**บทที่ 3**

วิธีการดำเนินโครงงาน

วิธีการดำเนินโครงงานระบบการจัดการผู้ใช้ มีวิธีในการดำเนินโครงงานดังต่อไปนี้ คือ

3.1 การศึกษาปัญหาของการแสดงผลข้อมูลประเภทกราฟแบบเดิม

3.2 การศึกษาความเป็นไปได้ และขอบเขตของการทำงาน

3.3 การวิเคราะห์ และออกแบบโปรแกรมรวมทั้งโครงสร้างข้อมูล

**3.1 การศึกษาปัญหาของการแสดงผลข้อมูลประเภทกราฟแบบเดิม**

เนื่องจากข้อมูลในปัจจุบันมีข้อมูลเชิงความสัมพันธ์อยู่มาก ซึ่งข้อมูลเหล่านั้นสามารถแทนให้อยู่ในรูปของกราฟได้ แต่ว่าแม้ว่าข้อมูลจะอยู่ในรูปแบบของกราฟแต่การที่ จะคิดดำเนินการกับข้อมูลดังกล่าวปัจจุบันนั้น ใช้การจินตนาการรูปกราฟต่างๆ ขึ้นมา หรือว่าวาดด้วยมือเอง ซึ่งอาจจะทำให้ไม่เห็นภาพของข้อมูลที่ชัดเจน ซึ่งการจินตนาการรูปของกราฟนั้นอาจจะทำให้เกิดความผิดพลาดได้ หรือว่าถ้าข้อมูลมีขนาดใหญ่การจินตนาการนั้นอาจจะไม่เพียงพอที่จะใช้ในการวิเคราะห์กราฟที่เกิดขึ้น หรือในการที่ใช้ อัลกอริทึมที่จะใช้จัดการกับกราฟนั้นอาจจะทำให้ได้ผลที่เข้าใจได้ยาก และการจะอธิบายผลต่อผู้อื่นก็เป็นเรื่องยากด้วย

**3.2 การศึกษาความเป็นไปได้**

จากการศึกษาการทำงานกับข้อมูลชนิดกราฟแบบเดิม จะพบว่าปัญหาเกี่ยวกับการนำเสนอข้อมูลเชิงความสัมพันธ์(หรือกราฟ) ในรูปแบบเดิมนั้น ยังไม่มีประสิทธิภาพเพียงพอสำหรับการนำไป แสดงผลให้ผู้ใช้ทั่วไปสามารถเข้าใจได้ ทำให้การนำไปคิดวิเคราะห์และพัฒนาต่อเป็นไปได้ยาก เพราะว่าผู้ใช้ไม่เข้าใจข้อมูล การปรับการแสดงผลข้อมูลชนิดกราฟให้อยู่ในรูปแบบกราฟิก อาจจะเป็นแนวทางหนึ่งในการช่วยให้ผู้ที่ดูข้อมูลเชิงความสัมพันธ์ (หรือกราฟ) สามารถเข้าใจข้อมูลเหล่านั้นได้มากขึ้น

ผู้จัดทำโครงงานจึงมีแนวคิดที่จะพัฒนาโปรแกรมแสดงผลข้อมูลชนิดกราฟให้อยู่ในรูปกราฟิกขึ้น เพื่อคอยให้บริการในส่วนของการแสดงผลข้อมูลเชิงความสัมพันธ์ต่างๆ ออกมาในรูปแบบกราฟิก โดยโปรแกรมจะมีความสามารถหลักๆดังนี้

1. สามารถอ่านไฟล์ข้อมูลเชิงความสัมพันธ์ขึ้นมาแสดงผลได้
2. สามารถที่จะสร้างหรือแก้ไขกราฟได้เอง เพื่อใช้ในการจัดการกราฟต่างๆ ได้ในระดับหนึ่ง เช่น การเพิ่มลดจุดตัด เคลื่อนย้ายจุดตัด การบันทึกกราฟหลังการเปลี่ยนแปลง
3. สามารถที่จะเพิ่มความสามารถในการจัดการกราฟด้วยอัลกอริทึมโดยจะแสดงผลออกมาโดยการเน้นสีที่ จุดตัดและเส้นที่เป็นผลลัพธ์
4. สามารถบันทึกกราฟที่แก้ไขหรือสร้างขึ้นมาได้โดยที่จะบันทึกแค่ส่วนของโครงสร้างเท่านั้น ไม่เก็บส่วนของตำแหน่งหรือสีของกราฟ

**3.3 การออกแบบโปรแกรม**

จากการศึกษาความเป็นไปได้ของการดำเนินการ แล้วจึงนำเข้าสู่ขั้นตอนของการออกแบบโปรแกรม โดยที่ผู้พัฒนาได้ทำการออกแบบโปรแกรมในรูปแบบของ Object Oriented โดยการมองส่วนประกอบของโปรแกรมเป็น Class และ Object เพื่อนำไปเขียนโปรแกรมต่อไปโดยส่วนประกอบหลักๆที่ต้องออกแบบมีดังนี้

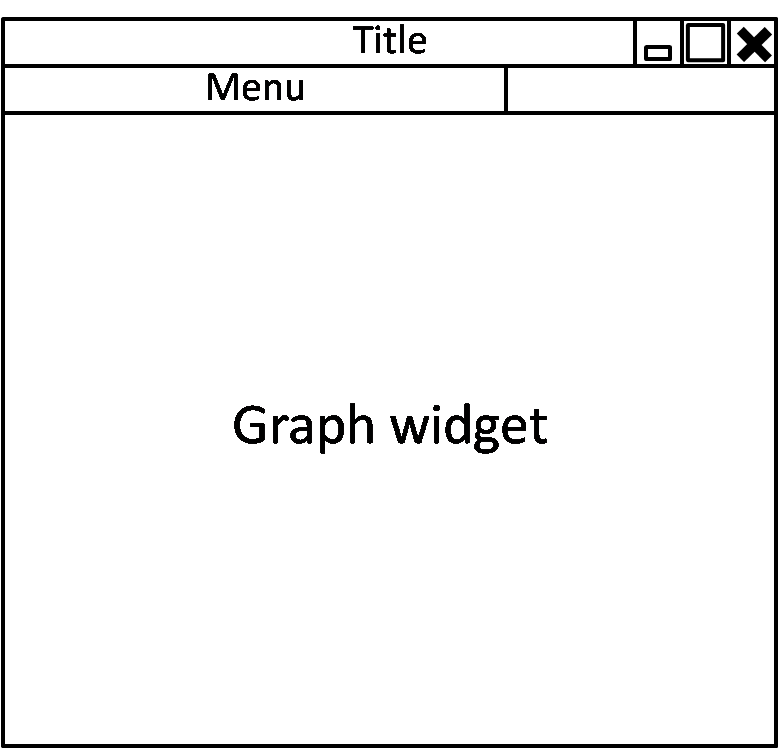
3.1.1 ออกแบบส่วนของการติดต่อกับผู้ใช้ (User Interface)

3.1.2 คลาสไดอะแกรม (Class Diagram)

**3.3.1 ออกแบบส่วนของการติดต่อผู้ใช้ (User Interface)**

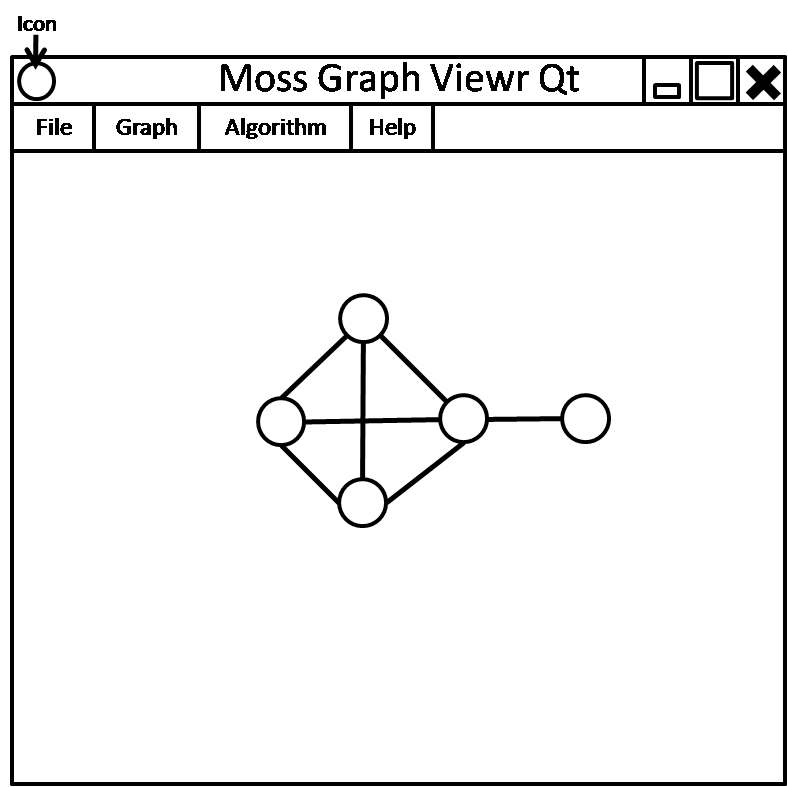
ออกแบบส่วนติดต่อผู้ใช้โปรแกรม เพื่อกำหนดส่งแนวทางในการพัฒนาโปรแกรม และสิ่งที่จำเป็นต้องมีในโปรแกรม

1. โครงสร้างหน้าตาหลักๆของโปรแกรมจะมีลักษณะดังรูป



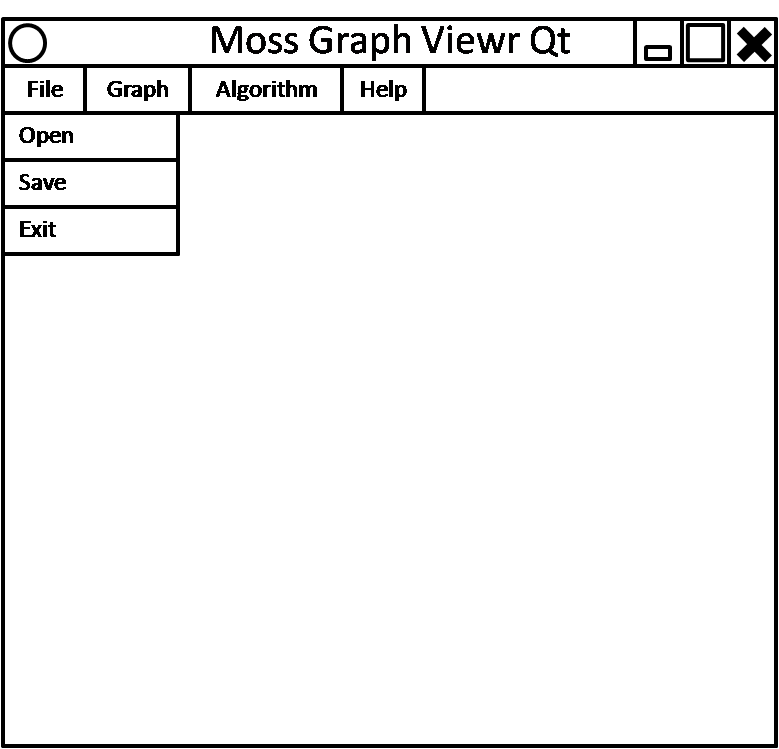
**รูปที่ 3-1** รูปโครงสร้างหลักของส่วนติดต่อผู้ใช้ของโปรแกรม

2. รูปร่างหน้าตาของโปรแกรมเมื่อใช้งานซึ่งประกอบไปด้วยการนำข้อมูลกราฟที่เปิดจากไฟล์ขึ้นมาแทนที่ Graph widget เมนูต่างๆ ที่จำเป็นต้องมี ชื่อโปรแกรมและไอคอนตกแต่ง



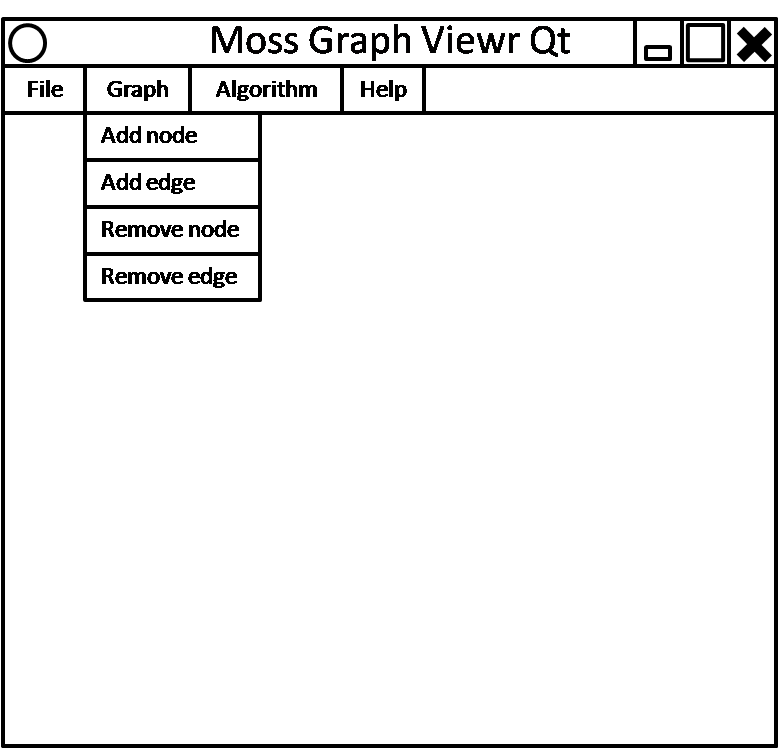
**รูปที่ 3-2** รูปส่วนติดต่อผู้ใช้ของโปรแกรมเมื่อใช้งาน

3. ส่วนติดต่อผู้ใช้เกี่ยวกับไฟล์จะมีความสามารถพื้นฐานในการเปิดไฟล์ การบันทึกไฟล์และออกจากโปรแกรม ส่วนหน้าต่างในการหาไฟล์(File manager, File Browser) จะเรียกใช้ของระบบโดยตรง (ขึ้นอยู่กับระบบปฏิบัติการ)



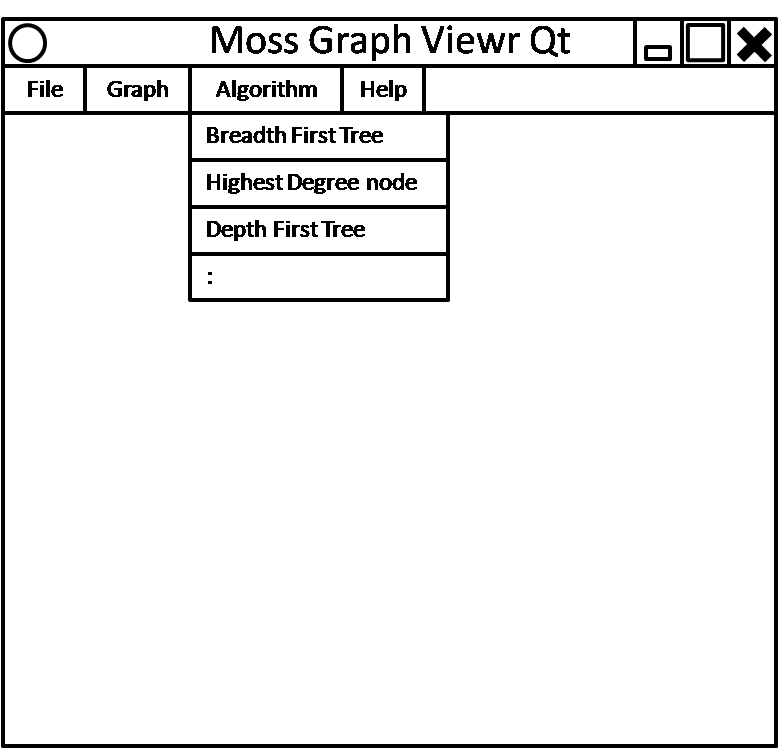
**รูปที่ 3-3** รูปส่วนติดต่อผู้ใช้ เมนูไฟล์

4. ส่วนติดต่อผู้ใช้ เมนูกราฟใช้เพื่อเก็บการจัดการหลักๆเกี่ยวกับกราฟไว้เช่น การเพิ่มลดจุดตัด(Node) การเพิ่มลดเส้นเชื่อม ( Edge)



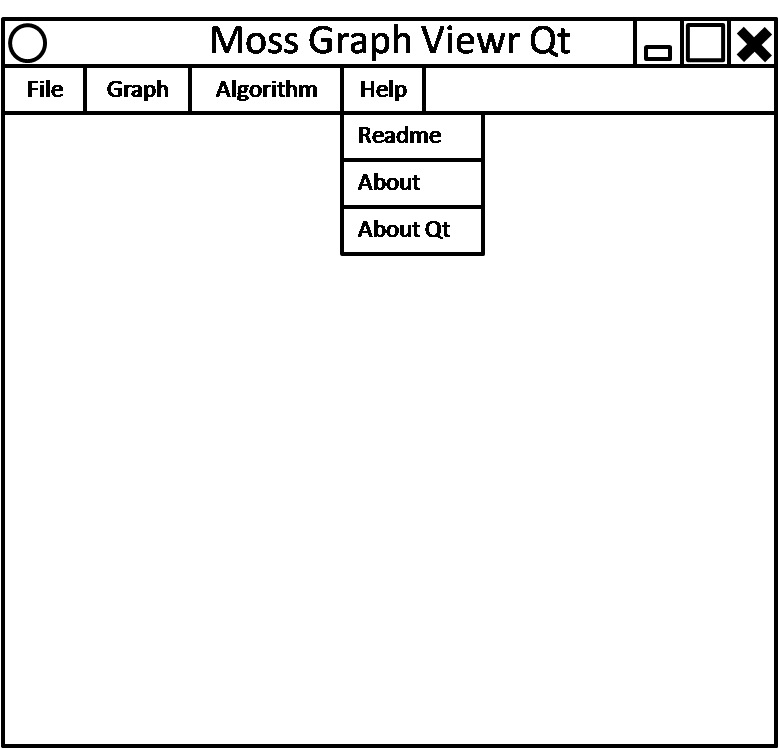
**รูปที่ 3-4** รูปส่วนติดต่อผู้ใช้ เมนูกราฟ

5. ส่วนติดต่อผู้ใช้ เมนูอัลกอริทึ่ม ตัวเลือกสำหรับการใส่การจัดการอัลกอริทึ่มซึ่งสามารถใส่เพิ่มขึ้นเข้าไปได้เรื่อยๆ โดยการแก้ไขปรับปรุงโปรแกรมส่วนนี้

****

**รูปที่ 3-5** รูปส่วนติดต่อผู้ใช้ เมนูอัลกอริทึม

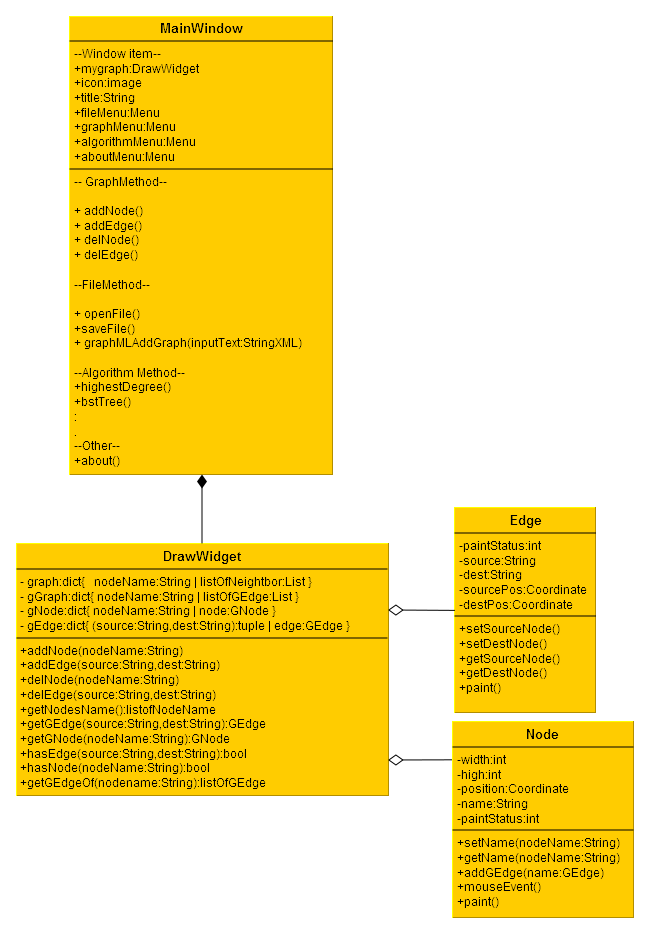
6. ส่วนติดต่อผู้ใช้ เมนูช่วยเหลือและเกี่ยวกับ ใช้สำหรับใส่ส่วนช่วยเหลือผู้ใช้ และแสดงข้อมูลเกี่ยวกับโปรแกรมและการพัฒนา



**รูปที่ 3-6** รูปส่วนติดต่อผู้ใช้ เมนูช่วยเหลือ

**3.3.2 คลาสไดอะแกรม (Class Diagram)**

คลาสไดอะแกรมต้นแบบของโปรแกรมแสดงผลข้อมูลประเภทกราฟออกมาในรูปแบบกราฟิก ได้ออกแบบส่วนประกอบสำคัญ ไว้เพื่อเป็นต้นแบบในการเขียนโปรแกรมจริง แสดงให้เห็นดังรูป



**รูปที่ 3-7** คลาสไดอะแกรมของโปรแกรมแสดงผลข้อมูลประเภทกราฟ