



บทที่ 1

บทนำ

1.1 ที่มาหลักการและเหตุผล

ในปัจจุบันการค้นหาข้อมูลต่างๆบนเว็บไซต์หรือโมบายแอพพลิเคชันนั้น โดยส่วนมากแล้วการป้อนคำค้น อาจจะเป็นเพียงการเทียบคำค้นที่ผู้ใช้งานป้อนเข้ามาเพียงชุดเดียว กับข้อมูลที่มีอยู่ในฐานข้อมูล ซึ่งผลลัพธ์ที่ได้ อาจจะไม่ตรงตามความต้องการของผู้ใช้งานอย่างแท้จริง เช่น การเลือกซื้อกีต้าร์บนเว็บไซต์ ผู้ใช้งานมีความ ต้องการที่จะเลือกซื้อกีต้าร์ยี่ห้อหนึ่ง และต้องการวัดสุดที่ทำกีต้าร์จากไม้ชนิดหนึ่ง นอกจากนั้นยังอาจต้องการ ซ่อมราคาไม่เกินจำนวนเงินที่ผู้ใช้มงบประมาณอยู่ ซึ่งแอพพลิเคชันโดยทั่วไปไม่สามารถตอบสนองต่อความ ต้องการเหล่านี้ได้

ด้วยเหตุนี้ ผู้จัดทำโครงการจึงได้พัฒนาระบบแอพพลิเคชันสำหรับการเลือกซื้อกีต้าร์ เพื่อให้ผู้ใช้งาน สามารถกำหนดคุณสมบัติของกีต้าร์ที่ต้องการ และระบุลำดับความสำคัญของคุณสมบัตินั้นได้อีกด้วย โดย ระบบจะนำข้อมูลที่ผู้ใช้กำหนดไปตรวจสอบกับข้อมูลในฐานข้อมูล และคำนวณเพื่อวัดความคล้าย แล้วจะ แสดงชุดข้อมูลพร้อมผลลัพธ์ของการวัดความคล้ายอุปกรณ์ในรูปแบบของร้อยละ ส่งผลให้ผู้ใช้ทั้งที่เป็นผู้หัด เล่นกีต้าร์และผู้เล่นกีต้าร์ระดับเชี่ยวชาญสามารถค้นหากีต้าร์ที่เหมาะสมกับความต้องการได้

1.2 วัตถุประสงค์

- เพื่อพัฒนาระบบแอพพลิเคชันสำหรับแนะนำการเลือกซื้อกีต้าร์ ให้เหมาะสมกับความต้องการของ ผู้ใช้งาน

1.3 ขอบเขตของโครงการ

- ระบบสามารถแนะนำการเลือกซื้อกีต้าร์ ให้เหมาะสมกับความต้องการของผู้ใช้งานได้ โดยใช้การ คำนวณเพื่อวัดความคล้าย ด้วยระบบarendroidแอพพลิเคชัน
- ระบบเว็บแอพพลิเคชันสามารถจัดการข้อมูลในฐานข้อมูลได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- สามารถลดระยะเวลาการค้นหาข้อมูล ก่อนการเลือกซื้อกีต้าร์ของผู้ใช้งานได้
- ผลลัพธ์ที่ได้สามารถช่วยให้ผู้ใช้งานสามารถเลือกซื้อกีต้าร์ได้ตรงตามความต้องการ



1.5 เครื่องมือที่ใช้ในการพัฒนาระบบแอปพลิเคชัน

1. Hardware

- หน่วยประมวลผล (CPU) 3rd Generation Intel Core i5-3337U processor
- ฮาร์ดดิสก์ HDD:750GB 5400RPM SATA Hard Drive
- หน่วยความจำ (RAM) 6GB 1 DIMM (4GB x1) DDR3 1600MHz
- การ์ดจอ (VGA) NVIDIA GeForce GT625M 1GB DDR3
- Smart Phone 1 เครื่อง

2. Software

- NetBeans IDE : ใช้สำหรับพัฒนาระบบเว็บแอปพลิเคชันในการจัดการฐานข้อมูล
- MySQL Workbench : ใช้สำหรับออกแบบฐานข้อมูล
- XAMPP : ใช้สำหรับจำลองฐานข้อมูล
- Android Studio : ใช้สำหรับพัฒนาระบบ แอนดรอยด์แอปพลิเคชัน
- Notepad + + : ใช้สำหรับเขียน Source Code เพื่อ Call Web service
- Photoshop : ใช้สำหรับออกแบบ User Interface

1.6 ภาษาและเทคนิคที่ใช้ในการพัฒนาระบบแอปพลิเคชัน

- Java Programming
- Java Servlet
- HTML
- JDBC
- Tomcat Server
- Java Script
- Ajax
- Bootstrap CSS Framework
- เทคนิคการคำนวณความคล้าย



1.7 ขั้นตอนการดำเนินงาน

- หาข้อมูลเกี่ยวกับการเลือกซื้อกีต้าร์ของบุคคลใกล้ตัวในการดูแลรักษา
- เสนอหัวข้อโครงงานต่ออาจารย์ที่ปรึกษา
- ค้นคว้าข้อมูลเกี่ยวกับกีต้าร์เพิ่มเติม
- นำเสนอหัวข้อโครงงานต่ออาจารย์ที่ปรึกษา
- วิเคราะห์และออกแบบขั้นตอนการทำงานในส่วนของ เว็บแอพพลิเคชันเพื่อการจัดการฐานข้อมูล
- นำเสนอขั้นตอนการทำงานในส่วนของ เว็บแอพพลิเคชันเพื่อการจัดการฐานข้อมูล ต่ออาจารย์ที่ปรึกษา
- เริ่มพัฒนาระบบจัดการฐานข้อมูลเว็บแอพพลิเคชัน
- นำเสนอและทดสอบ โปรแกรมเว็บแอพพลิเคชันเพื่อการจัดการฐานข้อมูล ต่ออาจารย์ที่ปรึกษา และแก้ไขโปรแกรม
- วิเคราะห์และออกแบบขั้นตอนการทำงานในส่วนของ แอนดรอยด์แอพพลิเคชัน
- นำเสนอขั้นตอนการทำงานในส่วนของ แอนดรอยด์แอพพลิเคชัน ต่ออาจารย์ที่ปรึกษา
- เริ่มพัฒนาระบบแอนดรอยด์แอพพลิเคชัน
- นำเสนอและทดสอบ โปรแกรมแอนดรอยด์แอพพลิเคชัน ต่ออาจารย์ที่ปรึกษา และแก้ไขโปรแกรม
- จัดทำเอกสารคู่มือการใช้งาน



ตารางที่ 1 : รายละเอียดการดำเนินงาน

ลำดับ	ขั้นตอนการดำเนินงาน	ระยะเวลาการดำเนินงาน				
		ก.ย.	ต.ค.	พ.ค.	ธ.ค.	ม.ค.
1	หาข้อมูลเกี่ยวกับการเลือกซื้อกีต้าร์ของบุคคล ใกล้ตัวในวงการดนตรี					
2	เสนอหัวข้อโครงงานต่ออาจารย์ที่ปรึกษา					
3	ค้นคว้าข้อมูลเกี่ยวกับกีต้าร์เพิ่มเติม					
4	นำเสนอหัวข้อโครงงานต่ออาจารย์ที่ปรึกษา					
5	วิเคราะห์และออกแบบขั้นตอนการทำงานใน ส่วนของ เว็บแอปพลิเคชันเพื่อการจัดการ ฐานข้อมูล					
6	นำเสนอขั้นตอนการทำงานในส่วนของ เว็บ แอปพลิเคชันเพื่อการจัดการฐานข้อมูล ต่อ อาจารย์ที่ปรึกษา					
7	เริ่มพัฒนาระบบจัดการฐานข้อมูลเว็บ แอปพลิเคชัน					
8	นำเสนอและทดสอบ โปรแกรมเว็บ แอปพลิเคชันเพื่อการจัดการฐานข้อมูล ต่อ อาจารย์ที่ปรึกษา และแก้ไขโปรแกรม					
9	วิเคราะห์และออกแบบขั้นตอนการทำงานใน ส่วนของ แอนดรอยด์แอปพลิเคชัน					
10	นำเสนอขั้นตอนการทำงานในส่วนของ แอน ดรอยด์แอปพลิเคชัน ต่ออาจารย์ที่ปรึกษา					
11	เริ่มพัฒนาระบบแอนดรอยด์แอปพลิเคชัน					
12	นำเสนอและทดสอบ โปรแกรมแอนดรอยด์ แอปพลิเคชัน ต่ออาจารย์ที่ปรึกษา และแก้ไข โปรแกรม					
13	จัดทำเอกสารคู่มือการใช้งาน					

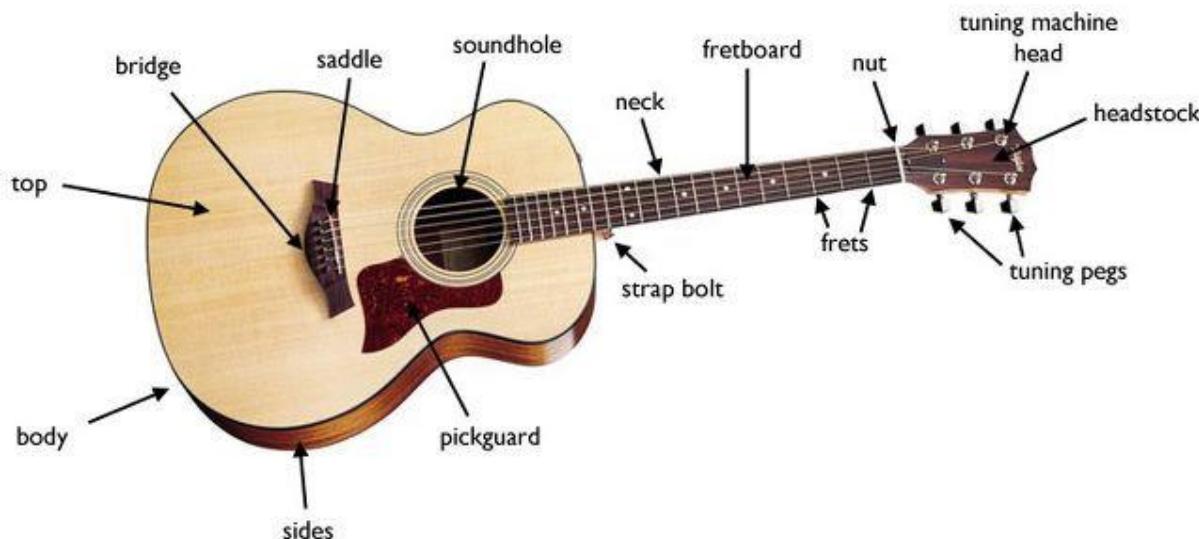


บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ระบบแนะนำการเลือกซื้อกีต้าร์ให้เหมาะสมกับความสามารถ หรือความต้องการ ความรู้พื้นฐานที่นำมาพัฒนาระบบ ได้แก่

2.1 ความรู้เกี่ยวกับองค์ประกอบของกีต้าร์ หลัก



1. ส่วนหัว ประกอบด้วย

1.1 ชุดลูกบิด โดยทั่วไปจะมี 2 แบบ ได้แก่แบบที่ตัวลูกบิดหันไปด้านหลังตั้งฉากกับตัวกีต้าร์ แกนหมุนสายเป็นพลาสติกซึ่งจะใช้กับกีต้าร์คลาสสิก หรือกีต้าร์ไฟฟ้า และอีกแบบจะขานานกับตัวกีต้าร์ หรือแกนหมุนสายตั้งฉากกับตัวกีต้าร์ซึ่งใช้กับกีต้าร์ไฟฟ้าและกีต้าร์ไฟฟ้าทั่วๆไป

1.2 นัท (Nut) หมายหรือสะพานสายบน มันจะติดอยู่ปลายบนสุดของฟิงเกอร์บอร์ดเพื่อรองรับสายกีต้าร์ให้ยกสูงจากฟิงเกอร์บอร์ด ซึ่งระยะความสูงของสายกับฟิงเกอร์บอร์ดดังกล่าวนี้เรียกว่า action ซึ่งมีความสำคัญมาก เพราะถ้ามันตั้งความสูงไว้ไม่เหมาะสมแล้วจะทำให้การเล่นกีตาร์นั้นยากลำบาก คือถ้าระยะดังกล่าวสูงไปผู้เล่นต้องออกแรงกดสายมากขึ้น แต่ถ้ามันตั้งไว้ต่ำไปก็จะทำให้เวลาดีดความสั่นของสายจะไปโดนเฟรตทำให้เกิดเสียงเปลก ๆ ออกมา



2. ส่วนคอกีตาร์ ประกอบด้วย

2.1 คอกีตาร์ คือส่วนที่ใช้จับคอร์ดเล่นเน็ตต่าง ๆ มีความสำคัญมากสำหรับคอกีตาร์ คอกีตาร์ควรจะทำมาจากไม้ มะขอก้านี้ หรือไม้ ซีด้า หลักการที่สำคัญที่สุดคือคอกีตาร์ต้องตรง ไม่มีรอยแตก หรือบริข่องเนื้อไม้

2.2 Finger board เป็นแผ่นไม้ที่ติดลงบนคอกีตาร์อีกชั้น เป็นตัวที่ใช้ยืดเพริต อาจมีลวดลาย มุกประดับให้สวยงาม ซึ่งโน๊ตต่างๆของคีตาร์จะอยู่บนหน้าพิงเกอร์บอร์ดนั้นเอง ไม่ที่นิยมใช้จะเป็นไม้ โรสวูด หรือไม้ อีบานี ซึ่งมีเนื้อไม้แข็งเกินไป คีตาร์โฟล์กบกีตาร์ไฟฟ้าส่วนใหญ่จะโครงเล็กน้อยเพื่อให้เข้ากับนิ้วเวลาทابบนคอ

2.3 เพริต (Fret) ทำมาจากโลหะฝังอยู่บนคอกีตาร์เป็นตัวที่จะกำหนดเสียงของโน๊ตดนตรี จากการกดสายคีตาร์ลงบนเพริตต่างๆ ซึ่งเมื่อติดลงไปทำให้เกิดเสียงมีความสั้นยาวต่างกันไปตามการกดสาย จำนวนของเพริตก็จะขึ้นกับความยาวของคอกีตาร์ซึ่งแต่ละผู้ผลิตก็จะต่างกันไป ปกติคีตาร์คลาสสิกจะมีประมาณ 18 ตัว คีตาร์โฟล์คประมาณ 20 ตัว แต่คีตาร์ไฟฟ้าซึ่งมักจะมีการเล่นโซโล่จึงมีช่องให้เล่นโน๊ตมากขึ้นประมาณ 22-24 ตัว และคีตาร์คลาสสิกซึ่งคอกีตาร์แบบราบ เพริตก็จะตรง แต่คีตาร์โฟล์คหรือคีตาร์ไฟฟ้านั้นส่วนใหญ่จะมีคอที่โครงเล็กน้อยก็จะมีเพริตที่โครงตามไปด้วย

3. ส่วนลำตัวคีตาร์ ประกอบด้วย

3.1 ลำตัวคีตาร์ (Body) หมายถึง 2 ส่วนได้แก่

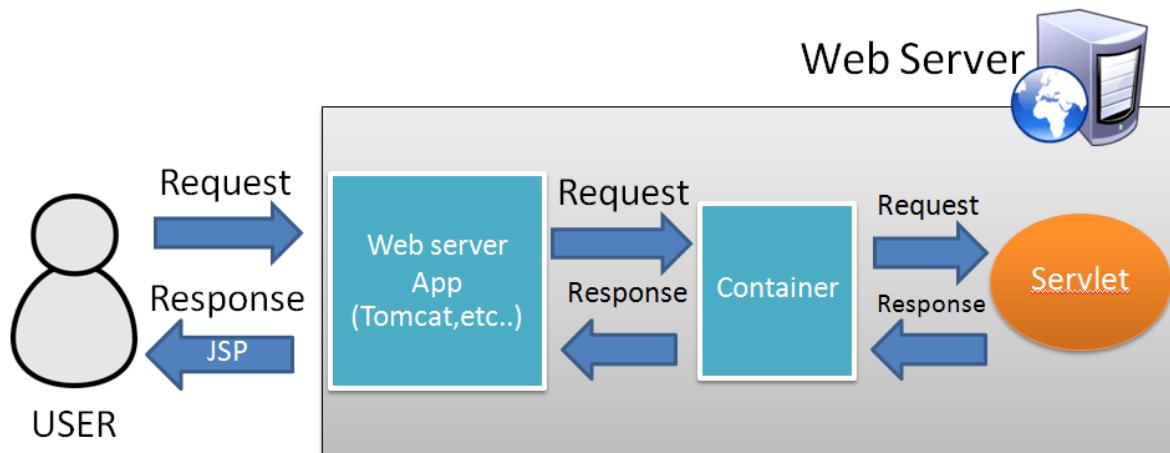
ไม้ด้านหน้า (Top Wood) หัวใจของเสียงที่ดี และทำให้ดูสวยงาม หรูหราและไม้ด้านหน้านี้ยังบ่งบอกถึงราคากีตาร์ตัวนั้นๆได้อีกด้วย

ไม้ด้านหลัง (Back) และไม้ด้านข้าง (Side) ความกันวานของเสียงจะเกิดจากด้านหลังและด้านข้าง ดังนั้นความนุ่มหรือกระด้างของเสียงจะเกิดจากการเลือกชนิดไม้หลังและไม้ข้างนี้เอง

และที่สำคัญคือลักษณะของไม้ต้องไม่มีรอยแตก ไม่มีตาไม้และมีลายไม้ที่ละเอียดไปตามความยาวจึงมีคุณภาพดี



2.2 ความรู้เกี่ยวกับ JSP และ Servlet



JavaServer Pages (JSP) เป็นเทคโนโลยีการเขียนโปรแกรมผ่านเซิร์ฟเวอร์ที่ช่วยให้การสร้างหน้าเว็บแบบไดนามิกและการประยุกต์ใช้ ซึ่งทำได้โดยการฝังรหัส Java เป็น HTML, XML, DHTML, หรือประเภทเอกสารอื่น ๆ เมื่อเว็บเบราว์เซอร์ทำให้คำขอไปยังที่เก็บโปรแกรม Java ซึ่งตามปกติจะเป็นเว็บเซิร์ฟเวอร์หน้าจะถูกแปลงอยู่เบื้องหลังและแสดงเป็นเนื้อหาแบบไดนามิกกับตัว JSP

Servlets เป็น Component ด้านเซิร์ฟเวอร์ที่ให้กลไกที่มีประสิทธิภาพ ในการพัฒนาโปรแกรมผ่านเซิร์ฟเวอร์ ทำหน้าที่ จัดการกับ request และ response ต่างๆ และส่งค่าไปแสดงผลใน JSP

ในภาพข้างบนนี้จะเป็นภาพที่อธิบายถึงการทำงานของ Servlet และ JSP เมื่อมีผู้ใช้งานหน้าเว็บผ่านเว็บ Browser ซึ่งการทำงานคือ เราจะมี Web server application ตัวนึงเช่น Apache Tomcat, Jboss เป็นต้น ไว้รับ request ต่างๆแล้วก็จะส่งต่อให้ container ไปเรียก servlet ให้ทำงานตามที่ request มา เช่น request ไปเพื่อขอข้อมูลลูกค้า servlet ก็จะไปทำการดึงข้อมูลลูกค้าออกมา (model) แล้วก็ส่งค่าต่างๆไปที่ JSP (view) การออกแบบของ JSP & Servlet เป็นการออกแบบที่เรียกว่า MVC (Model-View-Controller) มันแบ่งออกเป็นส่วนๆ แต่ละส่วนก็ทำหน้าที่ของของตัวเองเช่น Model ก็เก็บข้อมูล View ก็ทำหน้าที่แสดงผล และ Controller ก็ทำหน้าที่ควบคุมการทำงาน

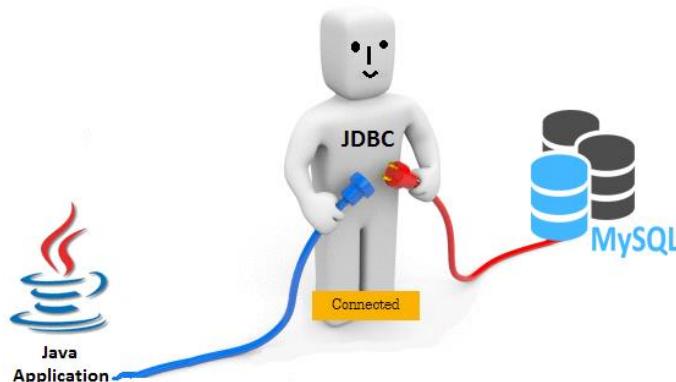


2.3 ความรู้เกี่ยวกับ MySQL



MySQL คือ ระบบจัดการฐานข้อมูลที่มีหน้าที่เก็บข้อมูล เพื่อใช้ในการเก็บข้อมูลที่ได้จากระบบงานที่โปรแกรมเมอร์ได้สร้างขึ้น โดยใช้ภาษา SQL โดย SQL คือภาษาที่ใช้ในการจัดการกับฐานข้อมูลโดยเฉพาะ เช่น สร้างฐานข้อมูล เพิ่มข้อมูล แก้ไขข้อมูล ลบข้อมูล เป็นต้น และ MySQL จะทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้งานกับฐานข้อมูล ซึ่ง MySQL เป็นระบบจัดการฐานข้อมูล (database management system DBMS) สำหรับฐานข้อมูลเชิงสัมพันธ์ โดยความสามารถติดต่อ กับ MySQL โดยการเขียนโปรแกรมภาษาต่าง ๆ ได้ เช่น PHP, Perl, Java, C#, C, Ruby, C++ เป็นต้น

2.4 ความรู้เกี่ยวกับ JDBC



JDBC เป็น Java API ที่ช่วยให้ Java Application สามารถเชื่อมต่อ กับฐานข้อมูลแบบ Relational Database เช่น MySQL ,SQL Server ,Oracle) โดย JDBC มีหน้าที่ 5 อย่างนี้

1. เชื่อมต่อ กับฐานข้อมูล
2. สร้าง Query statement
3. ส่ง (Execute) Query Statement เช่น SELECT, FROM, WHERE
4. รับค่าจาก ฐานข้อมูลที่ Query
5. ปิดการเชื่อมต่อ



2.5 ความรู้เกี่ยวกับ JAVA Programming For Android



Android คือ ระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ และแล็ตคอมพิวเตอร์ เน็ตบุ๊ก ทำงานบนลินุกซ์ เคอร์แนล ทางกูเกิลได้เปิดให้นักพัฒนาสามารถแก้ไขโค้ดต่างๆ ด้วยภาษาจาวา และ ควบคุมอุปกรณ์ผ่านทางชุด Java libraries ที่กูเกิลพัฒนาขึ้น โดยแอนดรอยด์ (Android) ถูกตั้งชื่อเลียนแบบ หุ่นยนต์ในเรื่อง สตาร์วอร์ส ที่ชื่อครอ>y> ซึ่งเป็นหุ่นยนต์ที่สร้างขึ้นมาเลียนแบบ



Java คือ ภาษา computer ขั้นสูงชนิดหนึ่ง ที่มีการทำงานเป็นลักษณะเชิงวัตถุ (OOP : Object-Oriented Programming) ถูกออกแบบให้มีการเข้ามายังกันภายในขั้นตอนน้อยที่สุดเท่าที่จะทำได้ เพื่อให้ application ที่ได้เขียนแล้ว compile เป็นครั้งเดียว แต่สามารถนำไปใช้ที่ไหนก็ได้ หรือว่าคือ รองรับทุก platform ที่ support Java โดยไม่จำเป็นต้อง recompile ใหม่ เพราะข้อมูลที่ compiled แล้วจะถูกเก็บในรูปของ bytecode ที่ใช้รันบน Java virtual machine (JVM) ตามสถาปัตยกรรมของเครื่องที่ Java ถูกนำมาใช้งานสูงมาก โดยเฉพาะงาน web application แบบ client-server



2.6 ความรู้เกี่ยวกับ เทคนิคการวัดความคล้าย

การวัดความคล้ายระหว่างวัตถุ สูตรที่ใช้คือ การวัดความคล้ายของวัตถุที่ถูกอธิบายด้วยคุณลักษณะหลายชนิด และกำหนดค่า俌หนักประจำคุณลักษณะ เพื่อใช้ในการแนะนำกีต้าร์ให้ตรงความต้องการของผู้ใช้งานมากที่สุด

คำนวณความคล้ายคลึงกันระหว่างสองวัตถุโดยใช้สูตรดังต่อไปนี้

$$\text{Similarity}(x, y) = \sum_{k=1}^n w_k s_k(x, y)$$

ขั้นตอนวิธี

- คุณลักษณะ k^{th} คำนวณความคล้ายคลึง $s_k(x, y)$ ในช่วง $[0, 1]$
- กำหนดค่า俌หนัก w_k เพื่อกำหนดจากความสำคัญของคุณลักษณะที่มีความสำคัญกว่าคุณลักษณะอื่นๆ ผลรวมของ俌หนัก w_k คือ 1

2.7 ตัวอย่างการคำนวณความคล้าย

- ผู้ใช้งานเลือกคุณลักษณะของกีต้าร์
- ผู้ใช้งานเลือกลำดับความสำคัญ หรือให้ค่า俌หนักแก่คุณลักษณะนั้นๆ

กำหนดคุณลักษณะของกีต้าร์ที่อยู่ในฐานข้อมูลมีดังนี้

ตารางที่ 2.1 ตัวอย่างชุดข้อมูลกีต้าร์ในฐานข้อมูล

	Skill	Brand	Guitar Type	Top Wood	Back & Side Wood	Price Range
Product 1	“Beginner”	“Epiphone”	“Electric”	non	non	“1,000-3,000”
Product 2	“Beginner”	“Fender”	“Acoustic”	non	non	“1,000-3,000”
Product 3	“Expert”	“Gibson”	“Acoustic”	“mahogany”	“mahogany”	“100,000-300,000”
Product 4	“Expert”	“PRS”	“Electric”	“red cedar”	“mahogany”	“10,000-30,000”

กำหนดค่า俌หนักประจำคุณลักษณะของกีต้าร์ดังต่อไปนี้

Beginer : (“***”= 0.4), (“**”=0.35), (“*”=0.25)

Expert : (“*****”=0.3), (“****”=0.25), (“***”= 0.2), (“**”=0.15), (“*”=0.1)



- Skill = Beginner :

คุณลักษณะที่ผู้ใช้งานเลือก

- Brand = “Epiphone” , “***”
- Guitar Type = “Electric” , “**”
- Price Range = “1,000-3,000” , “*”

ชุดข้อมูลที่ 1 Product 1

(ค่าน้ำหนักหรือลำดับความสำคัญ)*(Similarity(คุณลักษณะที่ผู้ใช้งานเลือก, Product 1))

$$= [(0.4)(1)+(0.35)(1)+(0.25)(1)]$$

$$= 1.0$$

ชุดข้อมูลที่ 2 Product 2

(ค่าน้ำหนักหรือลำดับความสำคัญ)*(Similarity(คุณลักษณะที่ผู้ใช้งานเลือก, Product 2))

$$= [(0.4)(0)+(0.35)(0)+(0.25)(1)]$$

$$= 0.25$$

สำหรับการแนะนำการเลือกซื้อกีต้าร์ แบบผู้เริ่มหัดเล่นนั้นจะเห็นว่าการวัดความคล้ายผลลัพธ์คือข้อมูล Product 1 เหมาะสมกับความต้องการของผู้ใช้งานระบบมากที่สุด ให้ผลลัพธ์การคำนวณความคล้ายเท่ากับ 1

- Skill = Expert :

คุณลักษณะที่ผู้ใช้งานเลือก

- Brand = “Gibson” , “*****”
- Guitar Type = “Acoustic” , “****”
- Top Wood = “mahogany” , “***”
- Back & Side = “mahogany” , “**”
- Price Range = “100,000-300,000” , “*”



ชุดข้อมูลที่ 1 Product 3

(ค่าน้ำหนักหรือลำดับความสำคัญ)*(Similarity(คุณลักษณะที่ผู้ใช้งานเลือก, Product 3))

$$= [(0.3)(1)+(0.25)(1)+(0.2)(1)+(0.15)(1)+(0.1)(1)]$$

$$= 1.0$$

ชุดข้อมูลที่ 2 Product 4

(ค่าน้ำหนักหรือลำดับความสำคัญ)*(Similarity(คุณลักษณะที่ผู้ใช้งานเลือก, Product 4))

$$= [(0.3)(0)+(0.25)(0)+(0.2)(0)+(0.15)(1)+(0.1)(0)]$$

$$= 0.15$$

สำหรับการแนะนำการเลือกซื้อกีต้าร์ แบบผู้เชี่ยวชาญนั้นจะเห็นว่าการวัดความคล้ายผลลัพธ์คือข้อมูล Product 3 เหมาะสมกับความต้องการของผู้ใช้งานระบบมากที่สุด ให้ผลลัพธ์การคำนวณความคล้ายเท่ากับ 1



บทที่ 3

วิเคราะห์และออกแบบระบบ

3.1 Requirement

3.1.1 Functional requirements

Front end For User.

- User สามารถเลือกการค้นหาข้อมูลของ **GuitarGuide** ผ่าน *Android Application* ได้
 - User สามารถเลือกการค้นหาข้อมูลแบบ **Beginner** (สำหรับผู้เริ่มต้น) ได้
 - User จะต้องเลือก *Guitar Types* (ชนิดกีต้าร์)
 - User จะต้องเลือก *Brands* (ยี่ห้อ)
 - User จะต้องเลือก *Price ranges* (ช่วงของราคา)
 - User จะต้องเลือกหนดลำดับความสำคัญของแต่ละคุณลักษณะด้วย
 - User สามารถเลือกการค้นหาข้อมูลแบบ **Expert** (สำหรับผู้เชี่ยวชาญ) ได้
 - User จะต้องเลือก *Guitar Type* (ชนิดกีต้าร์)
 - User จะต้องเลือก *Brands* (ยี่ห้อ)
 - User จะต้องเลือก *Top Tone Woods* (ไม้หน้า)
 - User จะต้องเลือก *Back & Side Tone Woods* (ไม้หลังและข้าง)
 - User จะต้องเลือก *Price range* (ช่วงของราคา)
 - User จะต้องเลือกหนดลำดับความสำคัญของแต่ละคุณลักษณะด้วย

Back end For Admin.

- Admin สามารถเข้าสู่ระบบของ **GuitarGuide** ผ่าน *Web Application* ได้
- Admin สามารถจัดการข้อมูลของ **GuitarGuide** ผ่าน *Web Application* ได้
 - Admin สามารถจัดการข้อมูล **Product** (กีต้าร์สำหรับแนะนำ) ได้
 - Admin สามารถเพิ่มข้อมูล **Product** ได้
 - Admin สามารถแก้ไขข้อมูล **Product** ได้
 - Admin สามารถลบข้อมูล **Product** ได้
 - Admin สามารถจัดการข้อมูล **Guitar Types** (ชนิดกีต้าร์) ได้
 - Admin สามารถเพิ่มข้อมูล **Guitar Type** ได้
 - Admin สามารถแก้ไขข้อมูล **Guitar Type** ได้
 - Admin สามารถลบข้อมูล **Guitar Type** ได้
 - Admin สามารถจัดการข้อมูล **Brands** (ยี่ห้อ) ได้
 - Admin สามารถเพิ่มข้อมูล **Brand** ได้



- Admin สามารถแก้ไขข้อมูล Brand ได้
- Admin สามารถลบข้อมูล Brand ได้
- Admin สามารถจัดการข้อมูล Top Tone Woods (ไม้หน้า) ได้
 - Admin สามารถเพิ่มข้อมูล Top Tone Wood ได้
 - Admin สามารถแก้ไขข้อมูล Top Tone Wood ได้
 - Admin สามารถลบข้อมูล Top Tone Wood ได้
- Admin สามารถจัดการข้อมูล Back & Side Tone Woods (ไม้หลังและข้าง) ได้
 - Admin สามารถเพิ่มข้อมูล Back & Side Tone Wood ได้
 - Admin สามารถแก้ไขข้อมูล Back & Side Tone Wood ได้
 - Admin สามารถลบข้อมูล Back & Side Tone Wood ได้
- Admin สามารถจัดการข้อมูล Necks (คอギต้าร์) ได้
 - Admin สามารถเพิ่มข้อมูล Neck ได้
 - Admin สามารถแก้ไขข้อมูล Neck ได้
 - Admin สามารถลบข้อมูล Neck ได้
- Admin สามารถจัดการข้อมูล Pickups (เครื่องกำเนิดเสียง) ได้
 - Admin สามารถเพิ่มข้อมูล Pickup ได้
 - Admin สามารถแก้ไขข้อมูล Pickup ได้
 - Admin สามารถลบข้อมูล Pickup ได้
- Admin สามารถจัดการข้อมูล Price ranges (ช่วงของราคา) ได้
 - Admin สามารถเพิ่มข้อมูล Price range ได้
 - Admin สามารถแก้ไขข้อมูล Price range ได้
 - Admin สามารถลบข้อมูล Price range ได้

3.1.2 Non-Functional requirements

- ระบบต้องมี User interface ที่สวยงาม
- ระบบต้องใช้งานง่ายไม่ซับซ้อน



3.2 ทฤษฎีการสร้าง Diagram

UML (Unified Modeling Language)

UML (Unified Modeling Language) เป็นภาษาแผนภาพที่ใช้แสดงการทำงานของระบบ ในการวิเคราะห์และออกแบบเชิงวัตถุ (Object Oriented Analysis and Design) ซึ่งแผนภาพที่ใช้แสดงนั้นมีหลายแบบด้วยกันได้แก่ Use Case Diagram, Class Diagram, Sequence Diagram และ Activity Diagram ซึ่งมีหลักการในการออกแบบดังต่อไปนี้

1. Use Case Diagram

Use Case Diagram คือแผนภาพที่แสดงการทำงานของผู้ใช้ระบบ (User) และความสัมพันธ์กับระบบย่อย (Sub systems) ภายในระบบใหญ่ ในการเขียน Use Case Diagram ผู้ใช้ระบบ (User) จะถูกกำหนดว่าให้เป็น Actor และระบบย่อย (Sub systems) คือ Use Case จุดประสงค์หลักของการเขียน Use Case Diagram ก็เพื่อเล่าเรื่องราวทั้งหมดของระบบว่ามีการทำงานอะไรบ้าง เป็นการดึง Requirement หรือเรื่องราวต่างๆ ของระบบจากผู้ใช้งาน ซึ่งถือว่าเป็นจุดเริ่มต้นในการวิเคราะห์และออกแบบระบบ สัญลักษณ์ที่ใช้ใน Use Case Diagram จะใช้สัญลักษณ์คนแทน Actor ใช้สัญลักษณ์วงรีแทน Use Case และใช้เส้นตรงในการเชื่อม Actor กับ Use Case เพื่อแสดงการใช้งานของ Use Case ของ Actor นอกเหนือไปนี้ Use Case ทุกๆ ตัวจะต้องอยู่ภายใต้สี่เหลี่ยมเดียวกันซึ่งมีชื่อของระบบระบุอยู่

ความสัมพันธ์ระหว่าง Use Case หมายถึงความสัมพันธ์ที่แต่ละ Use Case ภายในระบบเองมีความสัมพันธ์กันโดยความสัมพันธ์ของ Use Case นั้นสามารถแบ่งออกได้ 2 แบบคือ Include และ Extends

- ความสัมพันธ์แบบ Include หมายถึงการที่ Use Case หนึ่งเรียกใช้งาน Use Case อีกอันหนึ่งคล้ายๆ กับการเรียกใช้โปรแกรมย่อยโดยโปรแกรมหลัก การเขียนสัญลักษณ์แทนการ Include ของ Use Case นั้นใช้สัญลักษณ์เส้นประพร้อมหัวลูกศรซึ่งไปยัง Use Case ที่ถูกเรียกใช้งาน และมีคำว่า <<include>> กำกับอยู่บนเส้นลูกศร

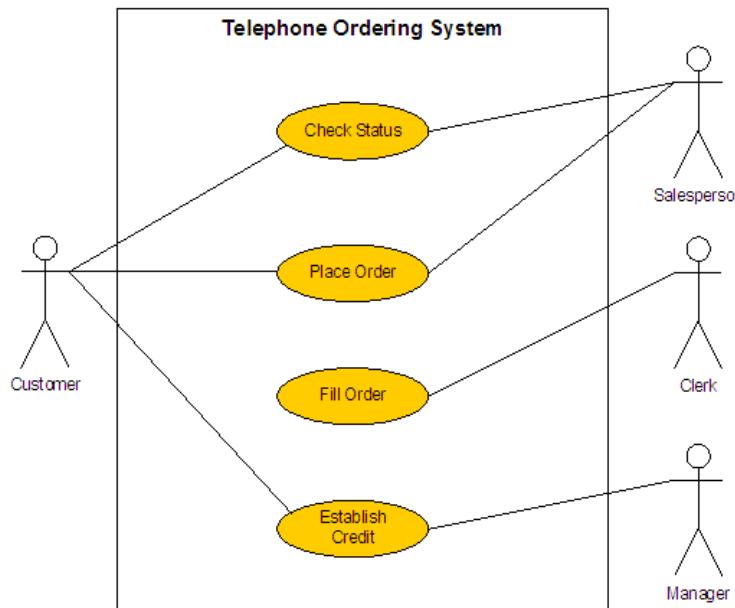
- ความสัมพันธ์แบบ Extend หมายถึงการที่ Use Case หนึ่งไปมีผลต่อการทำงานตามปกติของอีก Use Case หนึ่ง นั่นหมายถึง Use Case ที่มา Extend นั้นจะมีผลให้การทำงานของ Use Case ที่ถูก Extend ถูกรบกวนหรือมีการสะกดดูจาก Use Case ที่ Extend ไปยัง Use Case ที่ถูก Extend และมีคำว่า <<extend>> กำกับ

ประโยชน์ของ Use Case Diagram

1. ทราบความสามารถของระบบ
2. ทราบผู้ใช้งานในแต่ละส่วนของระบบ
3. จ่ายต่อการสื่อสารระหว่างลูกค้ากับผู้พัฒนาระบบ
4. ใช้ทดสอบระบบว่าตรงกับความต้องการของระบบหรือไม่
5. ช่วยให้ผู้พัฒนาระบบสามารถแยกแยะกิจกรรมที่อาจเกิดขึ้นในระบบ



6. เป็น Diagram พื้นฐานที่สามารถอธิบายสิ่งต่างๆได้โดยใช้รูปภาพไม่ซับซ้อน



ตัวอย่าง Use Case การสั่งซื้อสินค้าทางโทรศัพท์

2. Class Diagram

คือ แผนภาพที่ใช้แสดง Class และความสัมพันธ์ในแต่ละ Class ระหว่าง Class เหล่านั้นซึ่งความสัมพันธ์ที่กล่าวถึงใน Class Diagram นี้ถือเป็นความสัมพันธ์เชิงสถิต(Static Relation) หมายถึงความสัมพันธ์ที่มีอยู่แล้วเป็นปกติในระหว่าง Class ต่างๆซึ่งเรียกว่าความสัมพันธ์เชิงกิจกรรม(Dynamic Relationship)สิ่งที่ปรากฏในClass Diagram นั้นประกอบด้วยกลุ่มของ Class และกลุ่มของ Relationship โดยสัญลักษณ์ที่ใช้ในการแสดง Class นั้นจะแทนด้วยสี่เหลี่ยมที่แบ่งออกเป็น 3 ส่วน โดยแต่ละส่วนนั้นจะใช้ในการแสดงชื่อของ Class, Attribute, และฟังก์ชัน ตามลำดับ

ในการเขียนสัญลักษณ์แทน Class สิ่งที่ต้องคำนึงถึงอีกสิ่งหนึ่งคือ ระดับการเข้าถึงเรียกสัญลักษณ์ที่ใช้แทนการเข้าถึงนั่นว่า Visibility แบ่งออกได้เป็น 3 ประเภท ได้แก่

1. Private เขียนแทนด้วยสัญลักษณ์ - หมายถึง Attribute หรือ ฟังก์ชัน ที่ไม่สามารถมองเห็นได้จากภายนอก แต่สามารถมองเห็นได้จากภายในในตัวของ Class เองเท่านั้น
2. Protect เขียนแทนด้วยสัญลักษณ์ # หมายถึง Attribute หรือ ฟังก์ชัน ที่ส่วนไว้สำหรับการทำ Inheritance โดยเฉพาะ Attribute หรือ ฟังก์ชันเหล่านี้ จะเป็นของ Super class เมื่อทำการ Inheritance แล้ว Attribute หรือ ฟังก์ชัน ที่มี Visibility แบบ Protect จะถูกแปลงไปเป็น Private Attribute/ฟังก์ชัน หรือ Protected ขึ้นอยู่กับภาษา Programming ที่นำไปใช้

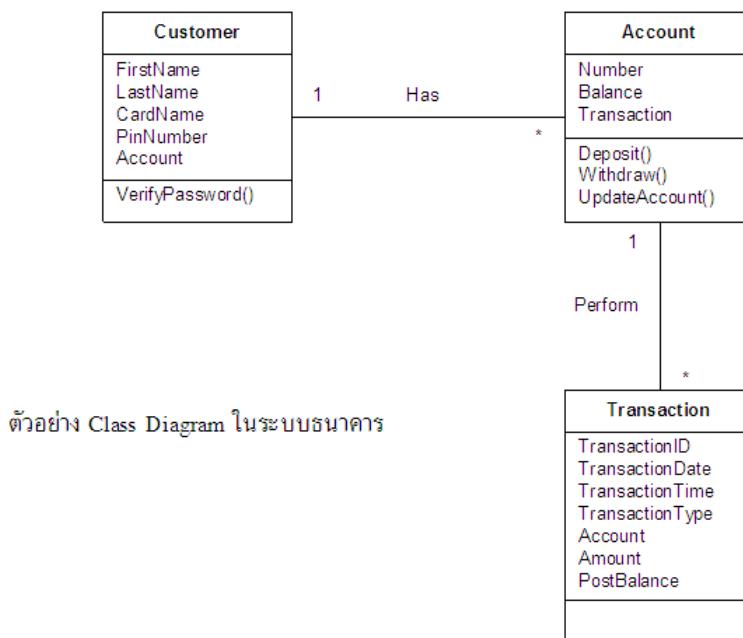


3. Public เขียนแทนด้วยสัญลักษณ์ + หมายถึง Attribute หรือ ฟังก์ชัน ที่สามารถมองเห็นได้จากภายนอก และสามารถเข้าไปเปลี่ยนค่า อ่านค่า หรือเรียกใช้งาน Attribute หรือ ฟังก์ชัน นั้นได้ทันทีโดยอิสระจากภายนอก (โดยทั่วไปแล้ว Visibility แบบ Public มักจะใช้กับฟังก์ชันมากกว่า Attribute)

หลักการในการสร้าง Class Diagram

1. กำหนดกรอบของ Problem Domain ให้ชัดเจน
2. พิจารณา Objects ที่สามารถจับต้องได้ เห็นได้ สัมผัสได้ (ถ้ามี)
3. พิจารณา Objects ที่ไม่สามารถจับต้องได้ (ถ้ามี)
4. ใช้ Classification Abstraction เพื่อแยกแยะและสร้าง Class จาก Objects ที่มีอยู่
5. หา Aggregation Abstraction โดยพิจารณา Class ที่ได้จากข้อ 4 ว่ามี Class ใด หรือไม่ที่มีความสัมพันธ์แบบเป็นส่วนหนึ่งหรือประกอบด้วย (Is Part of) กับ Class อื่นๆ
6. ใช้ Generalization มาพิจารณา Class ต่างๆ ใน Class Diagram
7. ใช้ Association มาพิจารณา Class ต่างๆ ใน Class Diagram
8. พิจารณา Class Diagram ที่สร้างมาทั้งหมดว่า ทุก Class และทุกกลุ่มของ Class มี ความสัมพันธ์ (Relationship) แบบใดแบบหนึ่งกับ Class หรือกลุ่มของ Class อื่นหรือไม่

Class Diagram





3. Sequence diagram

คือ การสร้างแบบจำลองเชิงกิจกรรม (Dynamic Model หรือ Behavioral Model) ซึ่งก็คือการจำลองกระบวนการที่ทำให้เกิดกิจกรรมของระบบ เกิดจากชุดของกิจกรรม ซึ่งกิจกรรมหนึ่ง ๆ นั้นเกิดจากการที่ Object หนึ่งโต้ตอบกับอีก Object หนึ่ง Sequence Diagram เป็น Diagram ที่ประกอบด้วย Class หรือ Object เส้นที่ใช้เพื่อแสดงลำดับเวลา และเส้นที่ใช้เพื่อแสดงกิจกรรมที่เกิดจาก Object หรือ Class ใน Diagram ภายใน Sequence Diagram จะใช้สีเหลี่ยมแทน Class หรือ Object ซึ่งภายในกรอบสีเหลี่ยมจะมีชื่อของ Object หรือ Class ประกอบอยู่ ในรูปแบบ Object: Class กิจกรรมที่เกิดขึ้นจะแทนด้วยลูกศร แนวอนที่ชี้จาก Class หรือ Object หนึ่งไปยัง Class หรือ Object ต่อไป

การระบุชื่อกิจกรรมนั้นจะอยู่ในรูปแบบ [Condition] พังก์ชัน ชื่อของกิจกรรมจะต้องเป็นพังก์ชัน ที่มีอยู่ใน Class หรือ Object ที่ลูกศรชี้ไป เส้นแสดงเวลาจะแทนด้วยเส้นตรงประแนvtting โดยเวลาจะเดินจากด้านบนลงมาสู่ด้านล่าง นั้นหมายถึงว่า ถ้าหากกิจกรรมที่เกิดขึ้นเกิดอยู่ด้านบนสุดกิจกรรมนั้นเป็นกิจกรรมแรก และกิจกรรมที่อยู่บริเวณต่ำลงมาจะเป็นกิจกรรมที่เกิดต่อจากนั้น

- Sequence Diagram เป็นแผนภาพที่ใช้อธิบายการทำงานของ Use Case เพื่อแสดงถึงขั้นตอนการทำงานและลำดับของการสื่อสาร (Message) ระหว่าง Object ที่ตอบโต้กัน
- Sequence Diagram จะแสดงอยู่ในรูปแบบ 2 มิติ โดยเส้นประแนvtting (Lifeline) จะนำเสนอด้วยเส้นในด้านเวลา ส่วนเส้นแนวอน (Message) จะนำเสนอกล่าวกับการโต้ตอบกันระหว่าง Object หรือ Class ต่างๆ

ประโยชน์ Sequence Diagram

-ช่วยในการพิจารณาว่าใน class diagram ที่สร้างขึ้นมี function ใดขาดหายไป หรือควรเพิ่มเติมเข้าไปอีกหรือไม่

-ทำให้ class ต่าง ๆ ที่สร้างขึ้นมีความสมบูรณ์มากยิ่งขึ้น

องค์ประกอบ Sequence Diagram

- Class / Object
- เส้นที่ใช้เพื่อแสดงลำดับเวลา
- เส้นที่ใช้เพื่อแสดงกิจกรรมที่เกิดขึ้นจาก object หรือ class

สัญลักษณ์ของ class/object

- แทนด้วย สีเหลี่ยม
- ภายในกรอบมีชื่อของ object/class ในรูปแบบ {Object};Class
- {Object} หมายถึง การระบุหรือไม่ระบุ object ก็ได้
- แทนด้วย ลูกศรแนวอน

สัญลักษณ์ของกิจกรรม

- ชี้จาก class/object หนึ่งไปยัง class/object ตัวต่อไป



-ชื่อของกิจกรรมจะต้องเป็น Function ที่มีอยู่ใน class/object ที่ลูกศรชี้ไป

-ชื่อจะอยู่ในรูปแบบ {[Condition]} Function

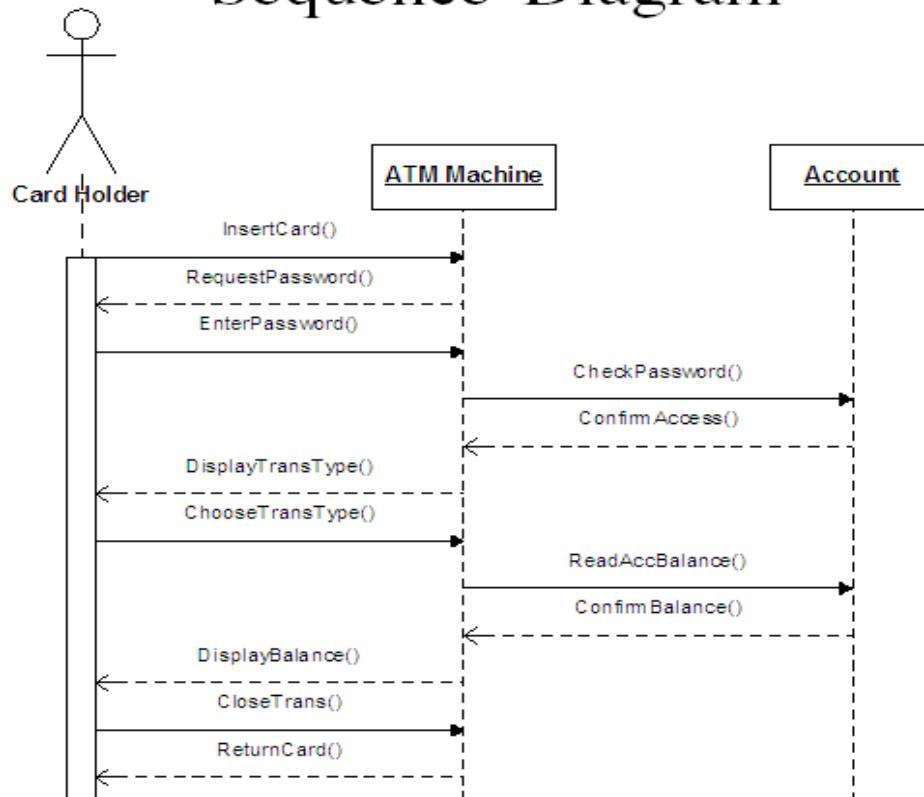
สัญลักษณ์ของเวลา

-แทนด้วย เส้นตรงประแนวตั้ง

-เวลาจะเดินจากด้านบนมาสู่ด้านล่าง

-กิจกรรมบนสุดถือเป็นกิจกรรมแรก และกิจกรรมที่อยู่ต่ำลงมาจะเป็นกิจกรรมที่เกิดขึ้นต่อจากนั้น

Sequence Diagram



4. Activity diagram

- Activity Diagram เป็นแผนภาพที่ใช้ที่แสดงขั้นตอนการทำงานของ use case (เช่นเดียวกับ Sequence Diagram) แต่จะเน้นไปที่งานย่อยของวัตถุ โดยจะมีกระบวนการทำงานคล้ายกับ Flowchart
- Activity Diagram บางครั้งมีลักษณะคล้าย Swimlane โดยจะแบ่งกลุ่มกิจกรรมที่เกิดขึ้นเป็นช่องโดยกำกับแต่ละช่องด้วยชื่อของ Object แต่ละ Swim lane แสดงถึงกิจกรรมที่เกิดขึ้นกับ Object นั้นๆ

Activities Diagram แสดงลำดับ กิจกรรมของการทำงาน (Work Flow) สามารถแสดงทางเลือกที่เกิดขึ้นได้ Activity Diagram จะแสดงขั้นตอนการทำงานในการปฏิบัติการ โดยประกอบไปด้วยสถานะต่างๆ ที่เกิดขึ้นระหว่างการทำงาน และผลจากการทำงานในขั้นตอนต่างๆ

ลักษณะของ Activity Diagram



- ໄດ້ອະແກນນີ້ມີລັກຊົນະເດືອກກັບໂພລວ່າຈັກ
- ໃຊ້ສໍາຫຼັບແສດງຂັ້ນຕອນກາರທຳການຂອງຮບບ
- ເຮັດວຽກຂັ້ນຕອນແຕ່ລະຂັ້ນຕອນວ່າ Activity ຈຶ່ງມີລັກຊົນການທຳການດັ່ງນີ້
- ກາຣຄໍານວນພລັພົບປະປະກາຣ
- ກາຣປ່ລິຍແປລັງສຖານະ (state) ຂອງຮບບ
- ກາຣສ່າງຄ່າບາງຍ່າງກລັບຄືນມາ
- ກາຣເຮັດວຽກໃໝ່Operation ອື່ນໆ ທຳການ
- ກາຣສ່າງສັນຍາລຸ່ມ
- ກາຣສ່າງ ຮີ່ອກາຣທຳລາຍອອບເຈັກຕົກ
- ວັກລົມສີດຳ ຄືວ ຈຸດເຮີມຕັນ
- ວັກລົມສີດຳ ມົງລ້ວມອີກຂັ້ນ ຄືວ ຈຸດສິ້ນສຸດ
- Swim lanes ເປັນກາຣແບ່ງກລຸ່ມ activity ເປັນຂ່ອງໃນແນວດິງ ແລະ ກຳນົດແຕ່ລະຂ່ອງດ້ວຍເຊື້ອ object ໄວດ້ານນີ້ ກາຣແບ່ງເປັນ Swim lanes ຜ່າຍໃຫ້ແຍກແຍະຜູ້ຮັບຜິດຂອບ ແຕ່ລະ ຈະໄດ້ວ່າໂຄຣຄວາມຈະເປັນຄົນທຳການໃນໜຸ່ມຫຸ່ມໄດ້

ກາຣໃຊ້ Activity Diagram ແບບທົ່ວໄປ

- ກາຣທຳລາຍງານພຣ້ອມກັນ

ກາຣທຳລາຍງານພຣ້ອມກັນໃຫ້ໃຊ້ເສັນຕຽນແນວນອນເສັນໜາທີ່ເຮັດວຽກ ສະແດງວ່າ ສົມ ມາເປັນສັນຍາລັກຊົນທີ່ໃຊ້ຈັດກລຸ່ມຈານທີ່ມີກາຣທຳການພຣ້ອມໆ ກັນ ຮີ່ອ ກາຣທຳກິຈກຽມໃນລັກຊົນຂ່າຍໆ

ກາຣແບ່ງກາຣທາງານໃຫ້ເປັນສັດສ່ວນ

- ຄຸນລັກຊົນຍ່າງໜຶ່ງຂອງກາຣໃຊ້ Activity Diagram ຄືວ ສາມາຮັດແສດງໃຫ້ເຫັນວ່າໂຄຣຮັບຜິດຂອບກິຈກຽມໃນສ່ວນໃໝ່ ສາຫຼັບ Activity Diagram ໃນ UML 2.0 ມີກາຣະບຸຄາມຮັບຜິດຂອບໄດ້ 2 ວິທີ

◦ Swim lanes

◦ Partition names

ກາຣກຳນົດຈຸດສິ້ນສຸດຂອງກິຈກຽມ

- ເມື່ອເຮົາຕ້ອງກາຣກຳນົດຈຸດສິ້ນສຸດຂອງໂພລວ່າກາຣທຳການ ຈະໃຊ້ສັນຍາລັກຊົນທັງນີ້

ກາຣເລືອກໃຊ້ Activity Diagram ກັບ State Diagram

- Activity Diagram ຈະມີລັກຊົນຄລ້າຍຄລື່ງກັບ Flow Chart ໂດຍຈະແສດງຄື່ງລາດັບຂັ້ນຕອນກາຣທາງານຈຸດທີ່ມີກາຣຕັດສິນໃຈ ແລະ ກາຣແຕກກາຣທາງານເປັນສ່ວນຍ່ອຍໆ

- Activity Diagram ຈຶ່ງມີປະໂຍບນໍາກສາຫຼັບໃຊ້ອື່ນບາຍຄື່ງກາຣທາງານຂອງອອບເຈັກຕົກຕ່າງໆ ລວມຄື່ງກະບວນກາຣທາງານຕ່າງໆ ດ້ວຍ

- ໃຊ້ Activity Diagram ເມື່ອເປັນກາຣປ່ລິຍແປລັງແຫຼຸກກາຣນີ່ຕ່າງໆ ອັນເປັນພລມາຈາກກາຣກະທາກາຍໃນຮະບບເອງ ມີລັກຊົນທີ່ຕ່ອນເນື່ອກັນ ແລະ ມັກຈະມີຈຸດທີ່ຮະບບານຕ້ອງມີກາຣຕັດສິນໃຈ



- ใช้ State Diagram เมื่อเป็นการเปลี่ยนแปลงเหตุการณ์ต่างๆ ในลักษณะไม่ต่อเนื่องกัน ให้ใช้ State Diagram เพื่อแสดงสถานะต่างๆที่เกิดขึ้นภายใน Diagram

คุณสมบัติของ Activity Diagram ที่ดี

- มุ่งเน้นการติดต่อสื่อสารของระบบในเชิงไดนามิก
- เนพาะอิเลิมิเนต์ที่มีความสำคัญต่อกระบวนการทางานเท่านั้น
- แสดงรายละเอียดในแต่ละระดับการทำงาน โดยเลือกแสดงเนพาะที่มีความสำคัญต่อการเข้าใจการทำงานของระบบเท่านั้น
- ถ้าการทำงานส่วนใดมีความสำคัญ ก็ควรเขียน Activity Diagram ไม่ควรละเจ้าไว้หรือแสดงเพียงอย่างย่อๆ

3.2.1 Use Case Diagram

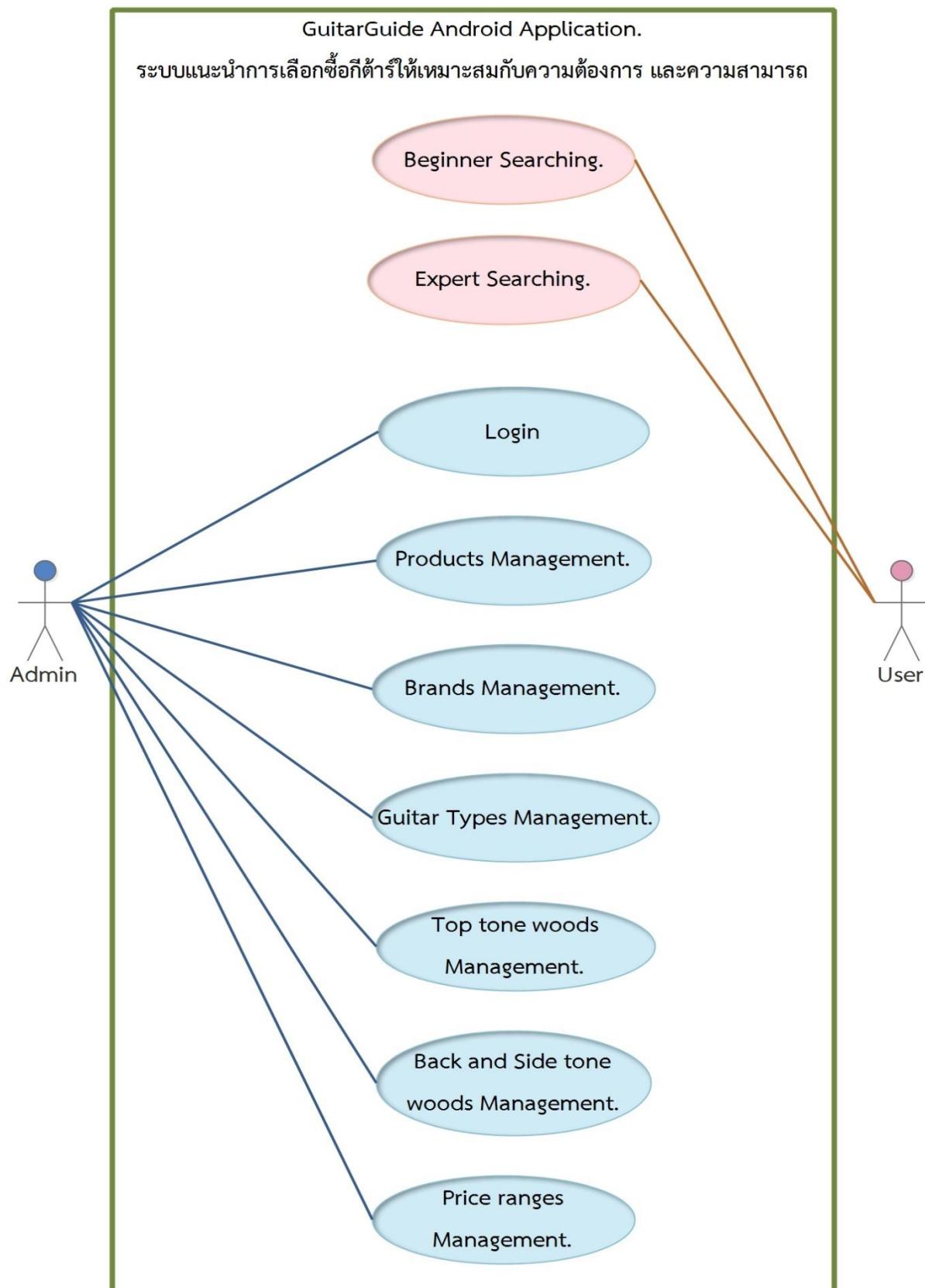
Use case main diagram

GuitarGuide Android Application ระบบแนะนำการเลือกซื้อ กีต้าร์ให้เหมาะสมกับความต้องการ และความสามารถ สามารถแบ่งระบบออกเป็น ส่วนหลักๆ ได้แก่ ระบบ Beginner Searching, ระบบ Expert Searching, ระบบ Login, ระบบ Products Management, ระบบ Brands Management, ระบบ Guitar Types Management, ระบบ Top tone woods Management, ระบบ Back and Side tone woods Management, ระบบ Price ranges Management ซึ่งมีรายละเอียดดังนี้



ตารางที่ 3.1 Use case Main Diagram

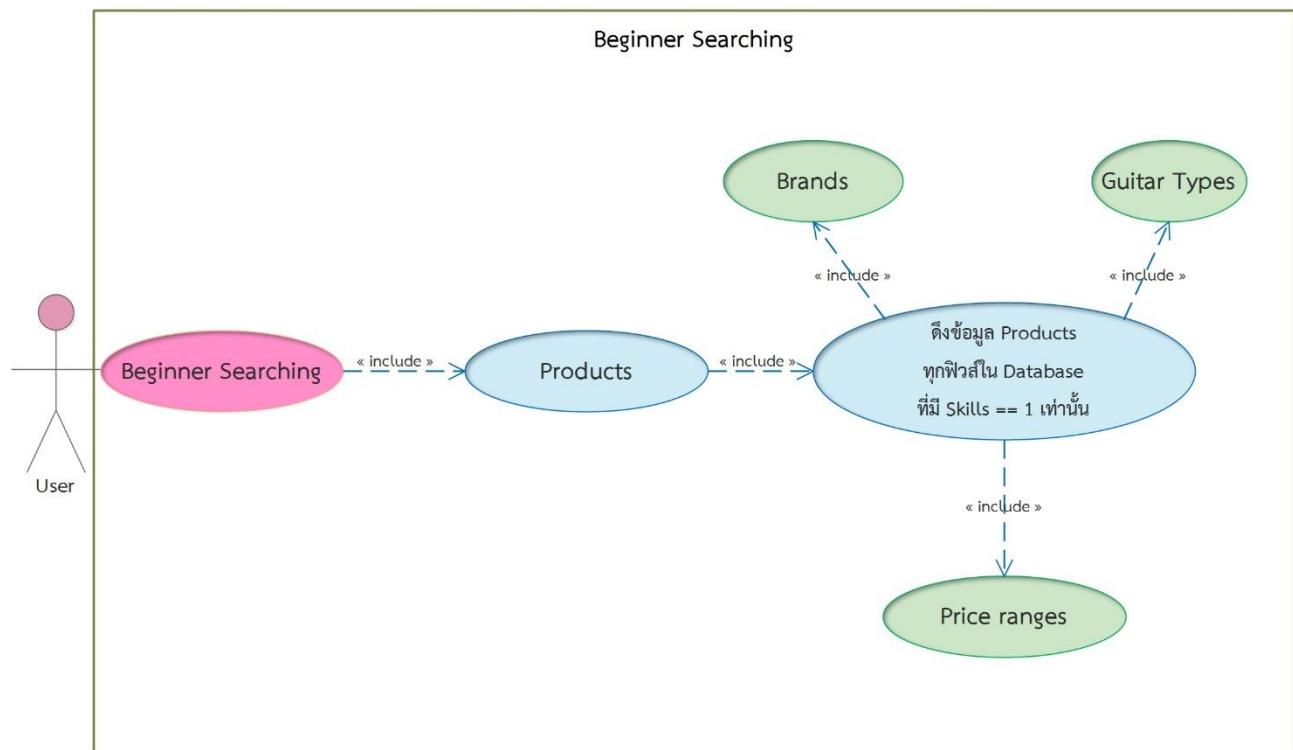
Diagram ID	Diagram Name	Diagram Description
UC_M	Main	Use Case Main Diagram
UC_01	Beginner Searching	ค้นหาข้อมูล สำหรับผู้เริ่มต้น
UC_02	Expert Searching	ค้นหาข้อมูล สำหรับ ผู้เชี่ยวชาญ
UC_03	Login	เข้าสู่ระบบ สำหรับการจัดการ
UC_04	Products Management	จัดการข้อมูล กีต้าร์สำหรับ แนะนำ
UC_05	Brands Management	จัดการข้อมูลยี่ห้อ
UC_06	Guitar Types Management	จัดการข้อมูล ชนิดกีต้าร์
UC_07	Top tone woods Management	จัดการข้อมูล ไม้หน้า
UC_08	Back and Side tone woods Management	จัดการข้อมูล ไม้หลังและไม่ ข้าง
UC_09	Price ranges Management	จัดการข้อมูล ช่วงราคา



รูปที่ 3.1 Use case main diagram



3.2.1 ค้นหาข้อมูลสำหรับผู้เริ่มต้น (Beginner Searching)



รูปที่ 3.2 Use case diagram UC_01:Beginner Searching



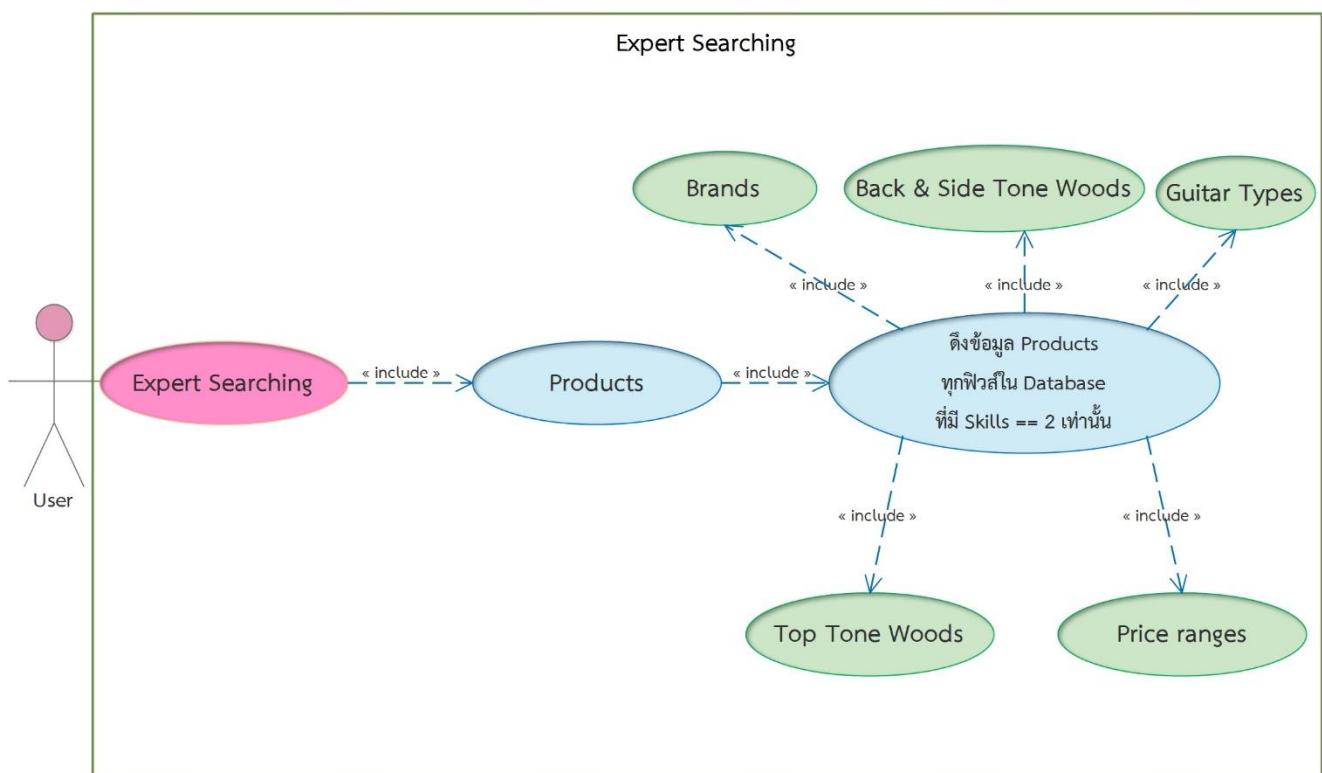
ตารางที่ 3.2 Use case description UC_01:Beginner Searching

Diagram ID	UC_01	
Use case Name	Beginner Searching	
Triggering Event	เมื่อผู้ใช้ต้องการค้นหาข้อมูลสำหรับผู้เริ่มต้น	
Brief Description	<p>ผู้ใช้ระบบ</p> <p>1. เมื่อต้องการค้นหาข้อมูลกีต้าร์สำหรับผู้เริ่มต้น</p>	
Actors	ผู้ใช้ระบบ	
Related Use Case	-	
Stakeholders	-	
Pre-Condition	<p>ผู้ใช้ระบบ</p> <p>1.ต้องทำการเลือกเมนู Beginner Searching</p>	
Post-Condition	1. สามารถค้นหาข้อมูลค้นหาข้อมูลกีต้าร์สำหรับผู้เริ่มต้น ได้สำเร็จ	
Flow of Event	Actor	System
(ลำดับขั้นตอนการ ทำงานระหว่าง User กับ System)	<p>ผู้ใช้ระบบ</p> <p><u>เลือกเมนู Beginner Searching</u></p> <p>1.กำหนดคุณลักษณะ และกำหนด ลำดับความสำคัญ</p> <p>1.1 เลือกคุณลักษณะของ Brands และกำหนดลำดับความสำคัญ</p> <p>1.2 เลือกคุณลักษณะของ Guitar types และกำหนดลำดับความสำคัญ</p> <p>1.3 เลือกคุณลักษณะของ Price ranges และกำหนดลำดับความสำคัญ</p> <p>2.กดปุ่ม SEARCH</p>	<p>แสดงหน้าฟอร์มตัวเลือก การค้นหา สำหรับผู้เริ่มต้น</p> <p>1.ทำการ Query ข้อมูลจากฐานข้อมูล จากนั้นทำการคำนวณความคล้ายคลึง</p>



	3.กดที่เมนูข้อมูล 4.กลับหน้าหลัก	2.แสดงผลอ กมาทางจอภาพ Smart Phone 1.ไปยังหน้าเว็บไซต์สำหรับขายกีต้าร์ตัวนั้นๆ 1.กลับหน้าหลัก
Exceptions	การเลือกการค้นหา ไม่จำเป็นจะต้องกดปุ่มเลือกทั้ง 3 เมนูย่อย สามารถกดปุ่มเลือกการค้นหาได้ตั้งแต่ 1 , 2 หรือ 3 เมนู	

3.2.2 ค้นหาข้อมูลสำหรับผู้เชี่ยวชาญ (Expert Searching)



รูปที่ 3.3 Use case diagram UC_02:Expert Searching



ตารางที่ 3.3 Use case description UC_02:Expert Searching

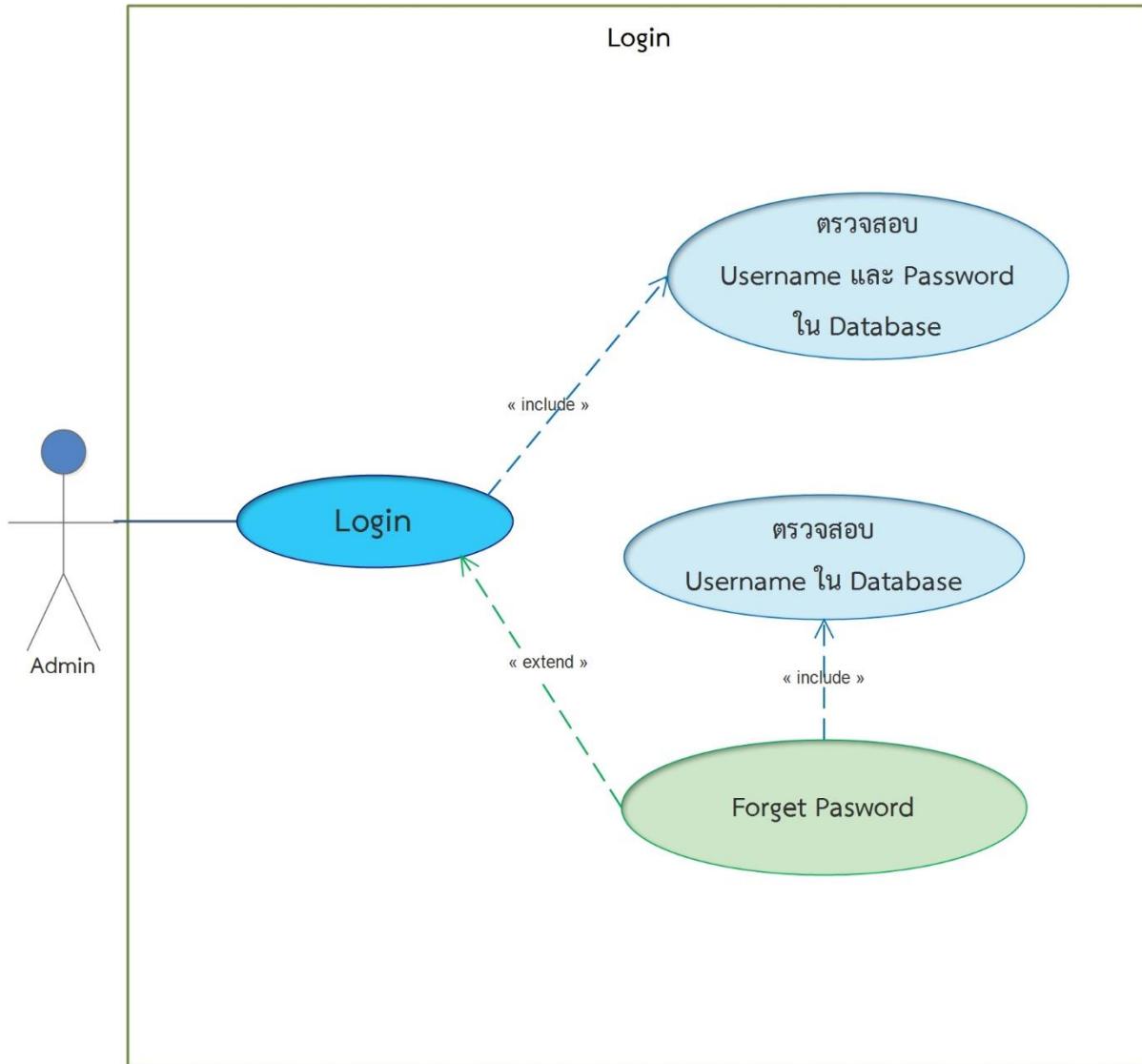
Diagram ID	UC_02	
Use case Name	Expert Searching	
Triggering Event	เมื่อผู้ใช้ต้องการค้นหาข้อมูลสำหรับผู้เชี่ยวชาญ	
Brief Description	<p>ผู้ใช้ระบบ</p> <p>1. เมื่อต้องการค้นหาข้อมูลเกี่ยวกับผู้เชี่ยวชาญ</p>	
Actors	ผู้ใช้ระบบ	
Related Use Case	-	
Stakeholders	-	
Pre-Condition	<p>ผู้ใช้ระบบ</p> <p>1. ต้องทำการเลือกเมนู Expert Searching</p>	
Post-Condition	สามารถค้นหาข้อมูลค้นหาข้อมูลเกี่ยวกับผู้เชี่ยวชาญ ได้สำเร็จ	
Flow of Event	Actor	System
(ลำดับขั้นตอนการ ทำงานระหว่าง User กับ System)	<p>ผู้ใช้ระบบ</p> <p><u>เลือกเมนู Expert Searching</u></p> <p>1. กดปุ่มเลือก ตัวเลือกที่ ต้องการค้นหา</p> <p>1.1 เลือกคุณลักษณะของ Brands และกำหนดลำดับความสำคัญ</p> <p>1.2 เลือกคุณลักษณะของ Guitar types และกำหนดลำดับความสำคัญ</p> <p>1.3 เลือกคุณลักษณะของ Price ranges และกำหนดลำดับความสำคัญ</p>	<p>แสดงหน้าฟอร์มตัวเลือก การค้นหา สำหรับผู้เชี่ยวชาญ</p>



	<p>1.4 เลือกคุณลักษณะของ Top tone woods และกำหนดลำดับความสำคัญ</p> <p>1.5 เลือกคุณลักษณะของ Back and Side tone woods และกำหนดลำดับความสำคัญ</p> <p>2. กดปุ่ม SEARCH</p> <p>3. กดที่เมนูข้อมูล</p> <p>4. กลับหน้าหลัก</p>	<p>1. ทำการ Query ข้อมูลจากฐานข้อมูล จากนั้นทำการคำนวณความคล้ายคลึง</p> <p>1. เปย়องหน้าเว็บไซต์สำหรับขายกีต้าร์ตัว นั้นๆ</p> <p>1. กลับหน้าหลัก</p>
Exceptions	การเลือกการค้นหา ไม่จำเป็นจะต้องกดปุ่มเลือกทั้ง 3 เมนูย่อย สามารถกดปุ่มเลือกการค้นหาได้ตั้งแต่ 1,2,3,4,5,6.1 หรือ 6.2 เมนู	



3.2.3 เข้าสู่ระบบสำหรับ Admin (Login)



รูปที่ 3.4 Use case diagram UC_03:Login



ตารางที่ 3.4 Use case description UC_03:Login

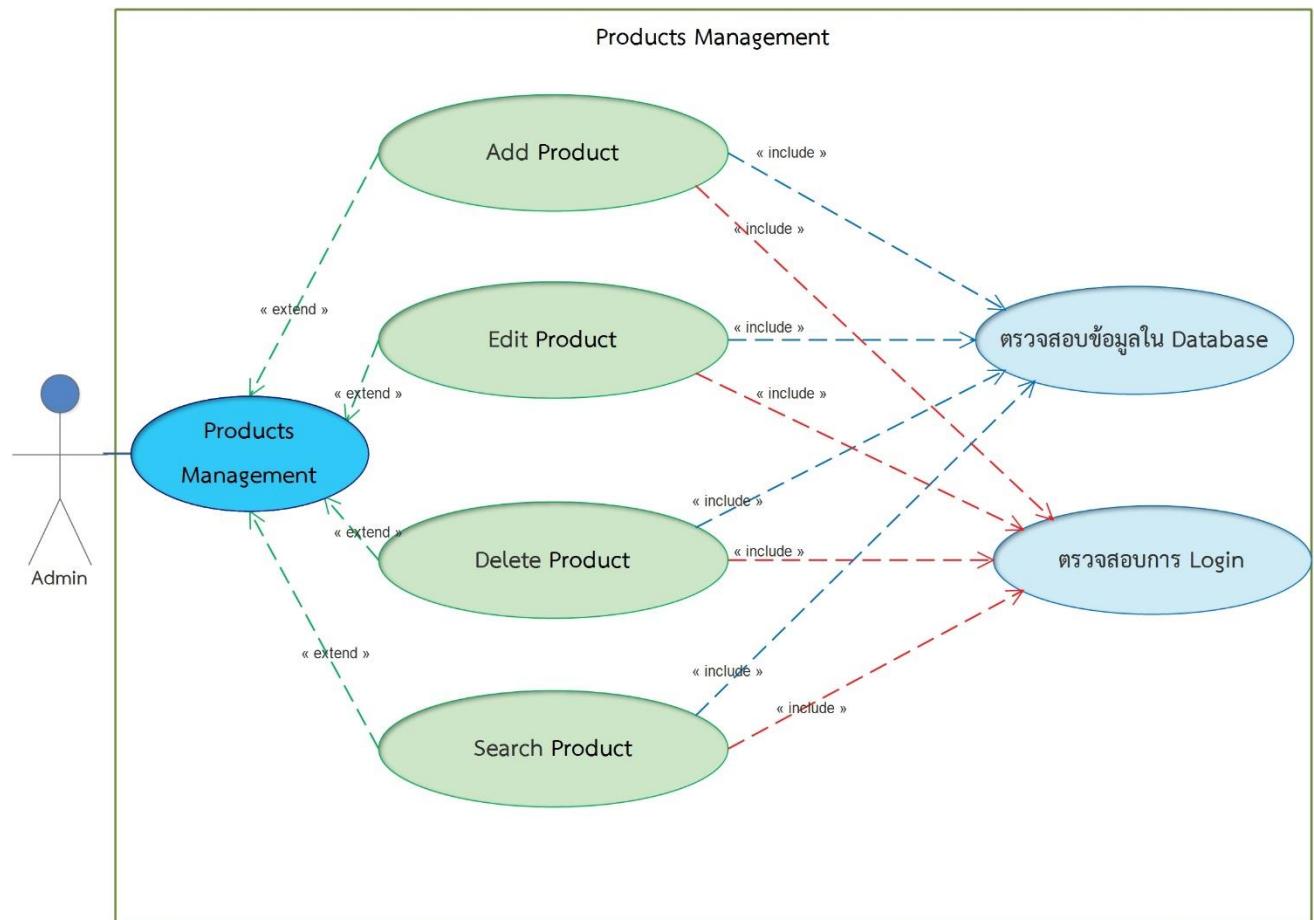
Diagram ID	UC_03	
Use case Name	Login	
Triggering Event	เมื่อผู้ดูแลระบบต้องการเข้าสู่ระบบ และลืมรหัสผ่าน	
Brief Description	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. เมื่อต้องการเข้าสู่ระบบ 2. เมื่อต้องการขอรับรหัสผ่านใหม่ (ลืมรหัสผ่าน) 	
Actors	ผู้ดูแลระบบ	
Related Use Case	UC_04,UC_05,UC_06, UC_07, UC_08, UC_09,UC_10,UC_11	
Stakeholders	-	
Pre-Condition	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1.ต้องทำการเลือกเมนูเข้าสู่ระบบ 2.ต้องทำการเลือกเมนูลืมรหัสผ่าน 3.ต้องเป็นผู้มีสิทธิ์เท่านั้น ที่จะสามารถเข้าใช้งานระบบได้ 	
Post-Condition	<ol style="list-style-type: none"> 1.สามารถเข้าสู่ระบบได้สำเร็จ 2.สามารถขอรับรหัสผ่านใหม่ (ลืมรหัสผ่าน) ได้สำเร็จ 	
Flow of Event	Actor	System
(ลำดับขั้นตอนการ ทำงานระหว่าง User กับ System)	<p><u>ผู้ดูแลระบบ</u></p> <p><u>เลือกเมนูเข้าสู่ระบบ</u></p> <p>1.ทำการกรอก Username และ Password จากนั้นกดปุ่ม OK</p> <p>1.1 ข้อมูลที่กรอกถูกต้อง</p>	<p>ระบบจะแสดงหน้าการเข้าสู่ระบบ เพื่อให้ Admin เข้าสู่ระบบ</p> <p>ระบบทำการตรวจสอบ Username และ Password นั้น จากฐานข้อมูลว่าถูกต้องหรือไม่</p> <p>ระบบจะเข้าสู่หน้าหลัก และ พร้อมสำหรับการจัดการต่อไป</p>



	<p>1.2 ข้อมูลที่กรอกไม่ถูกต้อง</p> <p>2.กดปุ่มลีมรหัสผ่าน</p> <p>2.1 ทำการกรอกUsername (เดิม) และ Password(ใหม่) และกรอก Password(ที่กรอกไว้ก่อนหน้า) อีกครั้งเพื่อเป็นการยืนยัน ลงในแบบฟอร์มแล้วกดปุ่ม OK</p> <p>2.1.1 กรณีกรอก Username และ Password ถูกต้อง</p> <p>2.1.2 กรณีกรอก Username และ Password ไม่ถูกต้อง</p> <p>2.1.3 กรณีการยกเลิก</p>	<p>1.ระบบจะแสดง “Username หรือ Password ไม่ถูกต้อง! กรุณากรอกข้อมูลให้ถูกต้อง”</p> <p>2.ระบบจะกลับไปหน้าการเข้าสู่ระบบ เพื่อกรอกข้อมูลใหม่</p> <p>ระบบแสดงหน้าแบบฟอร์มการลีมรหัสผ่าน</p> <p>ระบบทำการประมวลผล ตรวจสอบความถูกต้องของ Username และตรวจสอบความตรงกันของ Password ที่กรอกมา ทั้งสองครั้งว่าตรงกันหรือไม่</p> <p>1.ระบบจะเข้าสู่หน้าหลัก และพร้อมสำหรับการจัดการต่อไป</p> <p>1.ระบบจะแสดง “Username ไม่ถูกต้อง หรือ Password ไม่ตรงกัน! กรุณากรอกข้อมูลให้ถูกต้อง”</p> <p>2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่ ระบบจะกลับไปหน้าการ Login เพื่อทำการเข้าสู่ระบบ</p>
Exceptions	หากกรอก Username, Password ผิดพลาดหรือไม่มีอยู่ในระบบ ระบบจะแจ้งเตือนพร้อมแสดงข้อความให้ทราบ แล้วกลับไปหน้าแบบฟอร์มเพื่อกรอกข้อมูลใหม่	



3.2.4 จัดการข้อมูลของกีต้าร์สำหรับใช้ในการแนะนำ (Products Management)



รูปที่ 3.5 Use case diagram UC_04: Products Management

ตารางที่ 3.5 Use case description UC_04: Products Management

Diagram ID	UC_04
Use case Name	Products Management
Triggering Event	เมื่อผู้ดูแลระบบต้องการจัดการข้อมูลหลักของกีต้าร์ (Products)
Brief Description	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. เมื่อต้องการเพิ่ม 2. เมื่อต้องการแก้ไข 3. เมื่อต้องการลบ 4. เมื่อต้องการค้นหา



Actors	ผู้ดูแลระบบ	
Related Use Case	UC_05, UC_06, UC_07, UC_08, UC_09, UC_10, UC_11	
Stakeholders	-	
Pre-Condition	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. ต้องเพิ่มข้อมูลหลักของกีต้าร์ได้ 2. ต้องแก้ไขข้อมูลหลักของกีต้าร์ได้ 3. ต้องลบข้อมูลหลักของกีต้าร์ได้ 4. ต้องค้นหาข้อมูลหลักของกีต้าร์ได้ 	
Post-Condition	<ol style="list-style-type: none"> 1. สามารถเพิ่มข้อมูลหลักของกีต้าร์ได้สำเร็จ 2. สามารถแก้ไขข้อมูลหลักของกีต้าร์ได้สำเร็จ 3. สามารถลบข้อมูลหลักของกีต้าร์ได้สำเร็จ 4. สามารถค้นหาข้อมูลหลักของกีต้าร์ได้สำเร็จ 	
Flow of Event	Actor	System
(ลำดับขั้นตอนการ ทำงานระหว่าง User กับ System)	<p>ผู้ดูแลระบบ</p> <p><u>เลือกเมนู Products Management</u></p> <p>1. กดที่ปุ่มเพิ่ม</p> <p>1.1 ทำการกรอกข้อมูลให้ครบถ้วนถูกต้อง</p> <p>1.1.1 กรณีข้อมูลที่กรอกครบถ้วนถูกต้อง</p>	<p>ระบบจะแสดงหน้าการจัดการข้อมูลหลักของกีต้าร์ (Products Management)</p> <p>1. ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง</p> <p>2. ระบบจะทำการตรวจสอบ ว่าชื่อของข้อมูลที่ต้องการเพิ่มนี้ มีข้อมูลของสีเข้ากันหรือไม่</p> <p>1. ระบบจะทำการบันทึกลง Database</p>



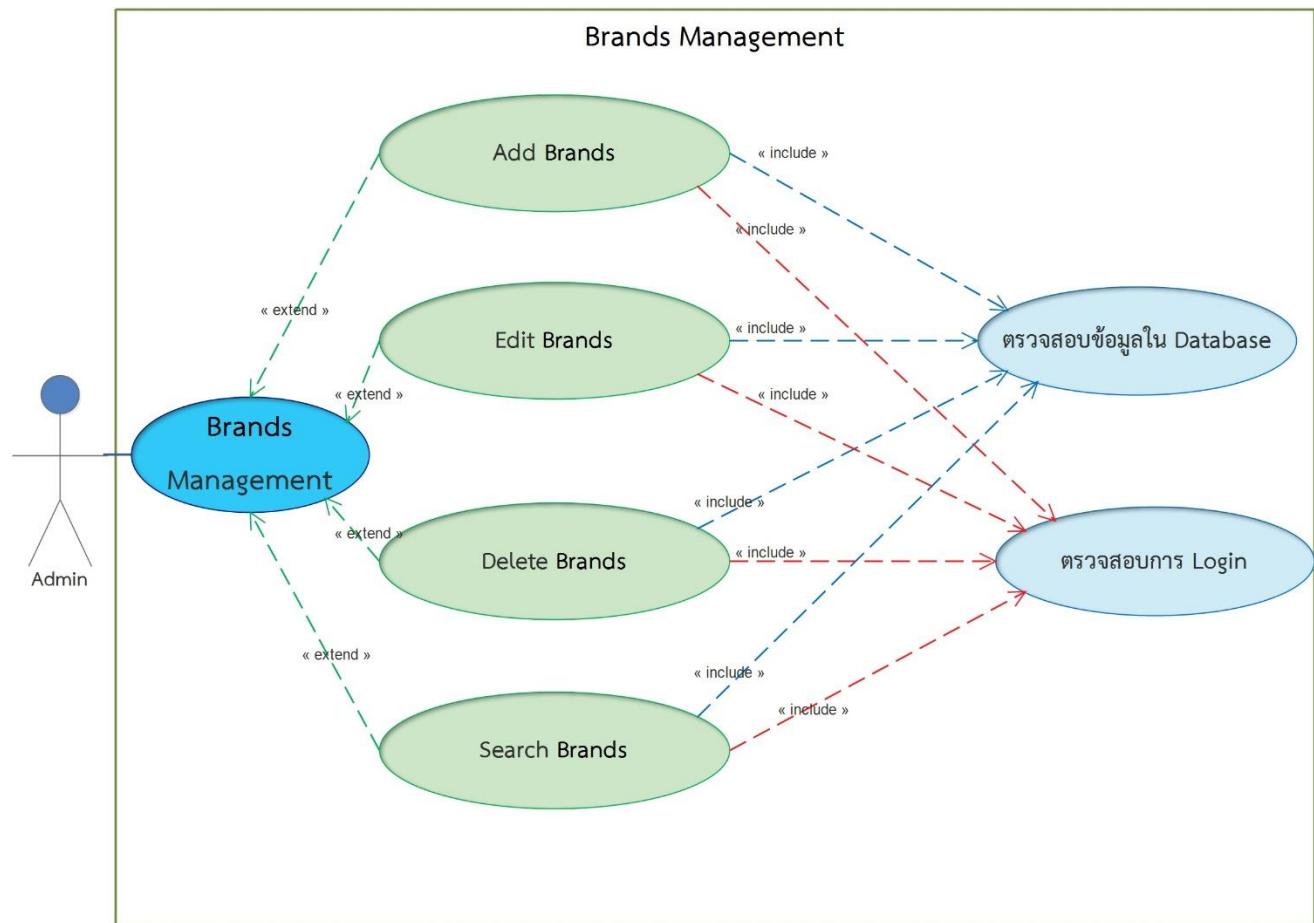
		2.ระบบจะแสดง “ทำการเพิ่มข้อมูลเรียบร้อยแล้ว”
	1.1.2 กรณีข้อมูลที่กรอกไม่ครบและไม่ถูกต้อง	1.ระบบจะแสดง “ทำการเพิ่มไม่สำเร็จ กรุณากรอกข้อมูลให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่
	1.1.3 กรณีกดปุ่มยกเลิก	ระบบจะกลับไปหน้าการจัดการข้อมูลหลักของกิ๊ต้าร์ (Products Management)
	2.กดปุ่มแก้ไข	
	2.1 ทำการแก้ไขข้อมูลและกรอกให้ครบถ้วนถูกต้อง	1.ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง 2.ระบบจะทำการตรวจสอบ ว่าชื่อของข้อมูลที่ต้องการเพิ่มนี้ มีข้อมูลของสีซ้ำกันหรือไม่
	2.1.1 กรณีข้อมูลที่กรอกครบถ้วนถูกต้อง	1.ระบบจะทำการบันทึกลง Database 2.ระบบจะแสดง “ทำการแก้ไขข้อมูลเรียบร้อยแล้ว”
	2.1.2 กรณีข้อมูลที่กรอกไม่ครบและไม่ถูกต้อง	1.ระบบจะแสดง “ทำการแก้ไขไม่สำเร็จ กรุณากรอกข้อมูลให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่
	2.1.3 กรณีกดปุ่มยกเลิก	ระบบจะกลับไปหน้าการจัดการข้อมูลหลักของกิ๊ต้าร์ (Products Management)



	<p>3. กดปุ่มลบ</p> <p>3.1 กรณีกดปุ่ม OK</p> <p>3.2 กรณีกดปุ่มยกเลิก</p> <p>4. กดปุ่มค้นหาข้อมูล</p> <p>4.1 พิมพ์คำค้น</p> <p>4.1.1 กรณีพบรหัสผ่าน</p> <p>4.1.2 กรณีไม่พบ</p>	<p>1. ระบบจะทำการลบข้อมูลที่อยู่ใน Database</p> <p>2. ระบบจะแสดง “ทำการลบข้อมูลเรียบร้อยแล้ว”</p> <p>3. ระบบจะกลับไปหน้าการจัดการข้อมูลหลักของกีต้าร์ (Products Management)</p> <p>ระบบจะกลับไปหน้าการจัดการข้อมูลหลักของกีต้าร์ (Products Management)</p> <p>1. ทำการตรวจสอบคำค้นนั้นใน Database</p> <p>ระบบจะทำการแสดงผลการค้นข้อมูลออกทางหน้าจอของระบบ</p> <p>ระบบจะทำการแสดง “ไม่พบคำค้น” และกลับไปหน้าหลัก</p>
Exceptions	หากกรอกข้อมูล ผิดพลาดหรือไม่ครบถ้วนและถูกต้อง ระบบจะแจ้งเตือน พร้อมแสดงข้อความให้ทราบ และกลับไปหน้าแบบฟอร์มเพื่อกรอกข้อมูลใหม่	



3.2.5 จัดการข้อมูลยี่ห้อกิ๊ต้าร์ (Brands Management)



รูปที่ 3.6 Use case diagram UC_05: Brands Management

ตารางที่ 3.6 Use case description UC_05: Brands Management

Diagram ID	UC_05
Use case Name	Brands Management
Triggering Event	เมื่อผู้ดูแลระบบต้องการจัดการกับข้อมูลยี่ห้อกิ๊ต้าร์ (Brands)
Brief Description	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. เมื่อต้องการเพิ่ม 2. เมื่อต้องการแก้ไข 3. เมื่อต้องการลบ 4. เมื่อต้องการค้นหา



Actors	ผู้ดูแลระบบ	
Related Use Case	UC_04	
Stakeholders	-	
Pre-Condition	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. ต้องเพิ่มข้อมูลได้ 2. ต้องแก้ไขข้อมูลได้ 3. ต้องลบข้อมูลได้ 4. ต้องค้นหาข้อมูลได้ 	
Post-Condition	<ol style="list-style-type: none"> 1. สามารถเพิ่มข้อมูลได้สำเร็จ 2. สามารถแก้ไขข้อมูลได้สำเร็จ 3. สามารถลบข้อมูลได้สำเร็จ 4. สามารถค้นหาข้อมูลได้สำเร็จ 	
Flow of Event	Actor	System
(ลำดับขั้นตอนการ ทำงานระหว่าง User กับ System)	<p><u>ผู้ดูแลระบบ</u></p> <p><u>เลือกเมนู Brans Management</u></p> <p>1. กดที่ปุ่มเพิ่ม</p> <p>1.1 ทำการกรอกชื่อให้ครบถ้วนถูกต้อง</p> <p>1.1.1 กรณีชีอที่ กรอกครบถ้วนถูกต้อง</p>	<p>ระบบจะแสดงหน้าการจัดการกับข้อมูลยี่ห้อกีต้าร์ (Brands Management)</p> <p>1. ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง</p> <p>2. ระบบจะทำการตรวจสอบว่าชื่อของข้อมูลที่ต้องการเพิ่มนี้ มีข้อมูลกันหรือไม่</p> <p>1. ระบบจะทำการบันทึกลง Database</p>



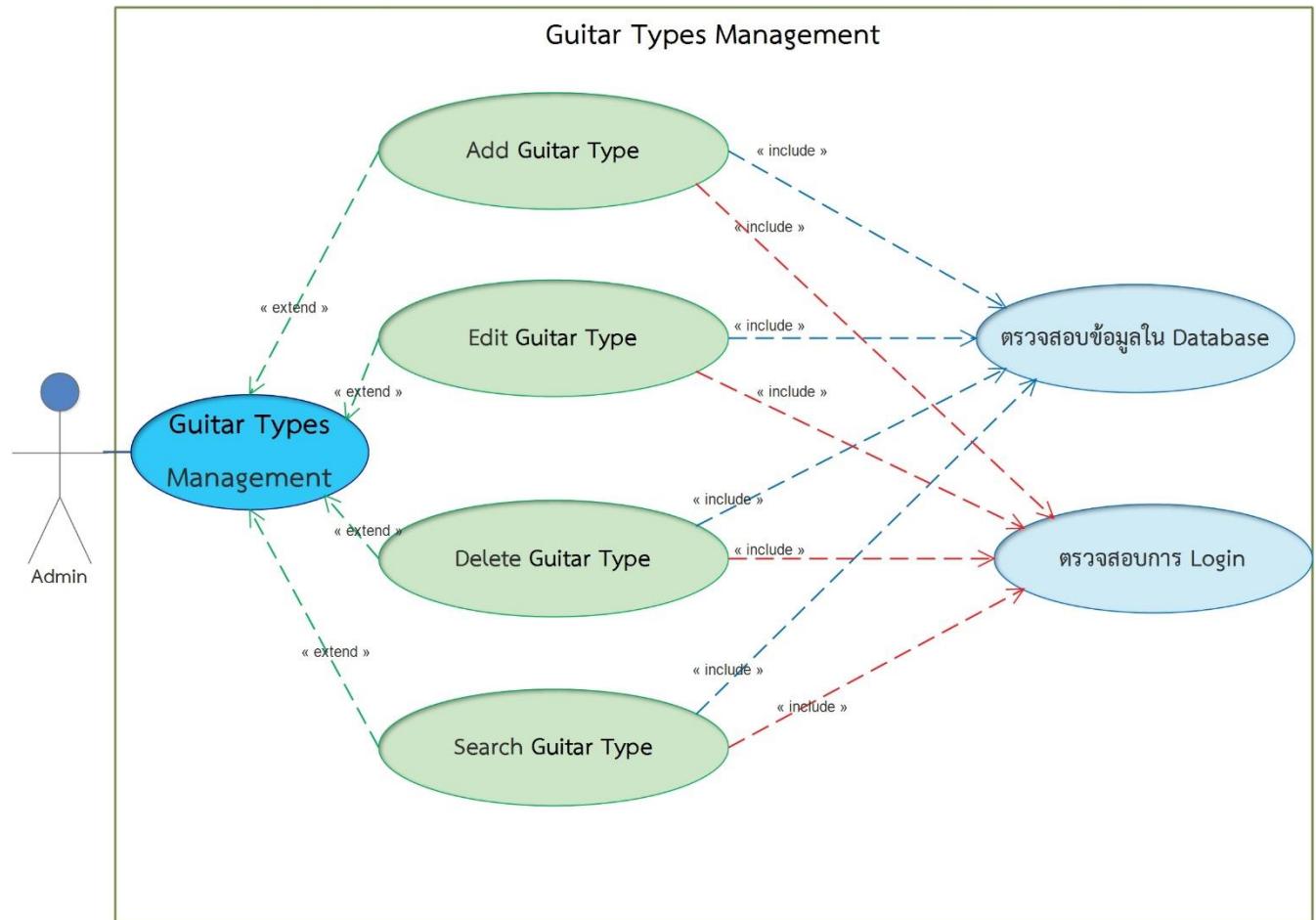
		2.ระบบจะแสดง “ทำการเพิ่มข้อมูลเรียบร้อยแล้ว”
	1.1.2 กรณีซื้อที่กรอกไม่ครบและไม่ถ้วนถูกต้อง	1.ระบบจะแสดง “ทำการเพิ่มไม่สำเร็จ กรุณากรอกซื้อให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่
	1.1.3 กรณีกดปุ่มยกเลิก	ระบบจะกลับไปหน้าการจัดการข้อมูล (Brands Management)
	2.กดปุ่มแก้ไข	
	2.1 ทำการแก้ไขซื้อและการให้ครบถ้วนถูกต้อง	1.ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง 2.ระบบจะทำการตรวจสอบ ว่าซื้อที่ต้องการเพิ่มนี้ มีซื้อซ้ำกันหรือไม่
	2.1.1 กรณีซื้อที่กรอกครบถ้วนถูกต้อง	1.ระบบจะทำการบันทึกลง Database 2.ระบบจะแสดง “ทำการแก้ไขข้อมูลเรียบร้อยแล้ว”
	2.1.2 กรณีซื้อที่กรอกไม่ครบและไม่ถ้วนถูกต้อง	1.ระบบจะแสดง “ทำการแก้ไขไม่สำเร็จ กรุณากรอกซื้อให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่ ระบบจะกลับไปหน้าการจัดการข้อมูล (Brands Management)
	2.1.3 กรณีกดปุ่มยกเลิก	
	3.กดปุ่มลบ	



	<p>3.1 กรณีกดปุ่ม OK</p> <p>3.2 กรณีกดปุ่มยกเลิก</p> <p>4. กดปุ่มค้นหาข้อมูล</p> <p>4.1 พิมพ์คำค้น</p> <p>4.1.1 กรณีพบ</p> <p>4.1.2 กรณีไม่พบ</p>	<p>1.ระบบจะทำการลบข้อมูลที่อยู่ใน Database</p> <p>2.ระบบจะแสดง “ทำการลบข้อมูล เรียบร้อยแล้ว”</p> <p>3.ระบบจะกลับไปหน้าการจัดการ ข้อมูล (Brands Management)</p> <p>ระบบจะกลับไปหน้าการจัดการ ข้อมูล (Brands Management)</p> <p>1.ทำการตรวจสอบคำค้นนั้นใน Database</p> <p>ระบบจะทำการแสดงผลการค้น ข้อมูลของทางหน้าจอของระบบ</p> <p>ระบบจะทำการแสดง “ไม่พบ คำค้น” และกลับไปหน้าหลัก</p>
Exceptions	หากกรอกข้อมูล ผิดพลาดหรือไม่ครบถ้วนและถูกต้อง ระบบจะแจ้ง เตือน พร้อมแสดงข้อความให้ทราบ และกลับไปหน้าแบบฟอร์มเพื่อ กรอกข้อมูลใหม่	



3.2.6 จัดการข้อมูลชนิดกีต้าร์ (Guitar Types Management)



รูปที่ 3.7 Use case diagram UC_06: Guitar Types Management

ตารางที่ 3.7 Use case description UC_06: Guitar Types Management

Diagram ID	UC_06
Use case Name	Guitar Types Management
Triggering Event	เมื่อผู้ดูแลระบบต้องการจัดการกับข้อมูลชนิดกีต้าร์ (Guitar Types)
Brief Description	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. เมื่อต้องการเพิ่ม 2. เมื่อต้องการแก้ไข 3. เมื่อต้องการลบ 4. เมื่อต้องการค้นหา



Actors	ผู้ดูแลระบบ	
Related Use Case	UC_04	
Stakeholders	-	
Pre-Condition	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. ต้องเพิ่มข้อมูลได้ 2. ต้องแก้ไขข้อมูลได้ 3. ต้องลบข้อมูลได้ 4. ต้องค้นหาข้อมูลได้ 	
Post-Condition	<ol style="list-style-type: none"> 1. สามารถเพิ่มข้อมูลได้สำเร็จ 2. สามารถแก้ไขข้อมูลได้สำเร็จ 3. สามารถลบข้อมูลได้สำเร็จ 4. สามารถค้นหาข้อมูลได้สำเร็จ 	
Flow of Event	Actor	System
(ลำดับขั้นตอนการ ทำงานระหว่าง User กับ System)	<p>ผู้ดูแลระบบ</p> <p><u>เลือกเมนู Guitar Types Management</u></p> <p>1. กดที่ปุ่มเพิ่ม</p> <p>1.1 ทำการกรอกชื่อให้ครบถ้วนถูกต้อง</p> <p>1.1.1 กรณีชีอที่ กรอกครบถ้วนถูกต้อง</p>	<p>ระบบจะแสดงหน้าการจัดการกับข้อมูลชนิดกีต้าร์ (Guitar Types Management)</p> <p>1. ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง</p> <p>2. ระบบจะทำการตรวจสอบ ว่าชื่อของข้อมูลที่ต้องการเพิ่มนี้ มีข้อมูล กันหรือไม่</p> <p>1. ระบบจะทำการบันทึกลง Database</p>



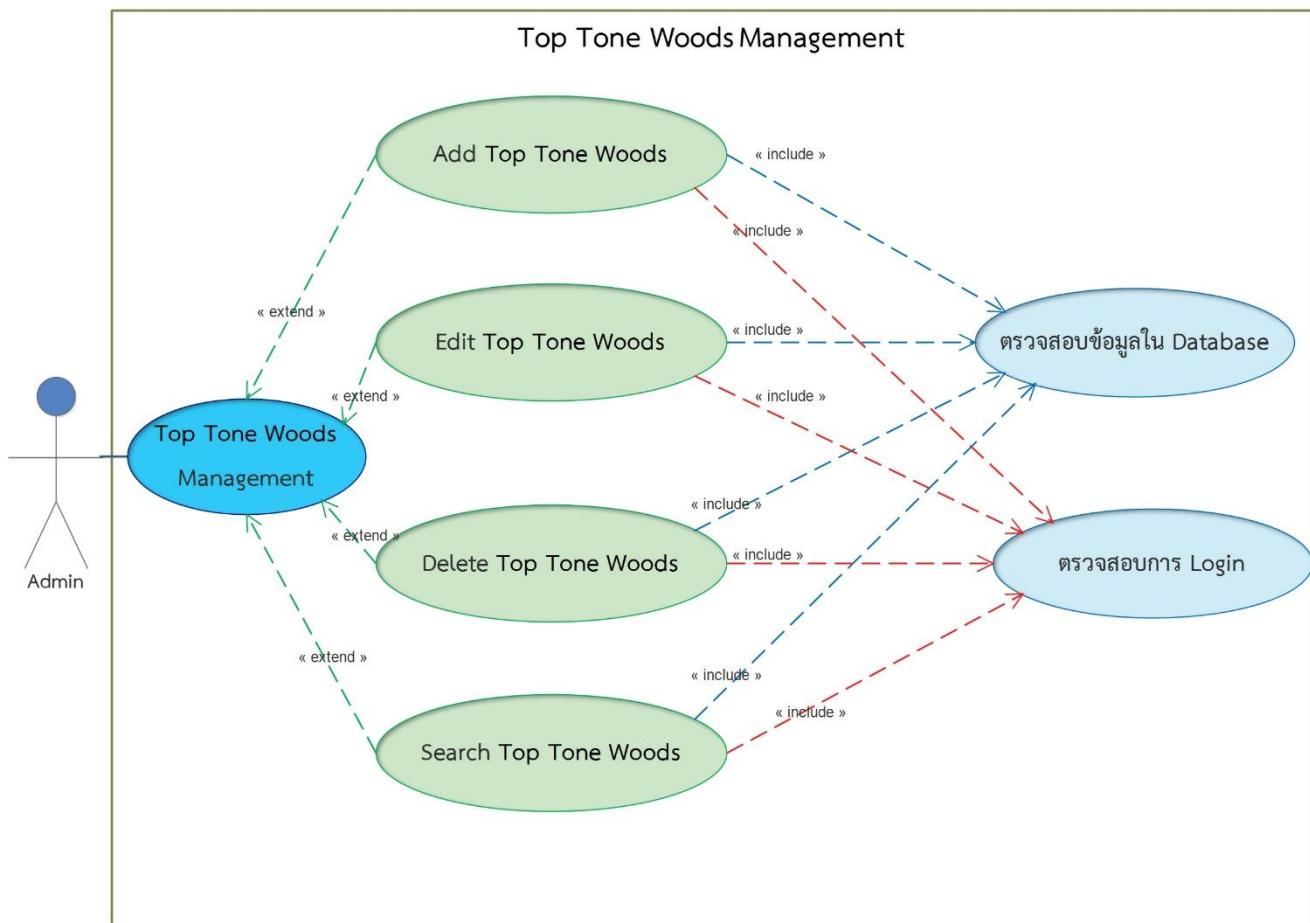
		2.ระบบจะแสดง “ทำการเพิ่มข้อมูลเรียบร้อยแล้ว”
	1.1.2 กรณีซื้อที่กรอกไม่ครบและไม่ถ้วนถูกต้อง	1.ระบบจะแสดง “ทำการเพิ่มไม่สำเร็จ กรุณากรอกซื่อให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่
	1.1.3 กรณีกดปุ่มยกเลิก	ระบบจะกลับไปหน้าการจัดการข้อมูลหลักของกีต้าร์ (Guitar Types Management)
	2.กดปุ่มแก้ไข	
	2.1 ทำการแก้ไขซื้อและกรอกให้ครบถ้วนถูกต้อง	1.ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง 2.ระบบจะทำการตรวจสอบ ว่าซื้อที่ต้องการเพิ่มนี้ มีซื้อซ้ำกันหรือไม่
	2.1.1 กรณีซื้อที่กรอกครบถ้วนถูกต้อง	1.ระบบจะทำการบันทึกลง Database 2.ระบบจะแสดง “ทำการแก้ไขข้อมูลเรียบร้อยแล้ว”
	2.1.2 กรณีซื้อที่กรอกไม่ครบและไม่ถ้วนถูกต้อง	1.ระบบจะแสดง “ทำการแก้ไขไม่สำเร็จ กรุณากรอกซื่อให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่ ระบบจะกลับไปหน้าการจัดการข้อมูล (Guitar Types Management)
	2.1.3 กรณีกดปุ่มยกเลิก	



	<p>3. กดปุ่มลบ</p> <p>3.1 กรณีกดปุ่ม OK</p> <p>3.2 กรณีกดปุ่มยกเลิก</p>	<p>1. ระบบจะทำการลบข้อมูลที่อยู่ใน Database</p> <p>2. ระบบจะแสดง “ทำการลบข้อมูลเรียบร้อยแล้ว”</p> <p>3. ระบบจะกลับไปหน้าการจัดการข้อมูล (Guitar Types Management)</p> <p>ระบบจะกลับไปหน้าการจัดการข้อมูล (Guitar Types Management)</p>
	<p>4. กดปุ่มค้นหาข้อมูล</p> <p>4.1 พิมพ์คำค้น</p> <p>4.1.1 กรณีพบรหัสผ่าน</p> <p>4.1.2 กรณีไม่พบคำค้น</p>	<p>1. ทำการตรวจสอบคำค้นนั้นใน Database</p> <p>ระบบจะทำการแสดงผลการค้นข้อมูลออกทางหน้าจอของระบบ</p> <p>ระบบจะทำการแสดง “ไม่พบคำค้น” และกลับไปหน้าหลัก</p>
Exceptions	หากกรอกข้อมูล ผิดพลาดหรือไม่ครบถ้วนและถูกต้อง ระบบจะแจ้งเตือน พร้อมแสดงข้อความให้ทราบ และกลับไปหน้าแบบฟอร์มเพื่อกรอกข้อมูลใหม่	



3.2.7 จัดการข้อมูลสำหรับ ไม้ด้านหน้าของตัวกีต้าร์ Top Tone Woods (Top Tone Woods Management)



รูปที่ 3.8 Use case diagram UC_07: Top Tone Woods Management

ตารางที่ 3.8 Use case description UC_07: Top Tone Woods Management

Diagram ID	UC_07
Use case Name	Top Tone Woods Management
Triggering Event	เมื่อผู้ดูแลระบบต้องการจัดการกับข้อมูลไม้หน้า (Top Tone Woods)
Brief Description	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. เมื่อต้องการเพิ่ม 2. เมื่อต้องการแก้ไข 3. เมื่อต้องการลบ



	4. เมื่อต้องการค้นหา	
Actors	ผู้ดูแลระบบ	
Related Use Case	UC_04	
Stakeholders	-	
Pre-Condition	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. ต้องเพิ่มข้อมูลได้ 2. ต้องแก้ไขข้อมูลได้ 3. ต้องลบข้อมูลได้ 4. ต้องค้นหาข้อมูลได้ 	
Post-Condition	<ol style="list-style-type: none"> 1. สามารถเพิ่มข้อมูลได้สำเร็จ 2. สามารถแก้ไขข้อมูลได้สำเร็จ 3. สามารถลบข้อมูลได้สำเร็จ 4. สามารถค้นหาข้อมูลได้สำเร็จ 	
Flow of Event	Actor	System
(ลำดับขั้นตอนการ ทำงานระหว่าง User กับ System)	<p>ผู้ดูแลระบบ</p> <p><u>เลือกเมนู Top Tone Woods Management</u></p> <p>1. กดที่ปุ่มเพิ่ม</p> <p>1.1 ทำการกรอกชื่อให้ครบถ้วนถูกต้อง</p> <p>1.1.1 กรณีชื่อที่ กรอกครบถ้วนถูกต้อง</p>	<p>ระบบจะแสดงหน้าการจัดการกับข้อมูล (Top Tone Woods Management)</p> <p>1. ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง</p> <p>2. ระบบจะทำการตรวจสอบ ว่าชื่อของข้อมูลที่ต้องการเพิ่มนี้ มีข้อมูลกันหรือไม่</p> <p>1. ระบบจะทำการบันทึกลง Database</p>



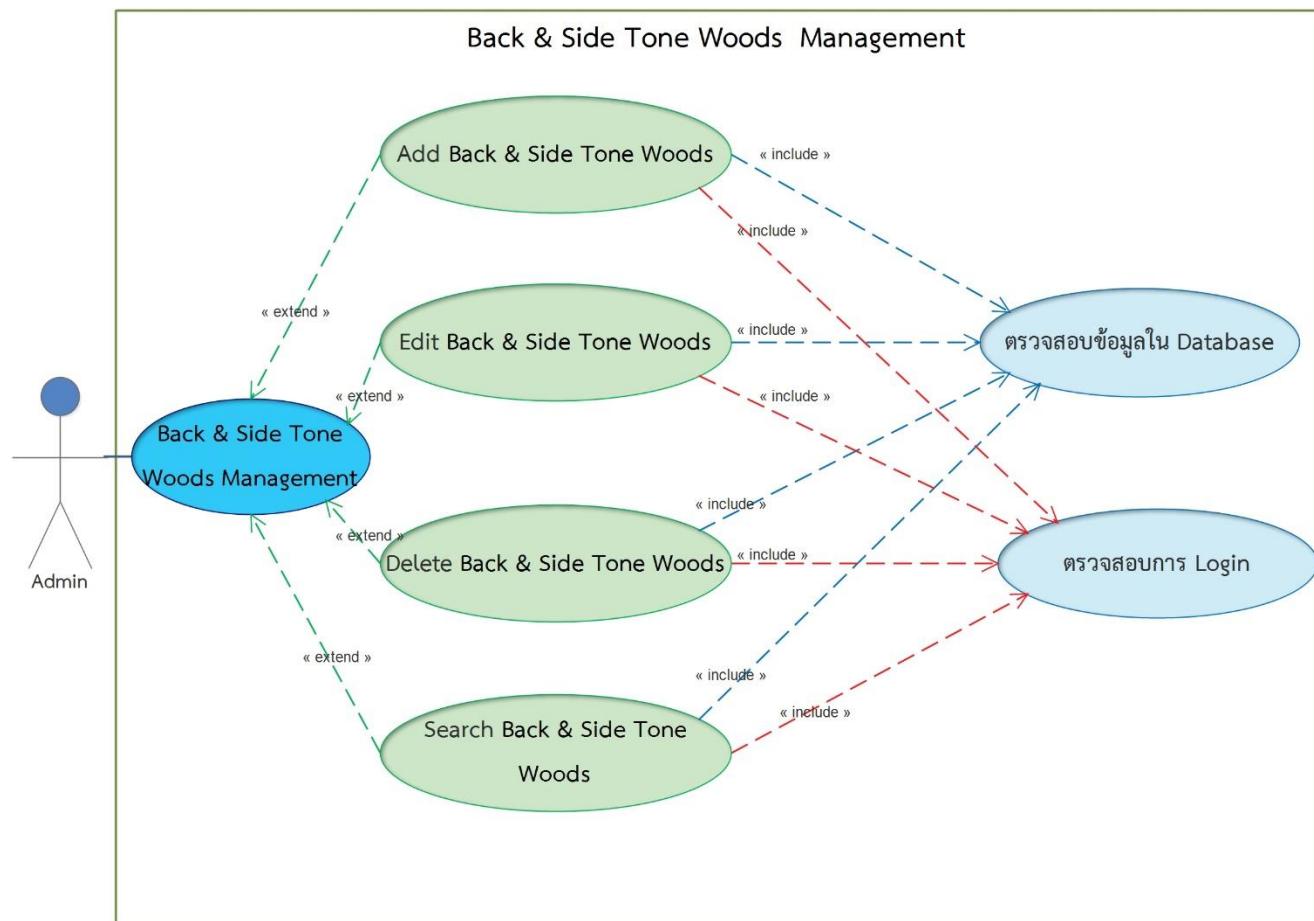
		2.ระบบจะแสดง “ทำการเพิ่มข้อมูลเรียบร้อยแล้ว”
	1.1.2 กรณีซื้อที่กรอกไม่ครบและไม่ถ้วนถูกต้อง	1.ระบบจะแสดง “ทำการเพิ่มไม่สำเร็จ กรุณากรอกซื่อให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่
	1.1.3 กรณีกดปุ่มยกเลิก	ระบบจะกลับไปหน้าการจัดการข้อมูล (Top Tone Woods Management)
	2.กดปุ่มแก้ไข	1.ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง 2.ระบบจะทำการตรวจสอบ ว่าซื้อที่ต้องการเพิ่มนี้ มีซื้อซ้ำกันหรือไม่
	2.1.1 กรณีซื้อที่กรอกครบถ้วนถูกต้อง	1.ระบบจะทำการบันทึกลง Database 2.ระบบจะแสดง “ทำการแก้ไขข้อมูลเรียบร้อยแล้ว”
	2.1.2 กรณีซื้อที่กรอกไม่ครบและไม่ถ้วนถูกต้อง	1.ระบบจะแสดง “ทำการแก้ไขไม่สำเร็จ กรุณากรอกซื่อให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่
	2.1.3 กรณีกดปุ่มยกเลิก	ระบบจะกลับไปหน้าการจัดการข้อมูล (Top Tone Woods Management)



	<p>3. กดปุ่มลบ</p> <p>3.1 กรณีกดปุ่ม OK</p> <p>3.2 กรณีกดปุ่มยกเลิก</p>	<p>1. ระบบจะทำการลบข้อมูลที่อยู่ใน Database</p> <p>2. ระบบจะแสดง “ทำการลบข้อมูลเรียบร้อยแล้ว”</p> <p>3. ระบบจะกลับไปหน้าการจัดการข้อมูล (Top Tone Woods Management)</p> <p>ระบบจะกลับไปหน้าการจัดการข้อมูล (Top Tone Woods Management)</p>
	<p>4. กดปุ่มค้นหาข้อมูล</p> <p>4.1 พิมพ์คำค้น</p> <p>4.1.1 กรณีพบรูป</p> <p>4.1.2 กรณีไม่พบ</p>	<p>1. ทำการตรวจสอบคำค้นนั้นใน Database</p> <p>ระบบจะทำการแสดงผลการค้นข้อมูลอุปกรณ์ทางหน้าจอของระบบ</p> <p>ระบบจะทำการแสดง “ไม่พบคำค้น” และกลับไปหน้าหลัก</p>
Exceptions	หากกรอกข้อมูล ผิดพลาดหรือไม่ครบถ้วนและถูกต้อง ระบบจะแจ้งเตือน พร้อมแสดงข้อความให้ทราบ และกลับไปหน้าแบบฟอร์มเพื่อกรอกข้อมูลใหม่	



3.2.8 จัดการข้อมูลไม้หลังและข้างสำหรับสร้างกีต้าร์ (Back & Side Tone Woods Management)



รูปที่ 3.9 Use case diagram UC_08: Back & Side Tone Woods Management

ตารางที่ 3.9 Use case description UC_08: Back & Side Tone Woods Management

Diagram ID	UC_08
Use case Name	Back & Side Tone Woods Management
Triggering Event	เมื่อผู้ดูแลระบบต้องการจัดการกับข้อมูลไม้หลังและไม้ข้าง(Back & Side Tone Woods)
Brief Description	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. เมื่อต้องการเพิ่ม 2. เมื่อต้องการแก้ไข



	3. เมื่อต้องการลบ 4. เมื่อต้องการค้นหา	
Actors	ผู้ดูแลระบบ	
Related Use Case	UC_04	
Stakeholders	-	
Pre-Condition	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. ต้องเพิ่มข้อมูลได้ 2. ต้องแก้ไขข้อมูลได้ 3. ต้องลบข้อมูลได้ 4. ต้องค้นหาข้อมูลได้ 	
Post-Condition	<ol style="list-style-type: none"> 1. สามารถเพิ่มข้อมูลได้สำเร็จ 2. สามารถแก้ไขข้อมูลได้สำเร็จ 3. สามารถลบข้อมูลได้สำเร็จ 4. สามารถค้นหาข้อมูลได้สำเร็จ 	
Flow of Event	Actor	System
(ลำดับขั้นตอนการ ทำงานระหว่าง User กับ System)	<p><u>ผู้ดูแลระบบ</u> <u>เลือกเมนู Back & Side Tone</u> <u>Woods Management</u></p> <p>1. กดที่ปุ่มเพิ่ม</p> <p>1.1 ทำการกรอกชื่อให้ครบถ้วนถูกต้อง</p> <p>1.1.1 กรณีซื้อที่ กรอกครบถ้วนถูกต้อง</p>	<p>ระบบจะแสดงหน้าการจัดการกับ ข้อมูล (Back & Side Tone Woods Management)</p> <p>1. ระบบจะทำการตรวจสอบความ ครบถ้วนและถูกต้อง</p> <p>2. ระบบจะทำการตรวจสอบว่าชื่อ ของข้อมูลที่ต้องการเพิ่มนี้ มีข้อมูล กันหรือไม่</p> <p>1. ระบบจะทำการบันทึกลง Database</p>



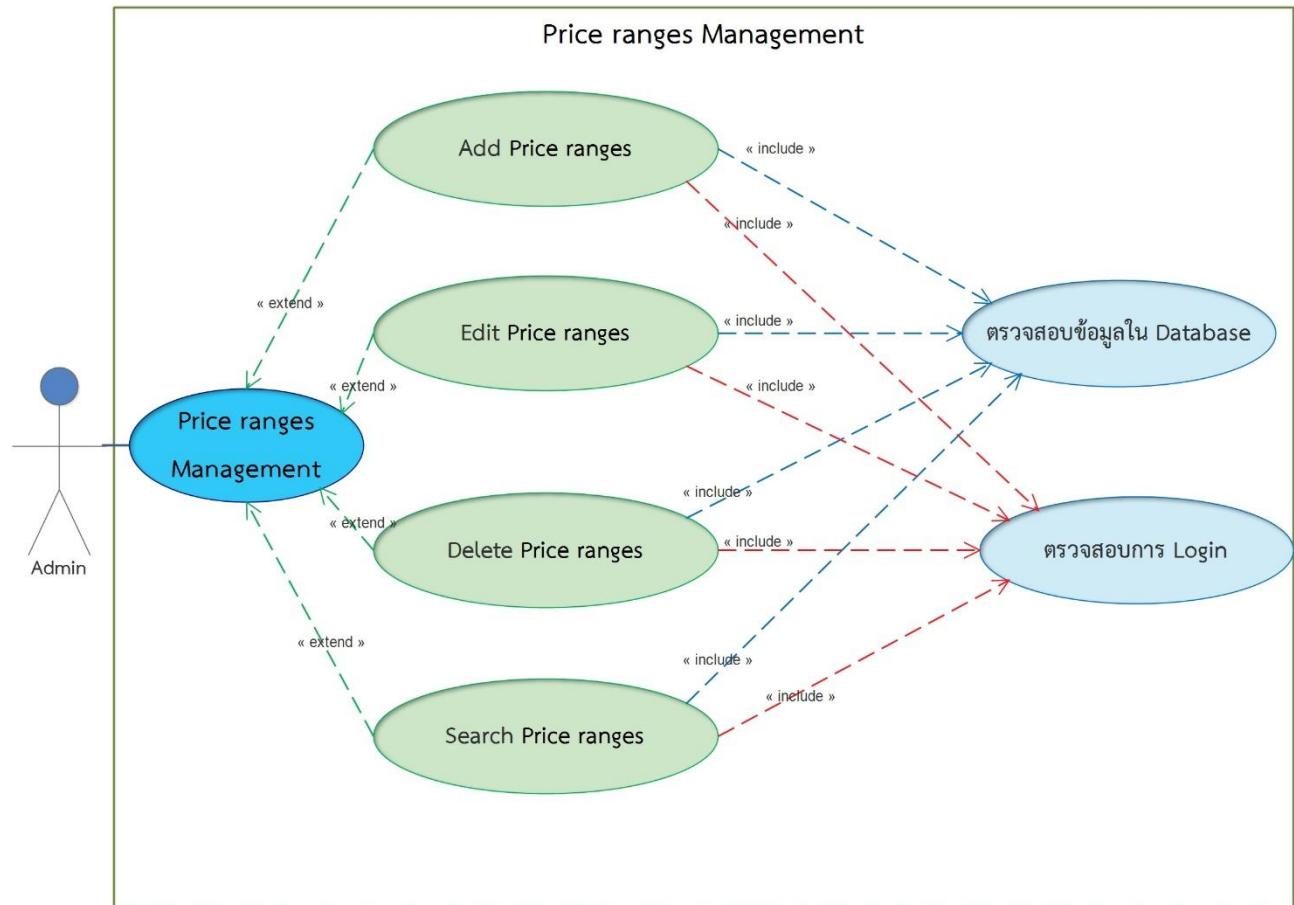
		2.ระบบจะแสดง “ทำการเพิ่มข้อมูลเรียบร้อยแล้ว”
	1.1.2 กรณีซื้อที่กรอกไม่ครบและไม่ถ้วนถูกต้อง	1.ระบบจะแสดง “ทำการเพิ่มไม่สำเร็จ กรุณากรอกชื่อให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่
	1.1.3 กรณีกดปุ่มยกเลิก	ระบบจะกลับไปหน้าการจัดการข้อมูล (Back & Side Tone Woods Management)
	2.กดปุ่มแก้ไข	
	2.1 ทำการแก้ไขซื้อและกรอกให้ครบถ้วนถูกต้อง	1.ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง 2.ระบบจะทำการตรวจสอบ ว่าซื้อที่ต้องการเพิ่มนี้ มีซื้อซ้ำกันหรือไม่
	2.1.1 กรณีซื้อที่กรอกครบถ้วนถูกต้อง	1.ระบบจะทำการบันทึกลง Database 2.ระบบจะแสดง “ทำการแก้ไขข้อมูลเรียบร้อยแล้ว”
	2.1.2 กรณีซื้อที่กรอกไม่ครบและไม่ถ้วนถูกต้อง	1.ระบบจะแสดง “ทำการแก้ไขไม่สำเร็จ กรุณากรอกชื่อให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่
		ระบบจะกลับไปหน้าการจัดการข้อมูล (Back & Side Tone Woods Management)



	<p>2.1.3 กรณีกดปุ่มยกเลิก</p> <p>3. กดปุ่มลบ</p> <p>3.1 กรณีกดปุ่ม OK</p> <p>3.2 กรณีกดปุ่มยกเลิก</p> <p>4. กดปุ่มค้นหาข้อมูล</p> <p>4.1 พิมพ์คำค้น</p> <p>4.1.1 กรณีพบรูป</p> <p>4.1.2 กรณีไม่พบ</p>	<p>1.ระบบจะทำการลบข้อมูลที่อยู่ใน Database</p> <p>2.ระบบจะแสดง “ทำการลบข้อมูลเรียบร้อยแล้ว”</p> <p>3.ระบบจะกลับไปหน้าการจัดการข้อมูล (Back & Side Tone Woods Management)</p> <p>ระบบจะกลับไปหน้าการจัดการข้อมูล (Back & Side Tone Woods Management)</p> <p>1.ทำการตรวจสอบคำค้นนั้นใน Database</p> <p>ระบบจะทำการแสดงผลการค้นข้อมูลอุปกรณ์ทางหน้าจอของระบบ</p> <p>ระบบจะทำการแสดง “ไม่พบคำค้น” และกลับไปหน้าหลัก</p>
Exceptions	หากกรอกข้อมูล ผิดพลาดหรือไม่ครบถ้วนและถูกต้อง ระบบจะแจ้งเตือน พร้อมแสดงข้อความให้ทราบ และกลับไปหน้าแบบฟอร์มเพื่อกรอกข้อมูลใหม่	



3.2.9 จัดการข้อมูลช่วงของราคา (Price ranges Management)



รูปที่ 3.10 Use case diagram UC_9: Price ranges Management

ตารางที่ 3.10 Use case description UC_9: Price ranges Management

Diagram ID	UC_9
Use case Name	Price ranges Management
Triggering Event	เมื่อผู้ดูแลระบบต้องการจัดการกับข้อมูลเครื่องกำเนิดเสียง(Price ranges Management)
Brief Description	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> เมื่อต้องการเพิ่ม เมื่อต้องการแก้ไข เมื่อต้องการลบ เมื่อต้องการค้นหา



Actors	ผู้ดูแลระบบ	
Related Use Case	UC_04	
Stakeholders	-	
Pre-Condition	<p>ผู้ดูแลระบบ</p> <ol style="list-style-type: none"> 1. ต้องเพิ่มข้อมูลได้ 2. ต้องแก้ไขข้อมูลได้ 3. ต้องลบข้อมูลได้ 4. ต้องค้นหาข้อมูลได้ 	
Post-Condition	<ol style="list-style-type: none"> 1. สามารถเพิ่มข้อมูลได้สำเร็จ 2. สามารถแก้ไขข้อมูลได้สำเร็จ 3. สามารถลบข้อมูลได้สำเร็จ 4. สามารถค้นหาข้อมูลได้สำเร็จ 	
Flow of Event	Actor	System
(ลำดับขั้นตอนการ ทำงานระหว่าง User กับ System)	<p>ผู้ดูแลระบบ</p> <p><u>เลือกเมนู Price ranges Management</u></p> <p>1. กดที่ปุ่มเพิ่ม</p> <p>1.1 ทำการกรอกช่วงราคาให้ครบถ้วนถูกต้อง</p> <p>1.1.1 กรณีช่วงราคาที่กรอกครบถ้วนถูกต้อง</p>	<p>ระบบจะแสดงหน้าการจัดการ กับข้อมูล (Price ranges Management)</p> <p>1. ระบบจะทำการตรวจสอบความ ครบถ้วนและถูกต้อง</p> <p>2. ระบบจะทำการตรวจสอบว่าช่วง ราคากำลังมีข้อมูลที่ต้องการเพิ่มนี้ มี ข้อมูลกันหรือไม่</p> <p>1. ระบบจะทำการบันทึกลง Database</p> <p>2. ระบบจะแสดง “ทำการเพิ่ม ข้อมูลเรียบร้อยแล้ว”</p>



	<p>1.1.2 กรณีซึ่งราคาที่กรอกไม่ครบและไม่ถ้วนถูกต้อง</p>	<p>1.ระบบจะแสดง “ทำการเพิ่มไม่สำเร็จ กรุณากรอกช่วงราคาให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่</p>
	<p>1.1.3 กรณีกดปุ่มยกเลิก</p>	<p>ระบบจะกลับไปหน้าการจัดการข้อมูล (Price ranges Management)</p>
	<p>2.กดปุ่มแก้ไข</p>	<p>1.ระบบจะทำการตรวจสอบความครบถ้วนและถูกต้อง</p>
	<p>2.1 ทำการแก้ไขช่วงราคาและกรอกให้ครบถ้วนถูกต้อง</p>	<p>2.ระบบจะทำการตรวจสอบ ว่าช่วงราคาที่ต้องการเพิ่มนี้ มีซึ่งซ้ำกันหรือไม่</p>
	<p>2.1.1 กรณีซึ่งราคาที่กรอกครบถ้วนถูกต้อง</p>	<p>1.ระบบจะทำการบันทึกลง Database 2.ระบบจะแสดง “ทำการแก้ไขข้อมูลเรียบร้อยแล้ว”</p>
	<p>2.1.2 กรณีซึ่งราคาที่กรอกไม่ครบและไม่ถ้วนถูกต้อง</p>	<p>1.ระบบจะแสดง “ทำการแก้ไขไม่สำเร็จ กรุณากรอกช่วงราคาให้ถูกต้อง” 2.ระบบจะกลับไปหน้าการกรอกข้อมูล เพื่อกรอกข้อมูลใหม่</p>
	<p>2.1.3 กรณีกดปุ่มยกเลิก</p>	<p>ระบบจะกลับไปหน้าการจัดการข้อมูล (Price ranges Management)</p>

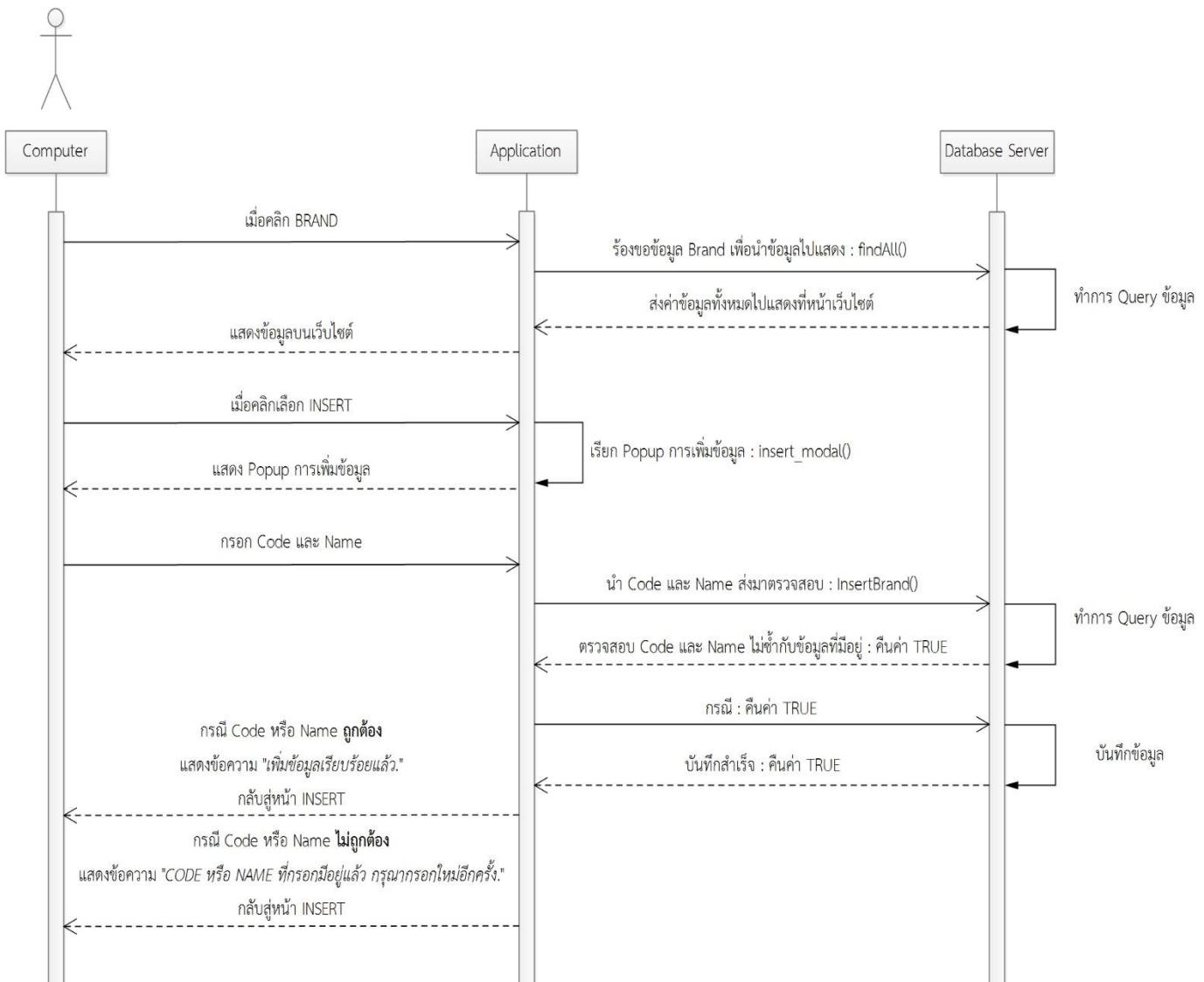


	<p>3. กดปุ่มลบ</p> <p>3.1 กรณีกดปุ่ม OK</p> <p>3.2 กรณีกดปุ่มยกเลิก</p>	<p>1. ระบบจะทำการลบข้อมูลที่อยู่ใน Database</p> <p>2. ระบบจะแสดง “ทำการลบข้อมูล เรียบร้อยแล้ว”</p> <p>3. ระบบจะกลับไปหน้าการจัดการข้อมูล (Price ranges Management)</p> <p>ระบบจะกลับไปหน้าการจัดการข้อมูล (Price ranges Management)</p>
	<p>4. กดปุ่มค้นหาข้อมูล</p> <p>4.1 พิมพ์คำค้น</p> <p>4.1.1 กรณีพบ</p> <p>4.1.2 กรณีไม่พบ</p>	<p>1. ทำการตรวจสอบคำค้นนั้นใน Database</p> <p>ระบบจะทำการแสดงผลการค้นข้อมูลของทางหน้าจอของระบบ</p> <p>ระบบจะทำการแสดง “ไม่พบคำค้น” และกลับไปหน้าหลัก</p>
Exceptions	หากกรอกข้อมูล ผิดพลาดหรือไม่ครบถ้วนและถูกต้อง ระบบจะแจ้งเตือน พร้อมแสดงข้อความให้ทราบ และกลับไปหน้าแบบฟอร์มเพื่อกรอกข้อมูลใหม่	



3.3 Sequence Diagram ขั้นตอนการทำงานของระบบมีดังนี้

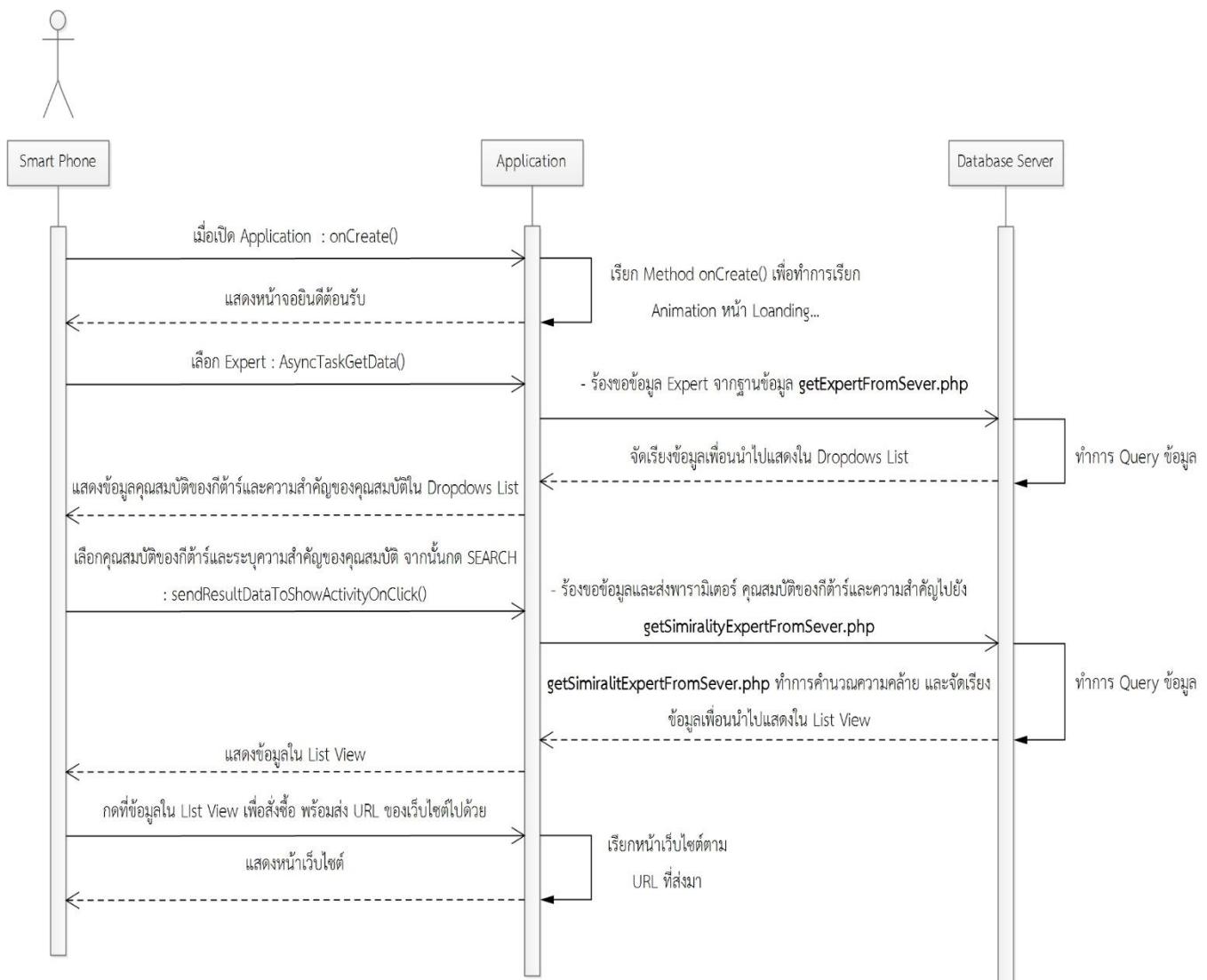
3.3.1 Sequence Diagram : Beginner Searching



รูปที่ 3.11 Sequence Diagram : Beginner Searching



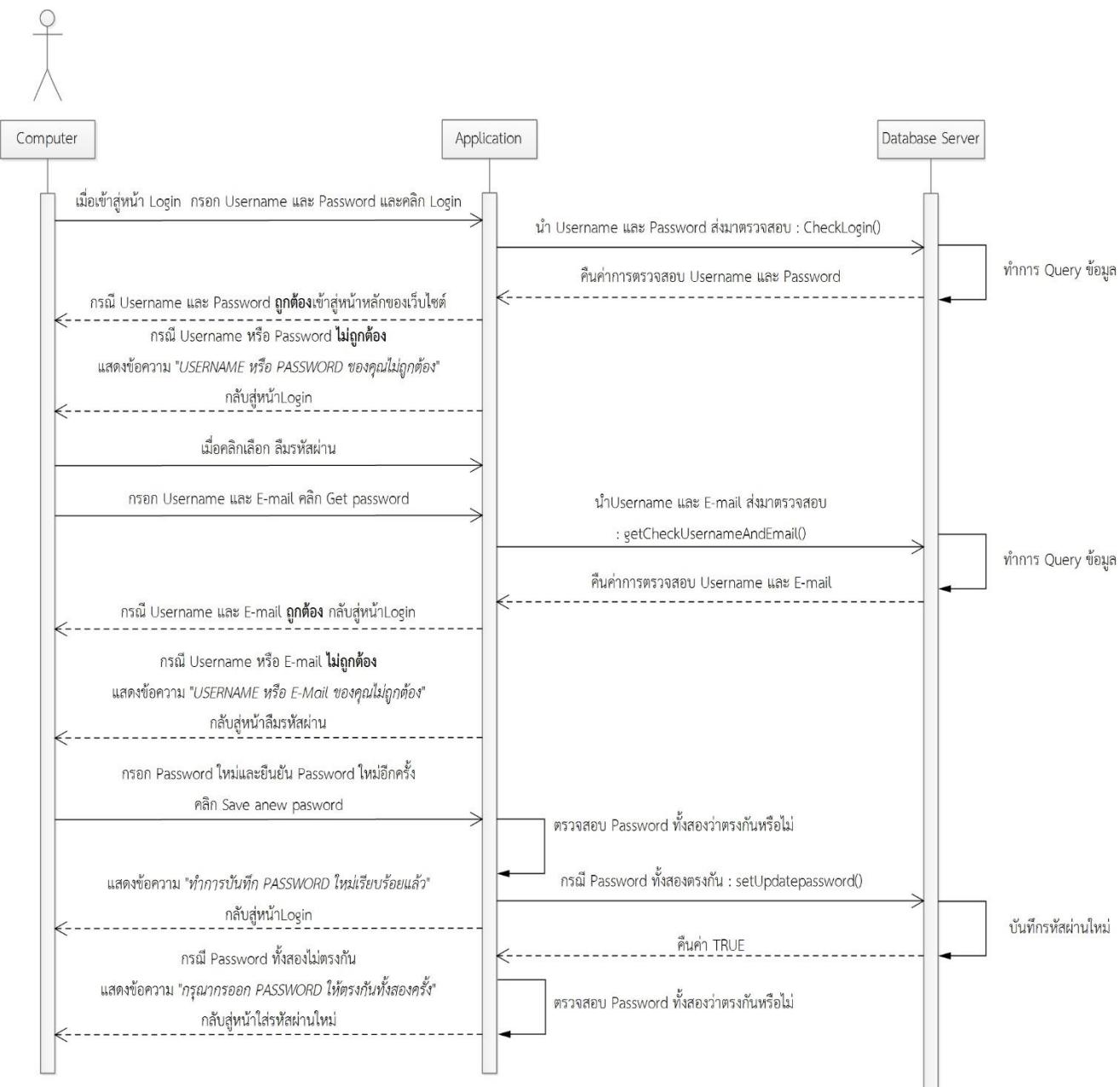
3.3.2 Sequence Diagram : Export Searching



รูปที่ 3.12 Sequence Diagram : Export Searching



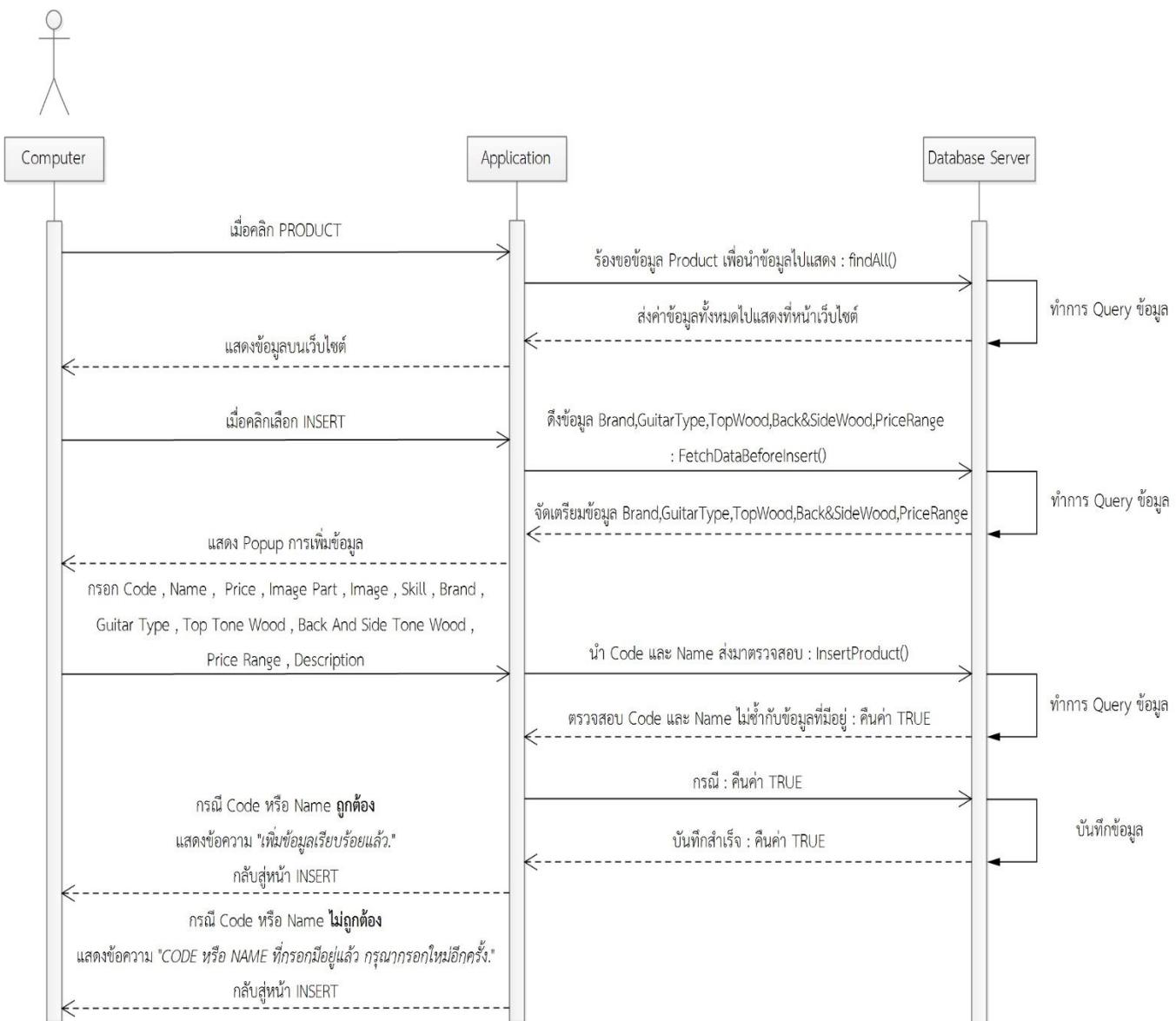
3.3.3 Sequence Diagram : Login

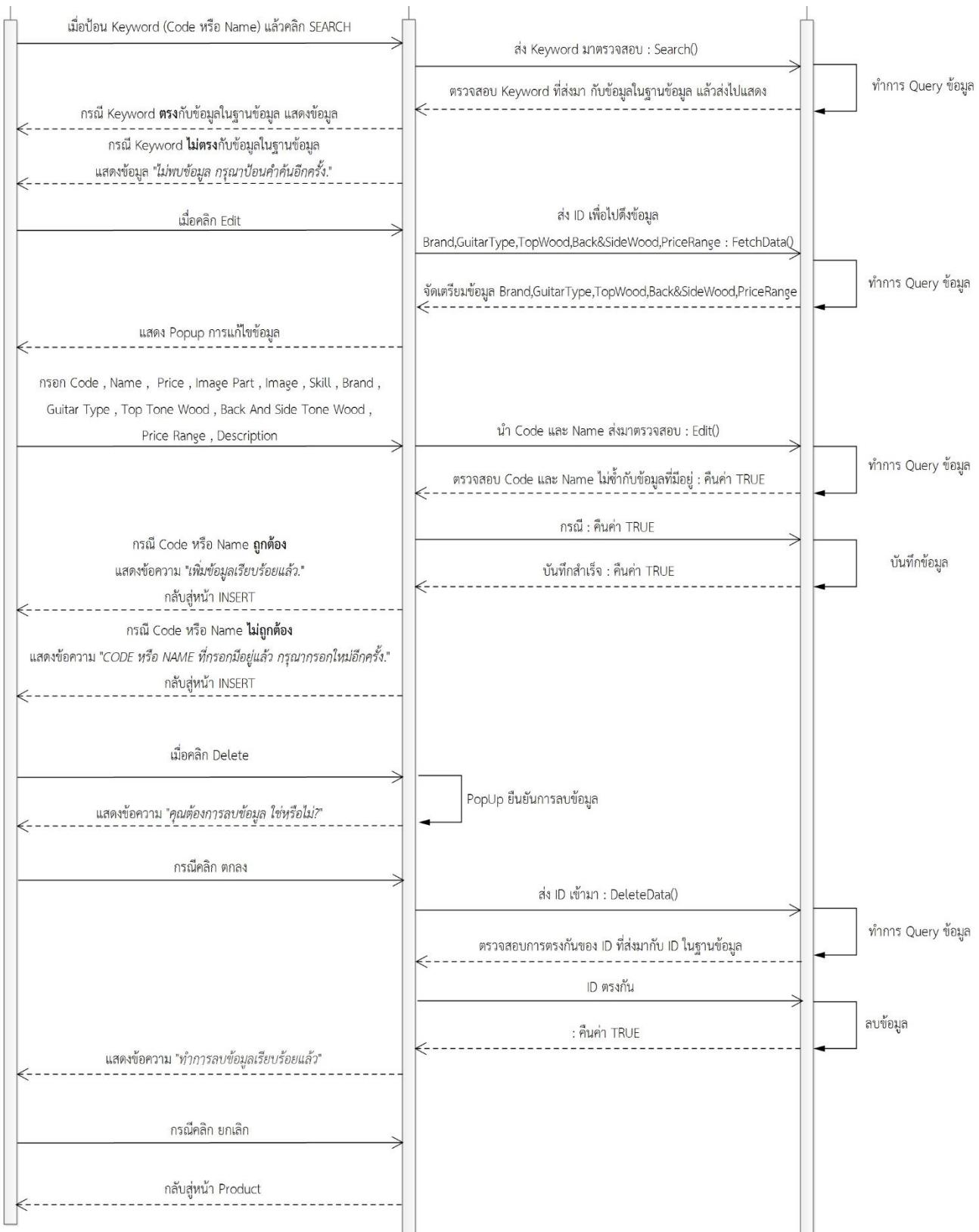


รูปที่ 3.13 Sequence Diagram : Login



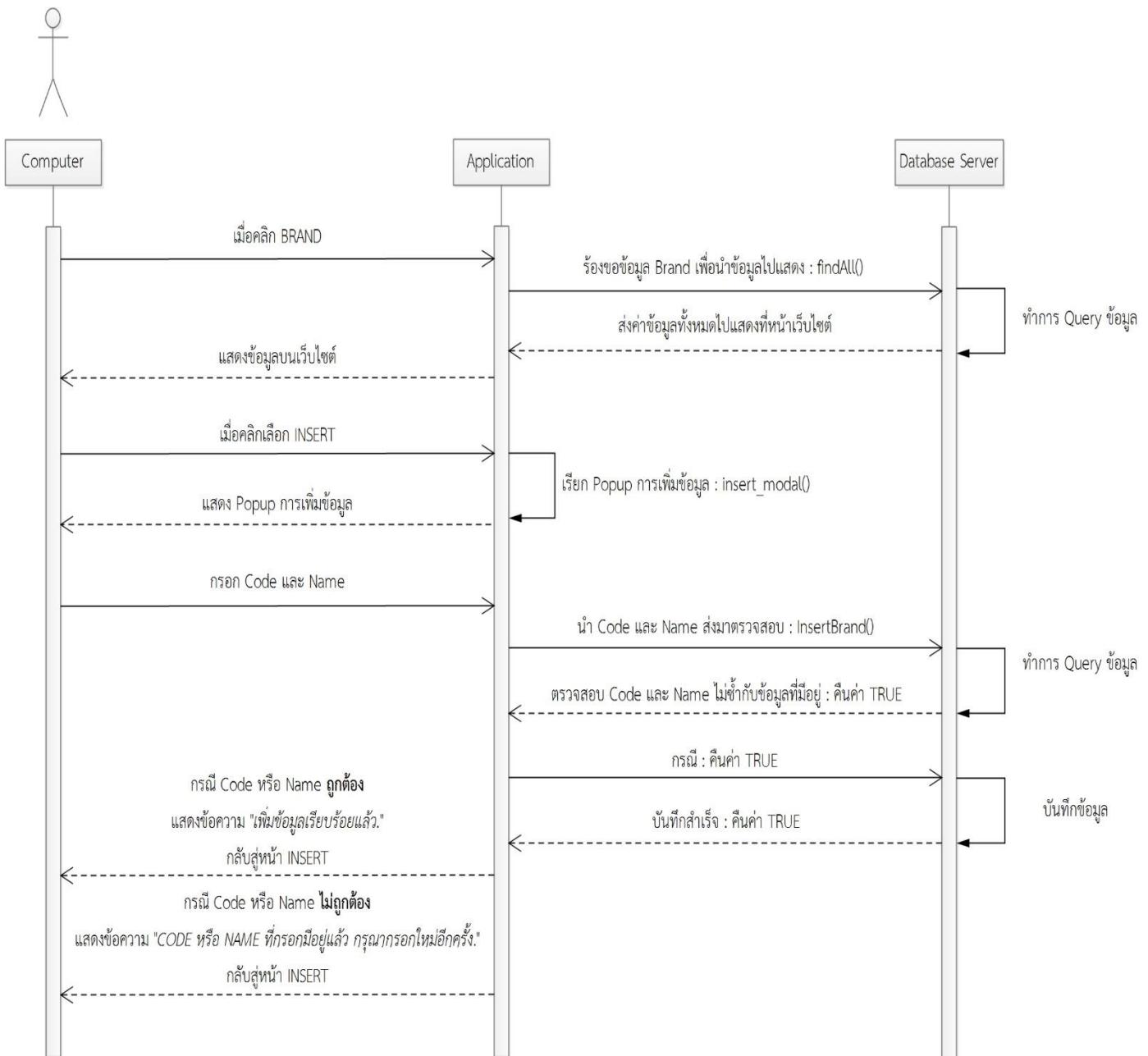
3.3.4 Sequence Diagram : Products Management



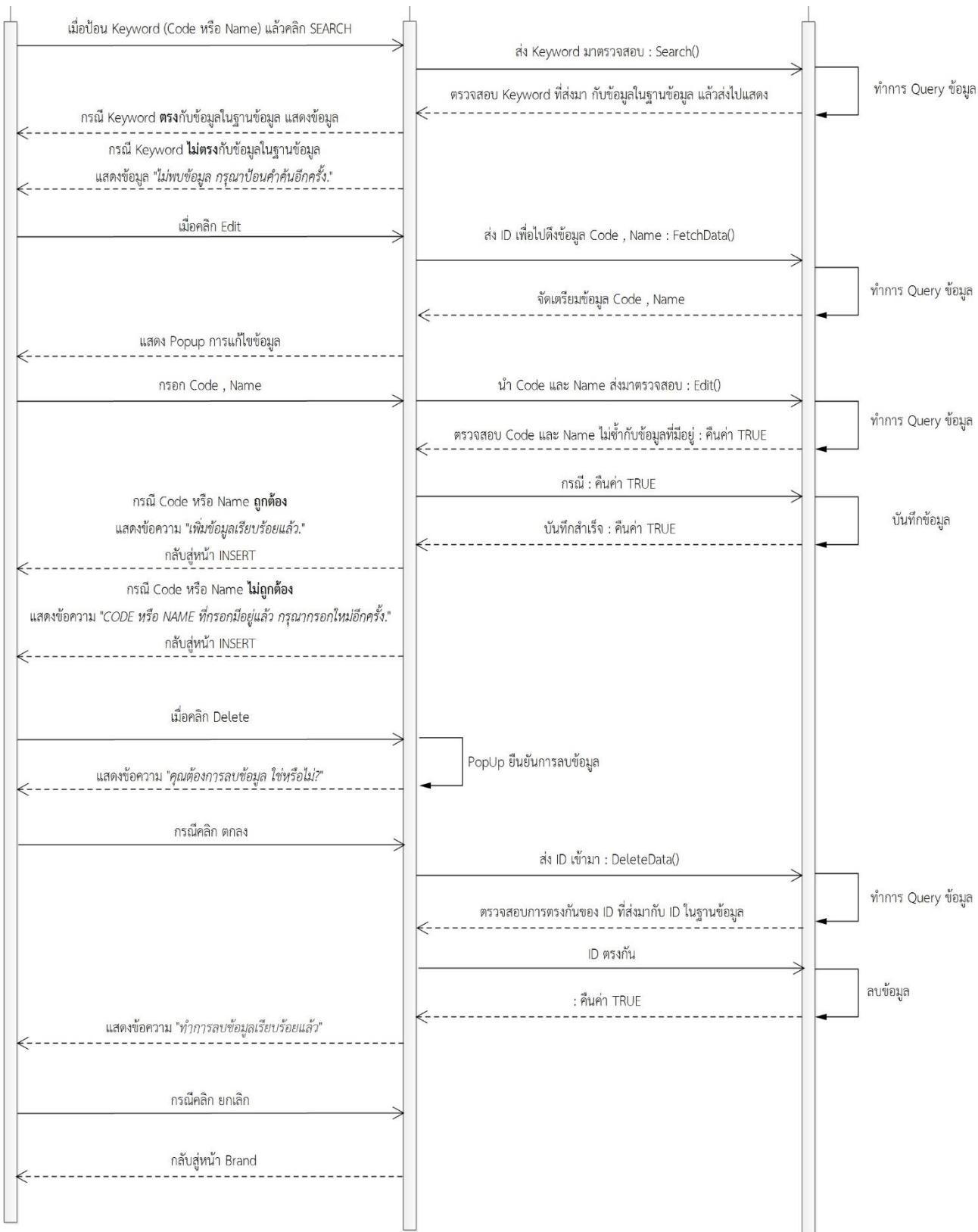




3.3.5 Sequence Diagram : Brands Management



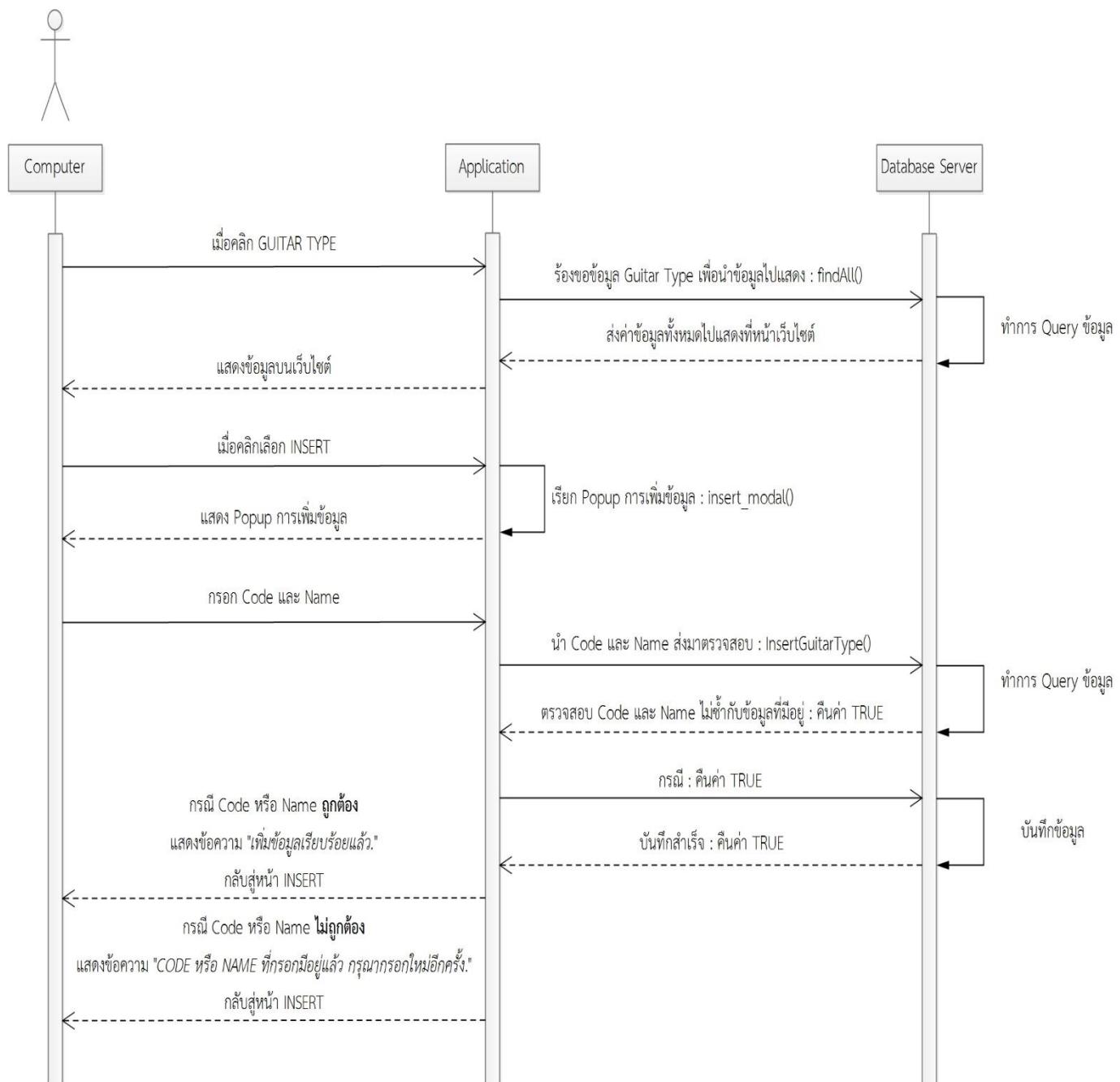
ต่อหน้าถัดไป



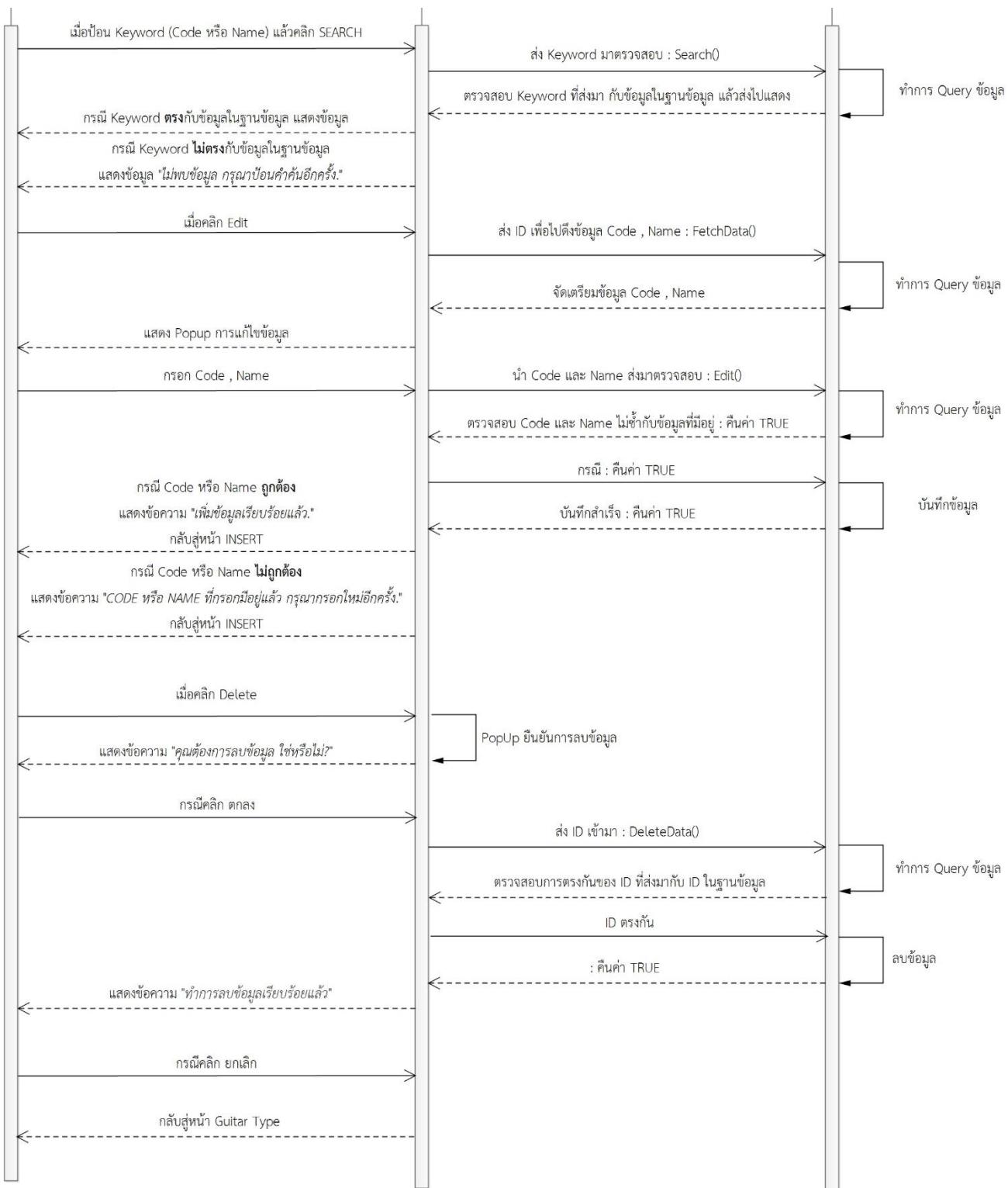
รูปที่ 3.15 Sequence Diagram : Brands Management



3.3.6 Sequence Diagram : Guitar Types Management



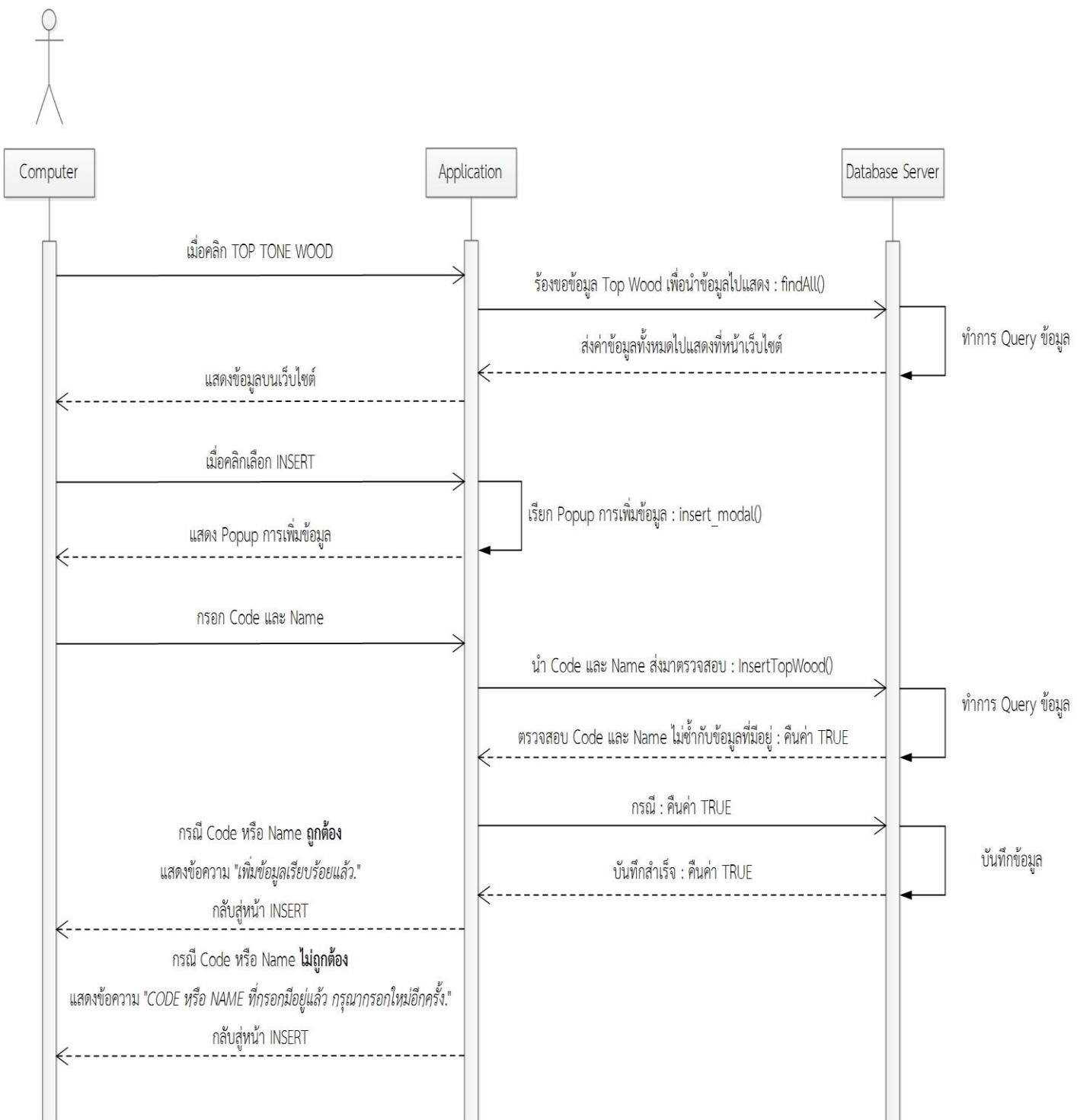
ต้องหน้าถัดไป

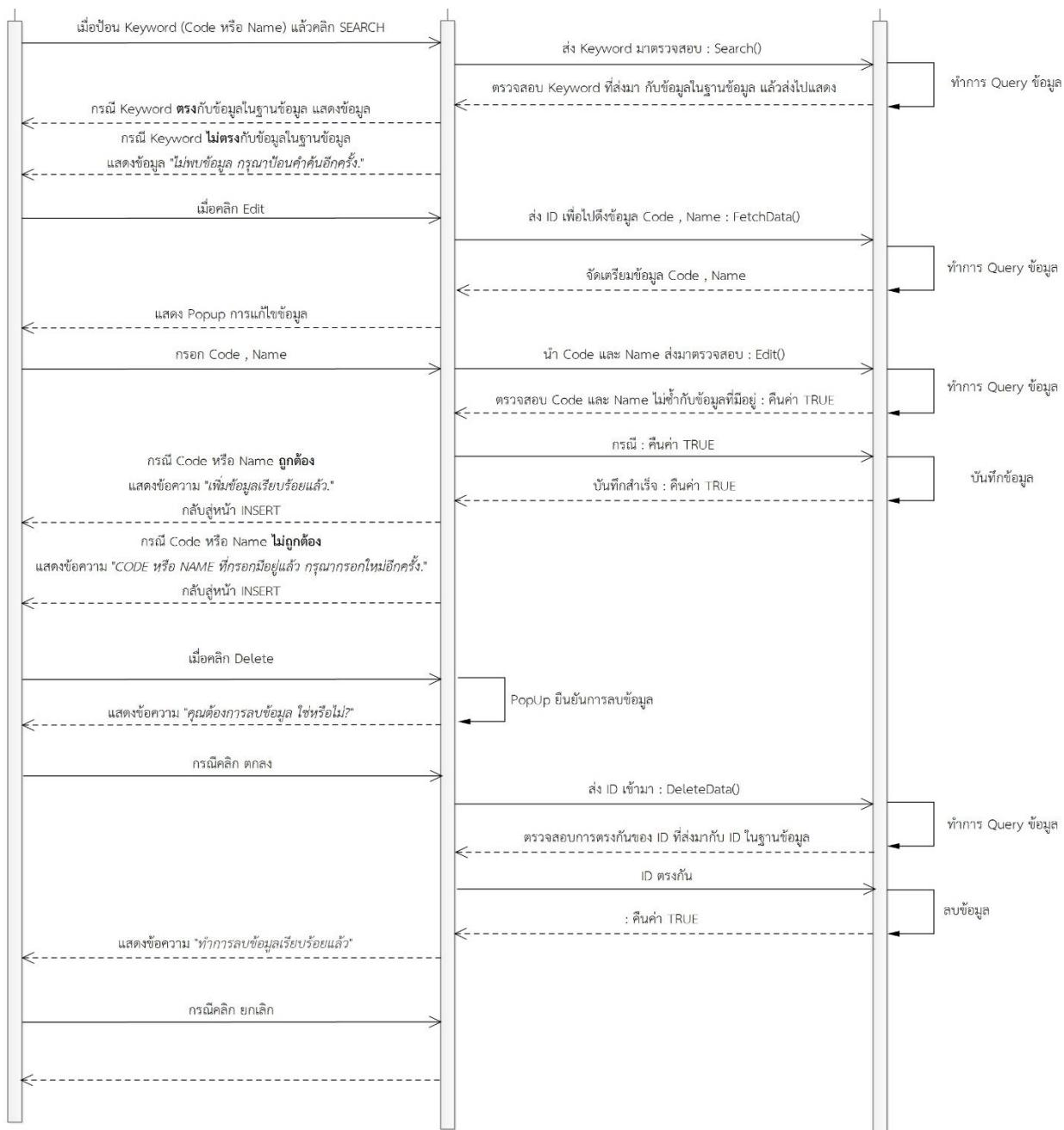


รูปที่ 3.16 Sequence Diagram : Guitar Types Management



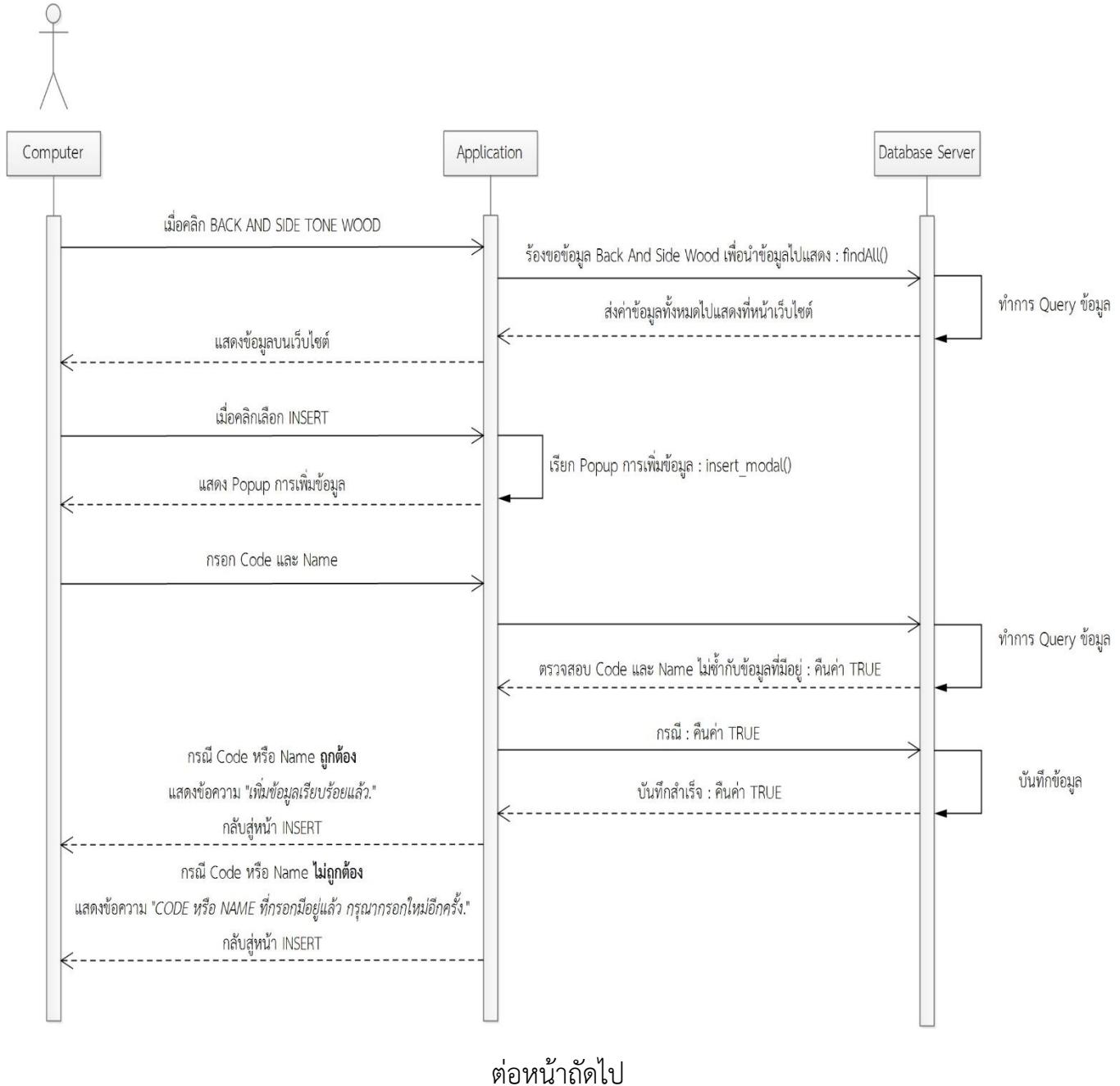
3.3.7 Sequence Diagram : Top tone woods Management

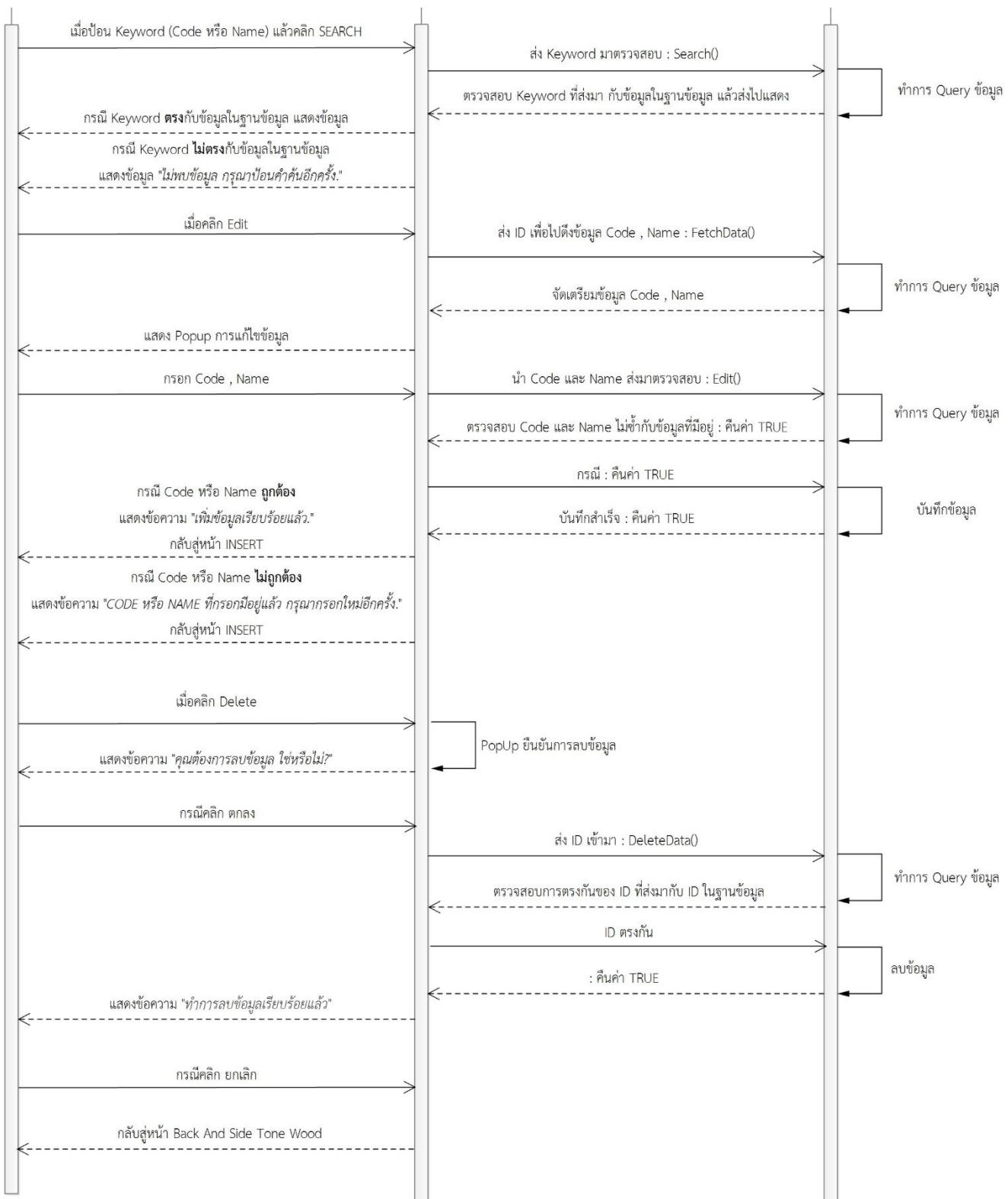




รูปที่ 3.17 Sequence Diagram : Top tone woods Management

3.3.8 Sequence Diagram : Back and Side tone woods Management

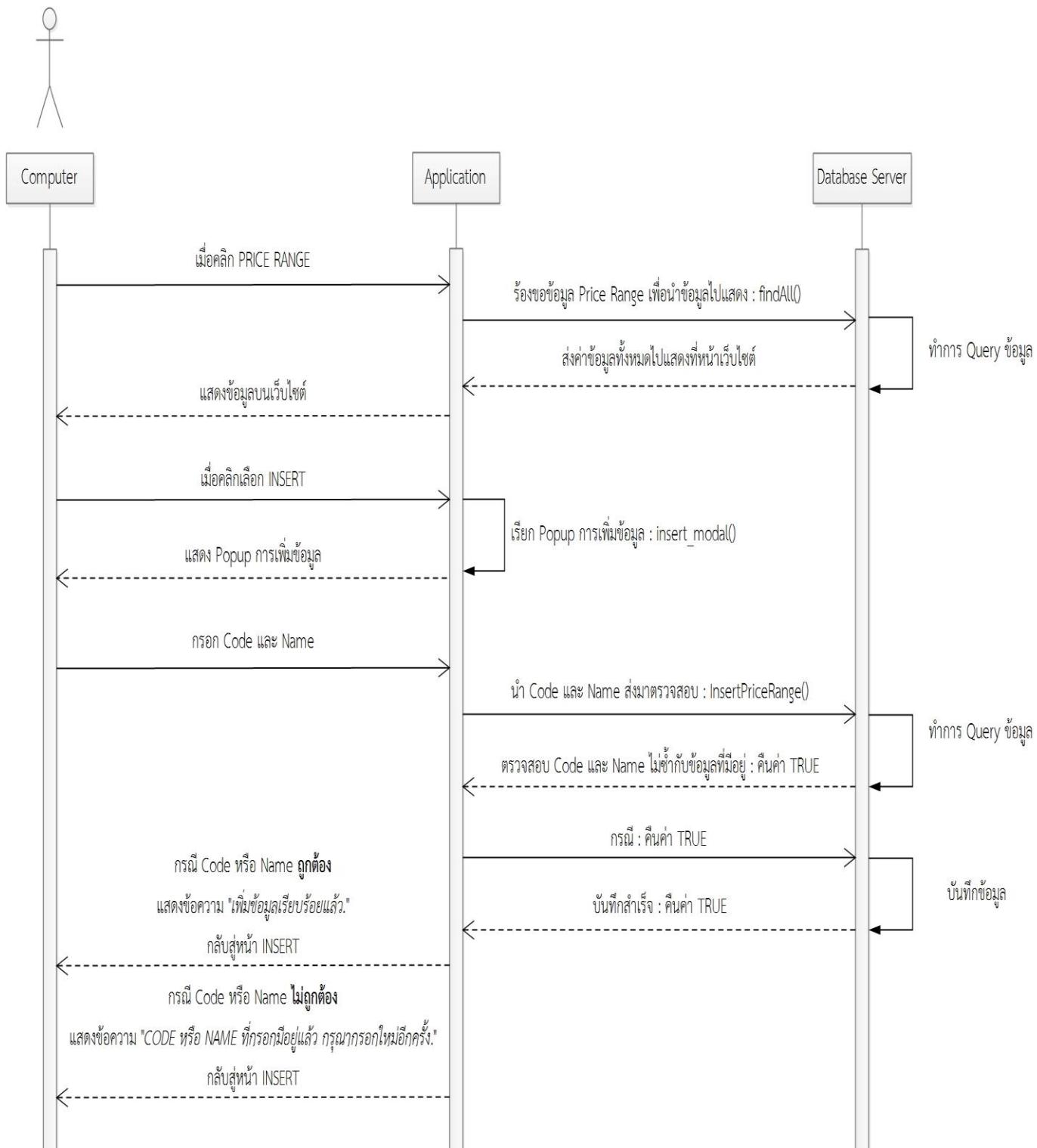




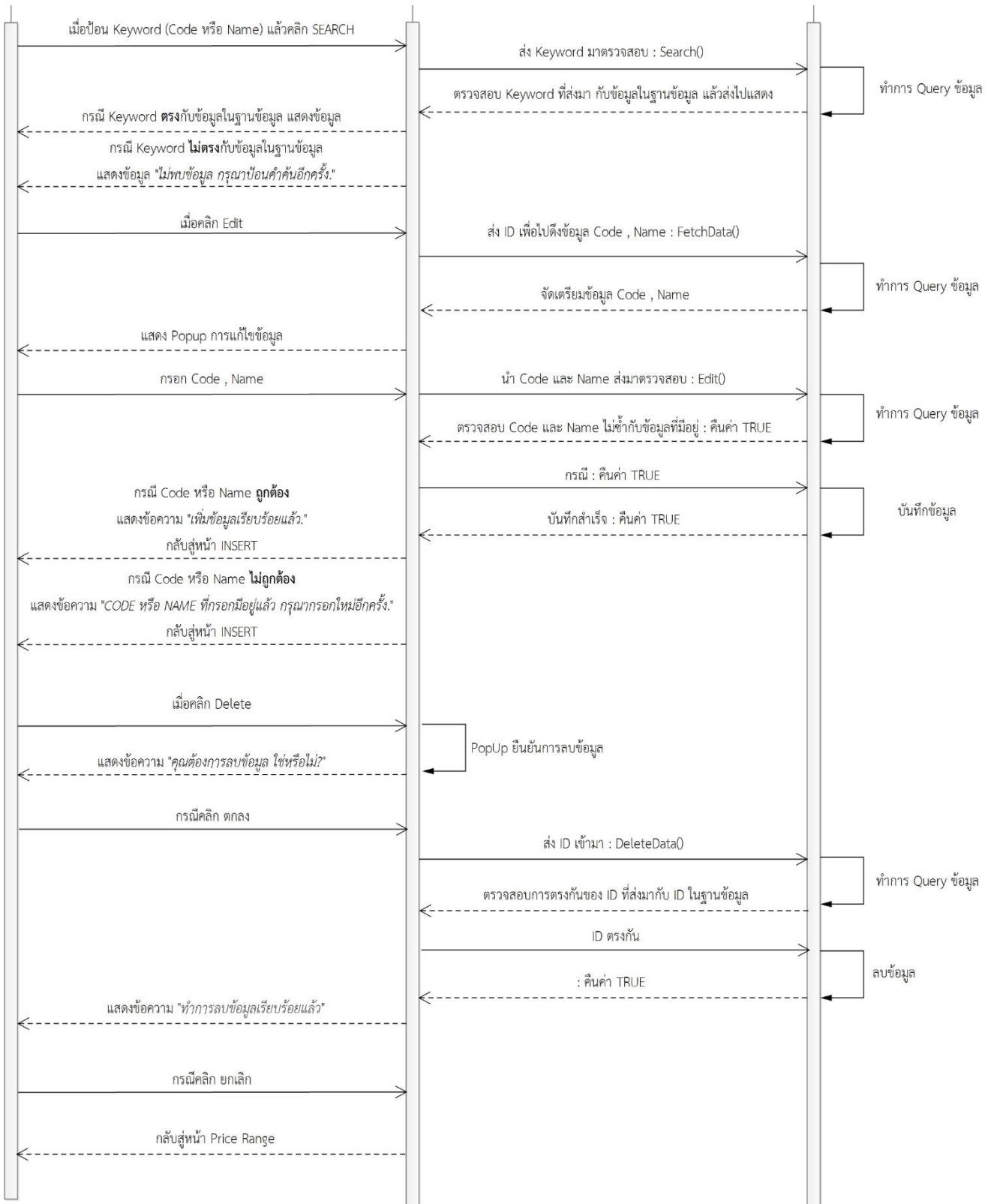
รูปที่ 3.18 Sequence Diagram : Back and Side tone woods Management



3.3.9 Sequence Diagram : Price ranges Management



ต่อหน้าถัดไป

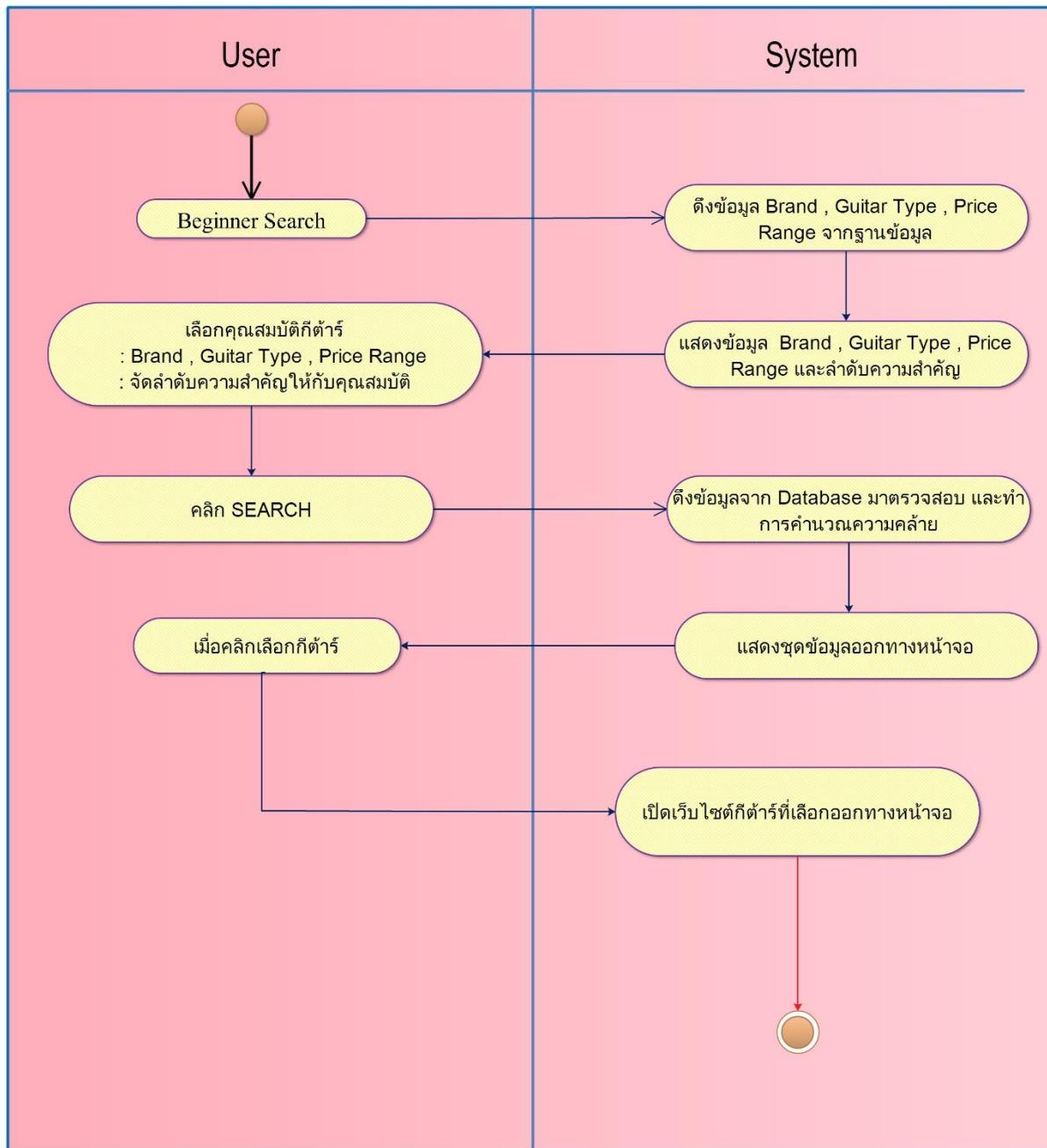


รูปที่ 3.19 Sequence Diagram : Price ranges Management



3.4 Activity Diagram กิจกรรมการทำงานของระบบมีดังต่อไปนี้

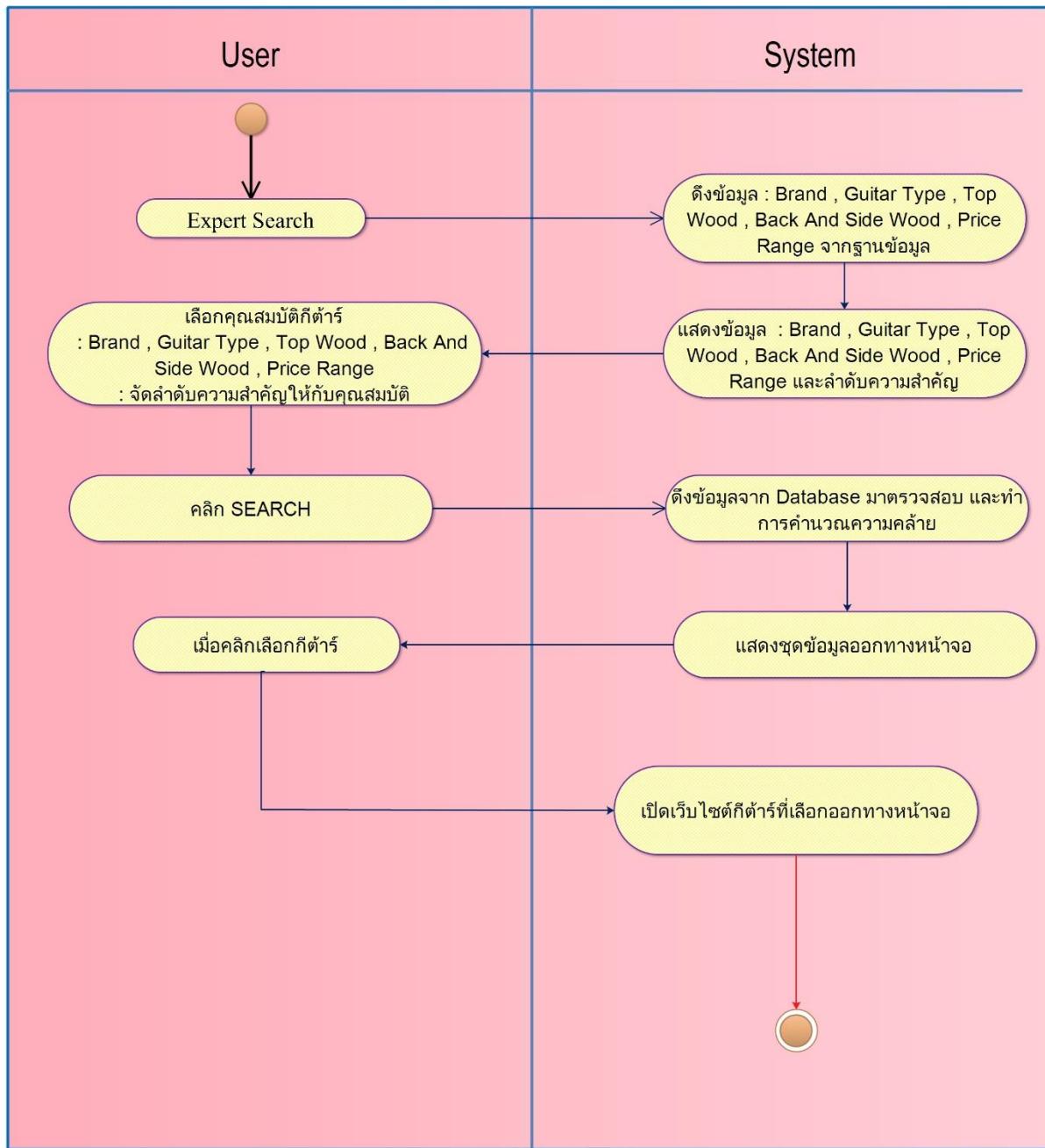
3.4.1 Activity Diagram : Beginner Searching



รูปที่ 3.20 Activity Diagram : Beginner Searching



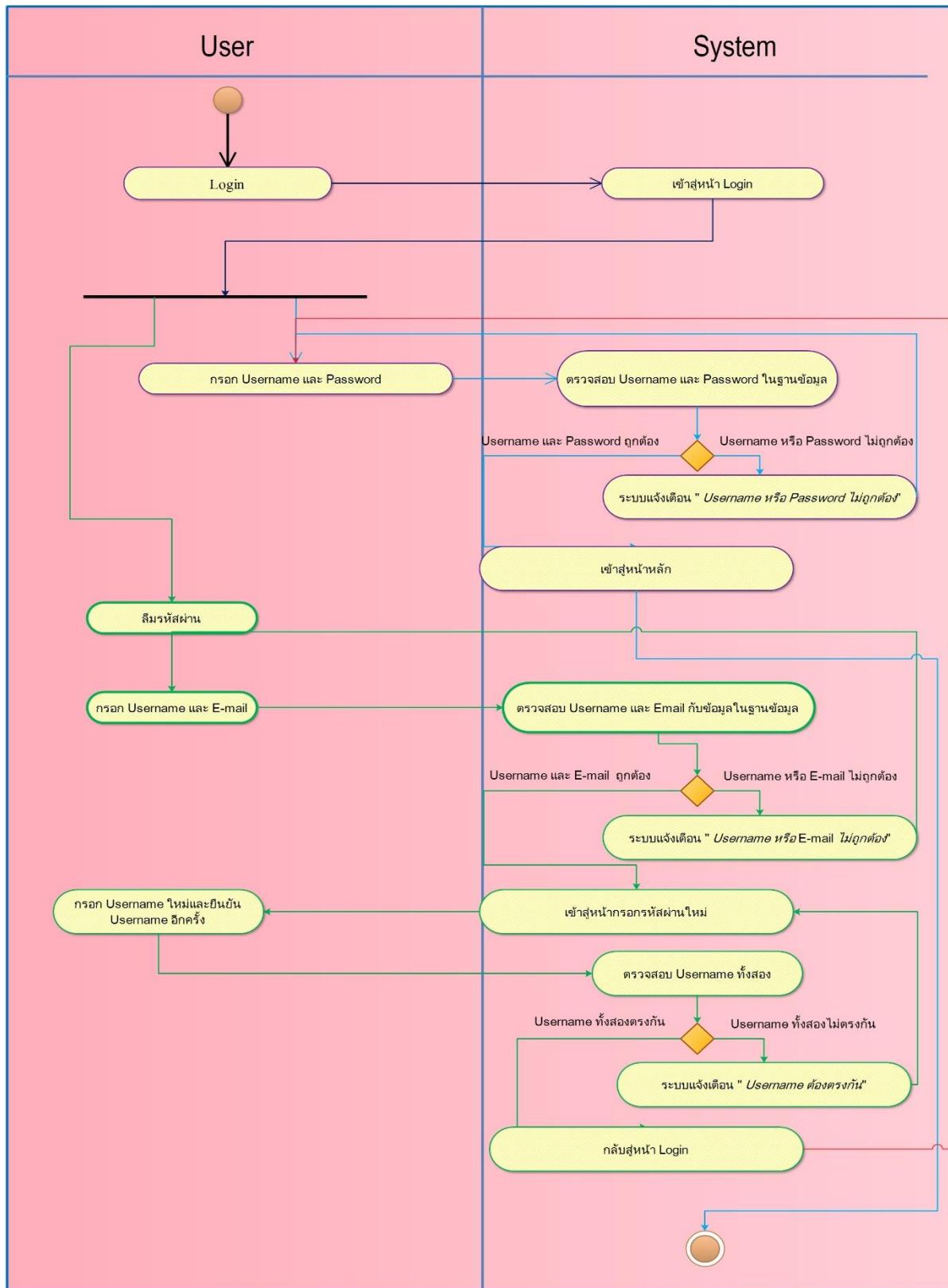
3.4.2 Activity Diagram : Expert Searching



รูปที่ 3.21 Activity Diagram : Expert Searching



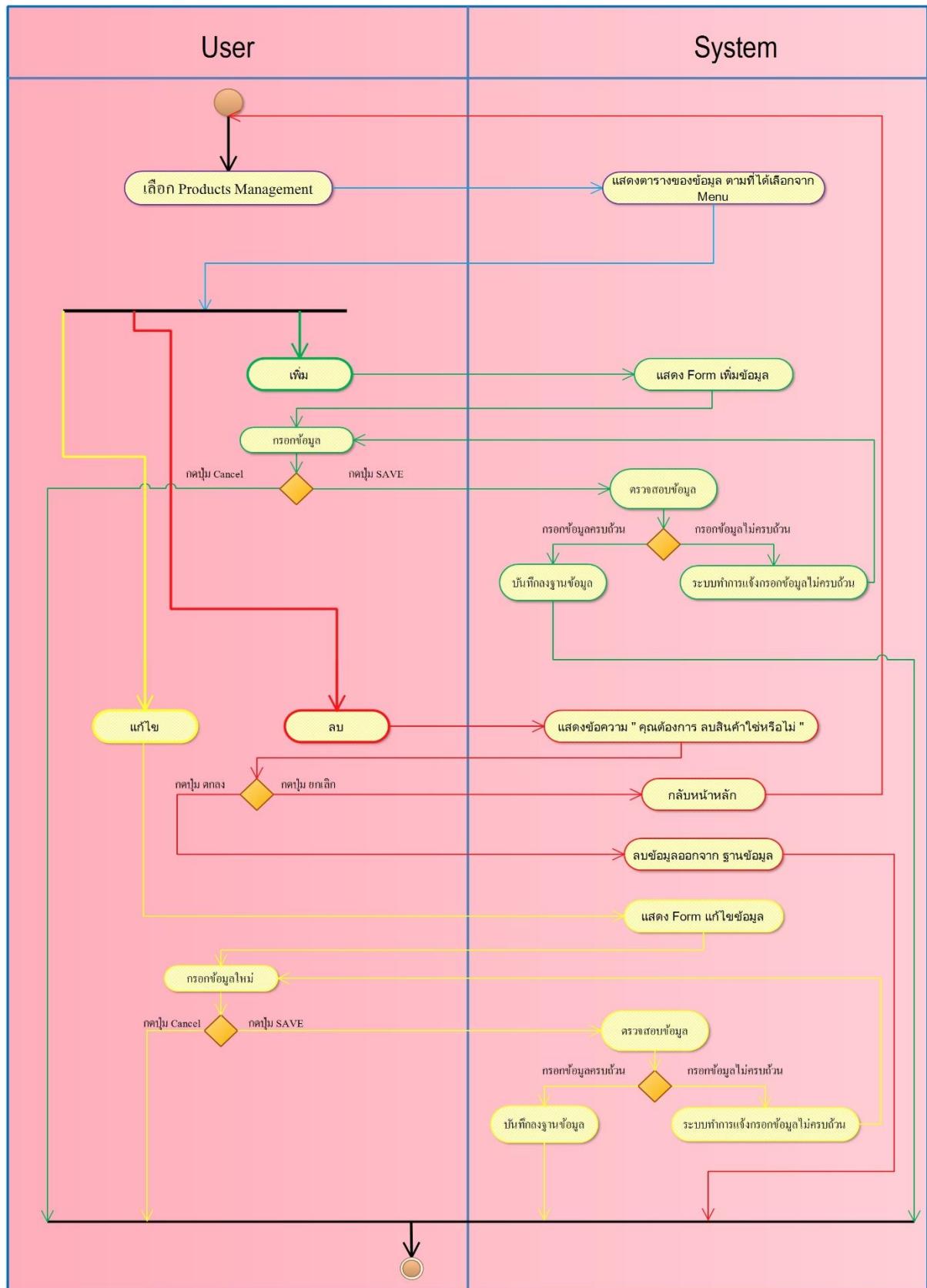
3.4.3 Activity Diagram : Login



รูปที่ 3.22 Activity Diagram : Login



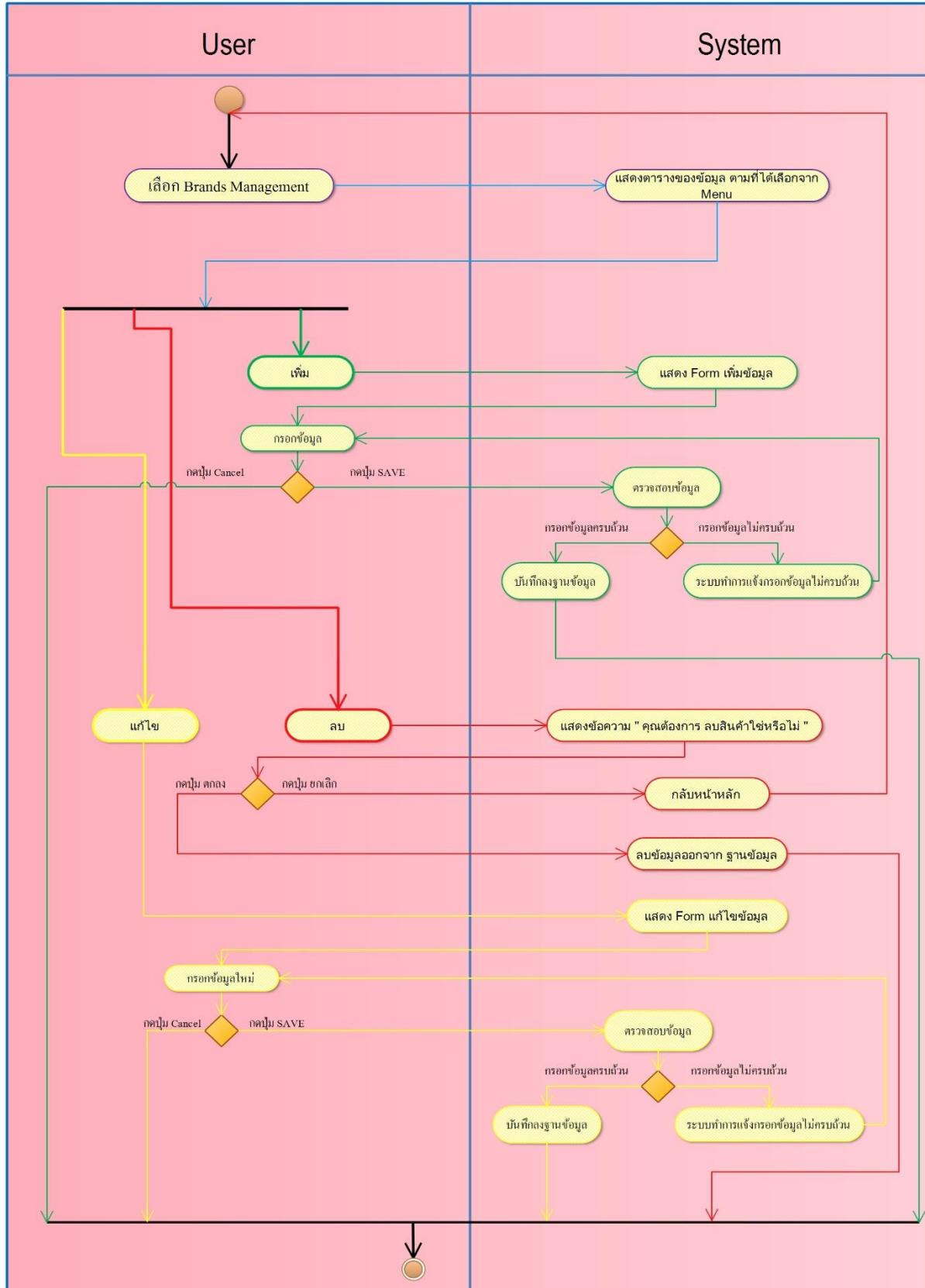
3.4.4 Activity Diagram : Products Management



รูปที่ 3.23 Activity Diagram : Products Management



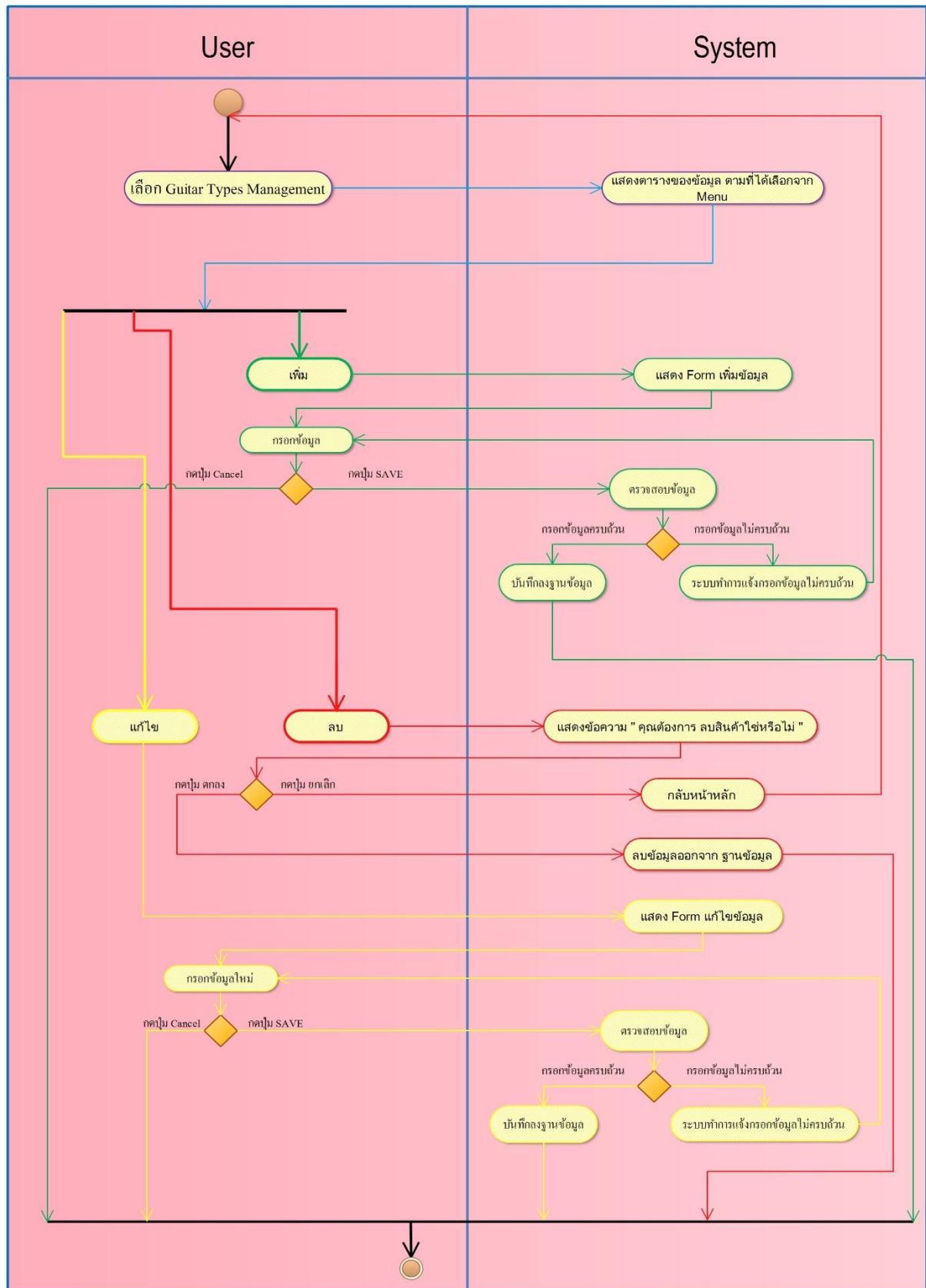
3.4.5 Activity Diagram : Brands Management



รูปที่ 3.24 Activity Diagram : Brands Management



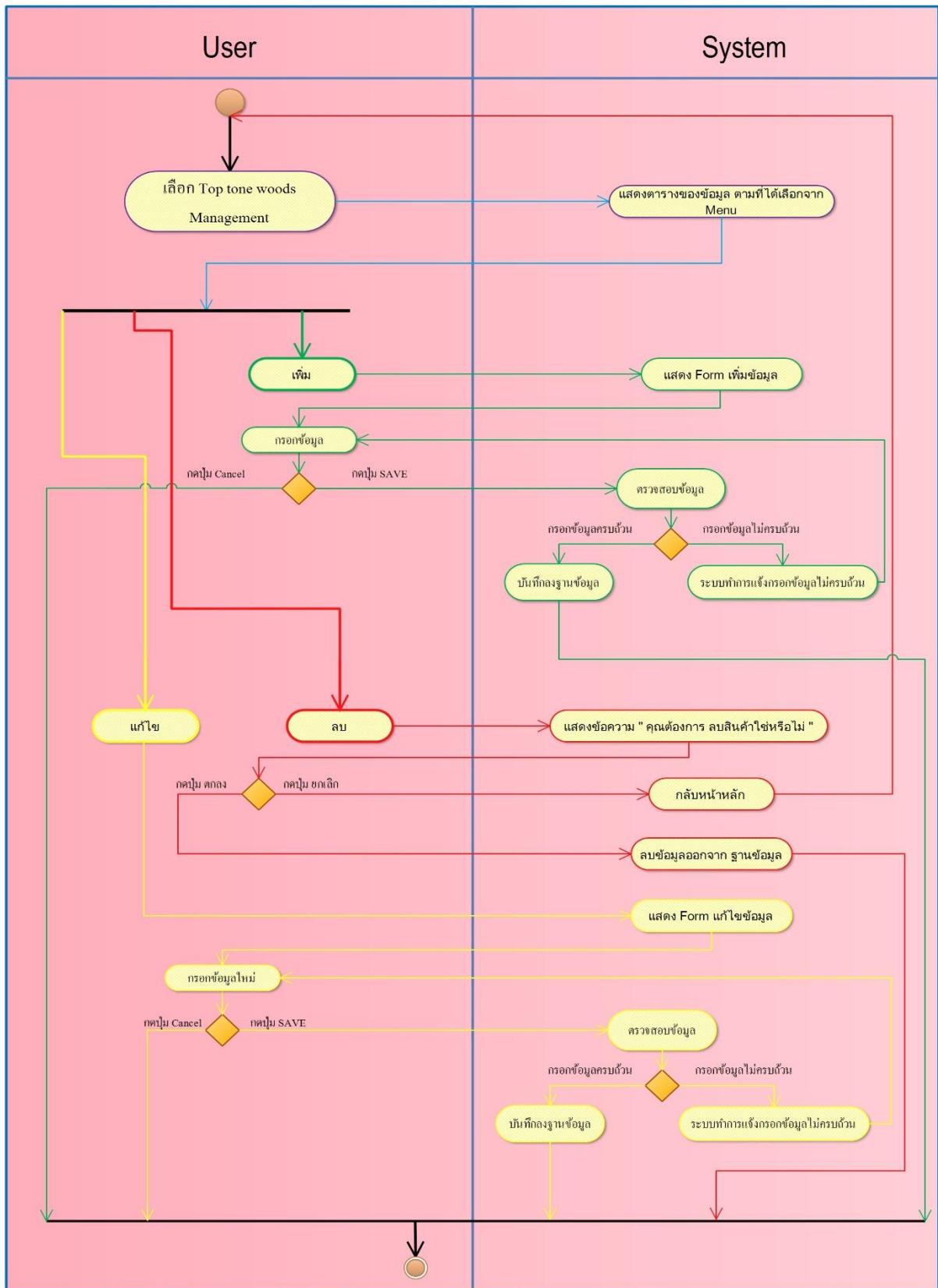
3.4.6 Activity Diagram : Guitar Types Management



รูปที่ 3.25 Activity Diagram : Guitar Types Management

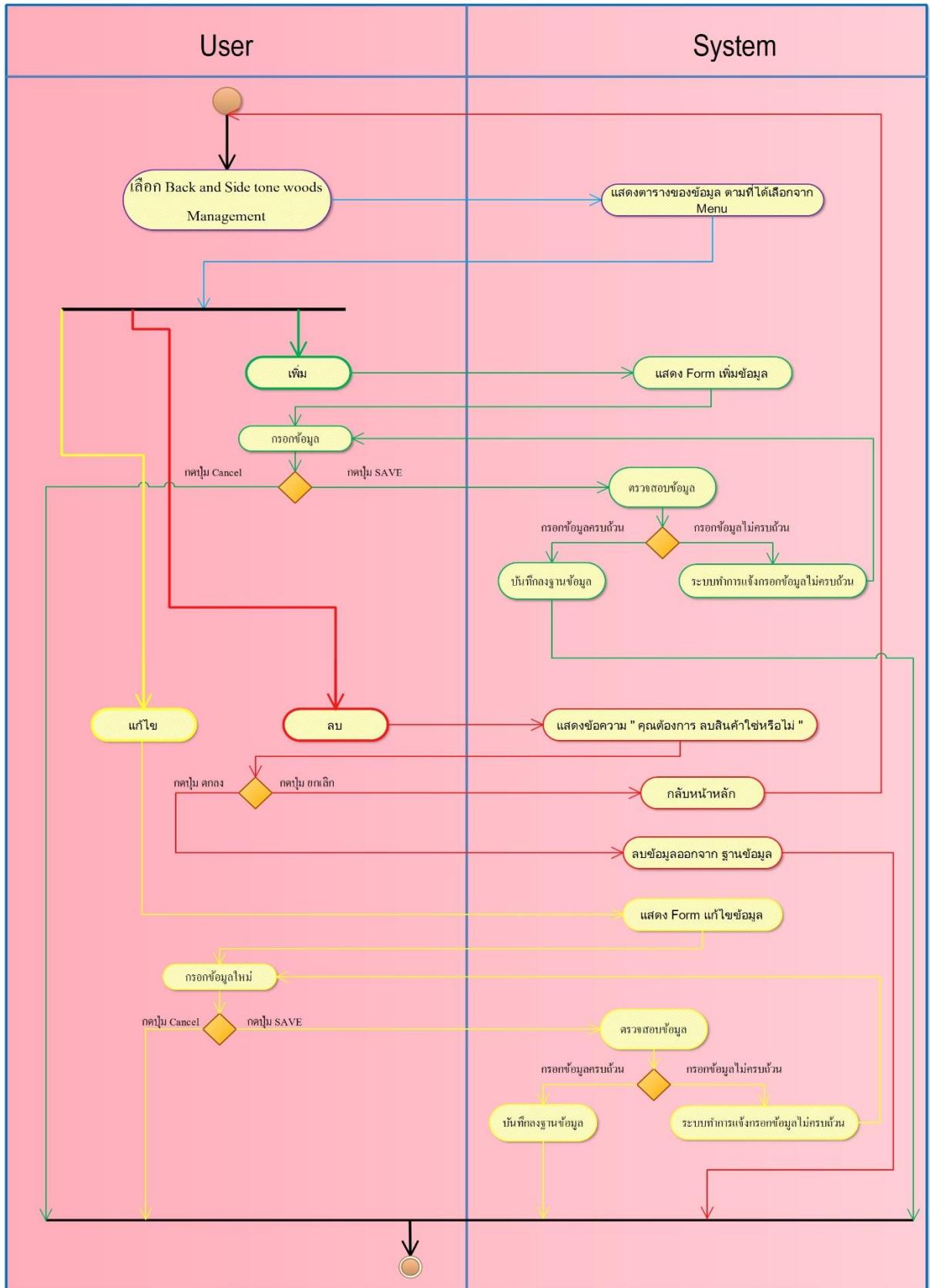


3.4.7 Activity Diagram : Top tone woods Management



ຮູບທີ 3.26 Activity Diagram : Top tone woods Management

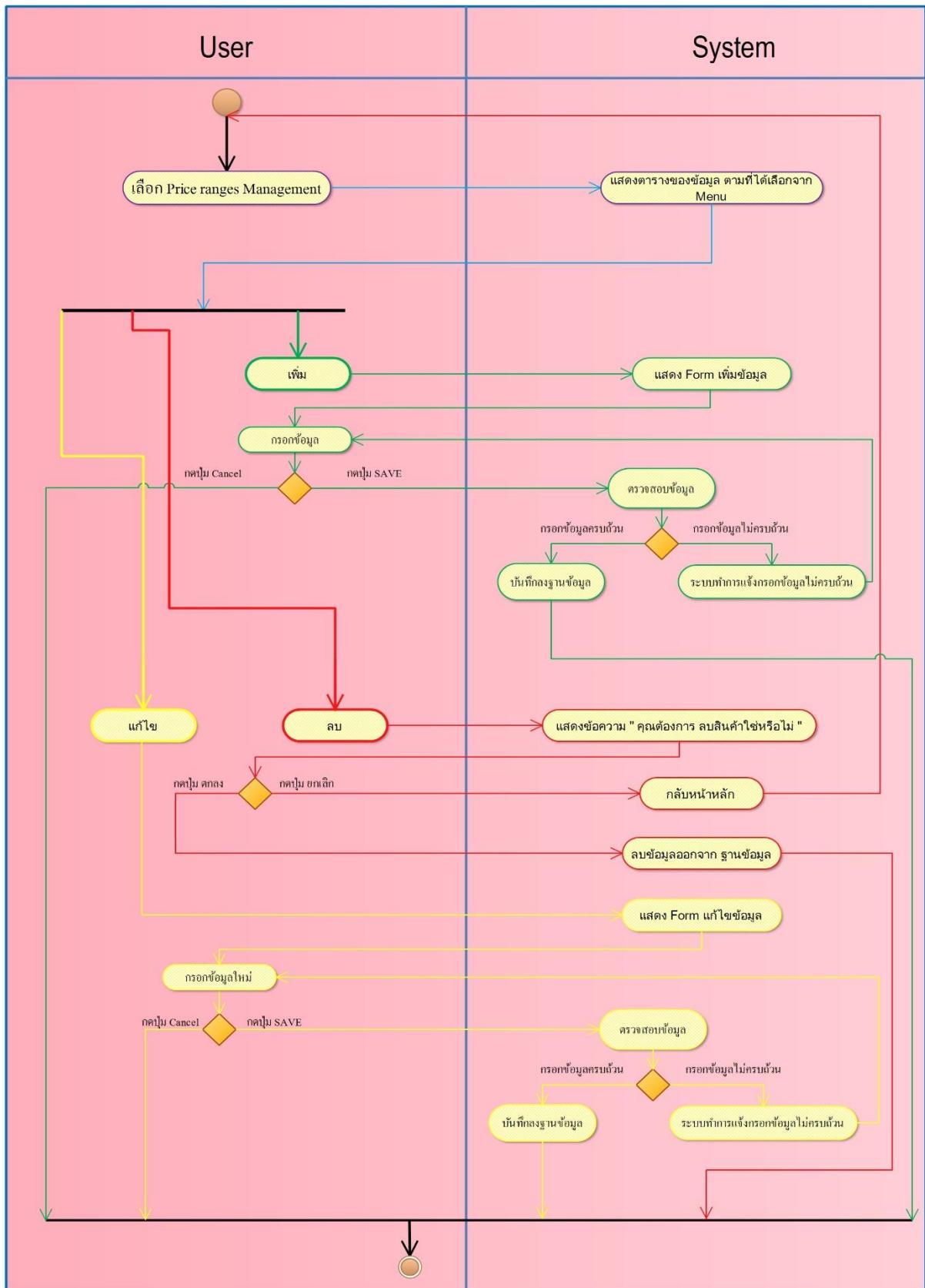
3.4.8 Activity Diagram : Back and Side tone woods Management



รูปที่ 3.27 Activity Diagram : Back and Side tone woods Management



3.4.9 Activity Diagram : Price ranges Management

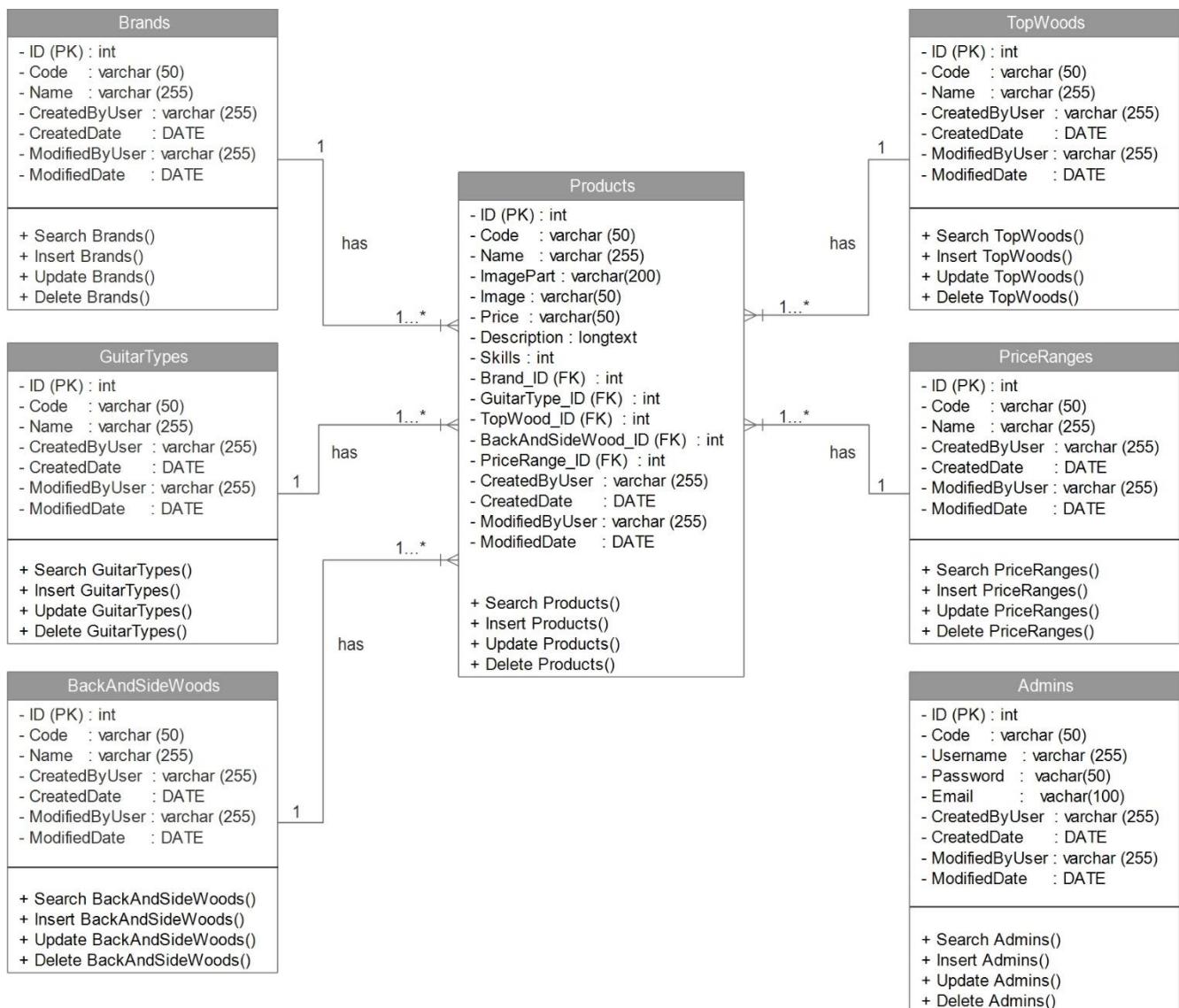


ຮູບທີ 3.28 Activity Diagram : Price ranges Management



3.5 Class Diagram ซึ่งเป็นแผนภาพที่ใช้แสดง Class ของความสัมพันธ์ (Relationship) ในแอ่งต่างๆระหว่าง Class ซึ่งแสดงรายละเอียดดังต่อไปนี้

3.5.1 Class Diagram : ระบบแนะนำการเลือกซื้อกีต้าร์



รูปที่ 3.29 Class Diagram : Price ranges Management



3.5.2 Database Dictionary

1. ตาราง Admins (ผู้ดูแลระบบ)

Attribute Name	Description	Data Type (Size)	Key Type	Reference Table
ID	เลขผู้ดูแลระบบ	Int (11)	Primary key	-
Code	รหัสประจำตัวผู้ดูแลระบบ	Varchar (50)	-	-
Username	ชื่อผู้ดูแลระบบ	Varchar (255)	-	-
Password	รหัสผ่านผู้ดูแลระบบ	Varchar (50)	-	-
Email	จดหมายอิเล็กทรอนิกส์	Varchar (100)	-	-
CreatedByUser	ชื่อผู้สร้างผู้ดูแลระบบนี้	Varchar (255)	-	-
CreatedDate	วันที่สร้างผู้ดูแลระบบนี้	DATE	-	-
ModifiedByUser	ชื่อผู้แก้ไขผู้ดูแลระบบนี้	Varchar (255)	-	-
ModifiedDate	วันที่แก้ไขผู้ดูแลระบบนี้	DATE	-	-

2. ตาราง Brands (ยี่ห้อ)

Attribute Name	Description	Data Type (Size)	Key Type	Reference Table
ID	เลขที่ห้อ	Int (11)	Primary key	-
Code	รหัสยี่ห้อ	Varchar (50)	-	-
Name	ชื่อยี่ห้อ	Varchar (255)	-	-
CreatedByUser	ชื่อผู้สร้างยี่ห้อ	Varchar (255)	-	-
CreatedDate	วันที่สร้างยี่ห้อ	DATE	-	-
ModifiedByUser	ชื่อผู้แก้ไขยี่ห้อ	Varchar (255)	-	-
ModifiedDate	วันที่แก้ไขยี่ห้อ	DATE	-	-

3. ตาราง GuitarTypes (ชนิดกีต้าร์)

Attribute Name	Description	Data Type (Size)	Key Type	Reference Table
ID	เลขชนิดกีต้าร์	Int (11)	Primary key	-
Code	รหัสชนิดกีต้าร์	Varchar (50)	-	-
Name	ชื่อชนิดกีต้าร์	Varchar (255)	-	-
CreatedByUser	ชื่อผู้สร้างชนิดกีต้าร์	Varchar (255)	-	-
CreatedDate	วันที่สร้างชนิดกีต้าร์	DATE	-	-
ModifiedByUser	ชื่อผู้แก้ไขชนิดกีต้าร์	Varchar (255)	-	-
ModifiedDate	วันที่แก้ไขชนิดกีต้าร์	DATE	-	-



4. ตาราง TopWoods (ไม้หน้า)

Attribute Name	Description	Data Type (Size)	Key Type	Reference Table
ID	เลขไม้หน้า	Int (11)	Primary key	-
Code	รหัสไม้หน้า	Varchar (50)	-	-
Name	ชื่อไม้หน้า	Varchar (255)	-	-
CreatedByUser	ชื่อผู้สร้างไม้หน้า	Varchar (255)	-	-
CreatedDate	วันที่สร้างไม้หน้า	DATE	-	-
ModifiedByUser	ชื่อผู้แก้ไขไม้หน้า	Varchar (255)	-	-
ModifiedDate	วันที่แก้ไขไม้หน้า	DATE	-	-

5. ตาราง BackAndSideWoods (ไม้หลังและไม้ข้าง)

Attribute Name	Description	Data Type (Size)	Key Type	Reference Table
ID	เลขไม้หลังและไม้ข้าง	Int (11)	Primary key	-
Code	รหัสไม้หลังและไม้ข้าง	Varchar (50)	-	-
Name	ชื่อไม้หลังและไม้ข้าง	Varchar (255)	-	-
CreatedByUser	ชื่อผู้สร้างไม้หลังและไม้ข้าง	Varchar (255)	-	-
CreatedDate	วันที่สร้างไม้หลังและไม้ข้าง	DATE	-	-
ModifiedByUser	ชื่อผู้แก้ไขไม้หลังและไม้ข้าง	Varchar (255)	-	-
ModifiedDate	วันที่แก้ไขไม้หลังและไม้ข้าง	DATE	-	-

6. ตาราง PriceRanges (ช่วงราคา)

Attribute Name	Description	Data Type (Size)	Key Type	Reference Table
ID	เลขช่วงราคา	Int (11)	Primary key	-
Code	รหัสช่วงราคา	Varchar (50)	-	-
Name	ชื่อช่วงราคา	Varchar (255)	-	-
CreatedByUser	ชื่อผู้สร้างช่วงราคา	Varchar (255)	-	-
CreatedDate	วันที่สร้างช่วงราคา	DATE	-	-
ModifiedByUser	ชื่อผู้แก้ไขช่วงราคา	Varchar (255)	-	-
ModifiedDate	วันที่แก้ไขช่วงราคา	DATE	-	-



7. ตาราง Products (กีต้าร์)

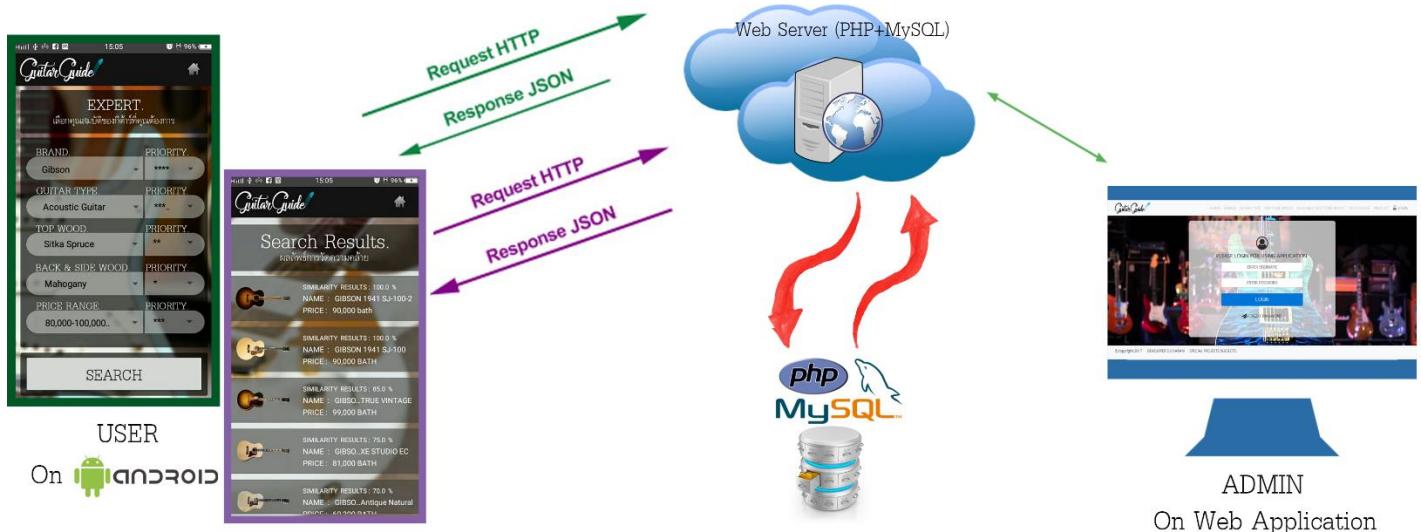
Attribute Name	Description	Data Type (Size)	Key Type	Reference Table
ID	เลขช่วงราคา	Int (11)	Primary key	-
Code	รหัสช่วงราคา	Varchar (50)	-	-
Name	ชื่อช่วงราคา	Varchar (255)	-	-
ImagePart	ชื่อที่เก็บภาพ	varchar(200)	-	-
Image	ชื่อรูปภาพ	varchar(50)	-	-
Price	ราคา	varchar(50)	-	-
Description	URL เว็บที่มาของกีต้าร์	longtext	-	-
Skills	ทักษะ	Int (11)	-	-
Brand_ID	เลขบัญชีห้อ	Int (11)	Foreign Key	Brands
GuitarType_ID	เลขชนิดกีต้าร์	Int (11)	Foreign Key	GuitarTypes
TopWood_ID	เลขไม้หน้า	Int (11)	Foreign Key	TopWoods
BackAndSideWood_ID	เลขไม้หลังและไม้ข้าง	Int (11)	Foreign Key	BackAndSideWoods
PriceRange_ID	เลขช่วงราคา	Int (11)	Foreign Key	RriceRanges
CreatedByUser	ชื่อผู้สร้างช่วงราคา	Varchar (255)	-	-
CreatedDate	วันที่สร้างช่วงราคา	DATE	-	-
ModifiedByUser	ชื่อผู้แก้ไขช่วงราคา	Varchar (255)	-	-
ModifiedDate	วันที่แก้ไขช่วงราคา	DATE	-	-



บทที่ 4

การสร้างระบบแอพพลิเคชัน

การทำงานของระบบแอพพลิเคชันทั้ง เว็บแอพพลิเคชัน และแอนดรอยด์แอพพลิเคชันนั้น มีหลักการดังนี้



รูปที่ 3.30 แสดงภาพรวมของระบบแอพพลิเคชัน

4.1 ระบบเว็บแอพพลิเคชัน (การจัดการฐานข้อมูล : Admin)

4.1.1 การเชื่อมต่อ Server

```

17 |     private String driver = "com.mysql.jdbc.Driver";
18 |     private String uri = "jdbc:mysql://localhost:3306/guitarguide?useUnicode=true&characterEncoding=UTF-8";
19 |     private String user = "root";
20 |     private String pass = "";
21 |     private Connection connect = null;
  
```

บรรทัดที่ 17-21 สร้างตัวแปรเพื่อกีบข้อมูลที่ใช้ในการเชื่อมต่อกับ Server



```

30     try {
31         Class.forName(driver);
32         connect = (Connection) DriverManager.getConnection(uri, user, pass);
33
34         if (connect != null) {
35             out.println("Database Connected.");
36         } else {
37             out.println("Database Connect Failed.");
38         }
39     } catch (Exception e) {
40         e.printStackTrace();
41     }

```

บรรทัดที่ 30-41 คำสั่งการเชื่อมต่อกับ Server

4.1.2 การจัดการล่วงของ Login

- HTML file : Login

```

51 <form class="form-signin" role="form" method="POST" action="CheckLoginController">
52     <input type="text" class="form-control text-center" name="txtUsername" placeholder="ENTER USERNAME" required autofocus>
53     <br>
54     <input type="password" class="form-control text-center" name="txtPassword" placeholder="ENTER PASSWORD" required>
55     <br>
56     <button class="btn btn-sm btn-lg btn-primary btn-block text-uppercase" type="submit" name="login">
57         <font size="6" face="TH SarabunPSK">Login</font>
58     </button>
59     <br>
60     <i class="fa d-inline fa-lg fa fa-paper-plane"></i><a class="text-dark" href="ForgotPassword.jsp">
61         <font size="5" face="TH SarabunPSK"><b>FORGOT PASSWORD.</b></font></a>
62 </form>

```

บรรทัดที่ 51-62 ทำการรับค่า Username และ Password แล้วส่งไปที่ Servlet file : CheckLoginController

- Servlet file : Login

```

19     HttpSession session = request.getSession(true);
20
21     if (request.getParameter("login") != null) {
22
23         CheckLoginDAO dao = new CheckLoginDAO();
24
25         String txtUsername = request.getParameter("txtUsername");
26         String txtPassword = request.getParameter("txtPassword");
27         boolean rs = dao.CheckLogin(txtUsername, txtPassword);
28
29         if(rs == true){
30             session.setAttribute("Username",txtUsername);
31
32             request.getRequestDispatcher("index.jsp").forward(request, response);
33         }else{
34             String CheckUsernameAndPassword_alert = "<script> alert('USERNAME หรือ PASSWORD ของคุณไม่ถูกต้อง'); </script>";
35             request.setAttribute("CheckUsernameAndPassword_alert", CheckUsernameAndPassword_alert);
36             request.getRequestDispatcher("Login.jsp").forward(request, response);
37         }
38     }

```

บรรทัดที่ 21-37 เมื่อมีการกดปุ่ม เข้าสู่ระบบมา เช็ค if condition จากนั้นส่ง Username และ Password ไปตรวจสอบที่ CheckLoginDAO



- DAO (Data Access Object) (ส่วน Query) : Login

```

55  public boolean CheckLogin(String txtUsername, String txtPassword) {
56
57      boolean result = false;
58
59      String sql = "SELECT * FROM admins WHERE username = '"+txtUsername+"' AND password = '"+txtPassword+"'";
60
61      try {
62
63          PreparedStatement ps = connect.prepareStatement(sql);
64          ResultSet rs = ps.executeQuery();
65
66          if(rs.next() == true){
67              result = true;
68          }
69
70
71          connect.close();
72      } catch (Exception e) {
73          e.printStackTrace();
74      }
75
76      return result;
77  }

```

บรรทัดที่ 55-77 รับค่า Username และ Password มาจาก Servlet file จากนั้นทำการตรวจสอบค่าทั้งสองในฐานข้อมูล กรณีข้อมูลที่ส่งมาตรงกับข้อมูลในฐานข้อมูล Return : true

4.1.3 การจัดการฐานข้อมูล Brands

1. การค้นหาข้อมูล : Brands

- HTML file : Brands

```

118  <form class="form-search form-inline" action="BrandSearchController" name="frm" method="POST">
119      <input class="form-control btn-sm" type="text" name="keyword" placeholder="ENTER KEYWORD" required>
120      &nbsp;&nbsp;&nbsp;
121      <button class="btn btn-light border btn-sm fa fa-search" type="submit" name="submit">
122          <font size="4" face="Calibri">&nbsp;SEARCH</font>
123      </button>
124  </form>

```

บรรทัดที่ 118-124 ทำการรับคำค้นหาจากผู้ใช้กรอก และทำการส่งคำค้นไปที่ Servlet file :

BrandSearchController



- Servlet file : Brands

```

38     if (request.getParameter("keyword") != "") {
39         keyword = request.getParameter("keyword");
40         searchList = dao.Search(keyword);
41
42         if (searchList != null) {
43             /*
44              - จากหัวเม่อได้ข้อมูลมาแล้วก็ทำการส่งไปยัง Web Page -> Brand.jsp
45            */
46             request.setAttribute("searchList", searchList);
47             request.getRequestDispatcher("/Brand.jsp").forward(request, response);
48         } else {
49
50             session.setAttribute("FindData", "DATA_NOT_FOUND");
51             /*
52              - จากหัวเม่อได้ข้อมูลมาแล้วก็ทำการส่งไปยัง Web Page -> BrandFindAllController
53            */
54             response.sendRedirect("BrandFindAllController");
55         }
56
57     } else {
58         session.setAttribute("FindData", "DATA_NOT_FOUND");
59         /*
60          - จากหัวเม่อได้ข้อมูลมาแล้วก็ทำการส่งไปยัง Web Page -> BrandFindAllController
61        */
62         response.sendRedirect("BrandFindAllController");
63     }

```

บรรทัดที่ 37- 62 รับค่าค้นหาที่ส่งมา จากนั้นส่งไปตรวจสอบในฐานข้อมูลที่ Search Method ใน BrandDAO file

- DAO (Data Access Object) (ส่วน Query) : Brands

```

137     public List<BrandInfo> Search(String keyword) {
138         List<BrandInfo> searchList = new ArrayList<BrandInfo>();
139         String sql = "SELECT * FROM brands WHERE id LIKE '%" +
140             + keyword + "%' OR code LIKE '%" +
141             + keyword + "%' OR name LIKE '%" +
142             + keyword + "%' ORDER BY id ASC    ";
143
144         try {
145             PreparedStatement ps = connect.prepareStatement(sql);
146             ResultSet result = ps.executeQuery();
147             int checkFind = 0;
148             while (result.next()) {
149                 checkFind++;
150                 BrandInfo b = new BrandInfo();
151                 b.setID(result.getInt("ID"));
152                 b.setCode(result.getString("Code"));
153                 b.setName(result.getString("Name"));
154                 b.setCreatedByUser(result.getString("CreatedByUser"));
155                 b.setCreatedDate(result.getDate("CreatedDate"));
156                 b.setModifiedByUser(result.getString("ModifiedByUser"));
157                 b.setModifiedDate(result.getDate("ModifiedDate"));
158                 searchList.add(b);
159             }

```

บรรทัดที่ 137-159 รับค่าค้นหามาแล้วทำการตรวจสอบข้อมูลในฐานข้อมูลแล้ว Return ค่าการค้นหากลับไปแสดง



2. การเพิ่มข้อมูล : Brands

- HTML file : Brands

```

32   <form class="form" method="POST" id="insert-form" name="insert-form">
33     <table class="table" align="center" style=" width: 100%; background-color: rgba(255,255,255,0.60);">
34       <tr class="border">
35         <td>
36           <div class="container-fluid">
37             <div class="row">
38               <div class="col-1"></div>
39               <div class="col-10">
40                 <div class="form-group">
41                   <label class="text-uppercase"><h5> code </h5></label>
42                   <input placeholder="ENTER CODE" style="background-color: rgba(255,255,255,0.90); " id="code" type="text" name="code">
43                 </div>
44                 <div class="form-group">
45                   <label class="text-uppercase"><h5> name </h5></label>
46                   <input placeholder="ENTER NAME" style="background-color: rgba(255,255,255,0.90); " id="name" type="text" name="name">
47                 </div>
48                 <div class="form-group" align="right">
49                   <button id="btn_insert" name="btn_insert" value="SAVE" type="button" class="btn btn-success text-dark border-light">SAVE</button>
50                   <button type="reset" value="Reset" class="btn btn-warning text-dark border-light">RESET</button>
51                 </div>
52               </div>
53             </div>
54           </td>
55         </tr>
56       </table>
57     </form>

```

บรรทัดที่ 31-57 ทำการส่งข้อมูลที่ทำการกรอกมาไปที่ javascript/ajax เพื่อนำไปเพิ่มข้อมูลในฐานข้อมูล

- Javascript : Brands

```

420   $('#btn_insert').click(function () {
421
422     $.ajax({
423       url: 'BrandInsertDataController',
424       method: 'POST',
425       data: $('#insert-form').serialize(),
426       dataType: 'json',
427       success: function (data) {
428
429         if (data.Insert_Complete === "Complete") {
430           alert('เพิ่มข้อมูลเรียบร้อยแล้ว.');
431           $('#insert-form')[0].reset();
432         } else if (data.Insert_Complete === "Not Complete") {
433           alert('CODE หรือ NAME ที่กรอกมีอยู่แล้ว กรุณากรอกใหม่อีกครั้ง.');
434         } else if (data.Code_Name === "No Data") {
435           alert('กรุณากรอกข้อมูล CODE และ NAME.');
436         }
437       }
438     });
439   });

```

บรรทัดที่ 420-439 ทำการส่งข้อมูลที่กรอกจาก insert form ไปที่ BrandInsertDataController



- Servlet file : Brands

```

31     if ((request.getParameter("name") != "") && (request.getParameter("code") != "")) {
32         LocalDate now = LocalDate.now();
33         String code = request.getParameter("code");
34         String name = request.getParameter("name");
35         String createdByUser = (String) session.getAttribute("Username");
36         Date createdDate = Date.valueOf(now);
37         String modifiedByUser = null;
38         Date modifiedDate = null;
39
40         // ใช้สำหรับแสดงผลการเพื่อตรวจสอบความถูกต้อง ในการ debug จุดตรวจสอบผ่านการ กlickขวาที่เริ่ม แล้วเลือกตรวจสอบ code
41         json.put("code", code);
42         json.put("name", name);
43         json.put("createdByUser", createdByUser);
44         json.put("createdDate", createdDate);
45         json.put("modifiedByUser", modifiedByUser);
46         json.put("modifiedDate", modifiedDate);
47         BrandDAO dao = new BrandDAO();
48         boolean Complete = dao.InsertBrand(code, name, createdByUser, createdDate, modifiedByUser, modifiedDate);
49
50         if (Complete == true) {
51             json.put("Insert_Complete", "Complete");
52         } else {
53             json.put("Insert_Complete", "Not Complete");
54         }
55         json.put("Code_Name", "Have Data");
56     } else {
57         json.put("Code_Name", "No Data");
58     }

```

บรรทัดที่ 31-57 รับค่าจาก ajax จากนั้นส่งไป insert ลงฐานข้อมูลที่ InsertBrand Method ใน BrandDAO file

- DAO (Data Access Object) (ส่วน Query) : Brands

```

307     public boolean InsertBrand(String code, String name, String createdByUser,
308                               Date createdDate, String modifiedByUser, Date modifiedDate) throws SQLException {
309         boolean Complete = false;
310         try {
311             Statement ps = connect.createStatement();
312             ResultSet rs = ps.executeQuery("SELECT * FROM brands WHERE Code = '" + code + "' OR Name = '" + name + "'");
313
314             int Duplicate = 0;
315             while (rs.next()) {
316
317                 Duplicate++;
318             }
319             if (Duplicate > 0) {
320                 Complete = false;
321             } else {
322                 PreparedStatement insert_Data = connect.prepareStatement("INSERT INTO brands "
323                             + "(Code,Name,CreatedByUser,CreatedDate,ModifiedByUser,ModifiedDate) VALUES "
324                             + "(" + code + "','" + name + "','" + createdByUser + "','" + createdDate + "',null,null)");
325                 insert_Data.executeUpdate();
326                 Complete = true;
327             }
328             connect.close();
329         } catch (Exception e) {
330             e.printStackTrace();
331         }
332         return Complete;
333     }

```

บรรทัดที่ 307-333 ทำการเพิ่มข้อมูล Brand ใหม่ลงในฐานข้อมูล



3. การแก้ไขข้อมูล : Brands

- HTML file : Brands

```

29  <form class="form" method="POST" id="update-form" name="update-form">
30      <legend class="text-uppercase"><h5 class="text-light"> Edit Data </h5></legend>
31      <br>
32      <table class="table" align="center" style=" width: 100%; background-color: rgba(255,255,255,0.60);">
33          <tr class="border">
34              <div class="form-group">
35                  <input type="hidden" id="id" name="id"/>
36              </div>
37              <div class="container-fluid">
38                  <div class="row">
39                      <div class="col-1"></div>
40                      <div class="col-10">
41                          <div class="form-group">
42                              <label class="text-uppercase"><h5> code </h5></label>
43                              <input style="background-color: rgba(255,255,255,0.90); " id="code" type="text" name="code" class="form-control text-uppe
44                          </div>
45                          <div class="form-group">
46                              <label class="text-uppercase"><h5> name </h5></label>
47                              <input style="background-color: rgba(255,255,255,0.90); " id="name" type="text" name="name" class="form-control text-uppe
48                          </div>
49                          <div class="form-group" align="right">
50                              <input id="btn_update" name="btn_update" value="SAVE" type="button" class="btn btn-success text-dark border-light">
51                              <button type="reset" value="Reset" class="btn btn-warning text-dark border-light">RESET</button>
52                          </div>
53                      </div>
54                  <div class="col-1"></div>
55              </div>

```

บรรทัดที่ 29-55 ทำการกรอกข้อมูลที่ต้องการแก้ไข แล้วกดปุ่ม บันทึก ค่าต่างๆจะถูกส่งไปยัง ajax เพื่อส่งต่อไปยัง Controller

- Javascript : Brands

```

387  $('#btn_update').click(function () {
388      $.ajax({
389          url: 'BrandEditDataController',
390          method: 'POST',
391          data: $('#update-form').serialize(),
392          dataType: 'json',
393          success: function (data) {
394
395              if (data.Edit_Complete === "Complete") {
396                  alert('แก้ไขข้อมูลเรียบร้อยแล้ว.');
397                  location.reload();
398                  //window.location='BrandFindAllController';
399              } else if (data.Edit_Complete === "Not Complete") {
400                  alert('CODE หรือ NAME ที่กรอกมีอยู่แล้ว กรุณากรอกใหม่อีกครั้ง.');
401              } else if (data.Code_Name === "No Data") {
402                  alert('กรุณากรอกข้อมูล CODE และ NAME.');
403              }
404          }
405      });
406  });

```

บรรทัดที่ 387-406 รับค่าจาก Edit Form และทำการส่งค่าไปที่ BrandEditDataController



- Servlet file : Brands

```

33     if (request.getParameter("id") != null) {
34         LocalDate now = LocalDate.now();
35         if ((request.getParameter("name") != "") && (request.getParameter("code") != "")) {
36             int id = Integer.parseInt(request.getParameter("id"));
37             String code = request.getParameter("code");
38             String name = request.getParameter("name");
39             String createdByUser = request.getParameter("createdByUser");
40             Date createdDate = Date.valueOf(request.getParameter("createdDate"));
41             String modifiedByUser = (String) session.getAttribute("Username");
42             Date modifiedDate = Date.valueOf(now);
43             json.put("id", id);
44             json.put("code", code);
45             json.put("name", name);
46             json.put("createdByUser", createdByUser);
47             json.put("createdDate", createdDate);
48             json.put("modifiedByUser", modifiedByUser);
49             json.put("modifiedDate", modifiedDate);
50             BrandDAO dao = new BrandDAO();
51             boolean Complete = dao.EditBrand(id, code, name, createdByUser, createdDate, modifiedByUser, modifiedDate);
52             if (Complete == true) {
53                 json.put("Edit_Complete", "Complete");
54             } else {
55                 json.put("Edit_Complete", "Not Complete");
56             }
57             json.put("Code_Name", "Have Data");
58         } else {
59             json.put("Code_Name", "No Data");
60         }
}

```

บรรทัดที่ 33-60 ทำการรับค่าจาก ajax ตรวจสอบ id ?ส่งมาต้องไม่ใช่ค่าว่าง จากนั้นทำการส่งข้อมูลการแก้ไขไปที่ EditBrand Method ใน BrandDAO file

- DAO (Data Access Object) (ส่วน Query) : Brands

```

273     public boolean EditBrand(int id, String code, String name, String createdByUser,
274                             Date createdDate, String modifiedByUser, Date modifiedDate) {
275         boolean Complete = false;
276         try {
277             Statement ps = connect.createStatement();
278             ResultSet rs = ps.executeQuery("SELECT * FROM brands WHERE (id != '" + id + "' "
279                                         + "AND (Code = '" + code + "' OR Name = '" + name + "')");
280             int Duplicate = 0;
281             while (rs.next()) {
282                 Duplicate++;
283             }
284             if (Duplicate > 0) {
285                 Complete = false;
286             } else {
287                 PreparedStatement update_Data = connect.prepareStatement("UPDATE brands SET Code ='" +
288                                         + code + "' ,Name ='" + name + "' ,CreatedByUser ='" + createdByUser + "' ,CreatedDate ='" +
289                                         + createdDate + "' ,ModifiedByUser ='" + modifiedByUser + "' ,ModifiedDate ='" + modifiedDate +
290                                         + "' WHERE ID ='" + id + "'");
291                 update_Data.executeUpdate();
292                 Complete = true;
293             }
294             connect.close();
295         } catch (Exception e) {
296             e.printStackTrace();
297         }
298         return Complete;
299     }
}

```

บรรทัดที่ 273-299 ทำการรับค่า parameter ที่ส่งมาเพื่อแก้ไขข้อมูลไปตรวจสอบความซ้ำซ้อน กรณ์ไม่ซ้ำก็ทำการบันทึกลงฐานข้อมูล



4. การลบข้อมูล : Brands

- HTML file : Brands

```

220 |     <button class="btn btn-danger text-white btn-sm fa fa-trash-o delete-data" id="${bList.ID}">
221 |         <font face="Calibri">&ampnbspDELETE</font>
222 |     </button>

```

บรรทัดที่ 220-222 เมื่อกดปุ่ม DELETE จะส่ง id ที่เป็น primary key ณ index ที่เราคลิกนั้นไปที่ ajax

- Javascript : Brands

```

322 |     $('.delete-data').click(function () {
323 |
324 |         var uid = $(this).attr("id");
325 |         var status = confirm('คุณต้องการลบข้อมูล ใช่หรือไม่?');
326 |
327 |         if (status) {
328 |             $.ajax({
329 |                 url: 'BrandDeleteDataController',
330 |                 method: 'POST',
331 |                 data: {id: uid},
332 |                 dataType: 'json',
333 |                 success: function (data) {
334 |
335 |                     if (data.Deleteed_Complete === "Not Complete") {
336 |                         alert('ไม่สามารถลบข้อมูลได้ PRODUCT ใช้งานข้อมูลห้อย !!!');
337 |                         location.reload();
338 |                     }
339 |                     if (data.Deleteed_Complete === "Complete") {
340 |                         alert('ทำการลบข้อมูลเรียบร้อยแล้ว');
341 |                         location.reload();
342 |                     }
343 |
344 |                 }
345 |             });
346 |         }
347 |

```

บรรทัดที่ 322-346 เมื่อได้รับ id จากHTML file และจึงทำการส่งต่อไปที่ Servlet file :

BrandDeleteDataController



- Servlet file : Brands

```

27     PrintWriter out = response.getWriter();
28     JSONObject json = new JSONObject();
29
30     int id = Integer.parseInt(request.getParameter("id"));
31
32     BrandDAO dao = new BrandDAO();
33     int deleteData = 0;
34     deleteData = dao.DeleteData(id);
35
36     if(deleteData == 0) {
37         json.put("Deleteeed_Complete", "Not Complete");
38     }
39     if(deleteData == 1) {
40         json.put("Deleteeed_Complete", "Complete");
41     }
42
43     out.print(json);

```

บรรทัดที่ 27-43 คือเมื่อรับค่าจาก ajax แล้วก็ทำการส่งค่าที่รับมานั้นไปที่ DeleteData Method :

BrandDAO file

- DAO (Data Access Object) (ส่วน Query) : Brands

```

171     public int DeleteData(int id) {
172         int deleteData = 0;
173         try {
174             PreparedStatement p = connect.prepareStatement("SELECT * FROM products WHERE brand_id = '" + id + "'");
175             ResultSet re = p.executeQuery();
176             int check = 0;
177             while (re.next()) {
178                 check = 1;
179             }
180             if (check == 0) {
181                 String sql = "DELETE FROM brands WHERE id = '" + id + "'";
182                 PreparedStatement ps = connect.prepareStatement(sql);
183                 ps.executeUpdate();
184                 deleteData = 1;
185             }
186             connect.close();
187         } catch (Exception e) {
188             e.printStackTrace();
189         }
190     }
191     return deleteData;
192 }

```

บรรทัดที่ 171-192 คือทำการรับ id parameter ที่ส่งมาแล้วทำการตรวจสอบความถูกต้องของ id ในฐานข้อมูลกรณีที่ตรงกันก็ Return : True กลับไป



4.1.4 การจัดการฐานข้อมูล Guitar Types

4.1.5 การจัดการฐานข้อมูล Top Tone Woods

4.1.6 การจัดการฐานข้อมูล Back & Side Tone Woods

4.1.7 การจัดการฐานข้อมูล Price Ranges

4.1.4 – 4.1.7 หลักการการทำงานเหมือนกับ Brands เพราะเป็นการเขียนแบบ reuse source code คือ เป็นการเขียนแยกส่วนการทำงานเอาไว้เป็นส่วนๆ เมื่อต้องการจะใช้จึงเรียกผ่านชื่อ file พร้อมส่งพารามิเตอร์ชื่อของ การจัดการนั้นไปด้วยจึงทำให้ทราบว่าขณะนี้ file ได้กำลังขอสิทธิ์ใช้งานอยู่

4.1.8 การจัดการฐานข้อมูล Products

1. การค้นหาข้อมูล : Products

- HTML file : Products

```

117  <form class="form-search form-inline" action="ProductSearchController" name="frm" method="POST">
118    <input class="form-control btn-sm" type="text" name="keyword" placeholder="ENTER KEYWORD" required/>
119    &nbsp;&nbsp;&nbsp;
120    <button class="btn btn-light border btn-sm fa fa-search" type="submit" name="submit">
121      <font size="4" face="Calibri">&nbsp;SEARCH</font>
122    </button>
123  </form>

```

บรรทัดที่ 117-123 กรอกคำค้นหาจากนั้น กดปุ่ม SEARCH



- Servlet file : Products

```

30     String keyword = null;
31     ProductDAO dao = new ProductDAO();
32     List<ProductInfo> searchList = new ArrayList<ProductInfo>();
33
34     if (request.getParameter("keyword") != "") {
35         keyword = request.getParameter("keyword");
36         searchList = dao.Search(keyword);
37
38         if (searchList != null) {
39             request.setAttribute("searchList", searchList);
40             request.getRequestDispatcher("/Product.jsp").forward(request, response);
41         } else {
42
43             session.setAttribute("FindData", "DATA_NOT_FOUND");
44             response.sendRedirect("ProductFindAllController");
45         }
46
47     } else {
48         session.setAttribute("FindData", "DATA_NOT_FOUND");
49         response.sendRedirect("ProductFindAllController");
50     }
51 }
```

บรรทัดที่ 30-51 รับค่าคำค้นหาจาก web browser จากนั้นส่งไป Search Method ใน ProductDAO file

- DAO (Data Access Object) (ส่วน Query) : Products

```

470 public List<ProductInfo> Search(String keyword) {
471     List<ProductInfo> searchList = new ArrayList<ProductInfo>();
472     String sql = "SELECT * FROM products WHERE id LIKE '%" + keyword + "%' OR code LIKE '%" + keyword + "%' OR name LIKE '%" + keyword + "%'";
473     try {
474         PreparedStatement ps = connect.prepareStatement(sql);
475         ResultSet result = ps.executeQuery();
476         int checkFind = 0;
477         while (result.next()) {
478             checkFind++;
479             ProductInfo b = new ProductInfo();
480             b.setID(result.getInt("ID"));
481             b.setCode(result.getString("Code"));
482             b.setName(result.getString("Name"));
483             b.setImagePart(result.getString("ImagePart"));
484             b.setImage(result.getString("Image"));
485             b.setPrice(result.getString("Price"));
486             b.setDescription(result.getString("Description"));
487             b.setSkills(result.getInt("Skills"));
488             b.setBrand_ID(result.getInt("Brand_ID"));
489             b.setGuitarType_ID(result.getInt("GuitarType_ID"));
490             b.setTopWood_ID(result.getInt("TopWood_ID"));
491             b.setBackAndSideWood_ID(result.getInt("BackAndSideWood_ID"));
492             b.setPriceRange_ID(result.getInt("PriceRange_ID"));
493             b.setCreatedByUser(result.getString("CreatedByUser"));
494             b.setCreatedDate(result.getDate("CreatedDate"));
495             b.setModifiedByUser(result.getString("ModifiedByUser"));
496             b.setModifiedDate(result.getDate("ModifiedDate"));
497
498             searchList.add(b);
499         }
500
501         // - กรองที่ keyword ที่ส่งมาไม่ถูก query และไม่มี
502         if (checkFind == 0) {
503             searchList = null;
504         }
505
506         connect.close();
507
508     } catch (Exception e) {
509         e.printStackTrace();
510     }
511
512     return searchList;
513 }
514 }
```

บรรทัดที่ 470-514 รับค่าพารามิเตอร์ที่ส่งมาจาก Servlet file จากนั้นทำการ Query ข้อมูลจากฐานข้อมูลเพื่อตรวจสอบความตรงกัน จากนั้นส่งผลลัพธ์การค้นหาไปแสดง

2. การเพิ่มข้อมูล : Products

- HTML file : Products

```
44 <form class="form" method="POST" id="insertForm" name="insertForm">
45     <legend class="text-uppercase"><h5 class="text-light"> Insert Data </h5></legend>
46     <table class="table" style=" width: 100%; background-color: rgba(255,255,255,0.50);">
47         <tr class="border">
48             <td width="50%">
49                 <div class="form-group">
50                     <label class="text-uppercase"><h6> code </h6></label>
51                     <input placeholder="ENTER CODE" style="background-color: rgba(255,255,255,0.60); " id="code" type="text" name="code" clas
52                 </div>
53                 <div class="form-group">
54                     <label class="text-uppercase"><h6> name </h6></label>
55                     <input placeholder="ENTER NAME" style="background-color: rgba(255,255,255,0.60); " id="name" type="text" name="name" clas
56                 </div>
57                 <div class="form-group">
58                     <label class="text-uppercase"><legend><h6> price </h6></legend></label>
59                     <input placeholder="ENTER PRICE" style="background-color: rgba(255,255,255,0.60); " id="price" type="text" name="price" c
60                 </div>
61                 <div class="form-group">
62                     <label class="text-uppercase"><legend><h6> image part </h6></legend></label>
63                     <input placeholder="ENTER IMAGE PART" style="background-color: rgba(255,255,255,0.60); " id="imagePart" type="text" name=
64                 </div>
65                 <div class="form-group" width="50%">
66                     <label class="text-uppercase"><legend><h6> image <img height="30px" id="out_put"/>&nbsp;<span class="fa fa-picture-o"></
67                     <input style="background-color: rgba(255,255,255,0.60); " onchange="load_File(event)" type="file" id="image" name="image"
</div>

69                 <div class="form-group" width="50%">
70                     <div class="form-group text-uppercase">
71                         <label class="text-uppercase"><legend><h6> Skills </h6></legend></label>
72                         <div style="background-color: rgba(255,255,255,0.60); " class="checkbox form-control form-inline" required>
73                             <input id="skills" type="radio" name="skills" value="1" checked="checked" > beginner &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
74                             <input id="skills" type="radio" name="skills" value="2" > expert
75                         </div>
76                     </div>
77                 <%
78                     //request.setAttribute("bName", session.getAttribute("brand_Name"));
79                     List brand_List = (List) session.getAttribute("brand_List");
80                 %>
81                 <div class="form-group" width="50%">
82                     <div class="form-group text-uppercase">
83                         <label class="text-uppercase"><legend><h6> Brand </h6></legend></label>
84                         <select id="brand_id" name="brand_id" style="background-color: rgba(255,255,255,0.60); " class="form-control text-up
85                             <c:forEach items="${brand_List}" var="brand_List">
86                                 <option style="color: black;" id="brand_id" name="brand_id" value="${brand_List.ID}"><c:out value="${brand_L
87                             </c:forEach>
88                         </select>
89                     </div>
90                 </div>
91             </td>
92         </tr>
```



```

97      <td width="50%">
98          <div class="form-group" width="50%">
99              <div class="form-group text-uppercase">
100                  <label class="text-uppercase"><legend><h6> Guitar Type </h6></legend></label>
101                  <select id="guitarType_id" name="guitarType_id" style="background-color: rgba(255,255,255,0.60); " class="form-control">
102                      <c:forEach items="${guitarType_List}" var="guitarType_List">
103                          <option style="color: black;" id="guitarType_id" name="guitarType_id" value="${guitarType_List.ID}"><c:out value="${guitarType_List.ID}" />
104                      </c:forEach>
105                  </select>
106          </div>
107      <*>
108          //request.setAttribute("tName", session.getAttribute("topWood_Name"));
109          List topWood_List = (List) session.getAttribute("topWood_List");
110      <*>
111          <div class="form-group" width="50%">
112              <div class="form-group text-uppercase">
113                  <label class="text-uppercase"><legend><h6> Top tone Wood </h6></legend></label>
114                  <select id="topWood_id" name="topWood_id" style="background-color: rgba(255,255,255,0.60); " class="form-control text">
115                      <c:forEach items="${topWood_List}" var="topWood_List">
116                          <option style="color: black;" id="topWood_id" name="topWood_id" value="${topWood_List.ID}"><c:out value="${topWood_List.ID}" />
117                      </c:forEach>
118                  </select>
119          </div>
120      </div>
121
126          <div class="form-group" width="50%">
127              <div class="form-group text-uppercase">
128                  <label class="text-uppercase"><legend><h6> Back And Side tone Wood </h6></legend></label>
129                  <select id="backAndSideWood_id" name="backAndSideWood_id" style="background-color: rgba(255,255,255,0.60); " class="form-control text">
130                      <c:forEach items="${backAndSideWood_List}" var="backAndSideWood_List">
131                          <option style="color: black;" id="backAndSideWood_id" name="backAndSideWood_id" value="${backAndSideWood_List.ID}"><c:out value="${backAndSideWood_List.ID}" />
132                      </c:forEach>
133                  </select>
134          </div>
135      <*>
136          //request.setAttribute("pName", session.getAttribute("priceRange_Name"));
137          List priceRange_List = (List) session.getAttribute("priceRange_List");
138      <*>
140          <div class="form-group" width="50%">
141              <div class="form-group text-uppercase">
142                  <label class="text-uppercase"><legend><h6> Price Range </h6></legend></label>
143                  <select id="priceRange_id" name="priceRange_id" style="background-color: rgba(255,255,255,0.60); " class="form-control text">
144                      <c:forEach items="${priceRange_List}" var="priceRange_List">
145                          <option style="color: black;" id="priceRange_id" name="priceRange_id" value="${priceRange_List.ID}"><c:out value="${priceRange_List.ID}" />
146                      </c:forEach>
147                  </select>
148          </div>
149      </div>
150
151          <div class="form-group">
152              <label class="text-uppercase"><legend><h6> Description </h6></legend></label>
153              <textarea id="description" name="description" style="background-color: rgba(255,255,255,0.60); " id="description" name="description" value=""></textarea>
154          </div>
155          <div class="form-group" align="right">
156              <button id="btn_insert" name="btn_insert" value="SAVE" type="button" class="btn btn-success text-dark border-light">SAVE</button>
157              <button type="reset" value="Reset" class="btn btn-warning text-dark border-light">RESET</button>
158          </div>
159
160          </td>
161      </tr>
162  </table>
163 </form>
164

```

บรรทัดที่ 44-164 กรอกข้อมูลของ Product ที่ต้องการบันทึกลงในฐานข้อมูลจากนั้นกดปุ่ม SAVE



- Javascript : Products

```

376  $('#btn_insert').click(function () {
377      $.ajax({
378          url: 'ProductInsertDataController',
379          method: 'POST',
380          data: $('#insertForm').serialize(),
381          dataType: 'json',
382          success: function (data) {
383
384              if (data.Insert_Complete === "Complete") {
385                  alert('ເພີ້ມຂໍ້ມູນລົງທຶນຮ່ອງແລ້ວ.');
386                  $('#insertForm')[0].reset();
387              } else if (data.Insert_Complete === "Duplicate Code And Name") {
388                  alert('*** CODE ແລະ NAME ຜິດຂູ້ແລ້ວ !!!');
389              } else if (data.Insert_Complete === "Duplicate Code") {
390                  alert('*** CODE ຜິດຂູ້ແລ້ວ !!!');
391              } else if (data.Insert_Complete === "Duplicate Name") {
392                  alert('*** NAME ຜິດຂູ້ແລ້ວ !!!');
393              } else if (data.INSERT_ALL_DATA === "No Data") {
394                  alert('ກຽດທາກໂຮກຂໍ້ມູນໃຫ້ຄົນກັບແທ້ງໝາດ !!!');
395              }
396
397          });
398      });
399  });

```

บรรทัดที่ 376-399 ทำการรับค่าจาก insert form ส่งไปที่ ProductInsertDataController file ในส่วนของ Servlet

- Servlet file : Products

```

24      HttpSession session = request.getSession(true);
25      LocalDate now = LocalDate.now();
26      PrintWriter out = response.getWriter();
27      JSONObject json = new JSONObject();
28      String code = request.getParameter("code");
29      String name = request.getParameter("name");
30      String price = request.getParameter("price");
31      String imagePart = request.getParameter("imagePart");
32      String image = "image.png";
33      int skills = Integer.parseInt(request.getParameter("skills"));
34      int brand_id = Integer.parseInt(request.getParameter("brand_id"));
35      int guitarType_id = Integer.parseInt(request.getParameter("guitarType_id"));
36      int topWood_id = Integer.parseInt(request.getParameter("topWood_id"));
37      int backAndSideWood_id = Integer.parseInt(request.getParameter("backAndSideWood_id"));
38      int priceRange_id = Integer.parseInt(request.getParameter("priceRange_id"));
39      String description = request.getParameter("description");
40      String createdByUser = (String) session.getAttribute("Username");
41      Date createdDate = Date.valueOf(now);
42

```



```

44     if (code != "" && name != "" && price != "" && imagePart != "" && image != "" && description != "") {
45         ProductDAO DAO = new ProductDAO();
46         int insertComplete;
47         insertComplete = DAO.Insert(code, name, imagePart, image, price, description, skills, brand_id, guitarType_id, topWood_id, backAndSideWood_id);
48
49         if (insertComplete == 999) {
50             json.put("Insert_Complete", "Complete");
51         } else if (insertComplete == 1) {
52             json.put("Insert_Complete", "Duplicate Code And Name");
53         } else if (insertComplete == 2) {
54             json.put("Insert_Complete", "Duplicate Code");
55         } else if (insertComplete == 3) {
56             json.put("Insert_Complete", "Duplicate Name");
57         }
58         json.put("INSERT_ALL_DATA", "Have Data");
59     } else {
60         json.put("INSERT_ALL_DATA", "No Data");
61     }
62     out.print(json);

```

บรรทัดที่ 24-61 รับค่าต่างๆที่ส่งมาจาก web browser และส่งต่อไปที่ Insert Method ใน ProductDAO file

- DAO (Data Access Object) (ส่วน Query) : Products

```

266     public boolean Insert(String Code, String Name, String ImagePart, String Image,
267                           String Price, String Description, int Skills, int Brand_ID, int GuitarType_ID,
268                           int TopWood_ID, int BackAndSideWood_ID, int PriceRange_ID, String CreatedByUser,
269                           Date CreatedDate, String modifiedByUser, Date modifiedDate) throws SQLException {
270         boolean Complete = false;
271         try {
272             Statement ps = connect.createStatement();
273             ResultSet rs = ps.executeQuery("SELECT * FROM products WHERE Code = '" + Code + "' OR Name = '" + Name + "'");
274             int Duplicate = 0;
275             while (rs.next()) {
276                 Duplicate++;
277             }
278             if (Duplicate > 0) {
279                 Complete = false;
280             } else {
281                 PreparedStatement insert_Data = connect.prepareStatement("INSERT INTO products (Code,Name,ImagePart,Image,Price,Description,Skill,Brand_ID,GuitarType_ID,TopWood_ID,BackAndSideWood_ID,PriceRange_ID,CreatedByUser,CreatedDate,ModifiedByUser,ModifiedDate) VALUES ('" + Code + "','" + Name + "','" + ImagePart + "','" + Image + "','" + Price + "','" + Description + "','" + Skills + "','" + Brand_ID + "','" + GuitarType_ID + "','" + TopWood_ID + "','" + BackAndSideWood_ID + "','" + PriceRange_ID + "','" + CreatedByUser + "','" + CreatedDate + "','" + modifiedByUser + "','" + modifiedDate + "')");
282                 insert_Data.executeUpdate();
283                 Complete = true;
284             }
285             connect.close();
286         } catch (Exception e) {
287             e.printStackTrace();
288         }
289         return Complete;
290     }

```

บรรทัดที่ 266-290 รับค่าพารามิเตอร์จาก Servlet file และทำการตรวจสอบความซ้ำซ้อนของ Code และ Name ที่รับมานั้นต้องไม่ซ้ำกับข้อมูลในฐานข้อมูล จากนั้นทำการบันทึก



3. การแก้ไขข้อมูล : Products

- HTML file : Products

```
47 <form class="form" method="POST" id="update-form" name="update-form" >
48     <legend class="text-uppercase"><h5 class=" text-light"> Edit Data </h5></legend>
49
50     <table class="table" style=" width: 100%; background-color: rgba(255,255,255, 0.80);">
51         <tr class="border">
52             <td width="50%">
53
54                 <div class="form-group">
55                     <label class="text-uppercase"><legend><h6> code <span id="sCode"></span></h6></legend></label>
56                     <input style="background-color: rgba(255,255,255, 0.85); " type="text" id="code" name="code" class="form-control text-uppe
57
58
59                 <div class="form-group">
60                     <label class="text-uppercase"><legend><h6> name <span id="sName"></span></h6></legend></label>
61                     <input style="background-color: rgba(255,255,255, 0.85); " type="text" id="name" name="name" class="form-control text-uppe
62
63
64                 <div class="form-group">
65                     <label class="text-uppercase"><legend><h6> price </h6></legend></label>
66                     <input style="background-color: rgba(255,255,255, 0.85); " type="text" id="price" name="price" class="form-control text-uppe
67
68
69                 <div class="form-group">
70                     <label class="text-uppercase"><legend><h6> image part </h6></legend></label>
71                     <input style="background-color: rgba(255,255,255, 0.85); " type="text" id="imagePart" name="imagePart" class="form-control
72
73
74             </div>
75
76             <div class="form-group" width="50%">
77                 <label class="text-uppercase"><legend><h6> image <img height="30px" id="output"/>&ampnbsp<span class="fa fa-picture-o"></span></h6></legend></label>
78                 <input type="text" hidden id="image" name="image">
79                 <input style="background-color: rgba(255,255,255, 0.85); " onchange="loadFile(event)" type="file" id="image" name="image"
80             </div>
81
82             <div class="form-group" width="50%">
83                 <div class="form-group text-uppercase">
84                     <label class="text-uppercase"><legend><h6> Skills </h6></legend></label>
85                     <div style="background-color: rgba(255,255,255, 0.85); " class="checkbox form-control form-inline" required>
86
87                         <%
88                             if (session.getAttribute("skill") == "1") {
89
89                         <input type="radio" name="skills" value="1" checked="checked"> beginner &ampnbsp&ampnbsp&ampnbsp&ampnbsp&ampnbsp&ampnbsp&ampnbsp
90                         <input type="radio" name="skills" value="2" > expert
91
92                         } else {
93
94                         <input type="radio" name="skills" value="1" > beginner &ampnbsp&ampnbsp&ampnbsp&ampnbsp&ampnbsp&ampnbsp&ampnbsp
95                         <input type="radio" name="skills" value="2" checked="checked"> expert
96
97                         }
98
99
100
101
102             <%
103                 request.setAttribute("bName", session.getAttribute("brand_Name"));
104             %>
105             <% List b_List = (List) session.getAttribute("b_List"); %>
106             <div class="form-group" width="50%">
107                 <div class="form-group text-uppercase">
108                     <label class="text-uppercase"><legend><h6> Brand </h6></legend></label>
109                     <select id="brand_id" name="brand_id" style="background-color: rgba(255,255,255, 0.85); " class="form-control text-uppe
110                         <c:forEach items="${b_List}" var="b_List">
111                             <c:if test="${b_List.name == bName }">
112                                 <option style="color: salmon;" value="${b_List.ID}" selected="selected"><c:out value="${b_List.name}"></c:out>
113                             </c:if>
114                             <c:if test="${b_List.name != bName }">
115                                 <option style="color: black;" value="${b_List.ID}"><c:out value="${b_List.name}"></c:out></option>
116                             </c:if>
117                         </c:forEach>
118                     </select>
119                 </div>
120             </div>
121         </td>
122
123             <%
124                 request.setAttribute("gName", session.getAttribute("guitarType_Name"));
125                 List gList = (List) session.getAttribute("gList");
126             %>
127                 <label class="text-uppercase"><legend><h6> top tone wood </h6></legend></label>
```



```
149         <select name="topWood_id" style="background-color: rgba(255,255,255,0.85); class="form-control text-uppercase" requ
150             <c:forEach items="${tList}" var="tList">
151                 <if test="${tList.name == tName}">
152                     <option style="color: salmon;" value="${tList.ID}" selected="selected"><c:out value="${tList.name}"></c:out>
153                 </if>
154                 <if test="${tList.name != tName}">
155                     <option style="color: black;" value="${tList.ID}"><c:out value="${tList.name}"></c:out></option>
156                 </if>
157             </c:forEach>
158         </select>
159     </div>
160 </div>
161 <%
162     request.setAttribute("bNsName", session.getAttribute("backAndSideWood_Name"));
163     List bNsList = (List) session.getAttribute("bNsList");
164 %>
165 <div class="form-group" width="50%">
166     <div class="form-group text-uppercase">
167         <label class="text-uppercase"><legend><h6> Back And Side tone Wood </h6></legend></label>
168         <select name="backAndSideWood_id" style="background-color: rgba(255,255,255,0.85); class="form-control text-uppercase
169             <c:forEach items="${bNsList}" var="bNsList">
170                 <if test="${bNsList.name == bNsName}">
171                     <option style="color: salmon;" value="${bNsList.ID}" selected="selected"><c:out value="${bNsList.name}">
172                 </if>
173                 <if test="${bNsList.name != bNsName}">
174                     <option style="color: black;" value="${bNsList.ID}"><c:out value="${bNsList.name}"></c:out></option>
175                 </if>
176             </c:forEach>
177         </select>
178     </div>
179 </div>
180 <%
181     request.setAttribute("p_Name", session.getAttribute("priceRange_Name"));
182     List p_List = (List) session.getAttribute("p_List");
183 %>
184 <div class="form-group" width="50%">
185     <div class="form-group text-uppercase">
186         <label class="text-uppercase"><legend><h6> Price Range </h6></legend></label>
187         <select name="priceRange_id" style="background-color: rgba(255,255,255,0.85); class="form-control text-uppercase" r
188             <c:forEach items="${p_List}" var="p_List">
189                 <if test="${p_List.name == p_Name}">
190                     <option style="color: salmon;" value="${p_List.ID}" selected="selected"><c:out value="${p_List.name}">
191                 </if>
192                 <if test="${p_List.name != p_Name}">
193                     <option style="color: black;" value="${p_List.ID}"><c:out value="${p_List.name}"></c:out></option>
194                 </if>
195             </c:forEach>
196         </select>
197     </div>
198 </div>
199 <%
200         <textarea id="description" name="description" style="background-color: rgba(255,255,255,0.85); i
201             <%="n" + dName%>
202         </textarea>
203     </div>
204     <div class="form-group">
205         <input class="form-group" type="hidden" id="id" name="id"/>
206     </div>
207     <div class="form-group">
208         <input type="hidden" id="createdByUser" name="createdByUser"/>
209     </div>
210     <div class="form-group">
211         <input type="hidden" id="createdDate" name="createdDate"/>
212     </div>
213     <div class="form-group">
214         <input type="hidden" id="modifiedByUser" name="modifiedByUser"/>
215     </div>
216     <div class="form-group">
217         <input type="hidden" id="modifiedDate" name="modifiedDate"/>
218     </div>
219     <div class="form-group" align="right">
220         <input id="btn_update" name="btn_update" value="SAVE" type="button" class="btn btn-success text-d
221         <button type="reset" value="Reset" class="btn btn-warning text-dark border-light">RESET</button>
222     </div>
223     </td>
224 </tr>
225 </table>
226 </form>
```

บรรทัดที่ 47-230 กรอกข้อมูลที่ต้องการแก้ไข จากนั้นกดปุ่ม SAVE



- Javascript : Products

```

319     $(".edit_data").click(function () {
320         var uid = $(this).attr("id");
321         $.ajax({
322             url: 'ProductFetchDataController',
323             method: 'POST',
324             data: {id: uid},
325             dataType: 'json',
326             success: function (data) {
327                 $('#id').val(data.id);
328                 $('#code').val(data.code);
329                 $('#name').val(data.name);
330                 $('#price').val(data.price);
331                 $('#imagePart').val(data.imagePart);
332                 $('#image').val(data.image);
333                 $('#skills').val(data.skills);
334                 $('#brand_id').val(data.brand_id);
335                 $('#brand_Name').val(data.brand_Name);
336                 $('#guitarType_id').val(data.guitarType_id);
337                 $('#topWood_id').val(data.topWood_id);
338                 $('#backAndSideWood_id').val(data.backAndSideWood_id);
339                 $('#priceRange_id').val(data.priceRange_id);
340                 $('#description').val(data.description);
341                 $('#createdByUser').val(data.createdByUser);
342                 $('#createdDate').val(data.createdDate);
343                 $('#modifiedByUser').val(data.modifiedByUser);
344                 $('#modifiedDate').val(data.modifiedDate);
345                 $('#EditProduct-Modal').modal('show');
346             }
347         });
348     });
349 
```

บรรทัดที่ 319-348 รับค่าข้อมูลที่ต้องการแก้ไขจาก Update form และส่งไปที่

ProductFetchDataController file ในส่วนการทำงานของ Servlet file



- Servlet file : Products

```

27     HttpSession session = request.getSession(true);
28
29     PrintWriter out = response.getWriter();
30     JSONObject json = new JSONObject();
31
32     if (request.getParameter("id") != null) {
33
34         LocalDate now = LocalDate.now();
35
36         if ((request.getParameter("name") != "") && (request.getParameter("code") != "") && (request.getParameter("price") != "") && (request.getParameter("image") != "") && (request.getParameter("skills") != "") && (request.getParameter("brand_id") != "") && (request.getParameter("guitarType_id") != "") && (request.getParameter("topWood_id") != "") && (request.getParameter("backAndSideWood_id") != "") && (request.getParameter("priceRange_id") != "") && (request.getParameter("description") != ""))
37
38             int id = Integer.parseInt(request.getParameter("id"));
39             String code = request.getParameter("code");
40             String name = request.getParameter("name");
41             String price = request.getParameter("price");
42             String imagePart = request.getParameter("imagePart");
43             String image = request.getParameter("image");
44             int skills = Integer.parseInt(request.getParameter("skills"));
45             int brand_id = Integer.parseInt(request.getParameter("brand_id"));
46             int guitarType_id = Integer.parseInt(request.getParameter("guitarType_id"));
47             int topWood_id = Integer.parseInt(request.getParameter("topWood_id"));
48             int backAndSideWood_id = Integer.parseInt(request.getParameter("backAndSideWood_id"));
49             int priceRange_id = Integer.parseInt(request.getParameter("priceRange_id"));
50             String description = request.getParameter("description");
51
52             String createdByUser = request.getParameter("createdByUser");
53             Date createdDate = Date.valueOf(request.getParameter("createdDate"));
54
55             String modifiedByUser = (String) session.getAttribute("Username");
56             Date modifiedDate = Date.valueOf(now);
57
58             json.put("id", id);
59             json.put("code", code);
60             json.put("name", name);
61             json.put("price", price);
62             json.put("imagePart", imagePart);
63             json.put("image", image);
64             json.put("skills", skills);
65             json.put("brand_id", brand_id);
66             json.put("guitarType_id", guitarType_id);
67             json.put("topWood_id", topWood_id);
68             json.put("backAndSideWood_id", backAndSideWood_id);
69             json.put("priceRange_id", priceRange_id);
70             json.put("description", description);
71
72             json.put("createdByUser", createdByUser);
73             json.put("createdDate", createdDate);
74             json.put("modifiedByUser", modifiedByUser);
75             json.put("modifiedDate", modifiedDate);
76
77             ProductDAO dao = new ProductDAO();
78             int Complete = dao.Edit(id, code, name, price, imagePart, image, skills, brand_id, guitarType_id, topWood_id, backAndSideWood_id, priceRange_id, description, createdByUser, createdDate, modifiedByUser, modifiedDate);
79
80             if (Complete == 999) {
81                 json.put("Edit_Complete", "Complete");
82
83             } else if(Complete == 1){
84                 json.put("Edit_Complete", "Duplicate Code And Name");
85
86             }
87             else if(Complete == 2){
88                 json.put("Edit_Complete", "Duplicate Code");
89
90             }
91             else if(Complete == 3){
92                 json.put("Edit_Complete", "Duplicate Name");
93
94             }
95
96             json.put("ALL_DATA", "Have Data");
97
98         } else {
99             json.put("ALL_DATA", "No Data");
100        }
101    }
102}

```

บรรทัดที่ 27-101 รับค่าพารามิเตอร์ที่ส่งมาจาก web browser และทำการส่งต่อไปที่ Edit Method ใน ProductDAO file เพื่อทำการบันทึกการแก้ไขข้อมูล



- DAO (Data Access Object) (ส่วน Query) : Products

```

330     public int Edit(int id, String code, String name, String price, String imagePart, String image,
331                 int skills, int brand_id, int guitarType_id, int topWood_id, int backAndSideWood_id, int priceRange_id, String description,
332                 String createdByUser, Date createdDate, String modifiedByUser, Date modifiedDate) {
333         int Complete = 0;
334         try {
335             Statement ps = connect.createStatement();
336             ResultSet rs = ps.executeQuery("SELECT * FROM products WHERE (id != '" + id + "') AND (Code = '" + code + "' AND Name = '" + name +
337                                         + " AND PriceRange_id = '" + priceRange_id + "' AND Description = '" + description + "' AND CreatedByUser = '" + createdByUser + "' AND ModifiedByUser = '" + modifiedByUser + "' AND CreatedDate = '" + createdDate + "' AND ModifiedDate = '" + modifiedDate + "'");
338             int DuplicateCodeAndName = 0;
339             while (rs.next()) {
340
341                 DuplicateCodeAndName = 1;
342             }
343             int DuplicateCode = 0;
344             if (DuplicateCodeAndName == 0) {
345                 Statement s = connect.createStatement();
346                 ResultSet se = s.executeQuery("SELECT * FROM products WHERE id != '" + id + "' AND Code = '" + code + "'");
347                 while (se.next()) {
348                     DuplicateCode = 2;
349                 }
350             }
351             int DuplicateName = 0;
352             if (DuplicateCodeAndName == 0 && DuplicateCode == 0) {
353                 Statement s = connect.createStatement();
354                 ResultSet se = s.executeQuery("SELECT * FROM products WHERE id != '" + id + "' AND Name = '" + name + "'");
355
356                 while (se.next()) {
357
358                     DuplicateName = 3;
359                 }
360             }
361             if (DuplicateCodeAndName > 0) {
362                 Complete = DuplicateCodeAndName;
363             } else if (DuplicateCode > 0) {
364                 Complete = DuplicateCode;
365             } else if (DuplicateName > 0) {
366                 Complete = DuplicateName;
367             } else {
368                 PreparedStatement update_Data = connect.prepareStatement("UPDATE products SET Code = '" + code + "', Name = '" + name + "', Image =
369                                         + ' WHERE id = '" + id + "'");
370                 update_Data.executeUpdate();
371                 Complete = 999;
372             }
373
374             connect.close();
375         } catch (Exception e) {
376             e.printStackTrace();
377         }
378
379         return Complete;
380     }

```

บรรทัดที่ 330-379 รับค่าพารามิเตอร์ที่ส่งมาจากการ Submit ของ Form แล้วทำการตรวจสอบความซ้ำซ้อนของ Code และ Name ที่รับมานั้นต้องไม่ซ้ำกับข้อมูลในฐานข้อมูล จากนั้นทำการบันทึก

4. การลบข้อมูล : Products

- HTML file : Products

```

206     <button class="btn btn-danger text-white btn-sm fa fa-trash-o delete-data" id="${pList.ID}">
207         <font face="Calibri">&ampnbspDELETE</font>
208     </button>

```

บรรทัดที่ 206-208 เมื่อกดปุ่ม DELETE จะทำการส่ง id ที่เป็น primary key ของแต่ละตัว Product ไป



- Javascript : Products

```

445      $('.delete-data').click(function () {
446
447          var uid = $(this).attr("id");
448          var status = confirm('คุณต้องการลบข้อมูล ใช่หรือไม่?');
449
450          if (status) {
451              $.ajax({
452                  url: 'ProductDeleteDataController',
453                  method: 'POST',
454                  data: {id: uid},
455                  success: function () {
456                      alert('ทำการลบข้อมูลเรียบร้อยแล้ว');
457                      location.reload();
458                  }
459              });
460          }
461
462      });
463

```

บรรทัดที่ 445-462 รับค่า id ที่ส่งมาจากนั้นส่งต่อไปทำการลบที่ ProductDeleteDataController file ในส่วนของ Servlet file

- Servlet file : Products

```

19         if(request.getParameter("id") != null){
20
21             int id = Integer.parseInt(request.getParameter("id"));
22
23             ProductDAO dao = new ProductDAO();
24             dao.DeleteData(id);
25
26         }// end if(request.getParameter("id") != null)

```

บรรทัดที่ 19-26 ทำการรับค่าพารามิเตอร์ id ที่ถูกส่งมา และส่งต่อไปที่ DeleteData Method ในส่วนของ ProductDAO Class



- DAO (Data Access Object) (ส่วน Query) : Products

```

1   public void DeleteData(int id) {
2
3       String sql = "DELETE FROM products WHERE id = '" + id + "' ORDER BY id ASC";
4
5       try {
6
7           PreparedStatement ps = connect.prepareStatement(sql);
8           ps.executeUpdate();
9
10          connect.close();
11
12      } catch (Exception e) {
13          e.printStackTrace();
14      }
15
16  }

```

บรรทัดที่ 445-458 รับค่าพารามิเตอร์ id มาตรวจสอบความตรงกันกับข้อมูลในฐานข้อมูล จากนั้นทำการลบข้อมูล

4.2 ระบบแออนดรอยด์แอพพลิเคชัน (การแนะนำการเลือกซื้อกีต้าร์ : User)

4.2.1 ส่วนของการแนะนำการเลือกซื้อกีต้าร์ ระดับผู้เริ่มต้น(ขั้นตอนการเลือกคุณสมบัติและจัดลำดับความสำคัญ)

- XML file

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:background="@drawable/beginner_activity"
8      tools:context="com.guitarguide.Beginner">
9
10 <Button
11     android:id="@+id/buttonBackHome"
12     android:onClick="backToHomeButtonOnClick"
13     android:layout_width="27dp"
14     android:layout_height="27dp"
15     android:layout_marginEnd="4dp"
16     android:layout_marginStart="8dp"
17     android:layout_marginTop="14dp"
18     android:background="@drawable/back_to_home_button2"
19     app:layout_constraintEnd_toEndOf="parent"
20     app:layout_constraintHorizontal_bias="0.932"
21     app:layout_constraintStart_toStartOf="parent"
22     app:layout_constraintTop_toTopOf="parent"
23     android:layout_marginRight="4dp"
24     android:layout_marginLeft="8dp" />
25

```



```
26 <ImageView  
27     android:id="@+id/imageView_branner"  
28     android:layout_width="384dp"  
29     android:layout_height="57dp"  
30     android:background="@drawable/guitarguide_logo_for_android"  
31     android:contentDescription="TODO"  
32     app:layout_constraintEnd_toEndOf="parent"  
33     app:layout_constraintHorizontal_bias="0.0"  
34     app:layout_constraintStart_toStartOf="parent"  
35     app:layout_constraintTop_toTopOf="parent" />  
36  
37  
38 <Spinner  
39     android:id="@+id/spinnerBrand"  
40     android:layout_width="187dp"  
41     android:layout_height="36dp"  
42     android:layout_marginBottom="8dp"  
43     android:layout_marginEnd="8dp"  
44     android:layout_marginStart="8dp"  
45     android:layout_marginTop="8dp"  
46     app:layout_constraintBottom_toTopOf="@+id/button_search"  
47     app:layout_constraintEnd_toEndOf="parent"  
48     app:layout_constraintHorizontal_bias="0.262"  
49     app:layout_constraintStart_toStartOf="parent"  
50     app:layout_constraintTop_toBottomOf="@+id/imageView_branner"  
51     app:layout_constraintVertical_bias="0.445" />  
52  
53 <Spinner  
54     android:id="@+id/spinnerGuitarType"  
55     android:layout_width="188dp"  
56     android:layout_height="34dp"  
57     android:layout_marginBottom="128dp"  
58     android:layout_marginEnd="8dp"  
59     android:layout_marginStart="8dp"  
60     android:layout_marginTop="8dp"  
61     app:layout_constraintBottom_toTopOf="@+id/button_search"  
62     app:layout_constraintEnd_toEndOf="parent"  
63     app:layout_constraintHorizontal_bias="0.266"  
64     app:layout_constraintStart_toStartOf="parent"  
65     app:layout_constraintTop_toBottomOf="@+id/imageView_branner"  
66     app:layout_constraintVertical_bias="0.972" />  
67  
68 <Spinner  
69     android:id="@+id/spinnerPriceRange"  
70     android:layout_width="188dp"  
71     android:layout_height="33dp"  
72     android:layout_marginEnd="8dp"  
73     android:layout_marginStart="8dp"  
74     android:layout_marginTop="8dp"  
75     app:layout_constraintBottom_toTopOf="@+id/button_search"  
76     app:layout_constraintEnd_toEndOf="parent"  
77     app:layout_constraintHorizontal_bias="0.266"  
78     app:layout_constraintStart_toStartOf="parent"  
79     app:layout_constraintTop_toBottomOf="@+id/imageView_branner"  
80     app:layout_constraintVertical_bias="0.885"  
81     tools:ignore="UnknownId" />
```

Help info



```

82      <Spinner
83          android:id="@+id/spinnerPriorityBrand"
84          android:layout_width="98dp"
85          android:layout_height="35dp"
86          android:layout_marginEnd="16dp"
87          app:layout_constraintBottom_toTopOf="@+id/button_search"
88          app:layout_constraintEnd_toEndOf="parent"
89          app:layout_constraintHorizontal_bias="0.0"
90          app:layout_constraintStart_toEndOf="@+id/spinnerBrand"
91          app:layout_constraintTop_toTopOf="@+id/imageView_branner"
92          app:layout_constraintVertical_bias="0.516" />
93
94
95      <Spinner
96          android:id="@+id/spinnerPriorityGuitarType"
97          android:layout_width="96dp"
98          android:layout_height="34dp"
99          android:layout_marginBottom="8dp"
100         android:layout_marginEnd="8dp"
101         android:layout_marginTop="8dp"
102         app:layout_constraintBottom_toTopOf="@+id/button_search"
103         app:layout_constraintEnd_toEndOf="parent"
104         app:layout_constraintHorizontal_bias="0.0"
105         app:layout_constraintStart_toEndOf="@+id/spinnerGuitarType"
106         app:layout_constraintTop_toTopOf="@+id/imageView_branner"
107         app:layout_constraintVertical_bias="0.706" />
108
109     <Spinner
110         android:id="@+id/spinnerPriorityPriceRange"
111         android:layout_width="97dp"
112         android:layout_height="32dp"
113         android:layout_marginBottom="8dp"
114         android:layout_marginTop="8dp"
115         app:layout_constraintBottom_toTopOf="@+id/button_search"
116         app:layout_constraintEnd_toEndOf="parent"
117         app:layout_constraintHorizontal_bias="0.0"
118         app:layout_constraintStart_toEndOf="@+id/spinnerPriceRange"
119         app:layout_constraintTop_toTopOf="@+id/imageView_branner"
120         app:layout_constraintVertical_bias="0.894" />
121
122     <Button
123         android:id="@+id/button_search"
124         android:layout_width="316dp"
125         android:layout_height="69dp"
126         android:layout_marginBottom="16dp"
127         android:layout_marginEnd="16dp" | highlight
128         android:layout_marginStart="16dp" |
129         android:text="SEARCH"
130         android:textSize="40sp"
131         android:onClick="sendResultDataToShowActivityOnClick"
132         app:layout_constraintBottom_toBottomOf="parent"
133         app:layout_constraintEnd_toEndOf="parent"
134         app:layout_constraintStart_toStartOf="parent" />
135
136
137     </android.support.constraint.ConstraintLayout>

```

บรรทัดที่ 1-137 เป็นการจัดวาง Layout ของหน้าจอ User Interface ผู้เริ่มต้น

บรรทัดที่ 10-24 ปุ่มสำหรับกลับหน้าหลัก

บรรทัดที่ 26-35 รูป Logo ของแอพพลิเคชัน

บรรทัดที่ 38-51 ส่วนแสดง List ข้อมูล Brand ทั้งหมดที่ทักษะเท่ากับผู้เริ่มต้น



บรรทัดที่ 53-66 ส่วนแสดง List ข้อมูล Guitar Type ทั้งหมดที่ทักษะเท่ากับผู้เริ่มต้น

บรรทัดที่ 68-81 ส่วนแสดง List ข้อมูล Price Range ทั้งหมดที่ทักษะเท่ากับผู้เริ่มต้น

บรรทัดที่ 83-93 ส่วนแสดง List ข้อมูล Priority ของ Brand

บรรทัดที่ 95-107 ส่วนแสดง List ข้อมูล Priority ของ Guitar Type

บรรทัดที่ 109-120 ส่วนแสดง List ข้อมูล Priority ของ Price Range

บรรทัดที่ 102-134 ส่วนแสดง ปุ่มการค้นหา

บรรทัดที่ 1-137 เมื่อทำการเลือกความต้องการและกำหนดลำดับความสำคัญแล้ว จากนั้นกดปุ่ม SEARCH ทำการส่งพารามิเตอร์ไปที่ Beginner : Java file เพื่อทำการค้นหาความคล้ายคลึง

- java file

```

67
68 ① ↗
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
626
627
627
628
629
629
630
631
632
633
634
635
635
636
637
637
638
639
639
640
641
642
643
643
644
645
645
646
647
647
648
649
649
650
651
652
653
653
654
655
655
656
657
657
658
659
659
660
661
661
662
663
663
664
665
665
666
667
667
668
669
669
670
671
671
672
673
673
674
675
675
676
677
677
678
679
679
680
681
681
682
683
683
684
685
685
686
687
687
688
689
689
690
691
691
692
693
693
694
695
695
696
697
697
698
699
699
700
701
701
702
703
703
704
705
705
706
707
707
708
709
709
710
711
711
712
713
713
714
715
715
716
717
717
718
719
719
720
721
721
722
723
723
724
725
725
726
727
727
728
729
729
730
731
731
732
733
733
734
735
735
736
737
737
738
739
739
740
741
741
742
743
743
744
745
745
746
747
747
748
749
749
750
751
751
752
753
753
754
755
755
756
757
757
758
759
759
760
761
761
762
763
763
764
765
765
766
767
767
768
769
769
770
771
771
772
773
773
774
775
775
776
777
777
778
779
779
780
781
781
782
783
783
784
785
785
786
787
787
788
789
789
790
791
791
792
793
793
794
795
795
796
797
797
798
799
799
800
801
801
802
803
803
804
805
805
806
807
807
808
809
809
810
811
811
812
813
813
814
815
815
816
817
817
818
819
819
820
821
821
822
823
823
824
825
825
826
827
827
828
829
829
830
831
831
832
833
833
834
835
835
836
837
837
838
839
839
840
841
841
842
843
843
844
845
845
846
847
847
848
849
849
850
851
851
852
853
853
854
855
855
856
857
857
858
859
859
860
861
861
862
863
863
864
865
865
866
867
867
868
869
869
870
871
871
872
873
873
874
875
875
876
877
877
878
879
879
880
881
881
882
883
883
884
885
885
886
887
887
888
889
889
890
891
891
892
893
893
894
895
895
896
897
897
898
899
899
900
901
901
902
903
903
904
905
905
906
907
907
908
909
909
910
911
911
912
913
913
914
915
915
916
917
917
918
919
919
920
921
921
922
923
923
924
925
925
926
927
927
928
929
929
930
931
931
932
933
933
934
935
935
936
937
937
938
939
939
940
941
941
942
943
943
944
945
945
946
947
947
948
949
949
950
951
951
952
953
953
954
955
955
956
957
957
958
959
959
960
961
961
962
963
963
964
965
965
966
967
967
968
969
969
970
971
971
972
973
973
974
975
975
976
977
977
978
979
979
980
981
981
982
983
983
984
985
985
986
987
987
988
989
989
990
991
991
992
993
993
994
995
995
996
997
997
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
16
```



```
236 public class AsyncTaskGetData extends AsyncTask<Void(Void,Void> {
237
238     @Override
239     protected Void doInBackground(Void... voids) {
240
241         Request request = new Request.Builder()
242             .url("http://android/getBeginnerFromServer.php")
243             .build();
244
245         Response response;
246         try{
247             response = okHttpClient.newCall(request).execute();
248             String jsonData = response.body().string();
249
250             JSONArray jsonArray = new JSONArray(jsonData);
251
252             for(int i = 0; i < jsonArray.length(); i++){
253                 JSONObject jsonObject = jsonArray.getJSONObject(i);
254                 BrandList.add(jsonObject.optString("name: "BRAND"));
255
256             }
257             for(int i = 0; i < jsonArray.length(); i++){
258                 JSONObject jsonObject = jsonArray.getJSONObject(i);
259                 GuitarTypeList.add(jsonObject.optString("name: "GUITARTYPE"));
260
261             }
262             for(int i = 0; i < jsonArray.length(); i++){
263                 JSONObject jsonObject = jsonArray.getJSONObject(i);
264                 PriceRangeList.add(jsonObject.optString("name: "PRICERANGE"));
265
266         }
267
268     } catch (IOException e) {
269         e.printStackTrace();
270     } catch (JSONException e) {
271         e.printStackTrace();
272     }
273     return null;
274 }
275
276     @Override
277     protected void onPostExecute(Void avoid){
278         super.onPostExecute(avoid);
279
280         final List<String> listB = new ArrayList<>();
281         final List<String> listGT = new ArrayList<>();
282         final List<String> listPR = new ArrayList<>();
283
284         for(int i = 0; i < BrandList.size(); i++){
285             if(i==0){
286                 listB.add("PLEASE SELECT.");
287             }
288             if(BrandList.get(i) != ""){
289                 listB.add(BrandList.get(i));
290             }
291         }
292
293         for(int i = 0; i < GuitarTypeList.size(); i++){
294             if(i==0){
295                 listGT.add("PLEASE SELECT.");
```

```
296
297     }
298     if(GuitarTypeList.get(i) != ""){
299         listGT.add(GuitarTypeList.get(i));
300     }
301 }
302
303 for(int i = 0; i < PriceRangeList.size(); i++){
304     if(i==0){
305         listPR.add("PLEASE SELECT.");
306     }
307     if(PriceRangeList.get(i) != ""){
308         listPR.add(PriceRangeList.get(i));
309     }
310 }
311
312 //simple_dropdown_item 	inline 	without Items 	normal
313 final ArrayAdapter<String> adapter = new ArrayAdapter<>( context: Beginner.this, android.R.layout.simple_spinner_dropdown_item
314 spinnerBrand.setAdapter(adapter);
315 // setOnItemSelectedListener
316 spinnerBrand.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
317     @Override
318     public void onItemSelected(AdapterView<> adapterView, View view, int i, long l) {
319         if(listB.get(i) == "PLEASE SELECT.") {
320             ChooseBrand = "";
321         }else{
322             ChooseBrand = listB.get(i);
323         }
324     }
325 }
326
327     public void onNothingSelected(AdapterView<> adapterView) {
328
329     }
330 });
331
332 //simple_spinner_dropdown_item 	inline 	without Items 	normal
333 final ArrayAdapter<String> adapter2 = new ArrayAdapter<>( context: Beginner.this, android.R.layout.simple_spinner_dropdown_item
334 spinnerGuitarType.setAdapter(adapter2);
335 // setOnItemSelectedListener
336 spinnerGuitarType.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
337     @Override
338     public void onItemSelected(AdapterView<> adapterView, View view, int i, long l) {
339         if(listGT.get(i) == "PLEASE SELECT.") {
340             ChooseGuitarType = "";
341         }else{
342             ChooseGuitarType = listGT.get(i);
343         }
344     }
345
346     @Override
347     public void onNothingSelected(AdapterView<> adapterView) {
348
349     }
350 });
351
352 //simple_spinner_item 	inline 	without Items 	normal
353 final ArrayAdapter<String> adapter3 = new ArrayAdapter<>( context: Beginner.this, android.R.layout.simple_spinner_dropdown_item
354 spinnerPriceRange.setAdapter(adapter3);
355 // setOnItemSelectedListener
356 spinnerPriceRange.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
357     @Override
358
359     public void onItemSelected(AdapterView<> adapterView, View view, int i, long l) {
360         if(listPR.get(i) == "PLEASE SELECT.") {
361             ChoosePriceRange = "";
362         }else{
363             ChoosePriceRange = listPR.get(i);
364         }
365     }
366
367     @Override
368     public void onNothingSelected(AdapterView<> adapterView) {
369
370     }
371 });
372
373 }
```

บรรทัดที่ 67-373 ทำการ Set ปุ่มต่างๆหน้า Layout และทำการร้องขอข้อมูลจากฐานข้อมูล โดยการ Call web service เมื่อได้ Brand,GuitarType,PriceRange มาแล้วทำการแสดงใน List ที่ได้จัดเตรียมไว้ที่หน้า Layout



-PHP call web service file

```

1 <?php
2
3     $HOST = 'localhost';
4     $USER = 'root';
5     $PASS = '';
6     $DBname = 'guitarguide';
7
8     ////////////// QUERY BRAND_ID FROM PRODUCTS /////////////
9     $conn = mysqli_connect($HOST,$USER,$PASS,$DBname) or die('error connecting to database');
10
11    $strBrandSQL = "SELECT Brand_ID FROM products WHERE Skills = 1";
12    $result = $conn->query($strBrandSQL) or die($conn->error);
13    $resultBrandArray = array();
14    while($row = $result->fetch_assoc()){
15        $resultBrandArray[] = $row;
16    }
17    $resultBrandArray = array_map("unserialize",array_unique(array_map("serialize",$resultBrandArray) ) );
18    sort($resultBrandArray);
19
20
21    ////////////// QUERY GUITARTYPE_ID FROM PRODUCTS ///////////
22    $strGSQL = "SELECT GuitarType_ID FROM products WHERE Skills = 1";
23    $resultGT = $conn->query($strGSQL) or die($conn->error);
24    $resultGTArray = array();
25    while($row = $resultGT->fetch_assoc()){
26        $resultGTArray[] = $row;
27    }
28    $resultGTArray = array_map("unserialize",array_unique(array_map("serialize",$resultGTArray) ) );
29    sort($resultGTArray);
30
31
32    ////////////// QUERY PRICERANGE_ID FROM PRODUCTS ///////////
33    $strPRSQL = "SELECT PriceRange_ID FROM products WHERE Skills = 1";
34    $resultPR = $conn->query($strPRSQL) or die($conn->error);
35    $resultPRArray = array();
36    while($row = $resultPR->fetch_assoc()){
37        $resultPRArray[] = $row;
38    }
39    $resultPRArray = array_map("unserialize",array_unique(array_map("serialize",$resultPRArray) ) );
40    sort($resultPRArray);
41
42
43    //***** CHECK BRAND_ID FROM PRODUCTS QUERY WITH BRANDS *****/
44    $Brand = "SELECT * FROM brands";
45    $reBrand = $conn->query($Brand) or die($conn->error);
46    $reBrandArray = array();
47    while($row = $reBrand->fetch_assoc()){
48        $reBrandArray[] = $row;
49    }
50    $brands = array();
51    $i = 0;

```



```

51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100 ?>

$brands = array();
foreach($resultBrandArray as $brandIDProduct) { //query from products
    foreach($reBrandArray as $brandsID) {
        if($brandIDProduct["Brand_ID"] === $brandsID["ID"]){
            $brands[$i] = array("BRAND"=>$brandsID["Name"]);
        }
    }
    $i++;
}

/*
***** CHECK GUITARTYPE_ID FROM PRODUCTS QUERY WITH GUITARTYPES *****/
$GuitarType = "SELECT * FROM guitartypes";
$reGuitarType = $conn->query($GuitarType) or die($conn->error);
$reGuitarTypeArray = array();
while($row = $reGuitarType->fetch_assoc()) {
    $reGuitarTypeArray[] = $row;
}
$guitarTypes = array();
$i = 0;
foreach($resultGTAarray as $guitarTypeIDProduct) { //query from products
    foreach($reGuitarTypeArray as $guitarTypeID) {
        if($guitarTypeIDProduct["GuitarType_ID"] === $guitarTypeID["ID"]){
            $guitarTypes[$i] = array("GUITARTYPE"=>$guitarTypeID["Name"]);
        }
    }
    $i++;
}

/*
***** CHECK PRICERANGE_ID FROM PRODUCTS QUERY WITH PRICERANGES *****/
$PriceRange = "SELECT * FROM priceranges";
$rePriceRange = $conn->query($PriceRange) or die($conn->error);
$rePriceRangeArray = array();
while($row = $rePriceRange->fetch_assoc()) {
    $rePriceRangeArray[] = $row;
}
$priceRanges = array();
$i = 0;
foreach($resultPRArray as $PriceRangeIDProduct) { //query from products
    foreach($rePriceRangeArray as $PriceRangeID) {
        if($PriceRangeIDProduct["PriceRange_ID"] === $PriceRangeID["ID"]){
            $priceRanges[$i] = array("PRICERANGE"=>$PriceRangeID["Name"]);
        }
    }
    $i++;
}

/*
***** BEGINNER *****/
$Beginner = array_merge($brands, $guitarTypes, $priceRanges);
$conn->close();
echo json_encode($Beginner);

```

บรรทัดที่ 1-100 ทำการ Query ฐานข้อมูล Products เพื่อต้องการข้อมูล Brand,GuitarType,PriceRange ที่มี Skills = 1 เท่านั้น (ระดับผู้เริ่มต้น) และทำการ Return : JSON file กลับไปยังตัวแอพพลิเคชันเพื่อนำข้อมูลดังกล่าวไปแสดงใน List



4.2.1 ส่วนของการแนะนำการเลือกซื้อกีต้าร์ ระดับผู้เริ่มต้น (ขั้นตอนการแสดงผลการวัดความคล้าย)

- XML file

```

8     <LinearLayout
9         android:layout_width="match_parent"
10        android:layout_height="84dp"
11        android:layout_marginBottom="5dp"
12        android:layout_marginEnd="5dp"
13        android:layout_marginStart="5dp"
14        android:layout_marginTop="5dp" style="background-color: #99404040; color: white;">
15    android:background="#99404040"
16    android:orientation="horizontal"
17    app:layout_constraintBottom_toBottomOf="parent"
18    app:layout_constraintEnd_toEndOf="parent"
19    app:layout_constraintHorizontal_bias="0.0"
20    app:layout_constraintStart_toStartOf="parent"
21    app:layout_constraintTop_toTopOf="parent"
22    app:layout_constraintVertical_bias="0.0">
23
24    <ImageView
25        android:id="@+id/list_item_img"
26        android:layout_width="103dp"
27        android:layout_height="62dp"
28        android:layout_margin="10dp"
29        android:adjustViewBounds="true"
30        app:srcCompat="@mipmap/ic_launcher" />
31
32    <LinearLayout
33        android:layout_width="match_parent"
34        android:layout_height="wrap_content"
35        android:orientation="vertical">
36
37        <LinearLayout
38            android:layout_width="match_parent"
39            android:layout_height="wrap_content"
40            android:layout_margin="10dp"
41            android:gravity="center_horizontal"
42            android:orientation="vertical">
43
44            <TextView
45                android:id="@+id/list_item_similarity"
46                android:layout_width="match_parent"
47                android:layout_height="wrap_content"
48                android:layout_margin="2dp"
49                android:layout_weight="1"
50                android:gravity="left|center_vertical"
51                android:text="100.0"
52                android:textColor="@android:color/background_light"
53                android:textSize="12sp" />
54
55            <TextView
56                android:id="@+id/list_item_name"
57                android:layout_width="match_parent"
58                android:layout_height="wrap_content"
59                android:layout_margin="2dp"
60                android:layout_weight="1"
61                android:ellipsize="middle"
62                android:gravity="left|center_vertical"
63                android:singleLine="true"
64                android:text="nameXXX"
65                android:textColor="@android:color/background_light"
66                android:textSize="14sp" />
67
68            <TextView
69                android:id="@+id/list_item_price"
70                android:layout_width="match_parent"
71                android:layout_height="wrap_content"
72                android:layout_margin="2dp"
73                android:layout_weight="1"
74                android:gravity="left|center_vertical"
75                android:text="pricexxx bath"
76                android:textColor="@android:color/background_light"
77                android:textSize="14sp" />
78
79        </LinearLayout>
80    </LinearLayout>
81 </LinearLayout>

```

บรรทัดที่ 24-30 Layout ส่วนแสดงรูปภาพที่ผ่านการคำนวณความคล้าย



บรรทัดที่ 45-54 Layout ส่วนแสดงผลลัพธ์ที่ผ่านการคำนวณความคล้าย

บรรทัดที่ 56-67 Layout ส่วนแสดงข้อมูลที่ผ่านการคำนวณความคล้าย

บรรทัดที่ 69-78 Layout ส่วนแสดงราคาที่ผ่านการคำนวณความคล้าย

- java file

```

115 public class AsyncTaskGetDataBeginner extends AsyncTask<Void, Void, Void> {
116     String similarity;
117     String name;
118     String image_part;
119     String image;
120     String price;
121     String web_url;
122     Bitmap bitmap;
123
124     @Override
125     protected Void doInBackground(Void... voids) {
126         Request request = new Request.Builder()
127             .url(hTTp + "android/getSimilarityBeginnerResult.php?Brand=" + brand + "&GuitarType=" + guitartype + "&PriceRange="
128             .build();
129
130         Response response;
131         try {
132             response = okHttpClient.newCall(request).execute();
133             String jsonData = response.body().string();
134
135             JSONArray jsonArray = new JSONArray(jsonData);
136
137             for (int i = 0; i < jsonArray.length(); i++) {
138                 JSONObject jsonObject = jsonArray.optJSONObject(i);
139
140                 similarity = jsonObject.optString( name: "SIMILARITY");
141                 name = jsonObject.optString( name: "NAME");
142                 image_part = jsonObject.optString( name: "IMAGE_PART");
143                 image = jsonObject.optString( name: "IMAGE");
144                 price = jsonObject.optString( name: "PRICE");
145
146                 web_url = jsonObject.optString( name: "WEB_URL");
147                 similarity = String.valueOf(Float.parseFloat(similarity) * 100);
148                 BeginnerModel beginnerModel = new BeginnerModel(similarity, name, image_part, image, price, web_url);
149                 //beginnerModel.compareSimilarity(Float.parseFloat(similarity));
150                 //if(Float.parseFloat(similarity) >= 0.5) {
151                 BeginnerList.add(beginnerModel);
152                 //}
153             }
154         } catch (JSONException e) {
155             e.printStackTrace();
156         } catch (IOException e) {
157             e.printStackTrace();
158         }
159         return null;
160     }
161
162     @Override
163     protected void onPostExecute(Void avoid) {
164     }
165
166     public class ListAdapter extends BaseAdapter {
167
168         private java.util.List<BeginnerModel> List = new ArrayList<>();
169         private Context context;
170
171         ListAdapter(Context context, List<BeginnerModel> BeginnerList) {
172             this.context = context;
173             this.List = BeginnerList;
174         }
175     }

```



```

176
177     @Override
178     public int getCount() {
179         //int temp = List.size();
180         return 10;
181     }
182
183     @Override
184     public Object getItem(int i) { return null; }
185
186     @Override
187     public long getItemId(int i) { return 0; }
188
189     @Override
190     public View getView(int i, View view, ViewGroup viewGroup) {
191         View viewl;
192         LayoutInflator inflater = (LayoutInflator) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
193         viewl = inflater.inflate(R.layout.activity_show_similarity_list_item, viewGroup, attachToRoot: false);
194
195         ImageView imageView = (ImageView) viewl.findViewById(R.id.list_item_img);
196         TextView tv1 = (TextView) viewl.findViewById(R.id.list_item_similarity);
197         TextView tv2 = (TextView) viewl.findViewById(R.id.list_item_name);
198         TextView tv3 = (TextView) viewl.findViewById(R.id.list_item_price);
199         TextView tv4 = (TextView) viewl.findViewById(R.id.list_item_url);
200
201         String url = hTTp + List.get(i).getImage();
202         Glide.with(ShowDataBeginnerActivity.this)
203             .load(url)
204             .crossFade()
205             .into(imageView);
206
207         tv1.setText("SIMILARITY RESULTS : " + List.get(i).getSimilarity() + "%");
208
209         tv2.setText("NAME : " + List.get(i).getName());
210         tv3.setText("PRICE : " + List.get(i).getPrice());
211         tv4.setText("URL : "+List.get(i).getWeb_url());
212
213         return viewl;
214     }
215 }
216
217 }
218
219 }
```

บรรทัดที่ 116-219 ทำการรับค่าพารามิเตอร์ที่ส่งมาจากการเลือกคุณสมบัติและลำดับความสำคัญ จากนั้นทำการ call web service เพื่อนำคุณสมบัติและลำดับความสำคัญไปทำการตรวจสอบกับข้อมูลในฐานข้อมูล และคำนวณความคล้ายคลึง แล้วทำการ Return : JSON file กลับมาแสดงใน List

-PHP call web service file

```

1 <?php
2
3     $HOST = 'localhost';
4     $USER = 'root';
5     $PASS = '';
6     $DBname = 'guitarguide';
7
8     $conn = mysqli_connect($HOST,$USER,$PASS,$DBname) or die('error connecting to database');
9
10    $Brand = $_GET['Brand'];//""Fender";
11    $GuitarType = $_GET['GuitarType'];//""Acoustic Guitar";
12    $PriceRange = $_GET['PriceRange'];//""10,000-30,000 bath";
13
14    $strResultSQL = "SELECT pro.Name AS pName, pro.ImagePart AS pImP, pro.Image AS pIm, pro.Price AS pPrice, pro.Description
15                           FROM products pro
16                           LEFT JOIN brands bra ON bra.ID = pro.Brand_ID
17                           LEFT JOIN guitaratypes gui ON gui.ID = pro.GuitarType_ID
18                           LEFT JOIN priceranges pr ON pr.ID = pro.PriceRange_ID
19                           WHERE pro.Skills = 1
20                           ";
21
22    $result = $conn->query($strResultSQL) or die($conn->error);
23    $resultArray = array();
24    while($row = $result->fetch_assoc()){
25        $resultArray[] = $row;
26    }

```



```

27
28     $priorityWieghtBrand = (float) $_GET['Bbrand'];//"0.4";//$_POST['Bbrand'];
29     $priorityWieghtGuitarType = (float) $_GET['Rguitartype'];//"0.35";//$_POST['Rguitartype'];
30     $priorityWieghtPriceRange = (float) $_GET['Pricerange'];//"0.25";//$_POST['Pricerange'];
31
32     $similarityBrand = 0.0;
33     $similarityGuitarType = 0.0;
34     $similarityPriceRange = 0.0;
35
36     $CALCULATE_SIMILARITY = 0.0;
37     $RESULT_SIMILARITY = array();
38     $i = 0;
39     foreach($resultArray as $result){
40         if($result["bName"] == $Brand){
41             $similarityBrand = 1.0;
42         }else{
43             $similarityBrand = 0.0;
44         }///////////
45
46         if($result["gName"] == $GuitarType){
47             $similarityGuitarType = 1.0;
48         }else{
49             $similarityGuitarType = 0.0;
50         }///////////
51
52         if($result["prName"] == $PriceRange){
53             $similarityPriceRange = 1.0;
54         }else{
55             $similarityPriceRange = 0.0;
56         }///////////
57
58         $CALCULATE_SIMILARITY = ($priorityWieghtBrand*$similarityBrand)+($priorityWieghtGuitarType*$similarityGuitarType)
59                         +($priorityWieghtPriceRange*$similarityPriceRange) ;
60         if($CALCULATE_SIMILARITY != 0.0){
61             $RESULT_SIMILARITY[$i] = array( "SIMILARITY" => (string)$CALCULATE_SIMILARITY,
62                                         "NAME"      => $result["pName"],
63                                         "IMAGE_PART" => $result["pImg"],
64                                         "IMAGE"      => $result["plm"],
65                                         "PRICE"      => $result["pPrice"],
66                                         "WEB_URL"   => $result["pURL"]);
67         }
68         $i++;
69     }
70     $conn->close();
71     rsort($RESULT_SIMILARITY);
72     echo json_encode($RESULT_SIMILARITY);
73
74
75 ?>
```

บรรทัดที่ 1-75 ทำการ Query ฐานข้อมูล Products เพื่อต้องการข้อมูล เพื่อนำข้อมูลมาเปรียบเทียบความคล้ายให้กับแต่ละตัวและที่ถูกส่งมาจากแอปพลิเคชันจากนั้นทำการคำนวณด้วย Algorithm ความคล้าย แล้วทำการ Return : JSON file กลับไปยังตัวแอปพลิเคชันเพื่อนำข้อมูลดังกล่าวไปแสดงใน List
 บรรทัดที่ 58-59 ทำการคำนวณความคล้าย



4.2.2 ส่วนของการแนะนำการเลือกซื้อกีต้าร์ ระดับผู้เชี่ยวชาญ(ขั้นตอนการเลือกคุณสมบัติ และจัดลำดับความสำคัญ)

- XML file

```

1 <?xml version="1.0" encoding="utf-8"?>
2   <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="@drawable/expert_activity"
8     tools:context="com.guitarguide.Expert">
9
10
11   <ImageView
12     android:id="@+id/imageView_branner"
13     android:layout_width="384dp"
14     android:layout_height="57dp"
15     android:background="@drawable/guitarguide_logo_for_android"
16     android:contentDescription="TODO"
17     app:layout_constraintEnd_toEndOf="parent"
18     app:layout_constraintStart_toStartOf="parent"
19     app:layout_constraintTop_toTopOf="parent" />
20
21   <Button
22     android:id="@+id/button_search"
23     android:layout_width="316dp"
24     android:layout_height="69dp"
25     android:layout_marginBottom="16dp"
26     android:layout_marginEnd="16dp"
27     android:layout_marginStart="16dp"
28     android:onClick="sendResultDataToShowActivityOnClick"
29     android:text="SEARCH"
30
31     android:textSize="40sp"
32     app:layout_constraintBottom_toBottomOf="parent"
33     app:layout_constraintEnd_toEndOf="parent"
34     app:layout_constraintStart_toStartOf="parent" />
35
36   <Spinner
37     android:id="@+id/spinnerBrand"
38     android:layout_width="198dp"
39     android:layout_height="36dp"
40     android:layout_marginBottom="8dp"
41     android:layout_marginEnd="8dp"
42     android:layout_marginStart="8dp"
43     app:layout_constraintBottom_toTopOf="@+id/button_search"
44     app:layout_constraintEnd_toEndOf="parent"
45     app:layout_constraintHorizontal_bias="0.251"
46     app:layout_constraintStart_toStartOf="parent"
47     app:layout_constraintTop_toTopOf="@+id/imageView_branner"
48     app:layout_constraintVertical_bias="0.383" />
49
50   <Spinner
51     android:id="@+id/spinnerGuitarType"
52     android:layout_width="196dp"
53     android:layout_height="36dp"
54     android:layout_marginBottom="8dp"
55     android:layout_marginEnd="8dp"
56     android:layout_marginStart="8dp"
57     android:layout_marginTop="8dp"
58     app:layout_constraintBottom_toTopOf="@+id/button_search"

```



```

58      app:layout_constraintEnd_toEndOf="parent"
59      app:layout_constraintHorizontal_bias="0.261"
60      app:layout_constraintStart_toStartOf="parent"
61      app:layout_constraintTop_toTopOf="@+id/imageView_banner"
62      app:layout_constraintVertical_bias="0.511" />
63
64  <Spinner
65      android:id="@+id/spinnerTopWood"
66      android:layout_width="192dp"
67      android:layout_height="36dp"
68      android:layout_marginBottom="8dp"
69      android:layout_marginEnd="8dp"
70      android:layout_marginStart="8dp"
71      android:layout_marginTop="8dp"
72      app:layout_constraintBottom_toTopOf="@+id/button_search"
73      app:layout_constraintEnd_toEndOf="parent"
74      app:layout_constraintHorizontal_bias="0.269"
75      app:layout_constraintStart_toStartOf="parent"
76      app:layout_constraintTop_toTopOf="@+id/imageView_banner"
77      app:layout_constraintVertical_bias="0.648" />
78
79  <Spinner
80      android:id="@+id/spinnerBackSideWood"
81      android:layout_width="192dp"
82      android:layout_height="37dp"
83      android:layout_marginBottom="8dp"
84      android:layout_marginEnd="8dp"
85      android:layout_marginStart="8dp"
86      android:layout_marginTop="8dp"
87
88      app:layout_constraintBottom_toTopOf="@+id/button_search"
89      app:layout_constraintEnd_toEndOf="parent"
90      app:layout_constraintHorizontal_bias="0.274"
91      app:layout_constraintStart_toStartOf="parent"
92      app:layout_constraintTop_toTopOf="@+id/imageView_banner"
93      app:layout_constraintVertical_bias="0.794" />
94
95  <Spinner
96      android:id="@+id/spinnerPriceRange"
97      android:layout_width="190dp"
98      android:layout_height="35dp"
99      android:layout_marginBottom="8dp"
100     android:layout_marginEnd="8dp"
101     android:layout_marginStart="8dp"
102     android:layout_marginTop="8dp"
103     app:layout_constraintBottom_toTopOf="@+id/button_search"
104     app:layout_constraintEnd_toEndOf="parent"
105     app:layout_constraintHorizontal_bias="0.279"
106     app:layout_constraintStart_toStartOf="parent"
107     app:layout_constraintTop_toTopOf="@+id/imageView_banner"
108     app:layout_constraintVertical_bias="0.937" />
109
110 <Spinner
111     android:id="@+id/spinnerPriorityBrand"
112     android:layout_width="96dp"
113     android:layout_height="35dp"
114     android:layout_marginEnd="16dp"
115     app:layout_constraintBottom_toTopOf="@+id/button_search"
116     app:layout_constraintEnd_toEndOf="parent"
117
118     app:layout_constraintHorizontal_bias="0.0"
119     app:layout_constraintStart_toEndOf="@+id/spinnerBrand"
120     app:layout_constraintTop_toTopOf="@+id/imageView_banner"
121     app:layout_constraintVertical_bias="0.375"
122     android:layout_marginRight="16dp" />
123
124  <Spinner
125      android:id="@+id/spinnerPriorityGuitarType"
126      android:layout_width="95dp"
127      android:layout_height="34dp"
128      android:layout_marginBottom="8dp"
129      android:layout_marginEnd="8dp"
130      android:layout_marginStart="8dp"
131      android:layout_marginTop="8dp"
132      app:layout_constraintBottom_toTopOf="@+id/button_search"
133      app:layout_constraintEnd_toEndOf="parent"
134      app:layout_constraintHorizontal_bias="0.0"
135      app:layout_constraintStart_toEndOf="@+id/spinnerGuitarType"
136      app:layout_constraintTop_toTopOf="@+id/imageView_banner"
137      app:layout_constraintVertical_bias="0.509"
138      android:layout_marginRight="8dp" />
139
140  <Spinner
141      android:id="@+id/spinnerPriorityTopWood"
142      android:layout_width="97dp"
143      android:layout_height="40dp"
144      android:layout_marginBottom="8dp"
145      android:layout_marginEnd="8dp"
146      android:layout_marginStart="8dp"
147      app:layout_constraintBottom_toBottomOf="parent"

```



```

144     app:layout_constraintBottom_toBottomOf="parent"
145     app:layout_constraintEnd_toEndOf="parent"
146     app:layout_constraintHorizontal_bias="0.0"
147     app:layout_constraintStart_toEndOf="@+id/spinnerTopWood"
148     app:layout_constraintTop_toTopOf="parent"
149     app:layout_constraintVertical_bias="0.545"
150     android:layout_marginRight="8dp" />
151
152     <Spinner
153         android:id="@+id/spinnerPriorityBackSideWood"
154         android:layout_width="96dp"
155         android:layout_height="36dp"
156         android:layout_marginBottom="8dp"
157         android:layout_marginEnd="8dp"
158         android:layout_marginTop="8dp"
159         app:layout_constraintBottom_toBottomOf="parent"
160         app:layout_constraintEnd_toEndOf="parent"
161         app:layout_constraintHorizontal_bias="0.0"
162         app:layout_constraintStart_toEndOf="@+id/spinnerBackSideWood"
163         app:layout_constraintTop_toTopOf="parent"
164         app:layout_constraintVertical_bias="0.671"
165         android:layout_marginRight="8dp" />
166
167     <Spinner
168         android:id="@+id/spinnerPriorityPriceRange"
169         android:layout_width="95dp"
170         android:layout_height="32dp"
171         android:layout_marginBottom="8dp"
172         android:layout_marginEnd="8dp"
173
174         android:layout_marginTop="8dp"
175         app:layout_constraintBottom_toTopOf="@+id/button_search"
176         app:layout_constraintEnd_toEndOf="parent"
177         app:layout_constraintHorizontal_bias="0.026"
178         app:layout_constraintStart_toEndOf="@+id/spinnerPriceRange"
179         app:layout_constraintTop_toTopOf="@+id/imageView_banner"
180         app:layout_constraintVertical_bias="0.93"
181         android:layout_marginRight="8dp" />
182
183     <Button
184         android:id="@+id/buttonBackHome"
185         android:layout_width="27dp"
186         android:layout_height="27dp"
187         android:layout_marginEnd="4dp"
188         android:layout_marginStart="8dp"
189         android:layout_marginTop="14dp"
190         android:background="@drawable/back_to_home_button2"
191         android:onClick="BackToHomeButtonOnClick"
192         app:layout_constraintEnd_toEndOf="parent"
193         app:layout_constraintHorizontal_bias="0.932"
194         app:layout_constraintStart_toStartOf="parent"
195         app:layout_constraintTop_toTopOf="parent"
196         android:layout_marginRight="4dp"
197         android:layout_marginLeft="8dp" />
198
199
200 </android.support.constraint.ConstraintLayout>

```

บรรทัดที่ 1-200 เป็นการจัดวาง Layout ของหน้าจอ User Interface ผู้เชี่ยวชาญ

บรรทัดที่ 11-19 รูป Logo ของแอพพลิเคชัน

บรรทัดที่ 21-33 ส่วนแสดง บุํมการค้นหา

บรรทัดที่ 35-47 ส่วนแสดง List ข้อมูล Brand ทั้งหมดที่ทักษะเท่ากับผู้เชี่ยวชาญ

บรรทัดที่ 49-62 ส่วนแสดง List ข้อมูล Guitar Type ทั้งหมดที่ทักษะเท่ากับผู้เชี่ยวชาญ

บรรทัดที่ 64-77 ส่วนแสดง List ข้อมูล Top Tone Wood ทั้งหมดที่ทักษะเท่ากับผู้เชี่ยวชาญ

บรรทัดที่ 79-92 ส่วนแสดง List ข้อมูล Back&Side Tone Wood ทั้งหมดที่ทักษะเท่ากับผู้เชี่ยวชาญ

บรรทัดที่ 94-107 ส่วนแสดง List ข้อมูล Price Range ทั้งหมดที่ทักษะเท่ากับผู้เชี่ยวชาญ



บรรทัดที่ 109-120 ส่วนแสดง List ข้อมูล Priority ของ Brand

บรรทัดที่ 122-135 ส่วนแสดง List ข้อมูล Priority ของ Guitar Type

บรรทัดที่ 137-150 ส่วนแสดง List ข้อมูล Priority ของ Top Tone Wood

บรรทัดที่ 152-165 ส่วนแสดง List ข้อมูล Priority ของ Back&Side Tone Wood

บรรทัดที่ 167-180 ส่วนแสดง List ข้อมูล Priority ของ Price Range

บรรทัดที่ 183-197 ปุ่มกลับหน้าหลัก

บรรทัดที่ 1-200 เมื่อทำการเลือกความต้องการและกำหนดลำดับความสำคัญแล้ว จากนั้นกดปุ่ม SEARCH ทำการส่งพารามิเตอร์ไปที่ Expert : Java file เพื่อทำการค้นหาความคล้ายคลึง

- java file

```

314
315
316
317 ①
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342

    public class AsyncTaskGetData extends AsyncTask<Void,Void(Void> {
        @Override
        protected Void doInBackground(Void... voids) {
            Request request = new Request.Builder()
                .url("HTTP+" "android/getExpertFromSever.php")
                .build();

            Response response;
            try{
                response = okHttpClient.newCall(request).execute();
                String jsonData = response.body().string();

                JSONArray jsonArray = new JSONArray(jsonData);

                for(int i = 0; i < jsonArray.length(); i++){
                    JSONObject jsonObject = jsonArray.optJSONObject(i);
                    BrandList.add(jsonObject.optString( name: "BRAND"));

                }
                for(int i = 0; i < jsonArray.length(); i++){
                    JSONObject jsonObject = jsonArray.optJSONObject(i);
                    GuitarTypeList.add(jsonObject.optString( name: "GUITARTYPE"));

                }
                for(int i = 0; i < jsonArray.length(); i++){
                    JSONObject jsonObject = jsonArray.optJSONObject(i);
                    TopWoodList.add(jsonObject.optString( name: "TOPTONEWOOD"));

                }
            }
        }
    }
}

```

```
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373

    }
    for(int i = 0; i < jsonArray.length(); i++){
        JSONObject jsonObject = jsonArray.optJSONObject(i);
        BackSideWoodList.add(jsonObject.optString( name: "BACKSIDEWOOD" ));
    }
    for(int i = 0; i < jsonArray.length(); i++){
        JSONObject jsonObject = jsonArray.optJSONObject(i);
        PriceRangeList.add(jsonObject.optString( name: "PRICERANGE" ));
    }
}
} catch (IOException e) {
    e.printStackTrace();
} catch (JSONException e) {
    e.printStackTrace();
}
return null;
}

@Override
protected void onPostExecute(Void avoid){
    super.onPostExecute(avoid);

    final List<String> listB = new ArrayList<>();
    final List<String> listGT = new ArrayList<>();
    final List<String> listTOP = new ArrayList<>();
    final List<String> listBS = new ArrayList<>();
    final List<String> listPR = new ArrayList<>();

    for(int i = 0; i < BrandList.size(); i++){
        if(i==0){
            listB.add("PLEASE SELECT.");
        }
        if(BrandList.get(i) != ""){
            listB.add(BrandList.get(i));
        }
    }
    for(int i = 0; i < GuitarTypeList.size(); i++){
        if(i==0){
            listGT.add("PLEASE SELECT.");
        }
        if(GuitarTypeList.get(i) != ""){
            listGT.add(GuitarTypeList.get(i));
        }
    }
    for(int i = 0; i < TopWoodList.size(); i++){
        if(i==0){
            listTOP.add("PLEASE SELECT.");
        }
        if(TopWoodList.get(i) != ""){
            listTOP.add(TopWoodList.get(i));
        }
    }
    for(int i = 0; i < BackSideWoodList.size(); i++){
        if(i==0){
            listBS.add("PLEASE SELECT.");
        }
        if(BackSideWoodList.get(i) != ""){
            listBS.add(BackSideWoodList.get(i));
        }
    }
}
```



```

405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434

        }

    for(int i = 0; i < PriceRangeList.size(); i++){
        if(i==0){
            listPR.add("PLEASE SELECT.");
        }
        if(PriceRangeList.get(i) != ""){
            listPR.add(PriceRangeList.get(i));
        }
    }

    final ArrayAdapter<String> adapter = new ArrayAdapter<>( context: Expert.this, android.R.layout.simple_spinner_dropdown_item,
    spinnerBrand.setAdapter(adapter);
    // setOnItemSelectedListener

    spinnerBrand.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener(){
        @Override
        public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
            if(listB.get(i) == "PLEASE SELECT.") {
                ChooseBrand = "";
            }else{
                ChooseBrand = listB.get(i);
            }
        }

        @Override
        public void onNothingSelected(AdapterView<?> adapterView) {
        }
    });

436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466

        final ArrayAdapter<String> adapter2 = new ArrayAdapter<>( context: Expert.this, android.R.layout.simple_spinner_dropdown_item
    spinnerGuitarType.setAdapter(adapter2);
    // setOnItemSelectedListener
    spinnerGuitarType.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener(){
        @Override
        public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
            if(listGT.get(i) == "PLEASE SELECT.") {
                ChooseGuitarType = "";
            }else{
                ChooseGuitarType = listGT.get(i);
            }
        }

        @Override
        public void onNothingSelected(AdapterView<?> adapterView) {
        }
    });

        final ArrayAdapter<String> adapter4 = new ArrayAdapter<>( context: Expert.this, android.R.layout.simple_spinner_dropdown_item
    spinnerTopWood.setAdapter(adapter4);
    // setOnItemSelectedListener
    spinnerTopWood.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener(){
        @Override
        public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
            if(listTOP.get(i) == "PLEASE SELECT.") {
                ChooseTopWood = "";
            }else{
                ChooseTopWood = listTOP.get(i);
            }
        }

        @Override
        public void onNothingSelected(AdapterView<?> adapterView) {
        }
    });

```



```

467
468
469
470
471     }
472     }
473     }
474     }
475
476
477     final ArrayAdapter<String> adapter5 = new ArrayAdapter<->( context: Expert.this, android.R.layout.simple_spinner_dropdown_item
478     spinnerBackSideWood.setAdapter(adapter5);
479     // setOnItemSelectedListener
480     spinnerBackSideWood.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
481         @Override
482         public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
483             if(listBS.get(i) == "PLEASE SELECT.") {
484                 ChooseBackSideWood = "";
485             }else{
486                 ChooseBackSideWood = listBS.get(i);
487             }
488         }
489
490         @Override
491         public void onNothingSelected(AdapterView<?> adapterView) {
492
493     });
494
495
496
497     final ArrayAdapter<String> adapter3 = new ArrayAdapter<->( context: Expert.this, android.R.layout.simple_spinner_dropdown_item
498
499
500
501     spinnerPriceRange.setAdapter(adapter3);
502     // setOnItemSelectedListener
503     spinnerPriceRange.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
504         @Override
505         public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
506             if(listPR.get(i) == "PLEASE SELECT.") {
507                 ChoosePriceRange = "";
508             }else{
509                 ChoosePriceRange = listPR.get(i);
510             }
511         }
512
513         @Override
514         public void onNothingSelected(AdapterView<?> adapterView) {
515
516     });
517
518 }

```

บรรทัดที่ 314-518 ทำการ Set ปุ่มต่างๆหน้า Layout และทำการร้องขอข้อมูลจากฐานข้อมูล โดยการ Call web service เมื่อได้ Brand,GuitarType,TopWood,BackAndSideWood,PriceRange มาแล้วทำการแสดงใน List ที่ได้จัดเตรียมไว้หน้า Layout



-PHP call web service file

```

1 <?php
2
3     $HOST = 'localhost';
4     $USER = 'root';
5     $PASS = '';
6     $DBname = 'guitarguide';
7
8     ////////////// QUERY Brand_ID FROM PRODUCTS /////////////
9     $conn = mysqli_connect($HOST,$USER,$PASS,$DBname) or die('error connecting to database');
10
11    $strBrandSQL = "SELECT Brand_ID FROM products WHERE Skills = 2";
12    $result = $conn->query($strBrandSQL) or die($conn->error);
13    $resultBrandArray = array();
14    while($row = $result->fetch_assoc()){
15        $resultBrandArray[] = $row;
16    }
17    $resultBrandArray = array_map("unserialize",array_unique(array_map("serialize",$resultBrandArray) ) );
18    sort($resultBrandArray);
19
20
21    ////////////// QUERY GuitarType_ID FROM PRODUCTS ///////////
22    $strGTSQl = "SELECT GuitarType_ID FROM products WHERE Skills = 2";
23    $resultGT = $conn->query($strGTSQl) or die($conn->error);
24    $resultGTArray = array();
25    while($row = $resultGT->fetch_assoc()){
26        $resultGTArray[] = $row;
27    }
28
29    $resultGTArray = array_map("unserialize",array_unique(array_map("serialize",$resultGTArray) ) );
30    sort($resultGTArray);
31
32
33    ////////////// QUERY TopWood_ID FROM PRODUCTS /////////////
34    $strTopWoodSQL = "SELECT TopWood_ID FROM products WHERE Skills = 2";
35    $resultTopWood = $conn->query($strTopWoodSQL) or die($conn->error);
36    $resultTopWoodArray = array();
37    while($row = $resultTopWood->fetch_assoc()){
38        $resultTopWoodArray[] = $row;
39    }
39    $resultTopWoodArray = array_map("unserialize",array_unique(array_map("serialize",$resultTopWoodArray) ) );
40    sort($resultTopWoodArray);
41
42
43    ////////////// QUERY BackAndSideWood_ID FROM PRODUCTS ///////////
44    $strBackSideWoodSQL = "SELECT BackAndSideWood_ID FROM products WHERE Skills = 2";
45    $resultBackSideWood = $conn->query($strBackSideWoodSQL) or die($conn->error);
46    $resultBackSideWoodArray = array();
47    while($row = $resultBackSideWood->fetch_assoc()){
48        $resultBackSideWoodArray[] = $row;
49    }
49    $resultBackSideWoodArray = array_map("unserialize",array_unique(array_map("serialize",$resultBackSideWoodArray) ) );
50    sort($resultBackSideWoodArray);
51
52
53    ////////////// QUERY PriceRange_ID FROM PRODUCTS ///////////
54    $strPRSQL = "SELECT PriceRange_ID FROM products WHERE Skills = 2";

```



```

55 $resultPR      = $conn->query($strPRSQL) or die($conn->error);
56 $resultPRArray = array();
57 while($row = $resultPR->fetch_assoc()){
58     $resultPRArray[] = $row;
59 }
60 $resultPRArray = array_map("unserialize",array_unique(array_map("serialize",$resultPRArray) ) );
61 sort($resultPRArray);
62 ///////////////////////////////////////////////////////////////////
63
64 //***** CHECK Brand_ID FROM PRODUCTS QUERY WITH BRANDS *****/
65 $Brand      = "SELECT * FROM brands";
66 $reBrand   = $conn->query($Brand) or die($conn->error);
67 $reBrandArray = array();
68 while($row = $reBrand->fetch_assoc()){
69     $reBrandArray[] = $row;
70 }
71 $brands = array();
72 $i = 0;
73 foreach($resultBrandArray as $brandIDProduct){ //equal query from products
74     foreach($reBrandArray as $brandsID){
75         if($brandIDProduct["Brand_ID"] === $brandsID["ID"]){
76             $brands[$i] = array("BRAND"=>$brandsID["Name"]);
77         }
78     }
79     $i++;
80 }
81
82 //***** CHECK GuitarType_ID FROM PRODUCTS QUERY WITH GUITARTYPES *****/
83 $GuitarType  = "SELECT * FROM guitarTypes";
84 $reGuitarType = $conn->query($GuitarType) or die($conn->error);
85 $reGuitarTypeArray = array();
86 while($row = $reGuitarType->fetch_assoc()){
87     $reGuitarTypeArray[] = $row;
88 }
89 $guitarTypes = array();
90 $i = 0;
91 foreach($resultGTArray as $guitarTypeIDProduct){ //equal query from products
92     foreach($reGuitarTypeArray as $guitarTypeID){
93         if($guitarTypeIDProduct["GuitarType_ID"] === $guitarTypeID["ID"]){
94             $guitarTypes[$i] = array("GUITARTYPE"=>$guitarTypeID["Name"]);
95         }
96     }
97     $i++;
98 }
99
100
101 //***** CHECK TopWood_ID FROM PRODUCTS QUERY WITH TOPTONEWOOD *****/
102 $TopWood    = "SELECT * FROM topwoods";
103 $reTopWood  = $conn->query($TopWood) or die($conn->error);
104 $reTopWoodArray = array();
105 while($row = $reTopWood->fetch_assoc()){
106     $reTopWoodArray[] = $row;
107 }

```



```

109 StopWoods = array();
110 $i = 0;
111 foreach($resultTopWoodArray as $TopWoodIDProduct) //query products
112 {
113     foreach($reTopWoodArray as $TopWoodID) {
114         if($TopWoodIDProduct["TopWood_ID"] === $TopWoodID["ID"]){
115             $StopWoods[$i] = array("TOPTONEWOOD"=>$TopWoodID["Name"]);
116         }
117     }
118     $i++;
119 }
120
121 /****** CHECK BackAndSideWood_ID FROM PRODUCTS QUERY WITH BACKSIDEWOOD *****/
122 $BackSideWood = "SELECT * FROM backandsidewoods";
123 $reBackSideWood = $conn->query($BackSideWood) or die($conn->error);
124 $reBackSideWoodArray = array();
125 while($row = $reBackSideWood->fetch_assoc()) {
126     $reBackSideWoodArray[] = $row;
127 }
128 $BackSideWoods = array();
129 $i = 0;
130 foreach($resultBackSideWoodArray as $BackSideWoodIDProduct) //query products
131 {
132     foreach($reBackSideWoodArray as $BackSideWoodID) {
133         if($BackSideWoodIDProduct["BackAndSideWood_ID"] === $BackSideWoodID["ID"]){
134             $BackSideWoods[$i] = array("BACKSIDEWOOD"=>$BackSideWoodID["Name"]);
135         }
136     }
137     $i++;
138 }
139 /****** CHECK PriceRange_ID FROM PRODUCTS QUERY WITH PRICERANGES *****/
140 $PriceRange = "SELECT * FROM priceranges";
141 $rePriceRange = $conn->query($PriceRange) or die($conn->error);
142 $rePriceRangeArray = array();
143 while($row = $rePriceRange->fetch_assoc()) {
144     $rePriceRangeArray[] = $row;
145 }
146 $PriceRanges = array();
147 $i = 0;
148 foreach($resultPRArray as $PriceRangeIDProduct) //query products
149 {
150     foreach($rePriceRangeArray as $PriceRangeID) {
151         if($PriceRangeIDProduct["PriceRange_ID"] === $PriceRangeID["ID"]){
152             $PriceRanges[$i] = array("PRICERANGE"=>$PriceRangeID["Name"]);
153         }
154     }
155 }
156
157 $Expert = array_merge($brands,$guitarTypes,$StopWoods,$BackSideWoods,$PriceRanges);
158
159 $conn->close();
160 echo json_encode($Expert);
161 ?>
```

บรรทัดที่ 1-161 ทำการ Query ฐานข้อมูล Products เพื่อต้องการข้อมูล ได้

Brand,GuitarType,TopWood,BackAndSideWood,PriceRange ที่มี Skills = 2 เท่านั้น (ระดับ

ผู้เชี่ยวชาญ) และทำการ Return : JSON file กลับไปยังตัวแอพพลิเคชันเพื่อนำข้อมูลดังกล่าวไปแสดงใน List



4.2.2 ส่วนของการแนะนำการเลือกซื้อกีต้าร์ ระดับผู้เชี่ยวชาญ(ขั้นตอนการแสดงผลการวัดความคล้าย)

- XML file

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      xmlns:app="http://schemas.android.com/apk/res-auto">
6
7      <LinearLayout
8          android:layout_width="match_parent"
9          android:layout_height="84dp"
10         android:layout_marginBottom="5dp"
11         android:layout_marginEnd="5dp"
12         android:layout_marginStart="5dp"
13         android:layout_marginTop="5dp"
14         android:background="#99404040"
15         android:orientation="horizontal"
16         app:layout_constraintBottom_toBottomOf="parent"
17         app:layout_constraintEnd_toEndOf="parent"
18         app:layout_constraintHorizontal_bias="0.0"
19         app:layout_constraintStart_toStartOf="parent"
20         app:layout_constraintTop_toTopOf="parent"
21         app:layout_constraintVertical_bias="0.0">
22
23         <ImageView
24             android:id="@+id/list_item_img"
25             android:layout_width="103dp"
26             android:layout_height="62dp"
27             android:layout_margin="10dp"
28             android:adjustViewBounds="true"
29             app:srcCompat="@mipmap/ic_launcher" />
30
31     <LinearLayout
32         android:layout_width="match_parent"
33         android:layout_height="wrap_content"
34         android:orientation="vertical">
35
36         <LinearLayout
37             android:layout_width="match_parent"
38             android:layout_height="wrap_content"
39             android:layout_margin="10dp"
40             android:gravity="center_horizontal"
41             android:orientation="vertical">
42
43             <TextView
44                 android:id="@+id/list_item_similarity"
45                 android:layout_width="match_parent"
46                 android:layout_height="wrap_content"
47                 android:layout_margin="2dp"
48                 android:layout_weight="1"
49                 android:gravity="left|center_vertical"
50                 android:text="100.0"
51                 android:textColor="@android:color/background_light"
52                 android:textSize="12sp" />
53
54             <TextView
55                 android:id="@+id/list_item_name"
56                 android:layout_width="match_parent"
57                 android:layout_height="wrap_content"
58             </TextView>

```



```

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82

```

```

    android:layout_height="wrap_content"
    android:layout_margin="2dp"
    android:layout_weight="1"
    android:ellipsize="middle"
    android:gravity="left|center_vertical"
    android:singleline="true"
    android:text="nameXXX"
    android:textColor="@android:color/background_light"
    android:textSize="14sp" />

    <TextView
        android:id="@+id/list_item_price"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="2dp"
        android:layout_weight="1"
        android:gravity="left|center_vertical"
        android:text="pricexxx bath"
        android:textColor="@android:color/background_light"
        android:textSize="14sp" />
</LinearLayout>
</LinearLayout>
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```

บรรทัดที่ 23-29 Layout ส่วนแสดงรูปภาพที่ผ่านการคำนวณความคล้าย

บรรทัดที่ 44-53 Layout ส่วนแสดงผลลัพธ์ที่ผ่านการคำนวณความคล้าย

บรรทัดที่ 55-66 Layout ส่วนแสดงชื่อกีต้าร์ที่ผ่านการคำนวณความคล้าย

บรรทัดที่ 68-77 Layout ส่วนแสดงราคาที่ผ่านการคำนวณความคล้าย

- java file

```

118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148

```

```

public class AsyncTaskGetDataExpert extends AsyncTask<Void, Void, Void> {
    String similarity;
    String name;
    String image_part;
    String image;
    String price;
    String web_url;

    @Override
    protected Void doInBackground(Void... voids) {
        RequestBody requestBody = new MultipartBody.Builder()
            .addFormDataPart("brand", brand)
            .addFormDataPart("guitartype", guitartype)
            .addFormDataPart("topwood", topwood)
            .addFormDataPart("backsidewood", backsidewood)
            .addFormDataPart("pricerange", pricerange)
            .addFormDataPart("PriorityBrand", Pbrand)
            .addFormDataPart("PriorityGuitartype", Pguitartype)
            .addFormDataPart("PriorityTopwood", Ptopwood)
            .addFormDataPart("PriorityBackSidewood", Pbacksidewood)
            .addFormDataPart("PriorityPriceRange", Ppricerange)
            .setType(MultipartBody.FORM)
            .build();
        Request request = new Request.Builder()
            .url(HTTP + "android/getSimilarityExpertResult.php")
            .post(requestBody)
            .build();

        Response response;
        try {
            response = okHttpClient.newCall(request).execute();
        }
    }
}

```



```

149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175 @Override
176 protected void onPostExecute(Void avoid) {
177     super.onPostExecute(avoid);
178 }

```

บรรทัดที่ 118-178 ทำการรับค่าพารามิเตอร์ที่ส่งมาจากการเลือกคุณสมบัติและลำดับความสำคัญ จากนั้นทำการ call web service เพื่อนำคุณสมบัติและลำดับความสำคัญ ไปทำการตรวจสอบกับข้อมูลในฐานข้อมูล และคำนวณความคล้ายคลึง แล้วทำการ Return : JSON file กลับมาแสดงใน List

-PHP call web service file

```

1 ok?php
2
3 $HOST = 'localhost';
4 $USER = 'root';
5 $PASS = '';
6 $DBname = 'guitarguide';
7
8 $conn = mysqli_connect($HOST,$USER,$PASS,$DBname) or die('error connecting to database');
9
10 $strResultSQL = "SELECT pro.Name AS proName,pro.Price AS proPrice,pro.ImagePart AS proImagePart,pro.Image AS proImage,
11 bra.Name AS braName,gui.Name AS guiName,top.Name AS topName,bs.Name AS bsName,pr.Name AS prName
12 FROM products pro
13 LEFT JOIN brands bra ON bra.ID = pro.Brand_ID
14 LEFT JOIN guitars gui ON gui.ID = pro.GuitarType_ID
15 LEFT JOIN topwoods top ON top.ID = pro.TopWood_ID
16 LEFT JOIN backandsidewoods bs ON bs.ID = pro.BackAndSideWood_ID
17 LEFT JOIN priceranges pr ON pr.ID = pro.PriceRange_ID
18 WHERE pro.Skills = 2
19 ";
20
21 $result = $conn->query($strResultSQL) or die($conn->error);
22 $resultArray = array();
23 while($row = $result->fetch_assoc()){
24     $resultArray[] = $row;
25 }

```



```

27     $brand = $_POST['brand'];      // "Gibson"; // "Fender"; //
28     $guitartype = $_POST['guitartype'];    // "Electric Guitar"; //
29     $topwood = $_POST['topwood'];      // "Maple"; //
30     $backsidewood = $_POST['backsidewood'];    // "Mahogany"; //
31     $pricerange = $_POST['pricerange'];    // "80,000-100,000 bath"; //
32
33     $priorityWieghtBrand = (float) $_POST['PriorityBrand']; // "0.3"; //
34     $priorityWieghtGuitarType = (float) $_POST['PriorityGuitartype']; // "0.25"; //
35     $priorityWieghtTopWood = (float) $_POST['PriorityTopwood']; // "0.2
36     $priorityWieghtBackSideWood = (float) $_POST['PriorityBackSidewood']; // "0.15"; //
37     $priorityWieghtPriceRange = (float) $_POST['PriorityPriceRange']; // "0.1"; //
38
39     $CALCULATE_SIMILARITY = 0.0;
40     $RESULT_SIMILARITY = array();
41     $i = 0;
42     foreach($resultArray as $result){
43         $similarityBrand = 0.0;
44         $similarityGuitarType = 0.0;
45         $similarityTopWood = 0.0;
46         $similarityBackSideWood = 0.0;
47         $similarityPriceRange = 0.0;
48         if($result["braName"] == $brand){
49             $similarityBrand = 1.0;
50         }
51         if($result["guiName"] == $guitartype){
52             $similarityGuitarType = 1.0;
53         }
54         if($result["topName"] == $topwood){
55             $similarityTopWood = 1.0;
56         }
57         if($result["bsName"] == $backsidewood){
58             $similarityBackSideWood = 1.0;
59         }
60         if($result["prName"] == $pricerange){
61             $similarityPriceRange = 1.0;
62         }
63         $CALCULATE_SIMILARITY = ($priorityWieghtBrand*$similarityBrand)+($priorityWieghtGuitarType*$similarityGuitarType)
64         +($priorityWieghtTopWood*$similarityTopWood)+($priorityWieghtBackSideWood*$similarityBackSideWood)
65         +($priorityWieghtPriceRange*$similarityPriceRange);
66         if($CALCULATE_SIMILARITY != 0.0){
67             $RESULT_SIMILARITY[$i] = array("SIMILARITY" => (string)$CALCULATE_SIMILARITY,
68                                         "NAME"      => $result["proName"],
69                                         "IMAGE_PART" => $result["proImagePart"],
70                                         "IMAGE"      => $result["proImage"],
71                                         "PRICE"      => $result["proPrice"],
72                                         "WEB_URL"   => $result["proWebURL"]);
73         }
74         $i++;
75     }
76     $conn->close();
77     rsort($RESULT_SIMILARITY);
78
79     echo json_encode($RESULT_SIMILARITY);
80 ?>
```

บรรทัดที่ 1-80 ทำการ Query ฐานข้อมูล Products เพื่อต้องการข้อมูล เพื่อนำข้อมูลมาเปรียบเทียบความคล้ายให้กับแต่ละตัวแปรที่ถูกส่งมาจากแอปพลิเคชันจากนั้นทำการคำนวณด้วย Algorithm ความคล้าย แล้วทำการ Return : JSON file กลับไปยังตัวแอปพลิเคชันเพื่อนำข้อมูลดังกล่าวไปแสดงใน List
 บรรทัดที่ 63-65 ทำการคำนวณความคล้าย



บทที่ 5

การทดสอบระบบ

เมื่อทำการสร้างระบบแอพพลิเคชันทั้งในส่วนของ ระบบเว็บแอพพลิเคชันและแอ nondroidแอพพลิเคชันแล้วในขั้นต่อไปก็คือการทดสอบการทำงานของระบบว่าจะเป็นไปตามวัตถุประสงค์ที่ได้ตั้งไว้หรือไม่

5.1 การทดสอบระบบเว็บแอพพลิเคชัน : Admin

5.1.1 ทดสอบการเข้าใช้ระบบ : Login

5.1.2 ทดสอบการจัดการข้อมูล : Brand,GuitarType,TopWood,Back&SideWood,PriceRange

5.1.3 ทดสอบการจัดการข้อมูล: Product

5.2 การทดสอบระบบแอ nondroidแอพพลิเคชัน : User

5.2.1 ทดสอบการใช้งานระบบแนะนำการเลือกซื้อกีต้าร์

5.3 User Interface

5.3.1 ระบบเว็บแอพพลิเคชัน

5.3.2 ระบบแอ nondroidแอพพลิเคชัน



5.1 การทดสอบระบบเว็บแอพพลิเคชัน : Admin

ตาราง 5.1.1 ทดสอบการเข้าใช้ระบบ : Login

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
กระบวนการเข้าใช้ระบบ	ไม่กรอก Username และ Password	แสดงข้อความ “โปรดกรอกฟิลด์นี้”
	กรอก Username และ Password ไม่ถูกต้อง	แสดงข้อความ “Username หรือ Password ไม่ถูกต้อง”
	กรอก Username และ Password ถูกต้อง	เข้าสู่หน้าจัดการข้อมูล

5.1.2 ทดสอบการจัดการข้อมูล : Brand,GuitarType,TopWood,Back&SideWood,PriceRange

ตาราง 5.1.2.1 ทดสอบการเพิ่มข้อมูล : Brand,GuitarType,TopWood,Back&SideWood,PriceRange

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
กระบวนการเพิ่มข้อมูล	กรอกข้อมูลไม่ครบ	แสดงข้อความ “โปรดกรอกฟิลด์นี้”
	กรอกข้อมูล Code หรือ Name ซ้ำกับข้อมูลที่มีอยู่แล้วในฐานข้อมูล	แสดงข้อความ “Code หรือ Name มีอยู่แล้ว กรุณารอกรหัสล็อกครั้ง”
	กรอกข้อมูล Code หรือ Name ถูกต้อง	แสดงข้อความบันทึกข้อมูลเรียบร้อย

ตาราง 5.1.2.2 ทดสอบการแก้ไขข้อมูล : Brand,GuitarType,TopWood,Back&SideWood,PriceRange

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
กระบวนการแก้ไขข้อมูล	กรอกข้อมูลไม่ครบ	แสดงข้อความ “โปรดกรอกฟิลด์นี้”
	กรอกข้อมูล Code หรือ Name ซ้ำกับข้อมูลที่มีอยู่แล้วในฐานข้อมูล	แสดงข้อความ “Code หรือ Name มีอยู่แล้ว กรุณารอกรหัสล็อกครั้ง”
	กรอกข้อมูล Code หรือ Name ถูกต้อง	แสดงข้อความบันทึกข้อมูลเรียบร้อย



ตาราง 5.1.2.3 ทดสอบการลบข้อมูล : Brand,GuitarType,TopWood,Back&SideWood,PriceRange

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
กระบวนการลบข้อมูล	เลือกฟิล์ด ส่ง id ที่เลือกไปเพื่อทำการลบข้อมูล	แสดงข้อความ “คุณต้องการลบข้อมูลนี้ใช่หรือไม่?”
	กดตกลง	แสดงข้อความ “ลบข้อมูลเรียบร้อย”
	กดยกเลิก	กลับสู่หน้าการจัดการ

5.1.3 ทดสอบการจัดการข้อมูล: Product

ตาราง 5.1.3.1 ทดสอบการเพิ่มข้อมูล : Product

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
กระบวนการเพิ่มข้อมูล Product	กรอกข้อมูลไม่ครบ	แสดงข้อความ “โปรดกรอกพิวด์นี”
	กรอกข้อมูล Code หรือ Name ซ้ำกับข้อมูลที่มีอยู่แล้วในฐานข้อมูล	แสดงข้อความ “Code หรือ Name มีอยู่แล้ว กรุณารอกรหัสอีกครั้ง”
	กรอกข้อมูล Code หรือ Name ถูกต้อง	แสดงข้อความบันทึกข้อมูลเรียบร้อย

ตาราง 5.1.3.2 ทดสอบการแก้ไขข้อมูล : Product

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
กระบวนการแก้ไขข้อมูล Product	กรอกข้อมูลไม่ครบ	แสดงข้อความ “โปรดกรอกพิวด์นี”
	กรอกข้อมูล Code หรือ Name ซ้ำกับข้อมูลที่มีอยู่แล้วในฐานข้อมูล	แสดงข้อความ “Code หรือ Name มีอยู่แล้ว กรุณารอกรหัสอีกครั้ง”
	กรอกข้อมูล Code หรือ Name ถูกต้อง	แสดงข้อความบันทึกข้อมูลเรียบร้อย



ตาราง 5.1.3.3 ทดสอบการลบข้อมูล : Product

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
กระบวนการลบข้อมูล Product	เลือกฟิล์ด ส่ง id ที่เลือกไปเพื่อทำการลบข้อมูล	แสดงข้อความ “คุณต้องการลบข้อมูลนี้ใช่หรือไม่?”
	กดตกลง	แสดงข้อความ “ลบข้อมูลเรียบร้อย”
	กดยกเลิก	กลับสู่หน้าการจัดการ

5.2 การทดสอบระบบแอนดรอยด์แอพพลิเคชัน : User

5.2.1 ทดสอบการใช้งานระบบแนะนำการเลือกซื้อกีต้าร์

ตาราง 5.2.1 ทดสอบค้นหา กีต้าร์ : Beginner

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
กระบวนการค้นหา กีต้าร์ : Beginner	ไม่ได้เลือกคุณสมบัติ และไม่ได้เลือกลำดับความสำคัญ	แสดงข้อความ “กรุณาเลือกคุณสมบัติ และเลือกลำดับความสำคัญ!”
	เลือกคุณสมบัติ และไม่ได้เลือกลำดับความสำคัญ	แสดงข้อความ “กรุณาเลือกลำดับความสำคัญ!”
	ไม่เลือกคุณสมบัติ และเลือกลำดับความสำคัญ	แสดงข้อความ “กรุณาเลือกคุณสมบัติ!”
	เลือกคุณสมบัติ และเลือกลำดับความสำคัญ	แสดงผลการคำนวณวัดความคล้ายและแสดงข้อมูลกีต้าร์



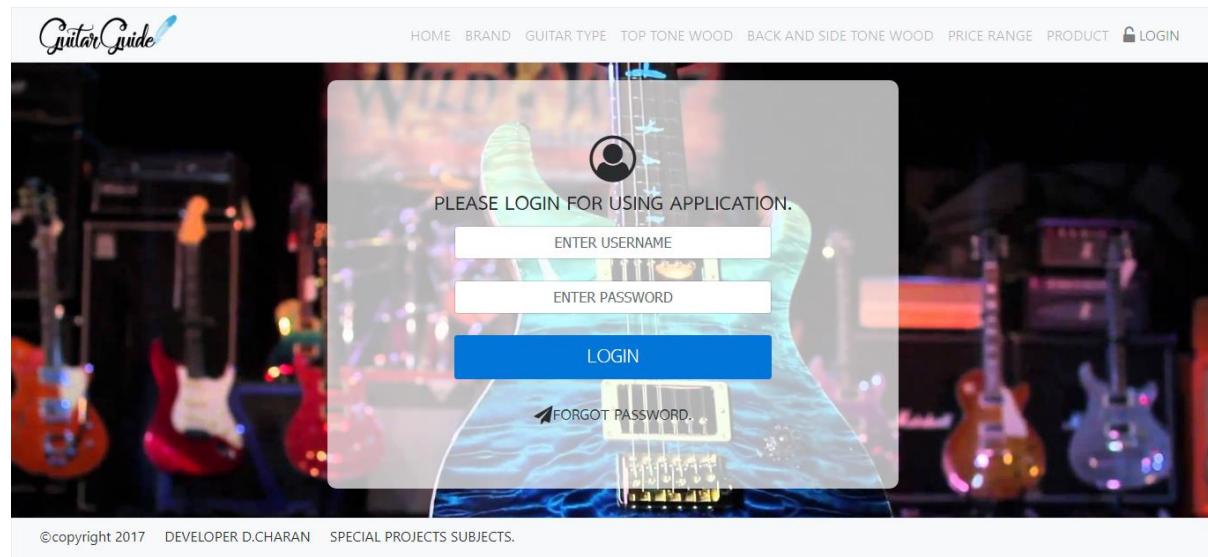
ตาราง 5.2.2 ทดสอบคันหากีต้าร์ : Expert

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
กระบวนการคันหากีต้าร์ : Expert	ไม่ได้เลือกคุณสมบัติ และไม่ได้เลือกลำดับความสำคัญ	แสดงข้อความ “กรุณาเลือกคุณสมบัติ และเลือกลำดับความสำคัญ!”
	เลือกคุณสมบัติ และไม่ได้เลือกลำดับความสำคัญ	แสดงข้อความ “กรุณาเลือกลำดับความสำคัญ!”
	ไม่เลือกคุณสมบัติ และเลือกลำดับความสำคัญ	แสดงข้อความ “กรุณาเลือกคุณสมบัติ!”
	เลือกคุณสมบัติ และเลือกลำดับความสำคัญ	แสดงผลการคำนวณวัดความคล้ายและแสดงข้อมูลกีต้าร์

5.3 User Interface

5.3.1 ระบบเว็บแอปพลิเคชัน

- Login



รูปที่ 3.31 แสดงภาพการ Login ของระบบเว็บแอปพลิเคชัน



- หน้าหลัก

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.32 แสดงภาพ หน้าหลัก ของระบบเว็บแอพพลิเคชัน

- หน้าหลักการจัดการ BRAND

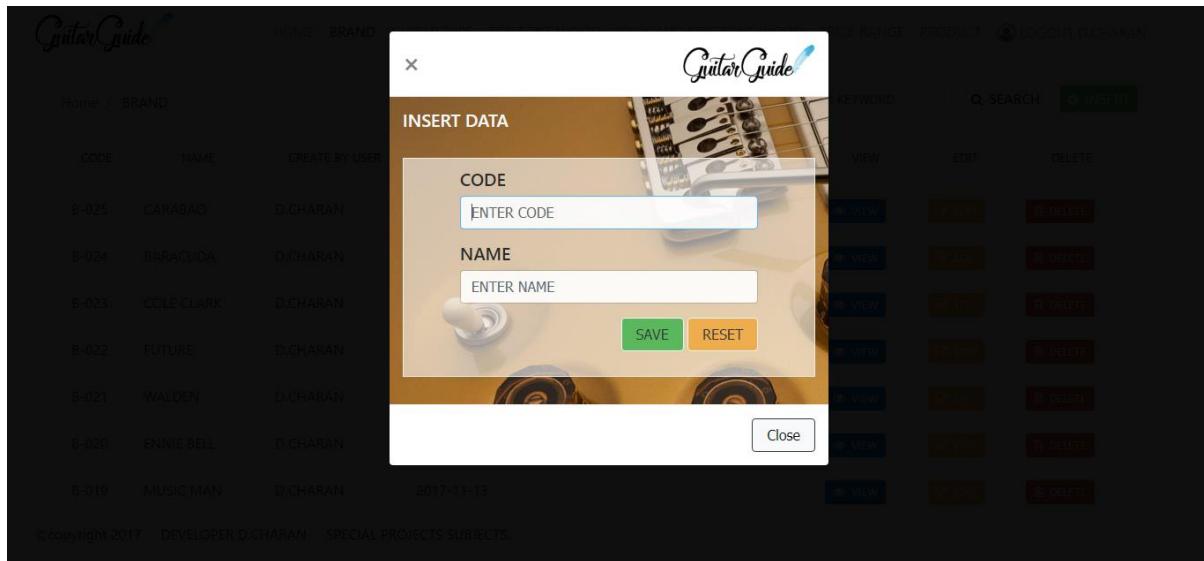
CODE	NAME	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
B-025	CARABAO	D.CHARAN	2017-11-13	D.CHARAN	2017-11-13	VIEW	EDIT	DELETE
B-024	BARACUDA	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-023	COLE CLARK	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-022	FUTURE	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-021	WALDEN	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-020	ENNIE BELL	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-019	MUSIC MAN	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.33 แสดงภาพหน้าหลักการจัดการ BRAND ของระบบเว็บแอพพลิเคชัน

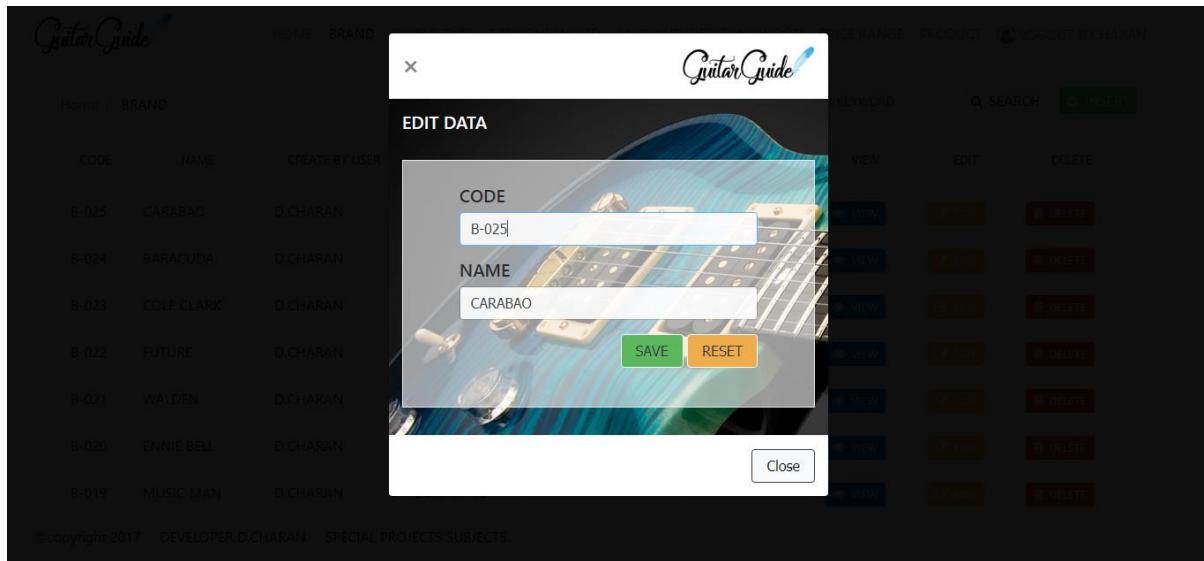


-การเพิ่มข้อมูล BRAND



รูปที่ 3.34 แสดงภาพการเพิ่มข้อมูล BRAND ของระบบเว็บแอพพลิเคชั่น

-การแก้ไขข้อมูล BRAND



รูปที่ 3.35 แสดงภาพการแก้ไขข้อมูล BRAND ของระบบเว็บแอพพลิเคชั่น



-การลบข้อมูล BRAND

CODE	NAME	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
B-025	CARABAO	D.CHARAN	2017-11-13	D.CHARAN	2017-11-13	VIEW	EDIT	DELETE
B-024	BARACUDA	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-023	COLE CLARK	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-022	FUTURE	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-021	WALDEN	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-020	ENNIE BELL	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE
B-019	MUSIC MAN	D.CHARAN	2017-11-13			VIEW	EDIT	DELETE

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.34 แสดงภาพการลบข้อมูล BRAND ของระบบเว็บแอพพลิเคชั่น

- หน้าหลักการจัดการ GUITAR TYPE

CODE	NAME	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
GT-001	ELECTRIC GUITAR	D.CHARAN	2017-11-14	D.CHARAN	2017-11-25	VIEW	EDIT	DELETE
GT-002	ACOUSTIC GUITAR	D.CHARAN	2017-11-14	D.CHARAN	2017-11-17	VIEW	EDIT	DELETE

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.35 แสดงภาพหน้าหลักการจัดการ GUITAR TYPE ของระบบเว็บแอพพลิเคชั่น



- การเพิ่มข้อมูล GUITAR TYPE

The screenshot shows a 'INSERT DATA' modal window overlaid on a list of guitar types. The modal has two input fields: 'CODE' and 'NAME', both with placeholder text 'ENTER CODE' and 'ENTER NAME' respectively. Below the fields are 'SAVE' and 'RESET' buttons. The background list includes columns for 'CODE', 'NAME', 'CREATE BY USER', and 'DATE'. The date column shows entries like '2017-11-17'.

รูปที่ 3.36 แสดงภาพการเพิ่มข้อมูล GUITAR TYPE ของระบบเว็บแอพพลิเคชัน

- การแก้ไขข้อมูล GUITAR TYPE

The screenshot shows an 'EDIT DATA' modal window overlaid on a list of guitar types. The modal displays the current values for 'CODE' (B-025) and 'NAME' (CARABAO). Below the fields are 'SAVE' and 'RESET' buttons. The background list includes columns for 'CODE', 'NAME', 'CREATE BY USER', and 'DATE'.

รูปที่ 3.37 แสดงภาพการแก้ไขข้อมูล GUITAR TYPE ของระบบเว็บแอพพลิเคชัน



- การลงทะเบียนข้อมูล GUITAR TYPE

CODE	NAME	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
GT-001	ELECTRIC GUITAR	D.CHARAN	2017-11-14	D.CHARAN	2017-11-25	VIEW	EDIT	DELETE
GT-002	ACOUSTIC GUITAR	D.CHARAN	2017-11-14	D.CHARAN	2017-11-17	VIEW	EDIT	DELETE

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.38 แสดงภาพการลงทะเบียนข้อมูล GUITAR TYPE ของระบบเว็บแอพพลิเคชัน

- หน้าหลักการจัดการ TOP TONE WOOD

CODE	NAME	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
TW-020	NONE	D.CHARAN	2017-12-05			VIEW	EDIT	DELETE
TW-019	SOLID	D.CHARAN	2017-12-05			VIEW	EDIT	DELETE
TW-018	SPRUCE	D.CHARAN	2017-12-04			VIEW	EDIT	DELETE
TW-017	ROSEWOOD	D.CHARAN	2017-12-04			VIEW	EDIT	DELETE
TW-016	ALDER	D.CHARAN	2017-12-04			VIEW	EDIT	DELETE
TW-015	MAPLE	D.CHARAN	2017-11-15			VIEW	EDIT	DELETE
TW-014	BUBINGA	D.CHARAN	2017-11-15			VIEW	EDIT	DELETE

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.39 แสดงภาพหน้าหลักการจัดการ TOP TONE WOOD ของระบบเว็บ
แอพพลิเคชัน



- การเพิ่มข้อมูล TOP TONE WOOD

The screenshot shows a 'GuitarGuide' website interface. A modal window titled 'INSERT DATA' is open in the center. It contains two input fields: 'CODE' with placeholder 'ENTER CODE' and 'NAME' with placeholder 'ENTER NAME'. Below these fields are 'SAVE' and 'RESET' buttons. The background of the modal shows a close-up image of a guitar's neck and headstock. The main page behind the modal lists various guitar brands with columns for 'CODE', 'NAME', 'CREATE BY USER', and 'VIEW', 'EDIT', 'DELETE' buttons. At the bottom of the page, there is a copyright notice: 'Copyright 2017 - DEVELOPER D.CHARAN - SPECIAL PROJECTS SUBJECTS.'

รูปที่ 3.40 แสดงภาพการเพิ่มข้อมูล TOP TONE WOOD ของระบบเว็บแอพพลิเคชัน

- การแก้ไขข้อมูล TOP TONE WOOD

The screenshot shows a 'GuitarGuide' website interface. A modal window titled 'EDIT DATA' is open in the center. It contains two input fields: 'CODE' with value 'B-025' and 'NAME' with value 'CARABAO'. Below these fields are 'SAVE' and 'RESET' buttons. The background of the modal shows a close-up image of a guitar's neck and headstock. The main page behind the modal lists various guitar brands with columns for 'CODE', 'NAME', 'CREATE BY USER', and 'VIEW', 'EDIT', 'DELETE' buttons. At the bottom of the page, there is a copyright notice: 'Copyright 2017 - DEVELOPER D.CHARAN - SPECIAL PROJECTS SUBJECTS.'

รูปที่ 3.41 แสดงภาพการแก้ไขข้อมูล TOP TONE WOOD ของระบบเว็บแอพพลิเคชัน



- การลบข้อมูล TOP TONE WOOD

CODE	NAME	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
TW-020	NONE	D.CHARAN	2017-12-05					
TW-019	SOLID	D.CHARAN	2017-12-05					
TW-018	SPRUCE	D.CHARAN	2017-12-04					
TW-017	ROSEWOOD	D.CHARAN	2017-12-04					
TW-016	ALDER	D.CHARAN	2017-12-04					
TW-015	MAPLE	D.CHARAN	2017-11-15					
TW-014	BUBINGA	D.CHARAN	2017-11-15					

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.42 แสดงภาพการลบข้อมูล TOP TONE WOOD ของระบบเว็บแอพพลิเคชั่น

- หน้าหลักการจัดการ BACK AND SIDE TONE WOOD

CODE	NAME	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
BS-024	NONE	D.CHARAN	2017-12-05					
BS-023	SOLID	D.CHARAN	2017-12-05					
BS-022	SPRUCE	D.CHARAN	2017-12-04					
BS-021	ROSEWOOD	D.CHARAN	2017-12-04					
BS-020	ALDER	D.CHARAN	2017-12-04					
BS-019	CHERRY	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15			
BS-018	IMBUIA	D.CHARAN	2017-11-15					

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.43 แสดงภาพหน้าหลักการจัดการ BACK AND SIDE TONE WOOD ของระบบเว็บแอพพลิเคชั่น



- การเพิ่มข้อมูล BACK AND SIDE TONE WOOD

The screenshot shows a 'Guitar Guide' web application interface. In the center, a modal window titled 'INSERT DATA' is open. It contains two input fields: 'CODE' with placeholder 'ENTER CODE' and 'NAME' with placeholder 'ENTER NAME'. Below these fields are 'SAVE' and 'RESET' buttons. The background of the main page lists various guitar brands with their codes and names, such as CARABAO, BARACUDA, COLE CLARK, FUTURE, WALDEN, ENNIE BELL, and MUSIC MAN. The entire interface is dark-themed.

รูปที่ 3.44 แสดงภาพการเพิ่มข้อมูล BACK AND SIDE TONE WOOD ของระบบเว็บแอปพลิเคชัน

- การแก้ไขข้อมูล BACK AND SIDE TONE WOOD

The screenshot shows the same 'Guitar Guide' web application interface as the previous one, but with a different modal window titled 'EDIT DATA'. This modal displays the existing data for a record: 'CODE' is B-025 and 'NAME' is CARABAO. There are 'SAVE' and 'RESET' buttons at the bottom. The background list of guitars remains the same. The interface is dark-themed.

รูปที่ 3.45 แสดงภาพการแก้ไขข้อมูล BACK AND SIDE TONE WOOD ของระบบเว็บแอปพลิเคชัน



- การลบข้อมูล BACK AND SIDE TONE WOOD

CODE	NAME	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
BS-024	NONE	D.CHARAN	2017-12-05					
BS-023	SOLID	D.CHARAN	2017-12-05					
BS-022	SPRUCE	D.CHARAN	2017-12-04					
BS-021	ROSEWOOD	D.CHARAN	2017-12-04					
BS-020	ALDER	D.CHARAN	2017-12-04					
BS-019	CHERRY	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15			
BS-018	IMBUIA	D.CHARAN	2017-11-15					

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.46 แสดงภาพการลบข้อมูล BACK AND SIDE TONE WOOD ของระบบเว็บแอพพลิเคชั่น

- หน้าหลักการจัดการ PRICE RANGE

CODE	PRICE RANGE	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
PR-008	100,000-300,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15			
PR-007	80,000-100,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15			
PR-006	60,000-80,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15			
PR-005	30,000-60,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15			
PR-004	10,000-30,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15			

1 2 [NEXT](#)

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.47 แสดงภาพหน้าหลักการจัดการ PRICE RANGE ของระบบเว็บแอพพลิเคชั่น



- หน้าหลักการจัดการ PRICE RANGE

The screenshot shows a dark-themed web application for managing guitar brands. At the top, there's a navigation bar with 'HOME', 'BRAND', 'PRICE RANGE', 'PRODUCT', and 'LOGOUT D.CHARAN'. Below the navigation, a table lists various guitar brands with columns for 'CODE', 'NAME', 'CREATE BY USER', and a date ('2017-11-17'). A modal window titled 'INSERT DATA' is centered over the table, containing fields for 'CODE' and 'NAME' with 'SAVE' and 'RESET' buttons. The background table shows rows for 'CARABAO', 'BARACUDA', 'COLE CLARK', 'FUTURE', 'WALDEN', 'ENNIE BELL', and 'MUSIC MAN', each created by 'D.CHARAN'.

รูปที่ 3.48 แสดงภาพการเพิ่มข้อมูล PRICE RANGE ของระบบเว็บแอพพลิเคชัน

- การแก้ไขข้อมูล PRICE RANGE

This screenshot shows the same dark-themed web application. A modal window titled 'EDIT DATA' is open, showing the details for a brand with code 'B-025' and name 'CARABAO'. The modal has 'CODE' (B-025) and 'NAME' (CARABAO) fields, along with 'SAVE' and 'RESET' buttons. The background table remains the same as in the previous screenshot.

รูปที่ 3.49 แสดงภาพการแก้ไขข้อมูล PRICE RANGE ของระบบเว็บแอพพลิเคชัน



- การลบข้อมูล PRICE RANGE

CODE	PRICE RANGE	CREATE BY USER	CREATE DATE	MODIFIED BY USER	MODIFIED DATE	VIEW	EDIT	DELETE
PR-008	100,000-300,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15	VIEW	EDIT	DELETE
PR-007	80,000-100,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15	VIEW	EDIT	DELETE
PR-006	60,000-80,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15	VIEW	EDIT	DELETE
PR-005	30,000-60,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15	VIEW	EDIT	DELETE
PR-004	10,000-30,000 BATH	D.CHARAN	2017-11-15	D.CHARAN	2017-11-15	VIEW	EDIT	DELETE

1 | 2 | [NEXT](#)

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.50 แสดงภาพการลบข้อมูล PRICE RANGE ของระบบเว็บแอพพลิเคชัน

- หน้าหลักการจัดการ PRODUCT

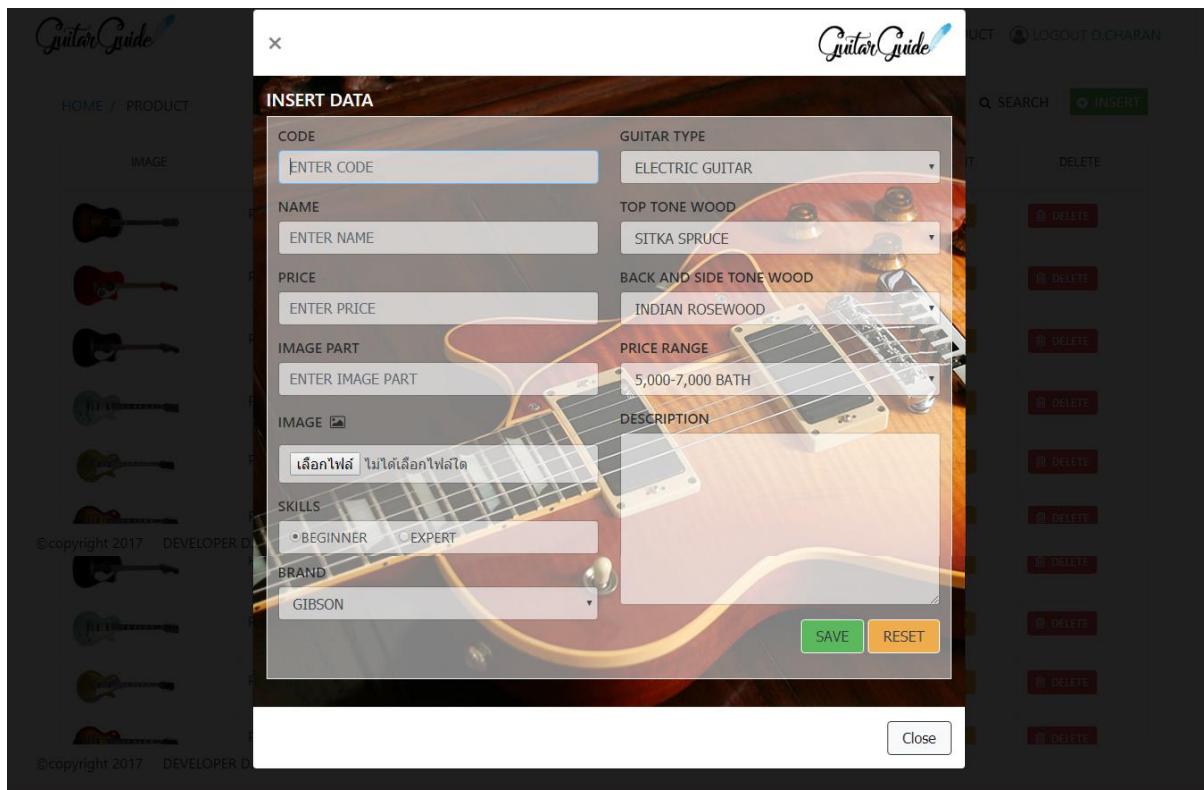
IMAGE	CODE	NAME	VIEW	EDIT	DELETE
	P-015	FENDER PM-1 DELUXE DREADNOUGHT	VIEW	EDIT	DELETE
	P-014	FENDER SONORAN SCE-AT001	VIEW	EDIT	DELETE
	P-013	FENDER SONORAN SCE-AT002	VIEW	EDIT	DELETE
	P-012	GIBSON 2014 LES PAUL CLASSIC	VIEW	EDIT	DELETE
	P-011	GIBSON LES PAUL 50S TRIBUTE 2016 T	VIEW	EDIT	DELETE
	P-010	GIBSON LES PAUL TRADITIONAL 2016 T	VIEW	EDIT	DELETE

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.51 แสดงภาพหน้าหลักการจัดการ PRODUCT ของระบบเว็บแอพพลิเคชัน

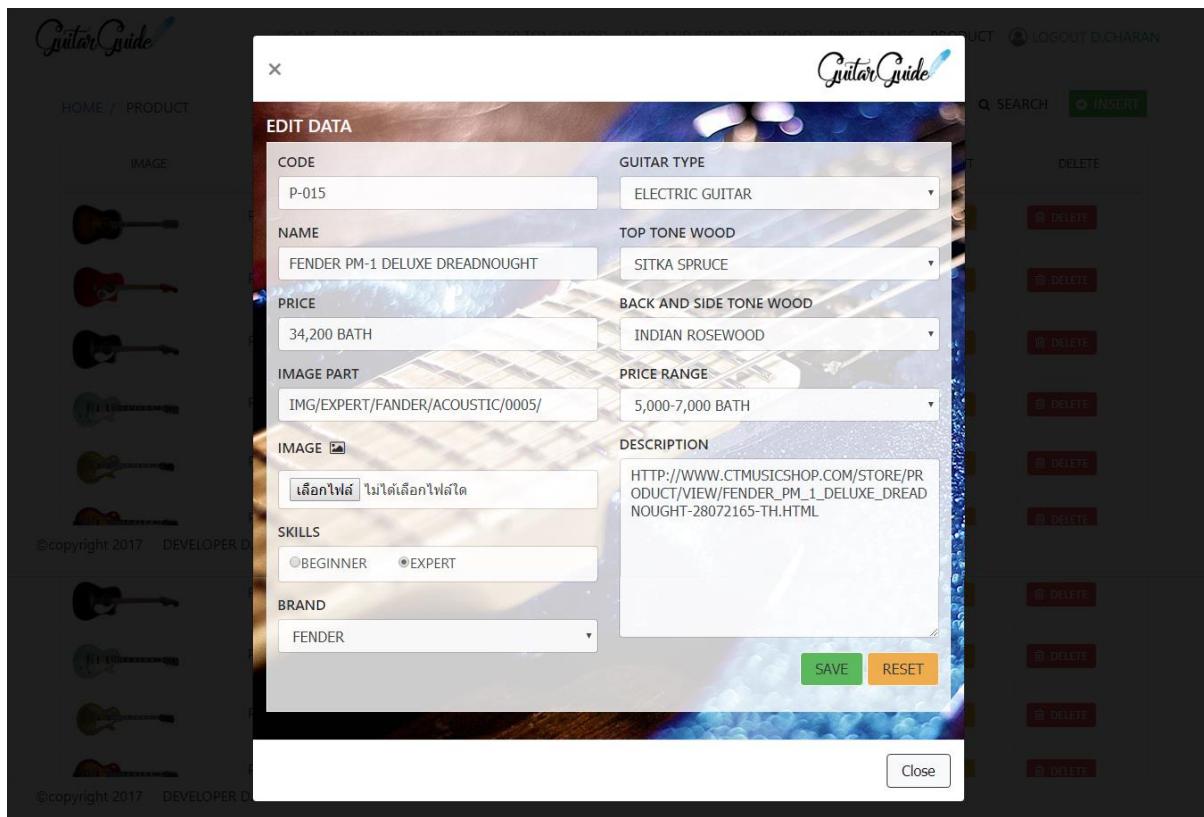


- การเพิ่มข้อมูล PRODUCT



รูปที่ 3.52 แสดงภาพการเพิ่มข้อมูล PRODUCT ของระบบเว็บแอพพลิเคชั่น

- การแก้ไขข้อมูล PRODUCT



รูปที่ 3.53 แสดงภาพการแก้ไขข้อมูล PRODUCT ของระบบเว็บแอพพลิเคชั่น



- การลบข้อมูล PRODUCT

IMAGE	CODE	NAME	VIEW	EDIT	DELETE
	P-015	FENDER PM-1 DELUXE DREADNOUGHT	<button>VIEW</button>	<button>EDIT</button>	<button>DELETE</button>
	P-014	FENDER SONORAN SCE-AT001	<button>VIEW</button>	<button>EDIT</button>	<button>DELETE</button>
	P-013	FENDER SONORAN SCE-AT002	<button>VIEW</button>	<button>EDIT</button>	<button>DELETE</button>
	P-012	GIBSON 2014 LES PAUL CLASSIC	<button>VIEW</button>	<button>EDIT</button>	<button>DELETE</button>
	P-011	GIBSON LES PAUL 50S TRIBUTE 2016 T	<button>VIEW</button>	<button>EDIT</button>	<button>DELETE</button>
	P-010	GIBSON LES PAUL TRADITIONAL 2016 T	<button>VIEW</button>	<button>EDIT</button>	<button>DELETE</button>

©copyright 2017 DEVELOPER D.CHARAN SPECIAL PROJECTS SUBJECTS.

รูปที่ 3.54 แสดงภาพการลบข้อมูล PRODUCT ของระบบเว็บแอพพลิเคชั่น

5.3.2 ระบบแอนดรอยด์แอพพลิเคชั่น

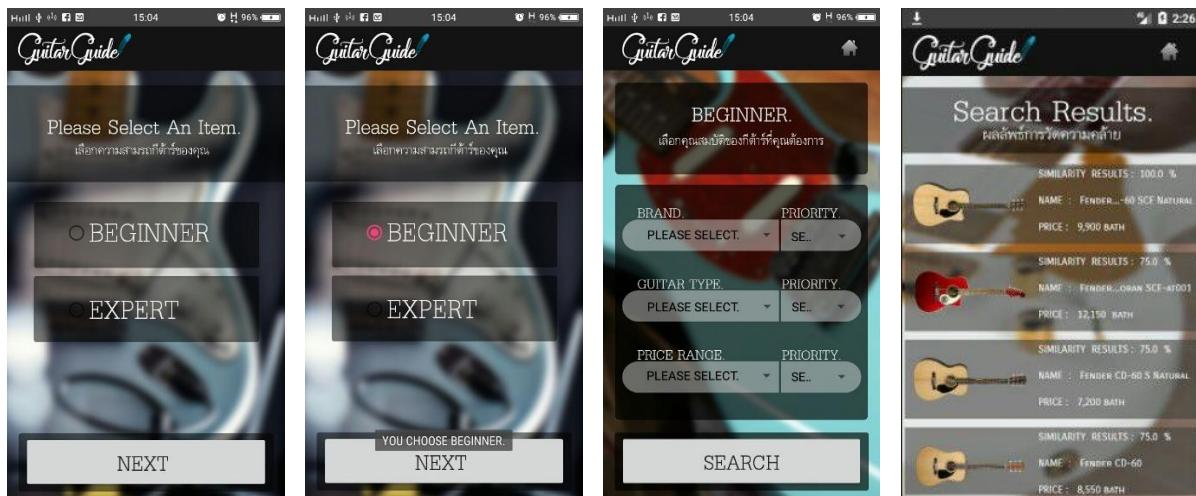
- แนะนำการเลือกชื่อกีต้าร์ : Loanding Activity



รูปที่ 3.55 แสดงภาพแนะนำการเลือกชื่อกีต้าร์ Loanding Activity ของระบบแอนดรอยด์ แอพพลิเคชั่น

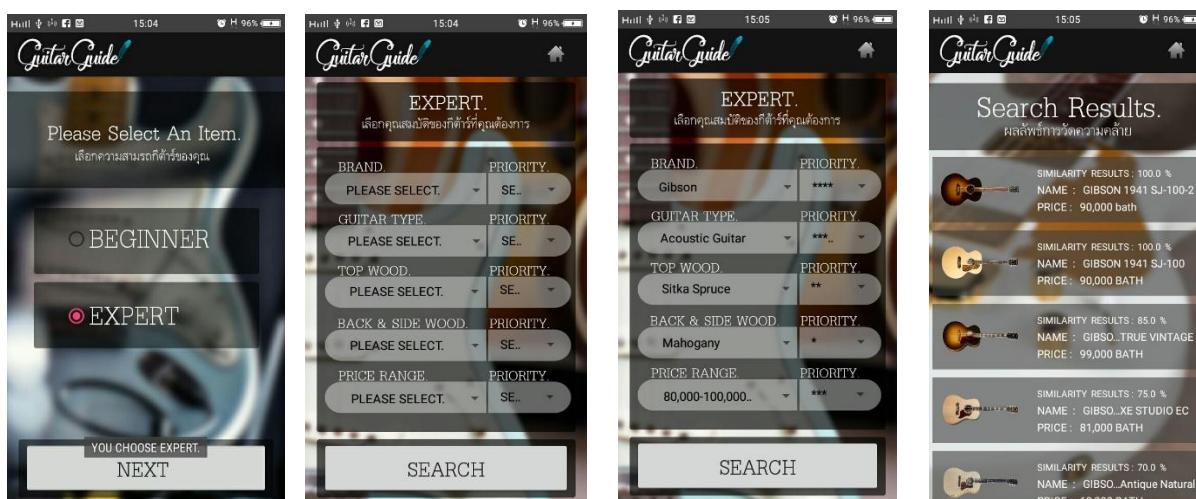


- แนะนำการเลือกซื้อกีต้าร์ : Beginner



รูปที่ 3.56 แสดงภาพแนะนำการเลือกซื้อกีต้าร์ Beginner ของระบบแอนดรอยด์แอพพลิเคชัน

- แนะนำการเลือกซื้อกีต้าร์ : Expert



รูปที่ 3.57 แสดงภาพแนะนำการเลือกซื้อกีต้าร์ Expert ของระบบแอนดรอยด์แอพพลิเคชัน



บทที่ 6

สรุปผลและข้อเสนอแนะ

ระบบเว็บแอพพลิเคชั่นการจัดการข้อมูล และระบบแนะนำการเลือกซื้อกีต้าร์ด้วยระบบแอนดรอยด์ แอพพลิเคชั่น ถูกพัฒนาขึ้นโดยผ่านกระบวนการ การวางแผน การวิเคราะห์ถึงปัญหาด้านต่างๆ การออกแบบ การเขียนโปรแกรมจนกระทั่งถึงขั้นตอนสุดท้าย คือการสรุปผลและมีข้อเสนอแนะดังนี้

6.1 สรุปผลความสามารถของระบบ

ระบบเว็บแอพพลิเคชั่นการจัดการข้อมูล และระบบแนะนำการเลือกซื้อกีต้าร์ด้วยระบบแอนดรอยด์ แอพพลิเคชั่น สรุปการใช้งานของระบบได้ดังต่อไปนี้

1. ระบบสามารถตรวจสอบสิทธิ์การเข้าใช้งานเว็บแอพพลิเคชั่นการจัดการข้อมูลได้
2. ผู้ดูแลระบบสามารถทำการค้นหา เพิ่ม แก้ไข ลบข้อมูล Brand,Guitar Type,Top Tone Wood,Back & Side Tone Wood,Price Range,Product ได้
3. ระบบสามารถแนะนำการเลือกซื้อกีต้าร์ให้แก่ผู้ใช้งานได้ โดยผู้ใช้ต้องเลือกคุณสมบัติและเลือก ลำดับความสำคัญให้กับคุณสมบัติตัว ระบบจะนำผลการค้นหาออกมาแสดงจำนวน 20 ชุดข้อมูล

6.2 ปัญหาที่พบรหัสการพัฒนาและแนวทางการแก้ไข

การพัฒนาทั้งระบบเว็บแอพพลิเคชั่นและแอนดรอยด์แอพพลิเคชั่น นักศึกษาผู้จัดทำโครงงานไม่มี ประสบการณ์การเขียนโปรแกรมภาษา JAVA มาก่อนจึงทำให้ค่อนข้างใช้เวลาในการศึกษาตัวภาษา ดังนั้น นักศึกษาผู้จัดทำโครงงานจึงต้องขอคำปรึกษาจากอาจารย์ผู้ที่สอนด้านภาษาจาวา ซึ่งช่วยให้การพัฒนาระบบ มีความคืบหน้าขึ้นมา

6.3 แนวทางในการพัฒนาต่อ

ระบบแนะนำการเลือกซื้อกีต้าร์โดยการใช้เทคนิคปริภูมิการคำนวณความคล้ายค่อนข้างมีประสิทธิภาพ ในอนาคตอาจจะพัฒนาต่อเพิ่มเติมในส่วนของการขาย การค้นหาในลักษณะนี้จะทำให้ผู้ใช้งานได้สิ่งที่ผู้ใช้งาน ต้องการมากกว่าการค้นหาแบบทั่วไป ในแต่การตลาดอาจทำให้ยอดขายเพิ่มขึ้นได้เนื่องจากผู้ใช้งานได้ข้อมูลที่ ตรงความต้องการ แต่ยังต้องปรับให้มีความสามารถที่หลากหลายและง่ายต่อการใช้งาน