveSp);
        $portSP = $resultSP['newSP'];
    }
}

```

ภาพที่ 3.13 Code ตรวจสอบว่ามีการย้ายเครือข่ายของเบอร์โทรศัพท์เคลื่อนที่

```

$selectSql = "SELECT prefixNumberStart,prefixNumberStop,sp FROM chk_number ORDER
BY prefixNumberStart ASC";

    $objsql = mysql_query($selectSql,$objConnect) or
die(mysql_error());

    while($result=mysql_fetch_array($objsql)){
        $sNumber = $result['prefixNumberStart'];
        $eNumber = $result['prefixNumberStop'];

        if($numbering>=$sNumber&&
$numbering<=$eNumber){

            $portSP = $result['sp'];
        }
    }

```

ภาพที่ 3.14 Code สำหรับตรวจสอบว่าเบอร์โทรศัพท์เคลื่อนที่อยู๋เครือข่ายใด

ขั้นตอนที่ 8 กลไกการตรวจสอบTopping (โปรโมชันเสริม) เนื่องจากบางเครือข่ายผู้ให้บริการมีโปรโมชันเสริมแบบโทรในเครือข่ายเดียวกันฟรี หลังจากระบบได้ตรวจสอบเบอร์โทรศัพท์แล้วว่าเป็นเครือข่ายใด จากนั้นจะทำการจับคู่เครือข่ายที่มีโปรโมชันเสริมแล้วใส่เครือข่ายนั้นลงไปในการจัดอันดับเพื่อใช้สำหรับโทรออก และโดยวิธีการทำงานของโปรแกรมจะเริ่มจากตรวจสอบว่ามีชื่อผู้ให้บริการอยู่ในตาราง promotion ในฐานข้อมูลหรือไม่ถ้าพบว่ามี จะเปรียบเทียบค่าวันและช่วงเวลาปัจจุบันกับวันและช่วงเวลาที่ได้กำหนดไว้อยู่ในเงื่อนไขหรือไม่ถ้าอยู่ในเงื่อนไข จะเก็บชื่อผู้ให้บริการมาไว้ในหมายเลข port อันดับแรกรูปแบบของ Code แสดงในภาพที่ 3.15

```
$sqlPromo = 'SELECT * FROM promotion WHERE sp = "'.$portSP.'"';
$objPromo = mysql_query($sqlPromo);
$rowsPromo = mysql_num_rows($objPromo);
if($rowsPromo!=0){
    while($resulrPromo = mysql_fetch_array($objPromo)){
        $days = explode(',',$resulrPromo['days']);
        $startTime = $resulrPromo['startTime'];
        $endTime = $resulrPromo['endTime'];
        if((strtotime($timesNow)>=strtotime($startTime))&&(strtotime($timesNow)<=strtotime($endTime))){
            if(in_array($daysNow, $days)){
                $portSP = $resulrPromo['sp'];
                $inTime = 'yes';
                break;
            }else{
                $inTime = 'no';
                if($inTime=="")
                    $inTime = 'no';
            }
        }
    }
}
```

ภาพที่ 3.15 Code ที่ใช้ตรวจสอบโปรโมชันเสริม

ขั้นตอนที่ 9 กลไกการเทียบเครือข่ายที่ยังไม่มีอยู่ในคิวการจัดอันดับกลไกจะตรวจสอบเครือข่ายที่เหลือที่ยังไม่ได้อยู่ในคิวการจัดอันดับการโทรออก โดยตรวจสอบเครือข่ายที่ยังไม่ได้ถูกเพิ่มลงไปคิว ด้วยวิธีตรวจสอบเครือข่ายที่ถูกใช้งานไปแล้ว หลังจากนั้นจะเรียงลำดับตามอัตราค่าใช้จ่ายที่น้อยที่สุดตามที่ผู้ดูแลระบบระบุโปรโมชันไว้ในระบบ ซึ่งเพื่อให้ระบบทำงานให้มีประสิทธิภาพยิ่งขึ้นระบบจะนำวิธีการตรวจสอบจำนวนการใช้งานของแต่ละช่องทางแล้วนำไปเปรียบเทียบกับค่าเริ่มต้นที่ผู้ดูแลระบบเป็นคนระบุไว้ด้วยเนื่องจาก ในบางกรณีโปรโมชันที่ใช้งานนั้นอาจอัตราค่าใช้จ่ายไม่เท่ากัน เช่น หลังจากใช้งานไปจำนวน 300 นาทีแล้ว อัตราค่าโทรศัพท์จะเปลี่ยนเป็น 3 บาท จากที่เริ่มต้น 1 บาท จึงจำเป็นต้องนำจุดนี้ไปคำนวณในระบบเพื่อให้ได้ข้อมูลเพื่อนำไปค้นหาเส้นทางที่มีอัตราค่าโทรศัพท์ต่ำที่สุดในเวลานั้น และหลังจากคำนวณเสร็จแล้วระบบจะเรียงลำดับแล้ว ก็จะส่งข้อมูลให้ Asterisk ต่อไปซึ่งตัวอย่าง Code ที่ใช้สำหรับตรวจสอบการใช้งานของแต่ละเครือข่ายจะแสดงดังภาพที่ 3.16 และ Code ที่ใช้สำหรับตรวจสอบอัตราค่าบริการตามที่ผู้ดูแลระบบระบุในส่วนของการเก็บข้อมูลการใช้งานนั้นระบบจำเป็นต้องจะเก็บข้อมูลการโทรเป็นวินาที แต่เนื่องจากการคิดค่าใช้จ่ายของผู้ให้บริการต่างๆจะทำเศษวินาทีของการใช้งานให้เป็นจำนวนเต็มผู้วิจัยจึงจำเป็นต้องทำให้เป็นจำนวนเต็มตามผู้บริการเพื่อให้ได้ฐานข้อมูลในการคำนวณเส้นทางโทรออกไปยังปลายทางถูกต้องตามค่าบริการจริงดังภาพที่ 3.17 ซึ่งลักษณะ Code ของการทำให้เป็นจำนวนวินาทีเป็นจำนวนนาทินั้นแสดงดัง ภาพที่ 3.18

```
$selectMin = mysql_query('SELECT * FROM `normalrate` where
`sp`="".'$portSP.'", $objConnect);
$resultMin = mysql_fetch_array($selectMin);
$fullMin = $resultMin['totalMin'];
$portNo = spToPortNo(portSP);
require('findSum.php');
if($portNo==1 && $port1>$fullMin){
    $full = 'over';
} else if($portNo==2 && $port2>$fullMin)
    $full = 'over';
```

ภาพที่ 3.16 Code สำหรับตรวจสอบจำนวนการใช้งานของแต่ละเครือข่าย

```

else if($portNo==3 && $port3>$fullMin)
    $full = 'over';
else if($portNo==4 && $port4>$fullMin)
    $full = 'over';
else if($portNo=="trunk" && $trunk>$fullMin)
    $full = 'over';
else
    $full = 'no';
if(($full=='no') && ($inTime=='no')){
    $portSP="";

    $sqlLeastCost = mysql_query('SELECT `port` FROM `normalrate` order by
perMinasc',$objConnect);

    while($resultLeastCost=mysql_fetch_array($sqlLeastCost)){
        if($portSP=="")
            $portSP = $resultLeastCost['port'];

    else
        $portSP = $portSP.'-'. $resultLeastCost['port'];
    } else if(($full=='no') && ($inTime=='yes')){
        $portSP=$portNo;
        $sqlLeastCost = mysql_query('SELECT `port` FROM `normalrate` where
`port`!="'.$portNo.'" order by perMinasc',$objConnect);

        while($resultLeastCost=mysql_fetch_array($sqlLeastCost)){

```

ภาพที่ 3.16 (ต่อ)

```

if($portSP=="")

        $portSP = $resultLeastCost['port'];

    else

        $portSP = $portSP.'-'. $resultLeastCost['port'];

$portSP="";

    $sqlLeastCost = mysql_query('SELECT `port` FROM `normalrate` order by overLimit
asc',$objConnect);

        while($resultLeastCost=mysql_fetch_array($sqlLeastCost)){

            if($portSP=="")

                $portSP = $resultLeastCost['port'];

        else

            $portSP = $portSP.'-'. $resultLeastCost['port'];}

```

ภาพที่ 3.17 Code สำหรับตรวจสอบอัตราค่าบริการตามที่คุณดูแลระบบระบุ

```

CREATE TRIGGER IntOfSec
BEFORE INSERT
ON cdr
FOR EACH ROW
BEGIN
    SET NEW.billsec = CEILING(`billsec`/60)*60;
    END IF;
END;

```

ภาพที่ 3.18 Code ที่ปัดจำนวนวินาทีให้เป็นจำนวนเต็มนาที

ขั้นตอนที่ 10 กลับส่งค่ากลับไปให้ Asterisk หลังจากนั้น Asterisk จะโทรออก ระบบจะส่งค่าคิวลำดับการโทรออกตามเงื่อนไขที่ได้รับมา ส่งไปให้ Asterisk เพื่อโทรออกเรียงตามลำดับอัตราค่าใช้จ่ายจากน้อยไปหามาก สาเหตุที่ต้องคิวลำดับการโทรส่งไปให้ Asterisk นั้นเนื่องจากในกรณีที่สายไม่ว่างระบบจะสามารถโทรออกตามลำดับค่าโทรที่มีค่าน้อยในลำดับถัดไป เพื่อให้เป็นไปตามเงื่อนไขของงานวิจัยที่สามารถโทรออกได้ในราคาที่ถูกลงที่สุด และจะต้องสามารถโทรออกได้ในกรณีที่มิ่วจรวงอยู่โดยรูปแบบเวลาโทรออกนั้นระบบจะใส่เป็นรูปแบบคิวตาม ช่องทางที่ใช้สำหรับโทรออกไปยังปลายทางที่งานวิจัยได้กำหนดไว้คือ 1 ช่องทางแบบ Trunk และ 4 ช่องทางแบบ PSTN โดยลักษณะการส่งรูปแบบการโทรไปยัง Asterisk จะแสดงดังภาพที่ 3.19 และลักษณะข้อมูลของผู้ใช้งานจะแสดงดังภาพที่ 3.20

```

= Using SIP RTP CoS mark 5
-- Executing [0812049998@default:1] Answer("SIP/100-00000016", "") in new stack
> 0xb6f147a0 -- Probation passed - setting RTP source address to 192.168.15.6:63776
-- Executing [0812049998@default:2] Set("SIP/100-00000016", "sp=1-4-trunk-2-3") in new stack
-- Executing [0812049998@default:3] NoOp("SIP/100-00000016", "1-4-trunk-2-3") in new stack
-- Executing [0812049998@default:4] AGI("SIP/100-00000016", "checkPort.php,0812049998,1-4-trunk-2-3") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/checkPort.php
-- AGI Script Executing Application: (Dial) Options: (dahdi/1-4-trunk-2-3/0812049998)
[Apr 15 19:28:57] WARNING[4558][C-00000016]: app_dial.c:2437 dial_exec_full: Unable to create channel of type 'dahdi' (cause 17 - User busy)
= Everyone is busy/congested at this time (1:1/0/0)
-- <SIP/100-00000016>AGI Script checkPort.php completed, returning 0
-- Executing [0812049998@default:5] Hangup("SIP/100-00000016", "") in new stack
= Spawn extension (default, 0812049998, 5) exited non-zero on 'SIP/100-00000016'
ubuntu11*CLI>
ubuntu11*CLI>
ubuntu11*CLI>
ubuntu11*CLI>

```

ภาพที่ 3.19 การเรียงลำดับการโทรออกไปยังปลายทางและวิธีการโทรออกของระบบ

โดยภาพที่ 3.19 แสดงถึงลักษณะการเรียงคิวโทรออกไปยังปลายทางตามลำดับ Port ที่มีอัตราค่าบริการถูกที่สุดในกรณีนี้ช่องทางที่ 1 ถูกที่สุดแล้วตามด้วย 4, Trunk, 2 และ 3 ตามลำดับโดยเบอร์ที่ต้องการโทรออกไปยังปลายทางนั้นคือเบอร์ 0812049998 ตามบรรทัดด้านล่าง

ลักษณะของการโทรออกของ ระบบนั้นจะแยกหมายเลขปลายทางออกจากหมายเลขช่องทางที่ต้องการโทรออก หลังจากนั้นจะตรวจสอบลำดับที่ Port ที่ได้รับมาและตรวจสอบช่องสัญญาณว่างหรือไม่ หาก Port ว่างระบบจะสร้างคำสั่งให้โทรออกไปทางช่องทางนั้น แต่ในกรณีที่ Port ที่ต้องการไม่ว่าง (มีการใช้สายอยู่) ระบบจะตรวจสอบในลำดับถัดไปจนครบทั้ง 4 ช่องทางที่ และกรณีที่ตรวจสอบครบทุก Port แล้วไม่มีคู่สายว่างระบบจะตัดสายนั้นทันที

ในส่วนของฐานข้อมูลจะประกอบด้วย 7 ตาราง chk_number, configuration, movesp, normalrate, prefixcountry, promotion, summarylog และรูปแบบการใส่ข้อมูลจะแสดงดังตารางที่ 3.1 - 3.7 ซึ่งความสัมพันธ์ของตารางทั้งหมดจะแสดงในภาพที่ 3.20

ตารางที่ 3.1 chk_number

Field Name	type	Description
prefixNumberStart	text	ช่วงของหมายเลขขึ้นต้น
prefixNumberStop	Text	ช่วงของหมายเลขสุดท้าย
Sp	varchar(100)	ชื่อผู้ให้บริการ

ตารางที่ 3.2 configuration

Field Name	type	Description
Id	Int(11)	Increase number
configName	Text	ชื่อการกำหนดค่า
Value	text	ค่าที่กำหนด

ตารางที่ 3.3 movesp

Field Name	type	Description
Numbering	text	หมายเลขโทรศัพท์
oldSP	Text	ชื่อผู้ให้บริการเดิม
newSP	text	ชื่อผู้ให้บริการใหม่

ตารางที่ 3.4 normalrate

Field Name	type	Description
Id	Int(11)	Increase Number
Port	varchar(200)	หมายเลข port
Sp	text	ชื่อผู้ให้บริการ
totalCost	float	อัตราค่าโทรทั้งหมด
totalMin	Int(11)	จำนวนเวลาที่โทรได้
perMin	float	อัตราค่าโทรต่อนาที
overLimit	float	อัตราค่าโทรต่อนาทีเมื่อใช้เวลากิน/อัตราค่าโทรจาก local-mobile

ตารางที่ 3.5 prefixcountry

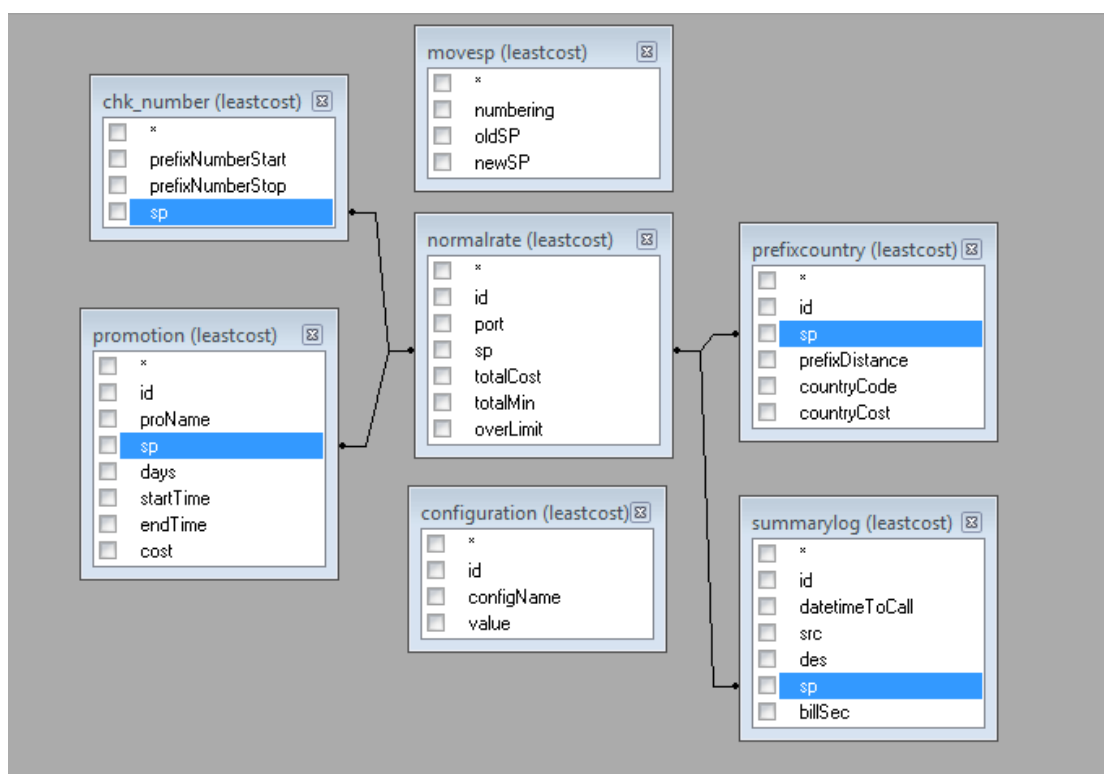
Field Name	type	Description
Id	Int(11)	Increase Number
Sp	Text	ชื่อผู้ให้บริการ
prefixDistance	text	หมายเลขนำสำหรับโทรทางไกลต่างประเทศ
countryCode	Int(5)	รหัสประเทศ
countryCost	Int(5)	อัตราค่าโทร

ตารางที่ 3.6 promotion

Field Name	type	Description
Id	Int(11)	Increase Number
proName	Text	ชื่อโปรโมชั่น
Sp	text	ชื่อผู้ให้บริการ
Days	text	ชื่อวันที่ใช้ได้
startTime	time	เวลาเริ่มต้น
endTime	time	เวลาสิ้นสุด
Cost	float	อัตราค่าโทร

ตารางที่ 3.7 summarylog

Field Name	type	Description
Id	Int(11)	Increase Number
datetimeToCall	datetime	วัน-เวลาที่ใช้งาน
Src	varchar(100)	หมายเลขต้นทาง
Des	varchar(100)	หมายเลขปลายทาง
Sp	text	ชื่อผู้ให้บริการ
billSec	Int(11)	เวลาที่ใช้ในการสนทนา



ภาพที่ 3.20 ความสัมพันธ์ของตารางทั้งหมด

จากความสัมพันธ์ตารางดังภาพที่ 3.20 อธิบายหลักการทำงานได้ดังต่อไปนี้ในการออกแบบฐานข้อมูลได้ทำการออกแบบ E-R diagram ดังภาพ แบ่งออกเป็น 6 ตารางดังนี้

1. ตาราง normalrate เป็นตารางหลักซึ่งใช้เก็บค่าโทรของแต่ละ port และมี field PORT เป็น primary key และมีความเชื่อมโยงกับตารางอื่นโดยใช้ field SP หรือ service provider เป็นตัวเชื่อมโยง
2. ตาราง promotion ทำหน้าที่เก็บโปรโมชั่นของแต่ละ port หรือแต่ละ sp โดยจะเก็บค่าวันและเวลาที่สามารถใช้โปรโมชั่นได้
3. ตาราง prefixcountry ทำหน้าที่เก็บรหัสโทรทางไกลของแต่ละเครือข่าย และรหัสประเทศเพื่อไว้คำนวณหาเส้นทางโทรที่ถูกสุด
4. ตาราง chk_number ทำหน้าที่เก็บหมายเลขโทรศัพท์เคลื่อนที่ของผู้ให้บริการแต่ละราย เพื่อตรวจสอบเบอร์ปลายทางว่าเป็นของผู้ให้บริการรายใด
5. ตาราง movesp ทำหน้าที่เก็บหมายเลขโทรศัพท์ที่มีการย้ายเครือข่ายผู้ให้บริการ เพื่อตรวจสอบว่าเบอร์ปลายทางเป็นของผู้ให้บริการรายใด

6. ตาราง configuration ทำหน้าที่เก็บค่า config ของระบบ เช่น หมายเลขขึ้นต้นของหมายเลขท้องถิ่น

7. ตาราง summarylog ทำหน้าที่เก็บข้อมูลการใช้งานของแต่ละ port ว่ามีการใช้งานอย่างไรบ้าง และเก็บจำนวนเวลาที่ใช้งานเพื่อนำไปคำนวณค่าโทรรายละเอียดความหมายของตารางแสดงดัง ตารางที่ 3.8

ตารางที่ 3.8 ความหมายของคำศัพท์ในตารางทั้งหมด

ตาราง	des
Normalrate	เก็บค่าโทรของแต่ละ port
Promotion	โปรโมชั่นเสริมของแต่ละเครือข่าย
prefixcountry	รหัสโทรทางไกลต่างประเทศ
chk_number	หมายเลขของผู้ให้บริการ
Movesp	หมายเลขที่ย้ายเครือข่าย
configuration	ค่า config ของระบบ
summarylog	รายละเอียดการใช้งาน

ขั้นตอนการสร้างรูปแบบในการโทรของระบบซึ่งจะเป็นลักษณะการเขียนรูปแบบสำหรับการโทรออกไปยังปลายทาง และรูปแบบของผู้ใช้งาน โทรศัพท์โดยไฟล์ที่ใช้สำหรับสร้างแผนการโทรและผู้ใช้งานคือ ไฟล์ extension.conf และ sip.conf นั้นลักษณะการเขียนแผนการโทรแสดงดังภาพที่ 3.21 และภาพที่ 3.22

```

exten=>_XXX,1,answer
same=>n,dial(sip/${EXTEN})
same=>n,hangup

exten=>_0.,1,answer
same=>n,set(sp=${CURL(http://127.0.0.1/leastcost/findport.php?numbering=${EXTEN}}))
same=>n,noop(${sp})
same=>n,agi(checkPort.php,${EXTEN},${sp})
;exten=>_0.,1,dial(sip/${EXTEN}@98100001355,30,r)
same=>n,hangup
;

```

ภาพที่ 3.21 แผนการโทรสำหรับใช้งานในระบบ IP-PBX จากไฟล์ extensions.conf

```

[100]
Type=friend
Secret=100
Host=dynamic
;context=default
nat=yes
[200]
Type=friend
Secret=200
Host=dynamic
;context=default
nat=yes

```

ภาพที่ 3.22 หน้าต่างการสร้างผู้ใช้งานสำหรับใช้งานในระบบ IP-PBX จากไฟล์ Sip.conf