

## แอปพลิเคชันเสียงเพื่อการผ่อนคลาย

Relax Sound

นายศุภกิตติ์ วงศ์แก้ว

โครงการนี้เป็นส่วนหนึ่งของการศึกษา

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2564

โครงงาน : แอปพลิเคชันเสียงเพื่อการผ่อนคลาย

Relax Sound

โดย : นายศุภกิตติ์ วงศ์แก้ว

อาจารย์ที่ปรึกษา : ผศ.ดร. สุภาวดี หรรษพงศ์สิน

ระดับการศึกษา : วิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

ปีการศึกษา : 2564

---

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา

ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

คณะกรรมการสอบประเมินความรู้โครงงานคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา

(ผศ.ดร. สุภาวดี หรรษพงศ์สิน)

..... กรรมการ

(ดร. ทศพร จุลิม)

..... กรรมการ

(อ. วานิชนา เหง้าเกษา)

..... หัวหน้าภาควิชา

(ผศ.ดร.สุพจน์ สีบุตร)

วันที่ ..../..../....

## กิตติกรรมประกาศ

โครงการเลี้ยงเพื่อการฝึกอบรมฯ สำหรับครุภัณฑ์ฯ ที่มีส่วนช่วยโครงการนี้

ขอขอบพระคุณอาจารย์ที่ปรึกษา พศ.ดร. สุวารดี หิรัญพงศ์สิน ที่เคยให้คำปรึกษาและช่วยแนะนำในการปรับแก้โครงการให้สมบูรณ์ ขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณบิดา มารดา ที่เคยให้กำลังใจ ความรักความห่วงใย ในการพัฒนาโครงการเสมอมา

นาย ศุภกิตติ์ วงศ์แก้ว

วันที่ 18 มิถุนายน 2565

โครงการ : แอปพลิเคชันเสียงเพื่อการผ่อนคลาย

### Relax Sound

โดย : นายศุภกิตติ์ วงศ์แก้ว

อาจารย์ที่ปรึกษา : ผศ.ดร. สุภารดี หรรษพงศ์สิน

ระดับการศึกษา : วิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

ปีการศึกษา : 2564

---

### บทคัดย่อ

ความเครียดสามารถส่งผลกระทบต่อบุคคลได้ทุกเพศทุกวัย บทความนี้นำเสนอโดยแอปพลิเคชันเสียงเพื่อช่วยลด

ความเครียดจากการทำงานหรือการเรียน โดยใช้ออนิคเฟรมเวิร์คและฐานข้อมูลไฟล์เบส โดยผู้ใช้สามารถเข้าใช้งาน และ

ระบบจะให้ผู้ใช้เลือกประเภทเสียงที่ผู้ใช้สนใจ จากนั้นระบบจะแสดงเสียงเพลงที่เลือกมาจากผู้ใช้อีกหนึ่งที่มีความสนุกในประเภทเสียง

เพลงเดียวกัน นอกจากนี้ ผู้ใช้ยังสามารถเพิ่มเพลงเป็นรายการที่ชอบ ตั้งเวลาปิดเสียง ตั้งเวลาอน และเลือกฟังเสียงได้หลากหลาย

**คำสำคัญ:** การผ่อนคลาย เสียงเพลง โน้ตบุ๊ก แอปพลิเคชัน อุปกรณ์เฟรมเวิร์ค

Topic : Relax Sound

Author : SUPAKIT WONGKAEW

Advisor : Asst.prof. Supawadee Hiranpongsin, Ph.D.

Degree : Bachelor of Science (Computer Science)

Academic Year : 2021

---

## ABSTRACT

Stress can affect people of all genders and ages. This article presents Relaxing Sound on the mobile application using ionic framework and firebase database. The proposed application may help to reduce the stress of work or study. The users can sign up to the application and choose the sound genre that they are interested in. Then, the matching sounds were selected from other users who are interested in the same sound music genre. Furthermore, users can also add a new sound as their favorite one, set a mute timer, set a sleep timer, and can choose a variety of sounds for listening.

**Keywords:** Relaxing, Sound Music, Mobile application, Ionic Framework

## สารบัญ

กิตติกรรมประกาศ	3
บทคัดย่อภาษาไทย	4
บทคัดย่อภาษาอังกฤษ	5
สารบัญ	6
สารบัญภาพ	9
สารบัญตาราง	13
บทที่ 1	14
บทนำ	14
1.1 ที่มาและเหตุผล	14
1.2 วัตถุประสงค์	14
1.3 ขอบเขตของโครงการ	14
1.4 ประโยชน์ที่ได้รับ	14
1.5 เครื่องมือที่ใช้ในการพัฒนา	15
1.5.1 ฮาร์ดแวร์	15
1.5.2 ซอฟต์แวร์ (Software)	15
1.5.3 แผนการดำเนินการ	16
บทที่ 2	17
ทฤษฎีที่เกี่ยวข้องและงานวิจัย	17
2.1 ความรู้พื้นฐาน Ionic Framework	17
2.2 ความรู้พื้นฐานเกี่ยวกับระบบปฏิบัติการแอนดรอยด์ (Android)	18
2.2.1 สถาปัตยกรรมของระบบแอนดรอยด์	18
2.3 ความรู้พื้นฐาน Firebase	21
2.3.1 Build Better Apps	21

2.3.2 Improve app quality	22
2.3.3 Grow your business	22
2.4 ความรู้พื้นฐาน Node.js	23
2.4.1 NPM Package management	23
2.5 แอปพลิเคชันที่เกี่ยวข้อง	23
บทที่ 3	30
การวิเคราะห์และออกแบบระบบ	30
3.1 โครงสร้างภาพรวมของระบบ (System Architecture)	31
3.2 การวิเคราะห์ความต้องการของระบบ (System Requirements)	32
3.2.1 Functional Requirements	32
3.2.2 Non-functional Requirements	32
3.3 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface Design)	32
3.3.1 หน้าสมัครสมาชิกและหน้าเข้าสู่ระบบ	34
3.3.2 หน้าหลักในการใช้งานแอปพลิเคชัน	36
3.3.3 หน้าแสดงรายการโปรด	39
3.3.4 ส่วนหน้าเว็บสำหรับผู้ดูแลระบบ	40
3.4 แผนภาพยุสเคส (Use Case Diagram)	43
3.5 แผนภาพคลาส (Class Diagram)	50
3.6 แผนภาพลำดับการทำงาน (Sequence Diagram)	55
บทที่ 4	65
การพัฒนาระบบ	65
4.1 การพัฒนาแอปพลิเคชัน	65
4.1.1 การเชื่อมต่อ Firestore database	65
4.1.2 การพัฒนาส่วนสมัครสมาชิกและเข้าสู่ระบบ	66

4.1.3 การพัฒนาส่วนแสดงเพลง	73
4.1.4 การพัฒนาส่วนโหลดข้อมูลมาแสดง	74
4.1.5 การพัฒนาส่วนแสดงเพลงที่ฟังบ่อຍ	78
4.1.6 การพัฒนาส่วนรับค่าข้อมูลสมาชิก	80
4.1.7 การพัฒนาส่วนเล่นเพลง	81
4.1.8 การพัฒนาส่วนเพิ่มเพลงรายการโปรด	83
4.1.9 การพัฒนาส่วนตั้งเวลาหยุดเพลง	84
4.2.0 การพัฒนาส่วนแจ้งเตือนเวลาอน	85
4.2.1 การพัฒนาส่วนส่งข้อมูลไปอีกหน้า	88
4.2.2 การพัฒนาส่วนหน้าเก็บเพลงโปรด	89
4.2.3 การพัฒนาส่วนผู้ดูแลระบบ	91
<b>บทที่ 5</b>	<b>94</b>
การทดสอบระบบ	94
5.1 การทดสอบใช้งานแอปพลิเคชัน	94
5.1.1 การทดสอบหน้าจอหลัก	94
5.1.2 การทดสอบหน้ารายการโปรด	94
5.1.3 การทดสอบส่วนเพิ่มเพลง	95
5.1.4 การทดสอบหน้าเล่นเพลง	95
<b>บทที่ 6</b>	<b>96</b>
สรุปและข้อเสนอแนะ	96
6.1 สรุปความสามารถแอปพลิเคชัน	96
6.1.1 เว็บ	96
6.1.2 แอปพลิเคชัน	96
6.2 ปัญหาและอุปสรรคในการพัฒนา	97

6.3 แนวทางพัฒนาต่อ	97
บรรณานุกรม	98
ภาคผนวก	100
ภาคผนวก ก	101
การติดตั้งเครื่องมือในการพัฒนาแอปพลิชัน	101
ก.1 การติดตั้ง Visual Studio Code	101
ก.2 การติดตั้ง Node.js	103
ก.3 การติดตั้ง Ionic Framework	106
ก.4 การติดตั้ง Android Studio	107
ภาคผนวก ข	110
คู่มือการติดตั้งแอปพลิเคชัน	110
ภาคผนวก ค	113
คู่มือการใช้งานระบบ	113
ประวัติผู้เขียน	123

## สารบัญภาพ

ภาพที่ 2.1 หลักการทำงาน Ionic Framework	18
ภาพที่ 2.2 โครงสร้างของระบบแออนดรอยด์	19
ภาพที่ 2.3 หน้าจอแอปพลิเคชัน Relax Melodies	24
ภาพที่ 2.4 หน้าจอแอปพลิเคชัน Sleep Sounds - relaxing sounds	25
ภาพที่ 2.5 หน้าจอแอปพลิเคชัน Sleepo: เสียงผ่อนคลาย, นอนหลับ	26
ภาพที่ 2.6 หน้าจอแอปพลิเคชัน ผ่อนคลายชันบ� - เสียงธรรมชาติ	27
ภาพที่ 2.7 หน้าจอแอปพลิเคชัน เสียงธรรมชาติ	28
ภาพที่ 3.1: System architecture แอปพลิเคชันเสียงเพื่อการผ่อนคลาย	31

ภาพที่ 3.2: หน้าจอสมัครสมาชิก	34
ภาพที่ 3.3: หน้าจอเข้าสู่ระบบสมาชิก	35
ภาพที่ 3.4: หน้าแสดงรายการเพลง	36
ภาพที่ 3.5: หน้าเล่นเสียงเพลง	37
ภาพที่ 3.6: หน้าแสดงรายเพลงยอดฮิต	38
ภาพที่ 3.7: หน้าตั้งเวลาเข้าอน	38
ภาพที่ 3.8: หน้าแสดงรายการโปรด	39
ภาพที่ 3.9: หน้าแสดงเพลงที่เพิ่ม	40
ภาพที่ 3.10: ส่วนเพิ่มเพลง	41
ภาพที่ 3.11: ส่วนลบเพลง	42
ภาพที่ 3.12: ยูสเคสของแอปพลิเคชันเสียงเพื่อการฟ่อนคลาย	44
ภาพที่ 3.13 : Class diagram แอปพลิเคชันเสียงเพื่อการฟ่อนคลาย	52
ภาพที่ 3.14 : Class diagram เว็บเพิ่มเพลงสำหรับผู้ดูและระบบ	54
ภาพที่ 3.15 : Sequence Diagram เข้าสู่ระบบแอปพลิเคชัน	57
ภาพที่ 3.16 : Sequence Diagram แสดงหน้าหลักแอปพลิเคชัน	58
ภาพที่ 3.17 : Sequence Diagram แสดงส่วนเพิ่มเพลง	59
ภาพที่ 3.18 : Sequence Diagram แสดงส่วนลบเพลง	60
ภาพที่ 3.19 : Sequence Diagram แสดงส่วนเพิ่มและลบเพลงรายการโปรด	61
ภาพที่ 3.20 : Sequence Diagram แสดงส่วนตั้งเวลาเดือนเข้าอน	62
ภาพที่ 3.21 : Sequence Diagram แสดงส่วนเพลงที่ชอบฟังมากที่สุด	63
ภาพที่ 3.22 : Sequence Diagram แสดงส่วนรายการเพลง	64
ภาพที่ 4.1 : ไฟล์ environment.ts	65
ภาพที่ 4.2 : ไฟล์ homes.ts ส่วนเข้ามบริการฐานข้อมูล	66
ภาพที่ 4.3 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ	68

ภาพที่ 4.4 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)	69
ภาพที่ 4.5 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)	71
ภาพที่ 4.6 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)	72
ภาพที่ 4.7 : ตัวอย่างส่วนเก็บลิสต์เพลง	73
ภาพที่ 4.8 : โครงสร้างส่วนໂທລດข้อมูลมาแสดง	74
ภาพที่ 4.9 : โครงสร้างส่วนໂທລດข้อมูลมาแสดง (ต่อ)	75
ภาพที่ 4.10 : โครงสร้างส่วนໂທລດข้อมูลมาแสดง (ต่อ)	76
ภาพที่ 4.11 : โครงสร้างส่วนเรียกใช้ฟังก์ชัน (ต่อ)	77
ภาพที่ 4.12 : โครงสร้างส่วนแสดงเพลงที่ฟังบอຍ	78
ภาพที่ 4.13 : ตัวอย่างโครงสร้างส่วนเก็บค่าจำนวนเพลง (ต่อ)	79
ภาพที่ 4.14 : โครงสร้างส่วนรับค่าข้อมูลสมาชิก	80
ภาพที่ 4.15 : โครงสร้างส่วนเล่นเพลง	81
ภาพที่ 4.16 : โครงสร้างส่วนเล่นเพลง (ต่อ)	82
ภาพที่ 4.17 : ตัวอย่างโครงสร้างส่วนเพิ่มเพลงเป็นรายโปรด	83
ภาพที่ 4.18 : โครงสร้างส่วนตั้งเวลาหยุดเพลง	84
ภาพที่ 4.19 : โครงสร้างส่วนแจ้งเตือนเวลาเข้านอน	85
ภาพที่ 4.20 : โครงสร้างส่วนแจ้งเตือนเวลาเข้านอน (ต่อ)	87
ภาพที่ 4.21 : โครงสร้างส่งข้อมูลไปอีกหน้า	88
ภาพที่ 4.22 : ตัวอย่างโครงสร้างหน้าเก็บเพลงโปรด	89
ภาพที่ 4.23 : ตัวอย่างโครงสร้างหน้าเก็บเพลงโปรด	90
ภาพที่ 4.24 : โครงสร้างหน้าเก็บเพลงโปรด	90
ภาพที่ 4.25 : โครงสร้างส่วนแสดงข้อมูล	91
ภาพที่ 4.26 : โครงสร้างส่วนลบเพลง	91
ภาพที่ 4.27 : โครงสร้างส่วนเพิ่มเพลง	92

ภาพที่ ก.1 : หน้าเว็บดาวน์โหลด Visual Studio Code	101
ภาพที่ ก.2 : ไฟล์โปรแกรม Visual Studio Code	102
ภาพที่ ก.3 : โปรแกรม Visual Studio Code	102
ภาพที่ ก.4 : หน้าเว็บดาวน์โหลด Node.js	103
ภาพที่ ก.5 : ติดตั้ง Node.js	104
ภาพที่ ก.6 : ติดตั้ง Node.js เสร็จสมบูรณ์	105
ภาพที่ ก.7 : คำสั่งติดตั้ง Ionic Framework	106
ภาพที่ ก.8 : ติดตั้ง Ionic Framework เสร็จสมบูรณ์	106
ภาพที่ ก.9 : หน้าเว็บดาวน์โหลด Android Studio	107
ภาพที่ ก.10 : หน้าติดตั้ง Android Studio	108
ภาพที่ ก.11 : ติดตั้ง Android Studio เสร็จสมบูรณ์	109
ภาพที่ ข.1 : หน้าดาวน์โหลดแอปพลิเคชัน Relax Sound	110
ภาพที่ ข.2 : หน้าแสดงการติดตั้ง	111
ภาพที่ ข.3 : หน้าแสดงการติดตั้งสำเร็จ	112
ภาพที่ ค.1 : หน้าจอแสดงเพลงและเพิ่มเพลง	113
ภาพที่ ค.2 : หน้าจอส่วนลับเพลง	114
ภาพที่ ค.3 : หน้าจอต้อนรับ	115
ภาพที่ ค.4 : หน้าสมัครสมาชิก	116
ภาพที่ ค.5 : หน้าเข้าสู่ระบบ	117
ภาพที่ ค.6 : หน้าจอหลัก	118
ภาพที่ ค.7 : หน้าจอส่วนตั้งเวลาเข้านอน	119
ภาพที่ ค.8 : ส่วนแจ้งเตือนเวลาเข้านอน	120
ภาพที่ ค.9 : หน้าจอตั้งเวลาหยุดเพลง	121

ภาพที่ ค.10 : หน้าจอรายการโปรด

122

## สารบัญตาราง

ตารางที่ 1.1 ขั้นตอนการดำเนินการ	16
ตารางที่ 3.1 สัญลักษณ์ Use Case Diagram	43
ตารางที่ 3.2 อธิบาย Use case หน้าที่ระบบแอป ในภาพที่ 3.8	45
ตารางที่ 3.3 Use Case สมัครเข้าใช้งาน	46
ตารางที่ 3.4 Use Case Login เข้าใช้งาน	46
ตารางที่ 3.5 เลือกเสียงที่ต้องการฟัง	47
ตารางที่ 3.6 เดือนเวลาอน	47
ตารางที่ 3.7 เพิ่มเสียงที่ชอบเป็นรายการโปรด	47
ตารางที่ 3.8 ดูเพลงยอดฮิต	48
ตารางที่ 3.9 ตั้งเวลาปิดเสียง	48
ตารางที่ 3.10 ลบเสียงรายการโปรด	48
ตารางที่ 3.11 เพิ่มรูป	49
ตารางที่ 3.12 เพิ่มเพลง	49
ตารางที่ 3.13 สัญลักษณ์ของ Class Diagram	50
ตารางที่ 3.14 สัญลักษณ์ของ Class Diagram (ต่อ)	51
ตารางที่ 3.15 อธิบาย Class Diagram ของคลาสพื้นฐานแอปพลิเคชัน	52
ตารางที่ 3.16 อธิบาย Class Diagram ของคลาสพื้นฐานแอปพลิเคชัน (ต่อ)	53
ตารางที่ 3.17 อธิบาย Class Diagram ของคลาสพื้นฐานแอปพลิเคชัน	54
ตารางที่ 3.18 สัญลักษณ์ Sequence Diagram	55

## บทที่ 1

### บทนำ

#### 1.1 ที่มาและเหตุผล

ปัจจุบันผู้คนเผชิญกับโรคไวรัสโคโรนา-19 เสียงเพลงได้รับความนิยมมาอย่างยาวนานโดยมีความเชื่อกันว่าเสียงเพลงจะช่วยให้คุณฟังผ่อนคลาย หรือลดความเครียด และยังช่วยให้หัวใจมีความเกี่ยวกับความเครียดลดลงมากกว่าใช้ยาลดความเครียด และยังช่วยลดภาวะซึมเศร้าทั้งยังช่วยปรับให้อารมณ์สงบขึ้นทำให้นอนหลับง่าย เรายกได้ว่าเสียงเพลงเปรียบเสมือนยาที่มีผลกับจิตใจ ช่วยกระตุ้นสมอง ปรับระดับคอร์ติซอลของรูปโนนที่เกี่ยวกับความเครียดให้คุณฟังรู้สึกผ่อนคลายได้ เสียงเพลงที่ใช้ในการผ่อนคลายนั้นไม่ได้ถูกจำกัดว่าจะต้องเพลงบรรเลงหรือเสียงธรรมชาติเท่านั้น เพลงทุกประเภทไม่ว่าจะเป็นคลาสสิก แจ๊ส และอีกหลากหลายแนว สามารถช่วยผ่อนคลายได้หมด ขึ้นอยู่กับความชอบของผู้ฟังแนวไหนแล้วมีอารมณ์ร่วมมากกว่า จึงเป็นที่มาในการพัฒนาโน้มบายแอปพลิเคชันในบทความนี้นำเสนอโน้มบายแอปพลิเคชันเสียงเพื่อช่วยลดความเครียดจากการทำงานหรือการเรียน โดยใช้ออนิคเฟรมเวิร์คและฐานข้อมูลไฟล์เบส โดยผู้ใช้สมัครสมาชิกเข้าใช้งาน ซึ่งระบบจะให้ผู้ใช้เลือกประเภทเสียงที่ผู้ใช้สนใจและจะแสดงเสียงเพลงให้ผู้ใช้ และจะแสดงเสียงเพลงจากผู้ใช้อื่นที่มีความสนใจในประเภทเสียงเพลงเดียวกัน และผู้ใช้ยังสามารถเพิ่มเพลงเป็นรายการที่ชอบ ตั้งเวลาปิดเสียง ตั้งเวลาอน และเลือกฟังเสียงได้หลากหลาย

#### 1.2 วัตถุประสงค์

##### 1.2.1 เพื่อพัฒนาโน้มบายแอปพลิเคชันเสียงเพลง

#### 1.3 ขอบเขตของโครงงาน

1. แอปพลิเคชันใช้งานได้บนสมาร์ทโฟนที่ใช้ระบบปฏิบัติการ Android เท่านั้น
2. ผู้ใช้ต้องเข้าสู่ระบบก่อนเข้าใช้งานแอปพลิเคชัน
3. มีเสียงให้เลือกฟัง 4 ประเภท 1.ธรรมชาติ 2.ผ่อนคลาย 3.เสียงเพลงและเสียงเครื่องดนตรี 4.เสียงยานพาหนะ

#### 1.4 ประโยชน์ที่ได้รับ

1. ช่วยประหยัดเวลาโดยที่ผู้ใช้ไม่ต้องเสียเวลาไปเปิดฟังบนอินเทอร์เน็ต
2. เพื่อช่วยให้เกิดการผ่อนคลาย

3. เพื่อช่วยให้ความสะดวกกับผู้ใช้ที่ต้องการใช้งานแอปพลิเคชันบนอุปกรณ์เคลื่อนที่

## 1.5 เครื่องมือที่ใช้ในการพัฒนา

### 1.5.1 ฮาร์ดแวร์

#### 1. สมาร์ทโฟน (Smart phone)

- ระบบปฏิบัติการ Android version 9
- หน่วยประมวลผลกลาง Octa Core Kirin 710 ความเร็ว 2.2 กิกะเฮิร์ตซ์ (Gigahertz: GHz)
- RAM 4 กิกะไบต์ (Gigabyte: GB)
- ตัวจัดเก็บข้อมูลขนาดความจุ 128 กิกะไบต์ (Gigabyte: GB)
- หน้าจอแสดงผลความละเอียด 2340 x 1080 พิกเซล (Pixel)
- หน้าจอแสดงผลขนาด 6.3 นิ้ว

#### 2. เครื่องคอมพิวเตอร์ส่วนบุคคล (Personal computer)

- ระบบปฏิบัติการ Windows
- หน่วยประมวลผล Intel Core i7 GEN 9th
- CPU 2.60 กิกะเฮิร์ตซ์ (Gigahertz: GHz)
- หน่วยความจำหลัก 16 กิกะไบต์ (Gigabyte: GB)
- ตัวจัดเก็บข้อมูลขนาดความจุ 1,000 กิกะไบต์ (Gigabyte: GB)

### 1.5.2 ซอฟต์แวร์ (Software)

#### 1. Ionic Framework ในการพัฒนาแอปพลิเคชัน

#### 2. Android Studio ในการพัฒนาแอปพลิเคชันบนเครือข่าย

#### 3. Firebase เป็นฐานข้อมูล

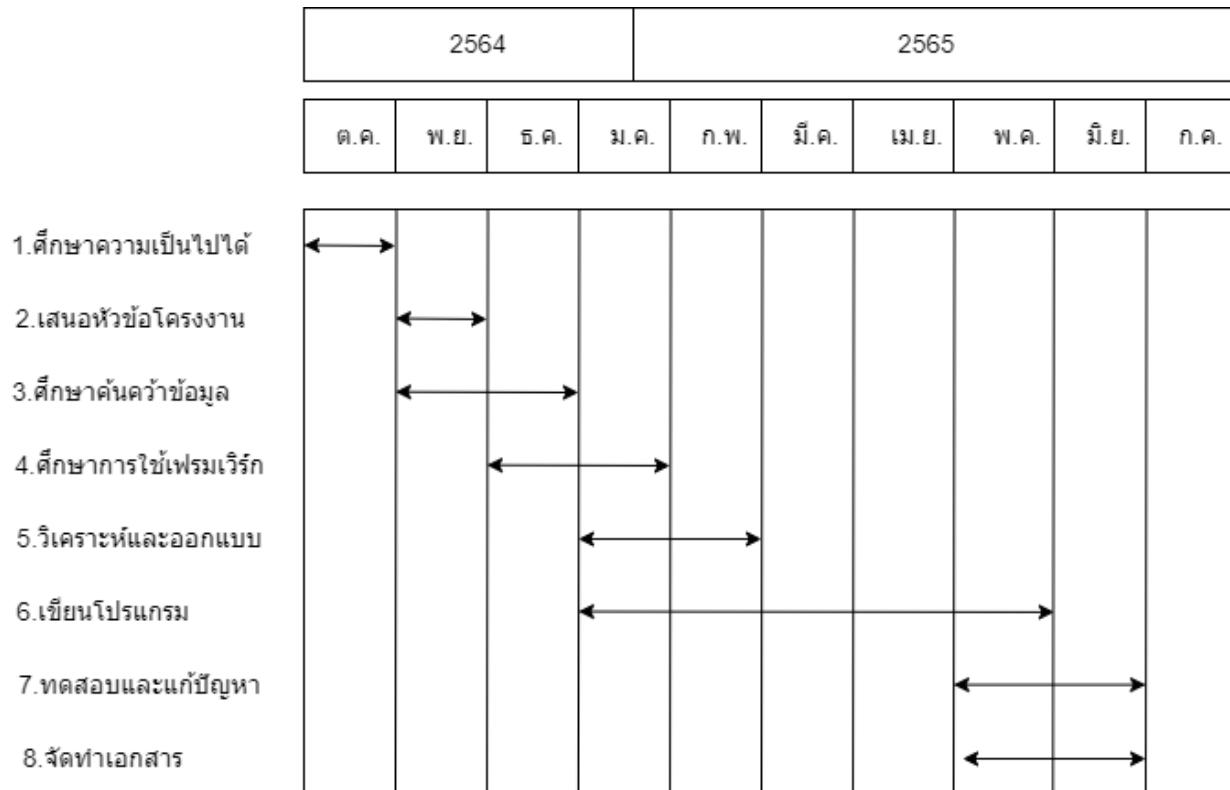
#### 4. Visual Studio Code ใช้สำหรับเปิดแก้ไขการเขียน Code โปรแกรม

#### 5. Node.js ใช้เป็นชุดคำสั่งในการติดต่อแลบารี

### 1.5.3 แผนการดำเนินการ

แผนการดำเนินการมี 8 ขั้นตอนดังนี้

ตารางที่ 1.1: ขั้นตอนการดำเนินการ



## บทที่ 2

### ทฤษฎีที่เกี่ยวข้องและงานวิจัย

บทนี้จะเป็นรายละเอียดเกี่ยวกับทฤษฎีและแอปพลิเคชันที่เกี่ยวข้องกับการพัฒนาแอปพลิเคชันในครั้งนี้ ประกอบด้วยหัวข้อหลัก หัวข้อมด 5 ข้อดังนี้

2.1 ความรู้พื้นฐาน Ionic Framework

2.2 ความรู้พื้นฐานเกี่ยวกับระบบปฏิบัติการแอนดรอยด์ (Android)

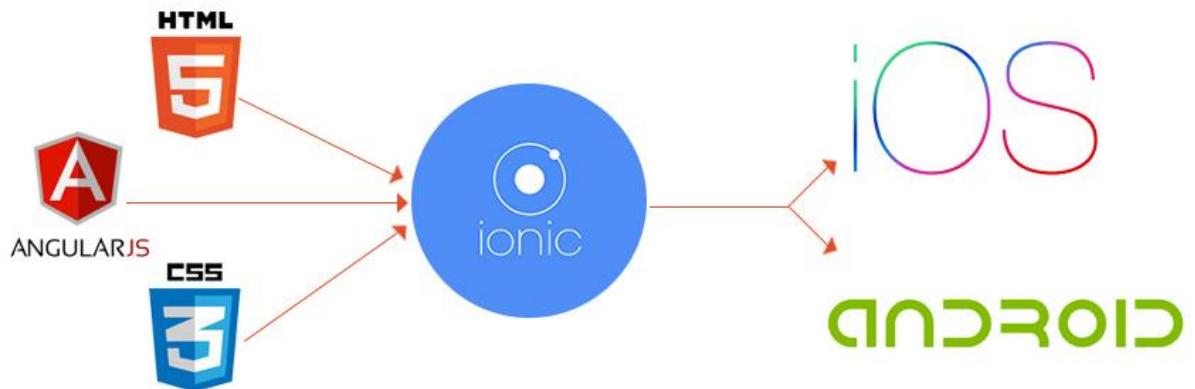
2.3 ความรู้พื้นฐาน Firebase

2.4 ความรู้พื้นฐาน Node.js

2.5 แอปพลิเคชันที่เกี่ยวข้อง

2.1 ความรู้พื้นฐาน Ionic Framework

Ionic Framework คือ [1] เครื่องมือในการสร้าง HTML , CSS และ JavaScript เพื่อใช้ในการสร้าง Mobile Application ซึ่งสามารถใช้งานได้ค่อนข้างง่าย Ionic Framework ยังสามารถใช้ Command-line interface(CLI) เข้ามาช่วยในการจัดการส่วนต่าง ๆ ของแอปพลิเคชัน เช่นการสร้างหน้าใหม่ หรือการเข้ามาช่วยด้านการติดตั้งให้เจ้ายันได้อีกด้วยซึ่ง ionic framework ยังเป็นเครื่องมือสร้างแอปพลิเคชันมือถือสามารถสร้างครั้งเดียวใช้งานได้หลายระบบปฏิบัติการ ในตอนสุดท้ายเพื่อให้ทั้งแอปพลิเคชันที่เขียนมาใช้ได้กับทุกระบบปฏิบัติการนั้นเอง ดังภาพ 2.1



ภาพที่ 2.1 หลักการทำงาน Ionic Framework

ที่มา: <http://www.mindphp.com/forums/download/file.php?id=24482>

## 2.2 ความรู้พื้นฐานเกี่ยวกับระบบปฏิบัติการแอนดรอยด์ (Android)

แอนดรอยด์ คือ [2] ระบบปฏิบัติการแบบเปิดเผยแพร่ต้นฉบับ (Open Source) โดยบริษัท กูเกิล (Google Inc.) ที่ได้รับความนิยมเป็นอย่างสูง เนื่องจากอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ มีจำนวนมาก อุปกรณ์มีหลากหลายระดับ หลายราคา ผู้บริโภคสามารถเข้าถึงได้ง่าย จึงเป็นที่นิยม

### 2.2.1 สถาปัตยกรรมของระบบแอนดรอยด์

ระบบ Android [3] มีรากฐาน มาจากระบบ Linux แต่ละแอปพลิเคชันรวมถึงแอปพลิเคชันที่พัฒนาขึ้นจะทำงานบนกระบวนการของตัวเอง โดยที่แอปพลิเคชันจะอยู่ส่วนขั้นบนสุด (Application) และทำการเรียกใช้งานส่วนสนับสนุน ขั้นล่างๆ ลงไป จนถึงขั้นล่างสุด คือ Linux Kernel ซึ่งเป็นส่วนติดต่อ กับอุปกรณ์

ภาษาหลักที่ใช้พัฒนา App คือ ภาษา JAVA หากเคยศึกษาเกี่ยวกับภาษาที่สามารถเข้าใจได้ไม่ยาก โดยเฉพาะหากเคยศึกษาการเขียนโปรแกรม เชิงวัตถุ (OOP) อุปกรณ์ซึ่งการทำงานของระบบแอนดรอยด์แบ่งการทำงานเป็นชั้น ๆ ดังภาพที่ 2.2



ภาพที่ 2.2 โครงสร้างของระบบแอนดรอยด์

ที่มา:

[https://sites.google.com/a/thantong.ac.th/android-operating-system/\\_/rsrc/1470104840517/khorngsrang-khxng-xaen-dr-xyd-android/CloudAndroid0\\_r1\\_c1.jpg](https://sites.google.com/a/thantong.ac.th/android-operating-system/_/rsrc/1470104840517/khorngsrang-khxng-xaen-dr-xyd-android/CloudAndroid0_r1_c1.jpg)

จากภาพที่ 2.2 เป็นการทำงานของแต่ละชั้นของโครงสร้างระบบแอนดรอยด์ จะสังเกตได้ว่าจะแบ่งการทำงานเป็นชั้น ๆ แต่ละชั้น มีความเกี่ยวเนื่องกัน ชั้นบนสุดจะเป็นชั้นแอปพลิเคชันที่ติดกับผู้ใช้งาน จากนั้นจะเป็นชั้นอื่น ๆ ถัดลงมาที่ทำงานในลักษณะต่าง ๆ ออกแบบมาให้สามารถอธิบายแต่ละชั้นได้ดังนี้

### 1. ชั้นแอปพลิเคชัน (Application)

ชั้นนี้จะเป็นชั้นที่อยู่บนสุดของโครงสร้างสถาปัตยกรรม Android ซึ่งเป็นส่วนของแอปพลิเคชันที่พัฒนาขึ้นมาใช้งาน เช่น แอปพลิเคชันรับ/ส่ง อีเมล, SMS, ปฏิทิน, แผนที่, เว็บเบราว์เซอร์, รายชื่อผู้ติดต่อ เป็นต้น ซึ่งแอปพลิเคชันจะอยู่ในรูปแบบของไฟล์ .apk โดยที่รีไซเคิลได้ เร็คทอรี data/app

### 2. ชั้นแอปพลิเคชันเฟรมเวิร์ค (Application Framework Layer)

ในชั้นนี้จะอนุญาตให้นักพัฒนาสามารถเข้าเรียกใช้งาน โดยผ่าน API (Application Programming Interface) ซึ่ง Android ได้ออกแบบไว้ เพื่อลดความซับซ้อนในการใช้งาน application component โดยในชั้นนี้ประกอบด้วยแอปพลิเคชันเฟรมเวิร์คดังนี้

- View System เป็นส่วนที่ใช้ในการควบคุมการทำงานสาหรับการสร้างแอปพลิเคชัน เช่น lists, grids, text boxes, buttons และ embeddable web browser
- Location Manager เป็นส่วนที่จัดการเกี่ยวกับตำแหน่งของอุปกรณ์พกพาเคลื่อนที่
- Content Provider เป็นส่วนที่ใช้ควบคุมการเข้าถึงข้อมูลที่มีการใช้งานร่วมกัน (Share data) ระหว่างแอปพลิเคชันที่แตกต่างกัน เช่น ข้อมูลผู้ติดต่อ (Contact)
- Resource Manager เป็นส่วนที่จัดการข้อมูลต่างๆ ที่ไม่ใช่ส่วนของโค้ดโปรแกรม เช่น รูปภาพ, localized strings, layout ซึ่งจะอยู่ในไฟล์ res/
- Notification Manager เป็นส่วนที่ควบคุมอีเวนต์ (Event) ต่างๆ ที่แสดงบนแถบสถานะ (Status bar) เช่น ในกรณีที่ได้รับข้อความหรือสายที่ไม่ได้รับและการแจ้งเตือนอื่นๆ
- Activity Manager เป็นส่วนควบคุม Life Cycle ของแอปพลิเคชัน

### 3. ชั้นไลบรารี (Library)

Android ได้รวบรวมกลุ่มของไลบรารีต่างๆ ที่สำคัญและมีความจำเป็นเอาไว้มากmany เพื่ออำนวยความสะดวกให้กับนักพัฒนาและง่ายต่อการพัฒนาโปรแกรม โดยตัวอย่างของไลบรารีที่สำคัญเช่น

- System C library เป็นกลุ่มของไลบรารีมาตรฐานที่อยู่บนพื้นฐานของภาษา C ไลบรารี (libc) สำหรับ embedded system ที่มีพื้นฐานมาจาก Linux
- Media Libraries เป็นกลุ่มการทำงานมัลติมีเดีย เช่น MPEG4, H.264, MP3, AAC, AMR, JPG, และ PNG
- Surface Manager เป็นกลุ่มการจัดการรูปแบบหน้าจอ การวาดหน้าจอ 2D/3D library เป็นกลุ่มของกราฟิกแบบ 2 มิติ หรือ SGL (Scalable Graphics Library) และแบบ 3 มิติ หรือ OpenGL
- FreeType เป็นกลุ่มของบิตแมป (Bitmap) และเวคเตอร์ (Vector) สำหรับการрендเรอร์ (Render) ภาพ
- SQLite เป็นกลุ่มของฐานข้อมูล โดยนักพัฒนาสามารถใช้ฐานข้อมูลนี้เก็บข้อมูลแอปพลิเคชันต่างๆ ได้
- Browser Engine เป็นกลุ่มของการแสดงผลบนเว็บเบราว์เซอร์โดยอยู่บนพื้นฐานของ Webkit ซึ่งจะมีลักษณะคล้ายกับ Google Chrome

สำหรับการเรียกใช้แอปพลิเคชันต่างๆ ในชั้นไลบรารี จะไม่สามารถเรียกใช้แอปพลิเคชันในตัวเองได้ โดยจะต้องเรียกใช้แอปพลิเคชันในชั้นที่สูงกว่าเท่านั้น จึงจะสามารถเรียกใช้ได้ นอกจากนี้ในชั้นไลบรารีนี้ Android ยังแบ่งเป็นชั้นอยู่ที่เรียกว่า Android Runtime ซึ่งจะประกอบไปด้วย 2 ส่วน คือ Dalvik VM (Virtual Machine) และ Core Java Library

- Dalvik VM (Virtual Machine) ส่วนนี้ถูกเขียนด้วยภาษา Java เพื่อใช้เฉพาะการใช้งานในอุปกรณ์เคลื่อนที่ Dalvik VM จะแทรกต่างจาก Java VM (Virtual Machine) คือ Dalvik VM จะรันไฟล์ .dex ที่คอมไพล์มาจากไฟล์ .class และ .jar โดยมี tool ที่ชื่อว่า dx ทำหน้าที่ในการบีบอัดคลาส Java ทั้งนี้ไฟล์ .dex จะมีขนาดกะทัดรัดและเหมาะสมกับอุปกรณ์เคลื่อนที่มากกว่า .class เพื่อต้องการใช้พลังงานจากแบตเตอรี่อย่างมีประสิทธิภาพสูงสุด
- Core Java Library ส่วนนี้เป็นไลบรารีมาตรฐาน แต่ก็มีความแตกต่าง

#### 4. ชั้นลีนักซ์เครื่องใน (Linux Kernel)

ระบบ Android นั้นถูกสร้างบนพื้นฐานของระบบปฏิบัติการ Linux โดยในชั้นนี้จะมีฟังก์ชันการทำงานหลายๆ ส่วน แต่โดยส่วนมากแล้วจะเกี่ยวข้องกับฮาร์ดแวร์โดยตรง เช่น การจัดการหน่วยความจำ (Memory Management) การจัดการโปรเซส (Process Management) การเชื่อมต่อเครือข่าย (Networking) เป็นต้น

ส่วนของ Linux Kernel นี้จะทำหน้าที่เป็น Hardware Abstraction Layer กล่าวคือเป็นตัวกลางระหว่าง Hardware และ Software ที่อยู่ตัดขึ้นไป และทำหน้าที่บริหารจัดการทรัพยากรต่างๆ ของเครื่อง เช่น การจัดการหน่วยความจำ การจัดการโปรเซส ฯลฯ ผู้ผลิตอุปกรณ์สามารถ “พอร์ต” แอนดรอยด์ให้ไปรันบนฮาร์ดแวร์แบบต่างๆ ได้โดยเปลี่ยนแปลงในส่วนของ Linux Kernel

ผู้ใช้มือถือหรืออุปกรณ์แอนดรอยด์จะไม่เห็นว่ามีลีนักซ์อยู่ในเครื่อง และแอปพลิเคชันที่เราพัฒนา ก็จะไม่ได้เรียกไปยังลีนักซ์โดยตรง แต่ในฐานะนักพัฒนาเราจำเป็นต้องรับรู้ว่ามีลีนักซ์อยู่ เนื่องจากโปรแกรม utility บางตัวที่ Android SDK เตรียมมาให้จะติดต่อกับลีนักซ์ในอุปกรณ์แอนดรอยด์ เช่น โปรแกรม adb ที่ช่วยให้เราเรียกใช้คำสั่งเพื่อสำรวจระบบไฟล์ของเครื่อง ดูโปรเซสที่รันอยู่ในขณะนั้นและอื่นๆ ได้

### 2.3 ความรู้พื้นฐาน Firebase

เป็นหนึ่งในผลิตภัณฑ์ของ Google โดย Firebase คือ [4] Platform ที่รวมเครื่องมือต่างๆ สำหรับการจัดการในส่วนของ Backend หรือ Server side ซึ่งทำให้สามารถ Build Mobile Application ได้อย่างมีประสิทธิภาพ และยังลดเวลาและค่าใช้จ่ายของการทำ Server side หรือการวิเคราะห์ข้อมูลให้อีกด้วย โดยมีทั้งเครื่องมือที่ฟรี และเครื่องมือที่มีค่าใช้จ่าย

Firebase มีบริการให้ใช้หลายอย่าง สามารถแบ่งเป็นหมวดหมู่ดังนี้

#### 2.3.1 Build Better Apps

- Cloud Firestore – จัดเก็บและจึงข้อมูลระหว่างผู้ใช้และอุปกรณ์โดยใช้ฐานข้อมูล NoSQL ที่ออกแบบมาสำหรับคลาวด์ Cloud Firestore ให้การเชื่อมโยงแบบสอดและการสนับสนุนอффไลน์พร้อมกับการสืบค้นข้อมูลที่มีประสิทธิภาพ

- Authentication – จัดการผู้ใช้งานด้วยวิธีที่ง่ายและปลอดภัย Firebase Auth มีหลายวิธีในการตรวจสอบสิทธิ์รวมถึงอีเมลและรหัสผ่านผู้ให้บริการบุคคลที่สามเช่น Google หรือ Facebook และใช้ระบบบัญชีที่มีอยู่โดยตรง สร้างอินเทอร์เฟซหรือใช้ประโยชน์จากโอลูนชอร์ส UI ที่ปรับแต่งได้
- Hosting – ลดความซับซ้อนของเว็บไซต์ด้วยเครื่องมือที่สร้างขึ้นเฉพาะสำหรับเว็บแอปสมัยใหม่ เมื่ออัปโหลดเนื้อหาเว็บ เว็บเหล่านั้นจะส่งไปยัง CDN ทั่วโลกโดยอัตโนมัติและมี SSL พร้อมเพื่อให้ผู้ใช้ได้รับความปลอดภัย
- Realtime Database – Realtime Database คือฐานข้อมูลดังเดิมของ Firebase เป็นโอลูนที่มีประสิทธิภาพและมีเวลาแฝงต่ำสำหรับแอปบนอุปกรณ์เคลื่อนที่ ที่ต้องการสถานะการซึ่งกันระหว่างคลื่อนตัวแบบเรียลไทม์

### 2.3.2 Improve app quality

- Crashlytics – ลดเวลาในการแก้ไขปัญหาด้วยการเปลี่ยนข้อข้อห้องให้เป็นรายการปัญหาที่จัดการได้ รับข้อมูลเชิงลึกที่ชัดเจนและนำไปปฏิบัติได้ว่าปัญหาใดที่ต้องจัดการก่อนโดยเห็นผลกราฟของผู้ใช้ในแดชบอร์ด Crashlytics การแจ้งเตือนแบบเรียลไทม์จะช่วยให้มีความเสถียรแม่นยำขณะเดินทาง Crashlytics เป็นตัวรายงานข้อข้อห้องหลักของ Firebase
- Performance Monitoring – วินิจฉัยปัญหาประสิทธิภาพของแอปที่เกิดขึ้นบนอุปกรณ์ของผู้ใช้ ใช้การติดตามเพื่อตรวจสอบประสิทธิภาพของบางส่วนของแอปและดูมุมมองสรุปในคอนโซล Firebase อยู่หนึ่งในเวลาเริ่มต้นของแอปและตรวจสอบคำขอ HTTP โดยไม่ต้องเขียนโค้ดใด ๆ
- Test Lab – เรียกใช้การทดสอบอัตโนมัติและกำหนดเองสำหรับแอปบนอุปกรณ์เสมือนและจริงที่โฮสต์โดย Google ใช้ Firebase Test Lab ตลอดจนการพัฒนาเพื่อค้นหาจุดบกพร่องและความไม่สอดคล้อง

### 2.3.3 Grow your business

- Google Analytics – วิเคราะห์คุณลักษณะและพฤติกรรมของผู้ใช้ในแดชบอร์ดเดียว รับข้อมูลเชิงลึกแบบเรียลไทม์จากการรายงานหรือส่องออกข้อมูลเหตุการณ์ดิบไปยัง Google BigQuery สำหรับการวิเคราะห์ที่กำหนดเอง
- Remote Config – กำหนดวิธีการแสดงผลแอปสำหรับผู้ใช้แต่ละคน เปลี่ยนรูปแบบได้ตัวฟีเจอร์ที่ลับน้อยเรียกใช้การทดสอบ A / B ส่งมอบเนื้อหาที่กำหนดเองให้กับผู้ใช้บางรายหรือทำการอัปเดตอื่น ๆ โดยไม่ต้องปรับใช้เวอร์ชันใหม่ทั้งหมดนี้ทำได้จากคอนโซล Firebase ตรวจสอบผลกระทบของการเปลี่ยนแปลงและทำการปรับเปลี่ยนในเวลาไม่กี่นาที
- Cloud Messaging – ส่งข้อความและการแจ้งเตือนไปยังผู้ใช้ข้ามแพลตฟอร์มทั้ง Android, iOS และเว็บได้ฟรี สามารถส่งข้อความไปยังอุปกรณ์เดียวกันอุปกรณ์หรือหัวข้อเฉพาะหรือกลุ่มผู้ใช้ Firebase Cloud Messaging (FCM) ปรับขนาดเป็นแอปที่ใหญ่ที่สุดโดยส่งข้อความหลายแผนล้านข้อความต่อวัน

## 2.4 ความรู้พื้นฐาน Node.js

Node.js คือ [5] สภาพแวดล้อมการทำงานของภาษา JavaScript นอกเว็บเบราว์เซอร์ที่ทำงานด้วย V8 engine นั่นหมายความว่า เราสามารถใช้ Node.js ในการพัฒนาแอปพลิเคชันแบบ Command line และแอปพลิเคชัน Desktop หรือแม้แต่เว็บไซต์ได้ โดยที่ Node.js จะมี APIs ที่เราสามารถใช้สำหรับทำงานกับระบบปฏิบัติการ เช่น การรับค่าและการแสดงผล การอ่านเขียนไฟล์ และการทำงานกับเน็ตเวิร์ก เป็นต้น

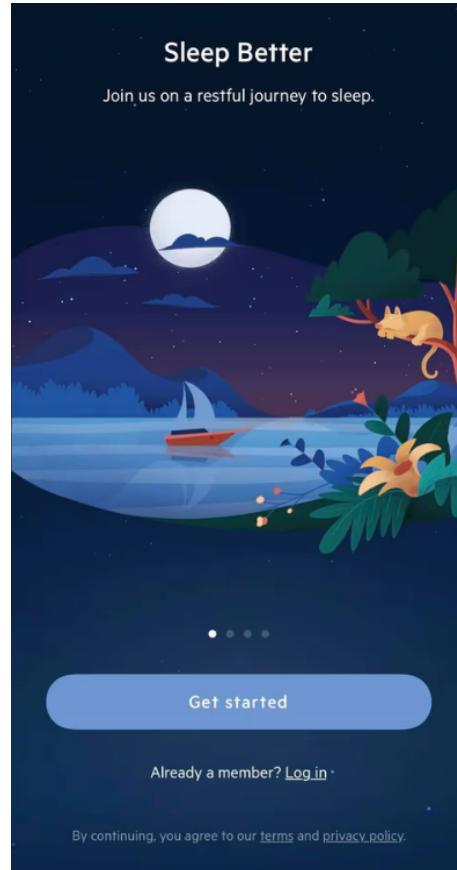
Node.js ยังเป็นโปรแกรมที่สามารถใช้ได้ทั้งบน Windows, Linux และ Mac OS นั่นหมายความว่าสามารถเขียนโปรแกรมในภาษา JavaScript และนำไปรันได้ทุกระบบปฏิบัติการที่สนับสนุนโดย Node.js นี้เป็นแนวคิดของการเขียนครั้งเดียวแต่ทำงานได้ทุกที่ (Write once, run anywhere) Node.js ยังมีข้อดีสำคัญในการติดตั้งไลบรารีดังนี้

### 2.4.1 NPM Package management

NPM คือ [5] ระบบจัดการแพ็คเกจ โมดูล หรือไลบรารีของภาษา JavaScript และ Node.js ใช้ npm สำหรับเป็นตัวจัดการแพ็จในปัจจุบัน npm มีมากกว่า 1 ล้านไลบรารีบน npmjs จากนักพัฒนาทั่วโลกที่สามารถนำมายัง Node.js โดยไลบรารีทั้งหมดบน npm ที่เป็นแบบสาธารณะ (Public repository) จะเป็นซอฟต์แวร์แบบโอเพ่นซอร์สและฟรี ในปัจจุบัน npm อยู่ภายใต้การดูแลของ Github และ Microsoft

## 2.5 แอปพลิเคชันที่เกี่ยวข้อง

Relax Melodies [6] เป็นแอปพลิเคชันช่วยให้ผ่อนคลายด้วยเสียงต่าง ๆ ที่มีให้เลือกฟังมากมาย ทำให้นอนหลับได้ง่ายขึ้น หมายความว่า สำหรับคนนอนไม่หลับ นอกจากตัวแอปพลิเคชัน จะมีเสียงต่าง ๆ ที่ช่วยให้หลับง่ายแล้วนั้น ยังสามารถสร้างเสียงขึ้นมาเองได้



ภาพที่ 2.3 หน้าจอแอปพลิเคชัน Relax Melodies

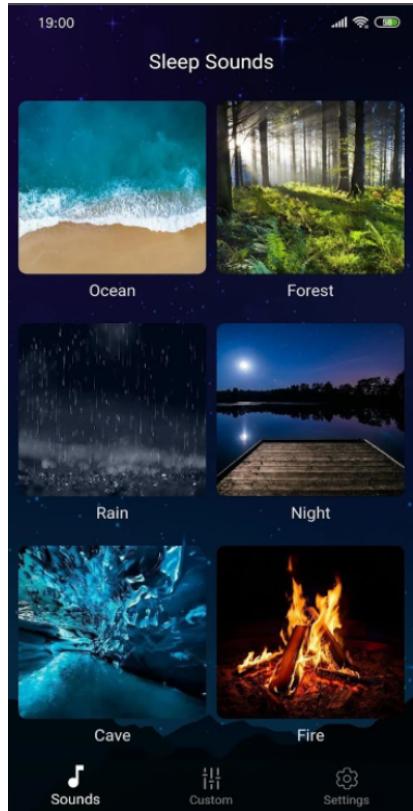
ที่มา:

[https://img-prod.api-onscene.com/cdn-cgi/image/format=auto%2Cwidth=1600%2Cheight=900/https://sls-prod.api-onscene.com/partner\\_files/trueidintrend/189029/1\\_4054.jpg](https://img-prod.api-onscene.com/cdn-cgi/image/format=auto%2Cwidth=1600%2Cheight=900/https://sls-prod.api-onscene.com/partner_files/trueidintrend/189029/1_4054.jpg)

คุณสมบัติการทำงานของแอปพลิเคชัน Relax Melodies มีดังนี้

- ใช้งานง่าย มีภาพประกอบเคลื่อนไหวสวยงาม
- สามารถสร้างเสียงขึ้นมาเองได้
- มีเรื่องสั้นที่เป็นภาษาอังกฤษให้ฟัง ช่วยให้ผ่อนคลายและฝึกความรู้ภาษาอังกฤษไปในตัว
- สามารถระบุข้อมูลอื่น ๆ ของเราได้ เช่น น้ำหนัก ความดันเลือด ระดับน้ำตาลในเลือด เป็นต้น
- แจ้งเตือนเวลาอนของเราได้
- สามารถตั้งเวลาปิดการเล่นเสียงได้
- สามารถตั้งเวลาปลุกได้

Sleep Sounds - relaxing sounds [7] แอปพลิเคชันสามารถช่วยให้ สามารถผ่อนคลายและนอนหลับไปกับเสียงการนอน เสียงธรรมชาติ เสียงฝน เสียงการทำสมาธิ และเสียงสีขาว และอื่น ๆ อีกมากมาย



ภาพที่ 2.4 หน้าจอแอปพลิเคชัน Sleep Sounds - relaxing sounds

ที่มา:

<https://play-lh.googleusercontent.com/4aO1zotY0u5VfVN4FsaAAHOwfND91FtpBMfqqisaW2AfFXtk3jLF35iTuwubPKYGqnS3=w2560-h1440-rw>

คุณสมบัติการทำงานของแอปพลิเคชัน Sleep Sounds - relaxing sounds มีดังนี้

- สร้างมิกซ์เสียงของคุณเองเพื่อผ่อนคลายและนอนหลับ
- ปรับระดับเสียงสำหรับแต่ละเสียงเมื่อสร้างมิกซ์ของคุณเอง
- เล่นเสียงในพื้นหลัง
- ตั้งเวลาปิดเครื่องเพื่อหยุดเสียงโดยอัตโนมัติ
- สามารถตั้งค่าการแจ้งเตือนเมื่อเข้าสู่โหมด睡眠

- ไม่ต้องใช้เครื่อข่าย
- การออกแบบที่สวยงามและเรียบง่าย
- เสียงผ่อนคลายคุณภาพสูง
- เสียงการนอนหลับฟรี

Sleepo: เสียงผ่อนคลาย, นอนหลับ [8] มีคอลเลกชันของเสียง HD อันมากมายที่สามารถนำมาผสมผสานให้เป็นบรรยากาศผ่อนคลายอันสมบูรณ์แบบได้ สามารถเลือกฟังเสียงที่แตกต่างกันของฝน เสียงธรรมชาติ เสียงเมือง white noise หรือเครื่องดนตรี บันทึกมิกซ์เสียงที่ชื่นชอบ สามารถตั้งเวลาเครื่องจับเวลาเพื่อหยุดการทำงานของแอปแบบอัตโนมัติ Sleepo และสามารถใช้งานโดยไม่เชื่อมต่อกับอินเทอร์เน็ต



ภาพที่ 2.5 หน้าจอแอปพลิเคชัน Sleepo: เสียงผ่อนคลาย, นอนหลับ

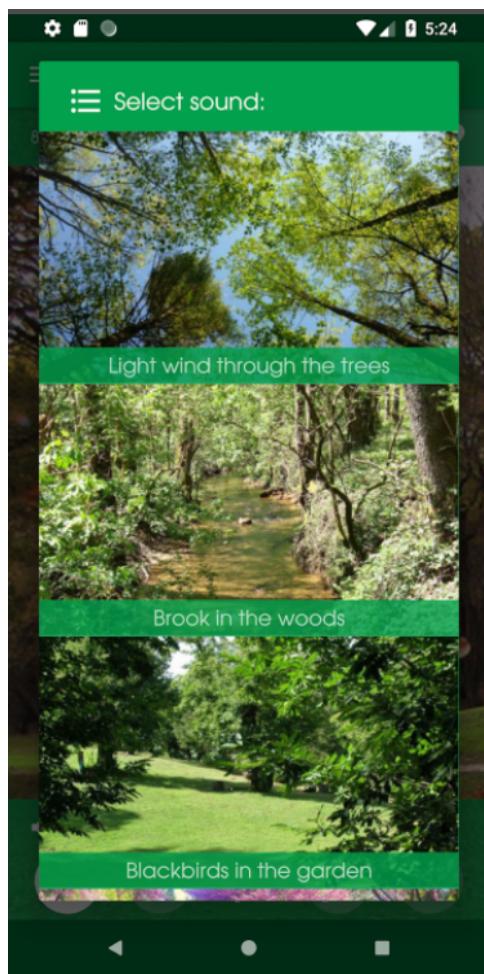
ที่มา:

[https://play-lh.googleusercontent.com/qpun2ElwLiu3fqA2qhCbut31erWHn9yuV1NlM3ed4RRFTnPZGd-XTxJhpORwnPl\\_u\\_g=w2560-h1440-rw](https://play-lh.googleusercontent.com/qpun2ElwLiu3fqA2qhCbut31erWHn9yuV1NlM3ed4RRFTnPZGd-XTxJhpORwnPl_u_g=w2560-h1440-rw)

คุณสมบัติการทำงานของแอปพลิเคชัน Sleepo: เสียงผ่อนคลาย, นอนหลับ มีดังนี้

- ผ่อนคลายเสียง
- เครื่องจับเวลาแบบกำหนดเองได้
- เสียงคุณภาพสูง
- สร้างรายการโปรด
- ทำงานแบบอฟไลน์

ผ่อนคลายชนบท - เสียงธรรมชาติ [9] สามารถตั้งเวลาและใช้งานแอปพลิเคชันในพื้นหลังหรือปิดหน้าจอ ที่เหมาะสมสำหรับการผ่อนคลายหลังจากวันที่เครียด ช่วยในการการทำสมาธิและการนอนหลับ



ภาพที่ 2.6 หน้าจอแอปพลิเคชัน ผ่อนคลายชนบท - เสียงธรรมชาติ

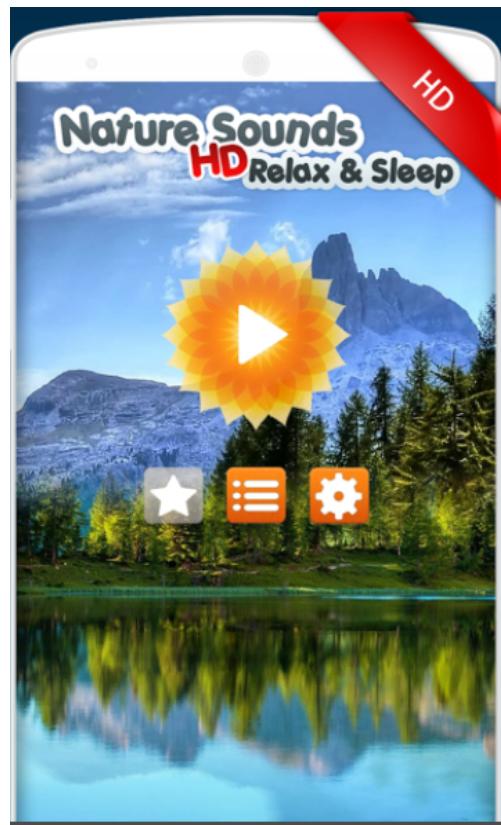
ที่มา:

[https://play-lh.googleusercontent.com/-O08xB01hZaJNYv9IpiQdESZB-6zo\\_n3QhfXASjmaZtUCXEWNGc8zz34pOqm6wywgg=w2560-h1440-rw](https://play-lh.googleusercontent.com/-O08xB01hZaJNYv9IpiQdESZB-6zo_n3QhfXASjmaZtUCXEWNGc8zz34pOqm6wywgg=w2560-h1440-rw)

คุณสมบัติการทำงานของแอปพลิเคชัน ผ่อนคลายชนบท - เสียงธรรมชาติ มีดังนี้

- เล่นเสียงหลากหลายประเภทของภูมิทัศน์ที่แตกต่างกัน
- ตั้งเวลาปิดเสียง
- เล่นในพื้นหลัง
- เพิ่มเสียงเป็นรายการโปรด

เสียงธรรมชาติ [10] แอปพลิเคชันสำหรับการพักผ่อนนอนหลับนั้นสามารถใช้ในการนอนหลับ แอปพลิเคชันสามารถทำงานในพื้นหลังหรือเมื่อมีการปิดหน้าจอโดยการเล่นเสียงของธรรมชาติ มีเสียงผ่อนคลายหลากหลายเสียง เช่น เสียงน้ำ, เสียงป่า, เสียงฝน, เสียงทะเล, เสียงสีขาว



ภาพที่ 2.7 หน้าจอแอปพลิเคชัน เสียงธรรมชาติ

ที่มา:

[https://play-lh.googleusercontent.com/LOdUcGzU\\_5ovdT\\_JRwH2Z7Cu0BYoU-eyVwxPNyKwNFoiOXcGMPq6GVyvMeTUQrtN5iY=w2560-h1440-rw](https://play-lh.googleusercontent.com/LOdUcGzU_5ovdT_JRwH2Z7Cu0BYoU-eyVwxPNyKwNFoiOXcGMPq6GVyvMeTUQrtN5iY=w2560-h1440-rw)

คุณสมบัติการทำงานของแอปพลิเคชัน เสียงบรรยายชาติ มีดังนี้

- ธรรมชาติและเสียงจริงของธรรมชาติที่มีคุณภาพสูง
- สนับสนุน 40 ภาษา
- ผสมเสียงของธรรมชาติกับเสียงผ่อนคลายและเงียบสงบ ๆ
- ภาพเต็มหน้าจอของภูมิทัศน์ในความละเอียด HD,
- ความสามารถในการตั้งค่าเสียงธรรมชาติเสียงเรียกเข้า
- ความสามารถในการเล่นในพื้นหลัง
- เสียงหารของมหาสมุทรเสียงอันงดงามของพายุลมและเสียงคลื่นและเสียงอื่น ๆ ของธรรมชาติ
- เพลงผ่อนคลายเสียงเพื่อการนอนหลับและการทำสมาธิ (ดนตรีบำบัด, การรักษาด้วยเสียง)

แอปพลิเคชัน Relax Sound ที่ผู้พัฒนา พัฒนาขึ้นนั้นมีความแตกต่างจากแอปพลิเคชันที่เกี่ยวข้องคือ สามารถตั้งค่าเพลงที่พึงพอใจได้

## บทที่ 3

### การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และออกแบบระบบ แบ่งออกเป็น 5 ขั้นตอนเพื่อให้เห็นการดำเนินงานอย่างมีระบบ ในหัวข้อแรกจะนำเสนอภาพรวมของระบบก่อนจะนำเสนอเอกสารแสดงความต้องการของระบบซึ่งจะทำให้เห็นถึงมาของหน้าต่าง ๆ ส่วนหัวข้อที่สามจะแสดง แผนภาพการการทำงานของระบบโดยใช้ UML diagram ซึ่งประกอบไปด้วย แผนภาพที่แสดง การทำงานของผู้ใช้ระบบ แผนภาพที่ใช้แสดงคลาส และ การเกิดปฏิสัมพันธ์ระหว่างออบเจกต์ เพื่อแสดงรายละเอียดของระบบก่อนนำไปเขียนคำสั่งด้วยภาษาโปรแกรมในบทต่อไป

3.1 โครงสร้างภาพรวมของระบบ (System Architecture) เป็นการออกแบบภาพรวมและเทคโนโลยีของระบบ

3.2 การวิเคราะห์ความต้องการของระบบ (System Requirements) คือ ความต้องการหรือสิ่งที่ระบบควรจะทำ หรือหน้าที่หลักของระบบที่จะต้องทำ

3.3 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface Design)

3.4 แผนภาพยุสเคส (Use Case Diagram) เป็นแผนภาพที่ใช้แสดงให้ทราบว่าระบบทำงาน

หรือมีหน้าที่ใดบ้าง

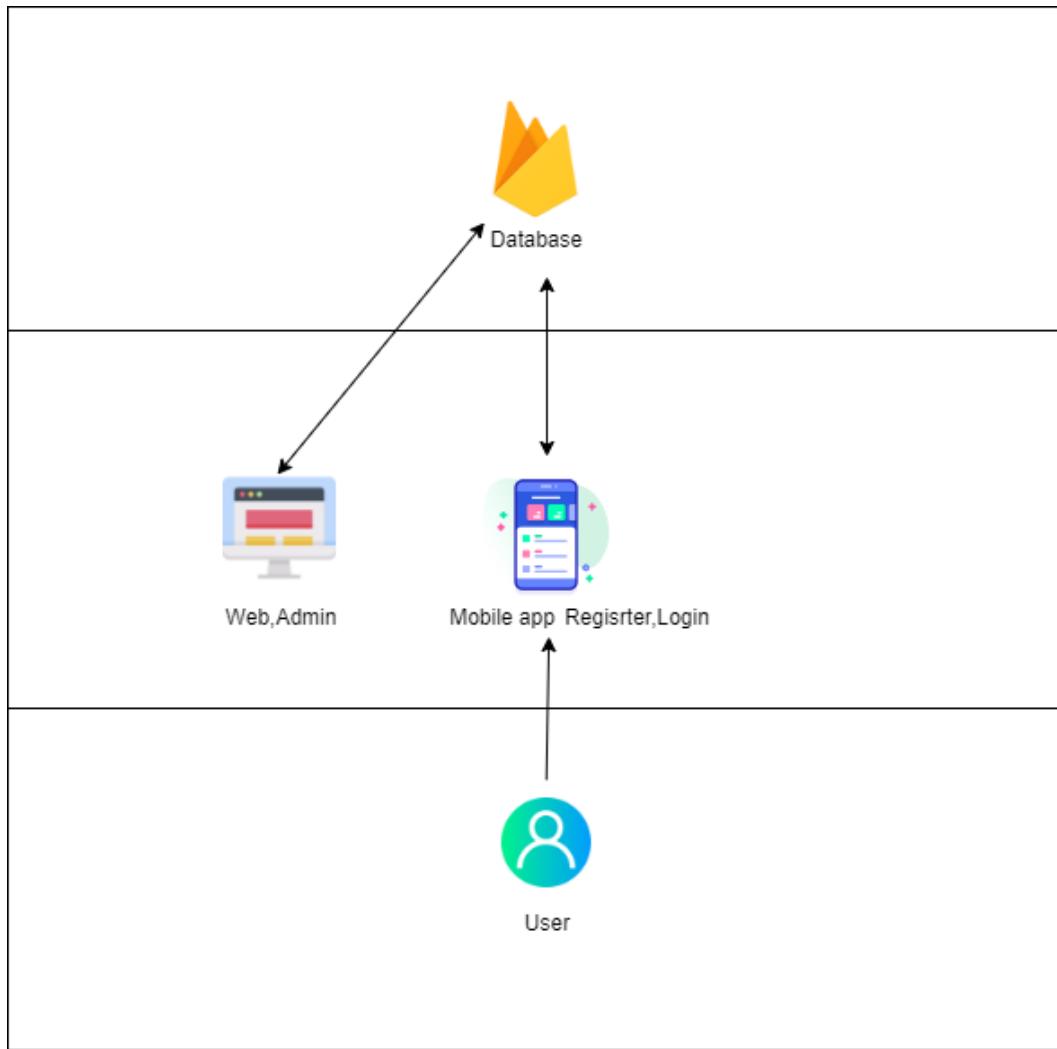
3.5 แผนภาพคลาส (Class Diagram) เป็นแผนภาพที่ใช้แสดง Class และความสัมพันธ์ระหว่าง Class

3.6 แผนภาพลำดับการทำงาน (Sequence Diagram) เป็นแผนภาพที่ใช้แสดงให้เห็นถึงการ

ตอบโต้ข้อมูลระหว่างคลาส เรียงตามลำดับของเวลาที่เกิดเหตุการณ์การน้อยไปมาก

### 3.1 โครงสร้างภาพรวมของระบบ (System Architecture)

การออกแบบ System architecture แสดงภาพรวมและเทคโนโลยีของแอปพลิเคชันเสียงเพื่อการผ่อนคลาย มีรายละเอียดดังภาพที่ 3.1



ภาพที่ 3.1: System architecture แอปพลิเคชันเสียงเพื่อการผ่อนคลาย

จากภาพที่ 3.1 สามารถอธิบายโครงสร้างระบบดังนี้

1. Firebase ใช้เป็นฐานข้อมูลในการเก็บข้อมูลผู้ใช้โดยผู้ใช้สมัครสมาชิกผ่าน

Mobile App

2. Mobile App สำหรับผู้ใช้ที่สมัครสมาชิกเข้าใช้งานแอปข้อมูลต่าง ๆ ของผู้ใช้จะเก็บบน

## Firebase

3. User สำหรับผู้ใช้ที่ยังไม่เป็นสมาชิกต้องสมัครเข้าใช้งานผ่าน Mobile App

4. Web สำหรับผู้ดูแลระบบเพิ่มเพลงไว้ที่ฐานข้อมูลและแสดงเพลงที่ผู้ดูแลระบบเพิ่มบน

โมบายแอปพลิเคชัน

### 3.2 การวิเคราะห์ความต้องการของระบบ (System Requirements)

#### 3.2.1 Functional Requirements

##### 1. ผู้ดูแลระบบ

- สามารถเพิ่มเพลงได้
- สามารถเพิ่มรูปได้

##### 2. สมาชิก

- สามารถสมัครสมาชิกและเข้าสู่ระบบได้
- สามารถตั้งเวลาเข้าอนได้
- สามารถเพิ่มเพลงเป็นรายการโปรดได้
- สามารถลบเพลงที่เป็นรายการโปรดได้
- สามารถดูเพลงย้อนหลังได้
- สามารถตั้งเวลาหยุดเพลงได้

#### 3.2.2 Non-functional Requirements

- พื้นแบบขอฟ์เลน์ได้เมื่อเข้าสู่ระบบ

### 3.3 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface Design)

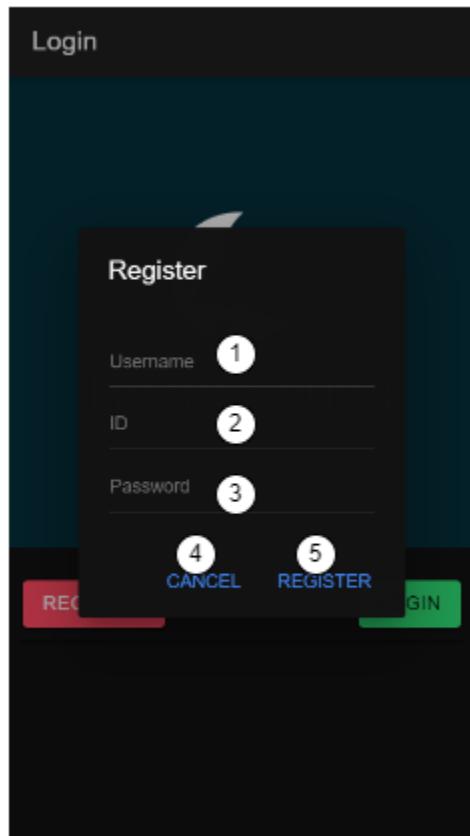
การออกแบบหน้าจอผู้ใช้งานแอปพลิเคชัน แบ่งเป็น 4 ส่วนดังนี้

- หน้าสมัครสมาชิกและหน้าเข้าสู่ระบบ
- หน้าหลักในการใช้งานแอปพลิเคชัน
- หน้าแสดงรายการโปรด
- ส่วนหน้าเว็บสำหรับผู้ดูแลระบบ

### 3.3.1 หน้าสมัครสมาชิกและหน้าเข้าสู่ระบบ

หน้าสมัครสมาชิกและเข้าสู่ระบบมีดังนี้

#### 1. หน้าจอสมัครสมาชิก มีดังนี้

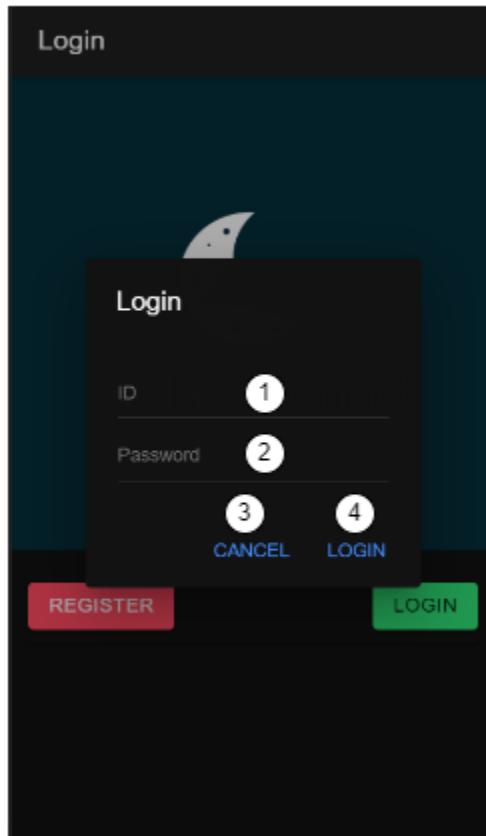


ภาพที่ 3.2: หน้าจอสมัครสมาชิก

จากภาพที่ 3.2 หน้าจอสมัครสมาชิก มี 5 ส่วนดังนี้

- ส่วนที่ 1 ช่องสร้างชื่อผู้ใช้
- ส่วนที่ 2 ช่องสร้างรหัสผ่าน
- ส่วนที่ 3 ช่องสร้างรหัสผ่านใหม่
- ส่วนที่ 4 ปุ่มยกเลิก
- ส่วนที่ 5 ปุ่มยืนยันการสมัครสมาชิกผู้ใช้

2. หน้าจอเข้าสู่ระบบสมาชิก มีดังนี้



ภาพที่ 3.3: หน้าจอเข้าสู่ระบบสมาชิก

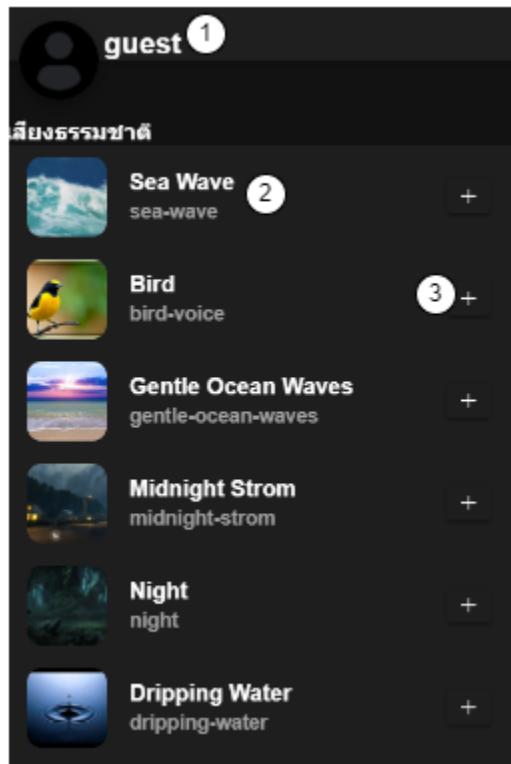
จากภาพที่ 3.3 หน้าจอเข้าสู่ระบบสมาชิก มี 4 ส่วนดังนี้

- ส่วนที่ 1 ช่องกรอกไอดีผู้ใช้
- ส่วนที่ 2 ช่องกรอกรหัสผ่านผู้ใช้
- ส่วนที่ 3 ปุ่มยกเลิก
- ส่วนที่ 4 ปุ่มยืนยันเข้าสู่ระบบผู้ใช้

### 3.3.2 หน้าหลักในการใช้งานแอปพลิเคชัน

หน้าหลักที่ให้สามารถใช้งานแอปพลิเคชัน มีดังนี้

#### 1. หน้าแสดงรายการเพลง

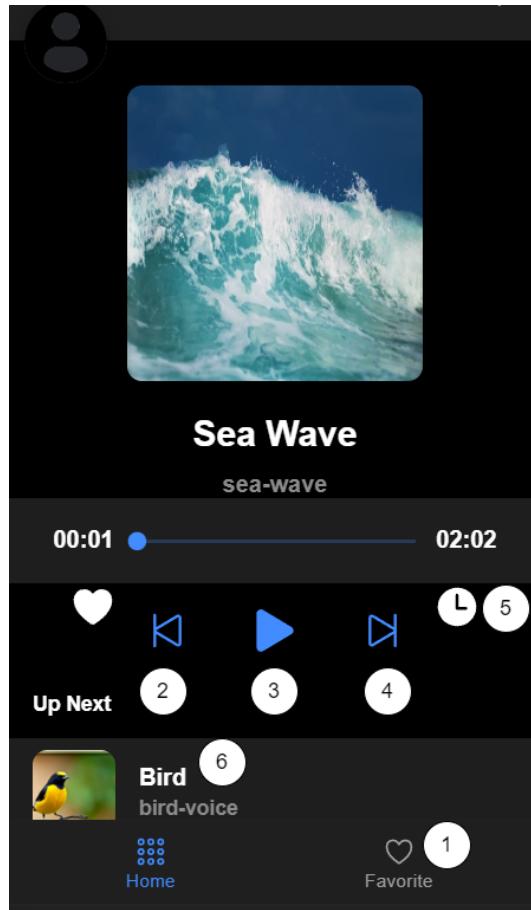


ภาพที่ 3.4: หน้าแสดงรายการเพลง

จากภาพที่ 3.4 หน้าแสดงรายการเพลง มี 3 ส่วนดังนี้

- ส่วนที่ 1 ข้อผู้ใช้
- ส่วนที่ 2 รายการเพลง
- ส่วนที่ 3 ปุ่มเพิ่มเพลงเป็นรายการโปรด

## 2. หน้าเล่นเสียงเพลง

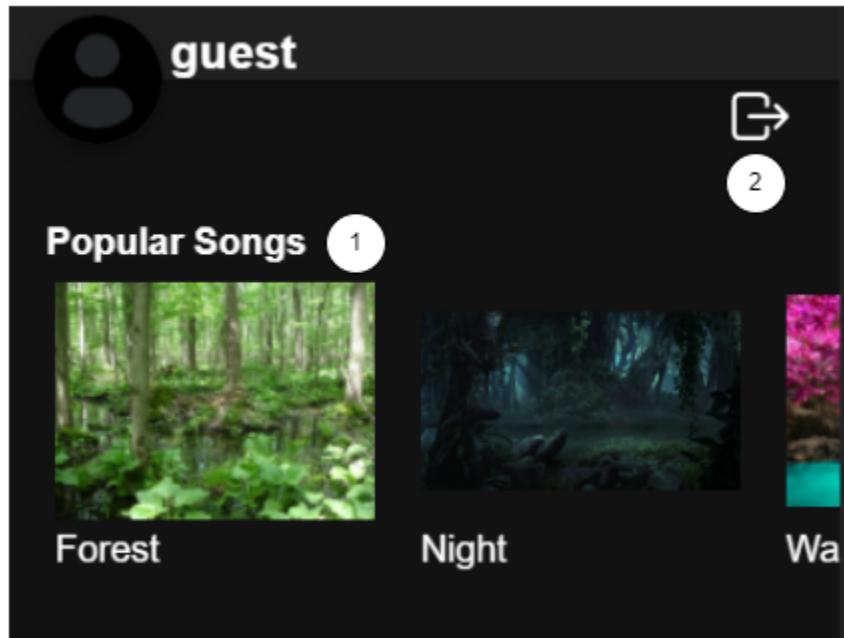


ภาพที่ 3.5: หน้าเล่นเสียงเพลง

จากภาพที่ 3.5 หน้าแสดงรายการเพลง มี 6 ส่วนดังนี้

- ส่วนที่ 1 ปุ่มดูรายเพลงโปรด
- ส่วนที่ 2 ปุ่มกลับไปเพลงก่อนหน้า
- ส่วนที่ 3 ปุ่มกดเล่นเพลง-หยุดเล่นเพลง
- ส่วนที่ 4 ปุ่มไปเพลงถัดไป
- ส่วนที่ 5 ปุ่มตั้งเวลาปิดเสียง
- ส่วนที่ 6 ส่วนแสดงเพลงถัดไป

### 3.หน้าแสดงรายเพลงยอดฮิต

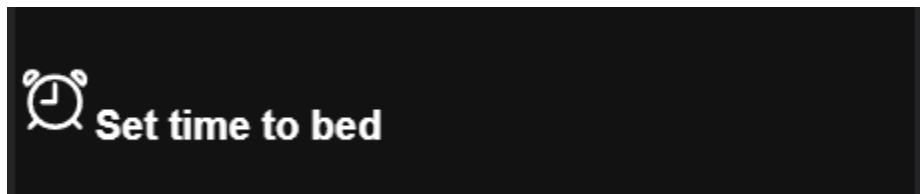


ภาพที่ 3.6: หน้าแสดงรายเพลงยอดฮิต

จากภาพที่ 3.6 หน้าแสดงรายการเพลง มี 2 ส่วนดังนี้

- ส่วนที่ 1 ส่วนแสดงเพลงที่มีการฟังมากที่สุด
- ส่วนที่ 2 ปุ่มออกจากระบบ

### 4.หน้าตั้งเวลาเข้านอน



ภาพที่ 3.7: หน้าตั้งเวลาเข้านอน

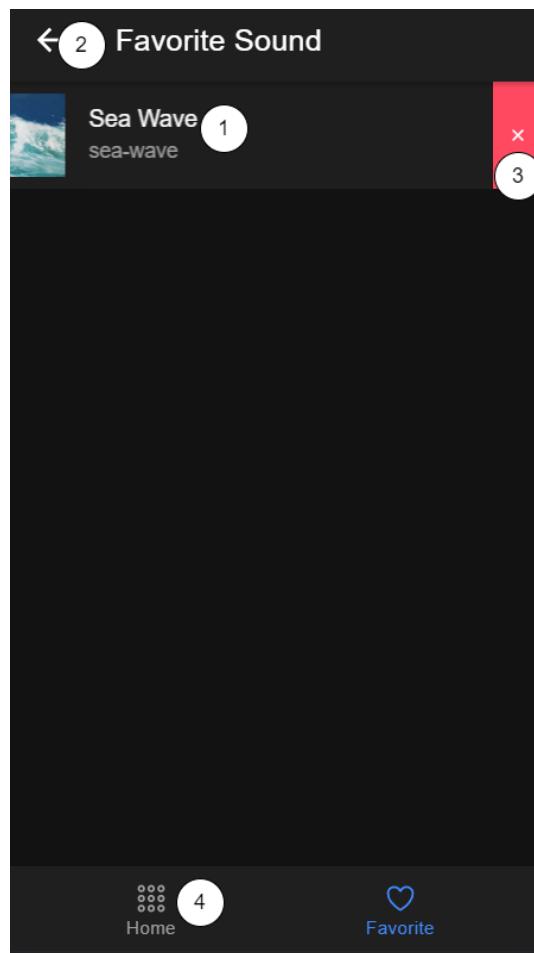
จากภาพที่ 3.7 หน้าแสดงรายการเพลง มีดังนี้

- ตั้งเวลาเข้านอน

### 3.3.3 หน้าแสดงรายการโปรด

หน้าแสดงเพลงรายโปรด มีดังนี้

#### 1. หน้ารายการโปรด



ภาพที่ 3.8: หน้าแสดงรายการโปรด

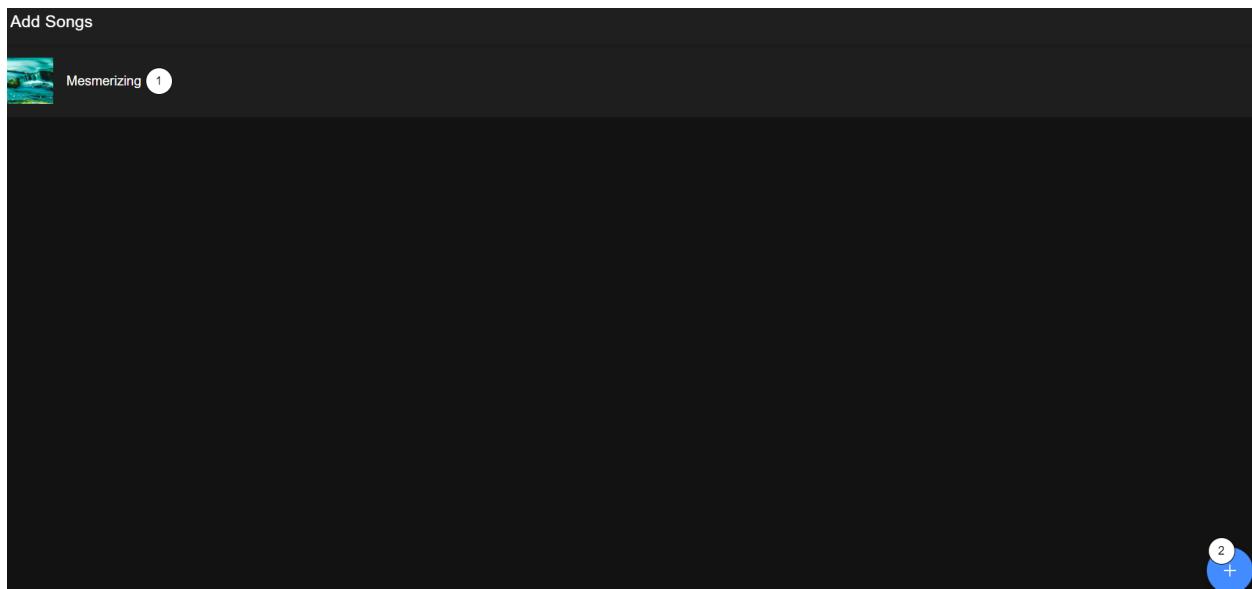
จากภาพที่ 3.8 หน้าแสดงรายการโปรด มีดังนี้

- ส่วนที่ 1 รายเพลงโปรด
- ส่วนที่ 2 ปุ่มกลับ
- ส่วนที่ 3 ปุ่มลบรายการเพลงโปรด
- ส่วนที่ 4 ปุ่มหน้าหลัก

### 3.3.4 ส่วนหน้าเว็บสำหรับผู้ดูและระบบ

ส่วนหน้าเว็บผู้ดูและระบบ มีดังนี้

#### 1. หน้าแสดงเพลงที่เพิ่ม

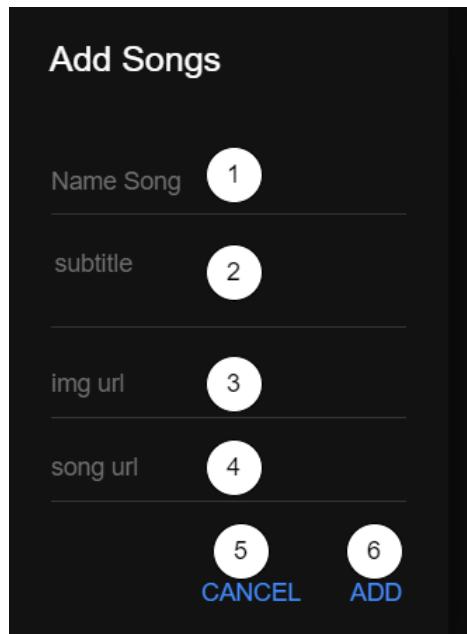


ภาพที่ 3.9: หน้าแสดงเพลงที่เพิ่ม

จากภาพที่ 3.9 หน้าแสดงรายการเพลงที่เพิ่ม มีดังนี้

- ส่วนที่ 1 รายเพลงที่เพิ่ม
- ส่วนที่ 2 ปุ่มเพิ่มเพลง

## 2. ส่วนเพิ่มเพลง

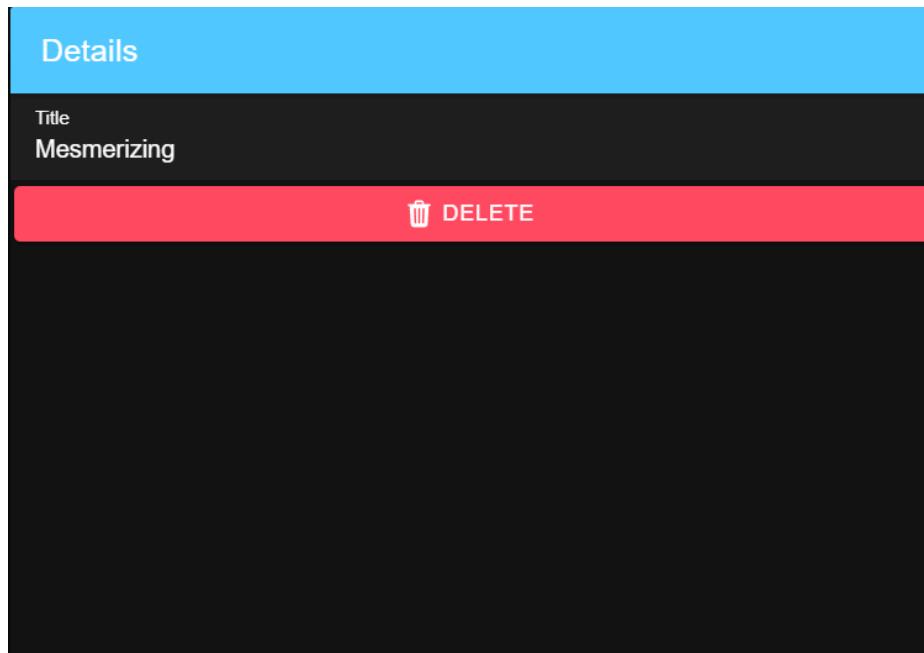


ภาพที่ 3.10: ส่วนเพิ่มเพลง

จากภาพที่ 3.10 ส่วนเพิ่มเพลง มีดังนี้

- ส่วนที่ 1 เพิ่มชื่อเพลง
- ส่วนที่ 2 เพิ่มชื่อร้องเพลง
- ส่วนที่ 3 เพิ่มที่อยู่รูปภาพ
- ส่วนที่ 4 เพิ่มที่อยู่เพลง
- ส่วนที่ 5 ปุ่มยกเลิก
- ส่วนที่ 6 ปุ่มเพิ่มเพลง

### 3. ส่วนลับเพลง



ภาพที่ 3.11: ส่วนลับเพลง

จากภาพที่ 3.11 ส่วนลับเพลง มีดังนี้

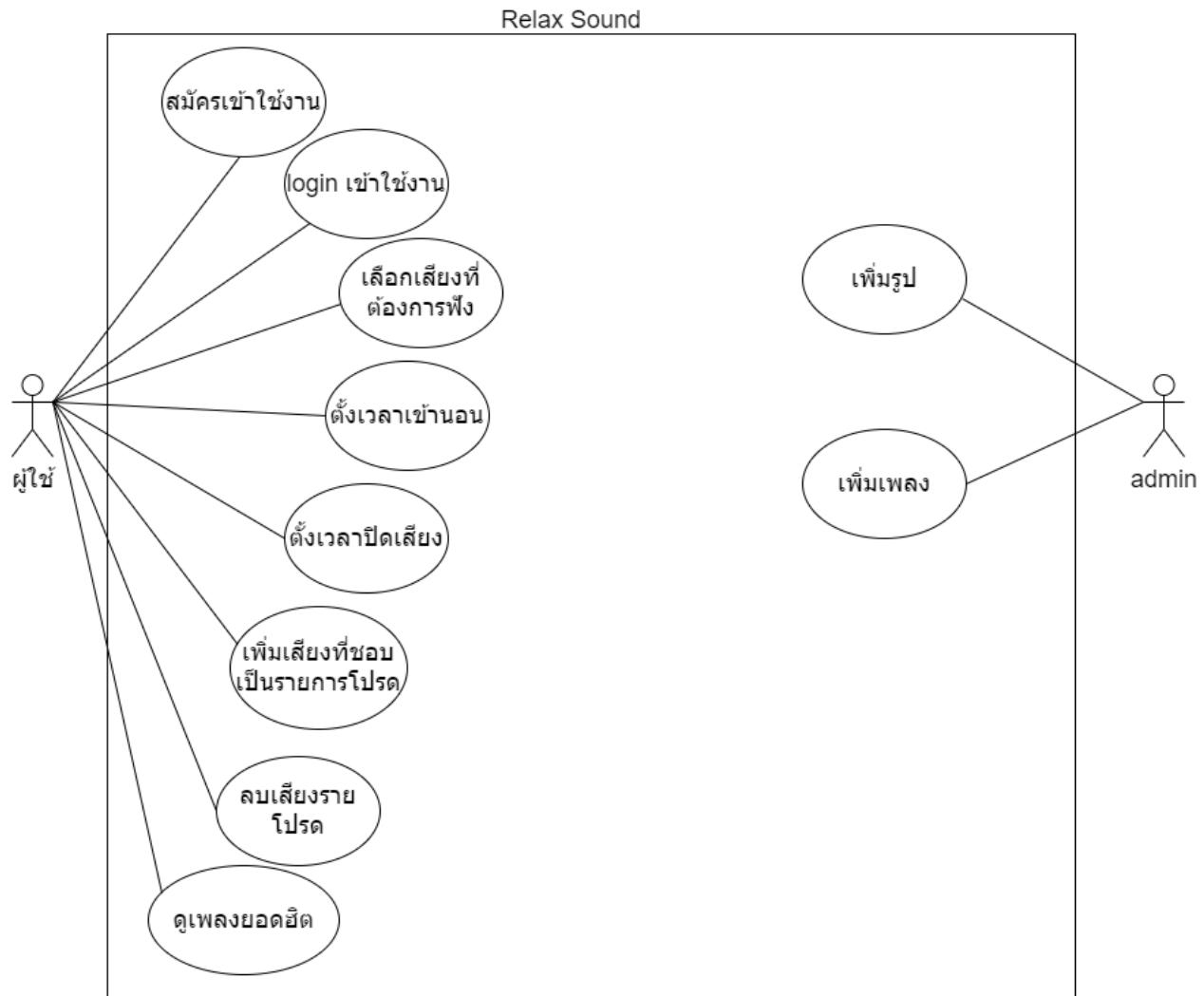
- ส่วนที่ 1 เพิ่มชื่อเพลง
- ส่วนที่ 2 เพิ่มชื่อร้องเพลง

### 3.4 แผนภาพยูสเคส (Use Case Diagram)

Use case Diagram คือ แผนภาพที่ใช้แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน และแสดงให้เห็นถึงส่วนประกอบทั้งหมด หรือ ภาพรวมของระบบ ที่เกิดขึ้นในแอปพลิเคชัน

ตารางที่ 3.1 สัญลักษณ์ Use Case Diagram

สัญลักษณ์	ความหมาย
Use case	หน้าที่ที่ระบบต้องกระทำ
	ผู้ที่กระทำการกับระบบ อาจเป็นผู้ที่ทำการส่งข้อมูล, รับข้อมูล หรือแลกเปลี่ยนข้อมูลกับระบบนั้นๆ
—	เส้นเชื่อมระหว่าง Actor กับ Use case
<b>System Name</b>	เส้นแบ่งขอบเขตระหว่างระบบกับผู้กระทำการต่อระบบ
----- <<include>> ----- ➔	กิจกรรมใน Use Case หนึ่ง อาจจะถูกผนวกเข้าไปรวมกับกิจกรรมของอีก Use Caseหนึ่ง
----- <<extend>> ----- ➔	Use Case หนึ่งทำหน้าที่ตามปกติ เมื่อเกิดเหตุการณ์ผิดปกติขึ้นจะต้องทำหน้าที่พิเศษเพิ่ม



ภาพที่ 3.12: ยูสเคสของแอปพลิเคชันเสียงเพื่อการผ่อนคลาย

ตารางที่ 3.2 อธิบาย Use case หน้าที่ระบบแอป ในภาพที่ 3.8

Use Case	คำอธิบาย
สมัครเข้าใช้งาน	สำหรับสมาชิกจะเข้าใช้งานต้องสมัครเข้าใช้งาน
Login เข้าใช้งาน	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน
เลือกเสียงที่ต้องการฟัง	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถเลือกเสียงที่ต้องการฟัง
ตั้งเวลาอน	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถตั้งเวลาอน สำหรับการเตือนเวลาเข้าอน
เพิ่มเสียงที่ชอบเป็นรายการโปรด	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถเพิ่มเสียงที่ชอบเป็นรายการโปรด
ดูเพลงยอดฮิต	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถดูเพลงที่มีการฟังมากที่สุด
ตั้งเวลาปิดเสียง	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถตั้งเวลาปิดเสียงได้
ลบเสียงรายการโปรด	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถลบเสียงที่เป็นรายการโปรดได้
เพิ่มรูป	สำหรับผู้ดูและระบบสามารถเพิ่มรูปปกเพลงได้
เพิ่มเพลง	สำหรับผู้ดูและระบบสามารถเพิ่มเพลงได้

ตารางที่ 3.3 Use Case สมัครเข้าใช้งาน

Use Case Title: สมัครเข้าใช้งาน	Use Case Id : 1
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : สมาชิกต้องสมัครเข้าใช้งานเมื่อใช้งาน แอปพลิเคชันครั้งแรก	

ตารางที่ 3.4 Use Case Login เข้าใช้งาน

Use Case Title: Login เข้าใช้งาน	Use Case Id : 2
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : สมาชิกจะเข้าใช้งานโดยกดปุ่ม Login	

ตารางที่ 3.5 เลือกเสียงที่ต้องการฟัง

Use Case Title: เลือกเสียงที่ต้องการฟัง	Use Case Id : 3
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : สมาชิกเลือกฟังเสียงที่ต้องการฟัง	

ตารางที่ 3.6 เดือนเวลาnoon

Use Case Title: เดือนเวลาnoon	Use Case Id : 4
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : สมาชิกเลือกดึงเวลาnoon	

ตารางที่ 3.7 เพิ่มเสียงที่ขอบเป็นรายการโปรด

Use Case Title: เพิ่มเสียงที่ขอบเป็นรายการโปรด	Use Case Id : 5
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : สมาชิกเพิ่มเสียงที่ขอบเป็นรายการโปรด	

ตารางที่ 3.8 ดูเพลงยอดธิค

Use Case Title: ดูเพลงยอดธิค	Use Case Id : 6
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : สมาชิกดูเพลงที่ฟังมากที่สุด	

ตารางที่ 3.9 ตั้งเวลาปิดเสียง

Use Case Title: ตั้งเวลาปิดเสียง	Use Case Id : 7
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : สมาชิกตั้งเวลาปิดเสียงเพลง	

ตารางที่ 3.10 ลบเสียงรายการโปรด

Use Case Title: ลบเสียงรายการโปรด	Use Case Id : 8
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : สมาชิกลบเสียงที่ชอบจากการโปรด	

## ตารางที่ 3.11 เพิ่มรูป

Use Case Title: เพิ่มรูป	Use Case Id : 9
Primary Actor : ผู้ดูแลระบบ	
Main Flow : ผู้ดูแลระบบเพิ่มรูปปกเพลง	

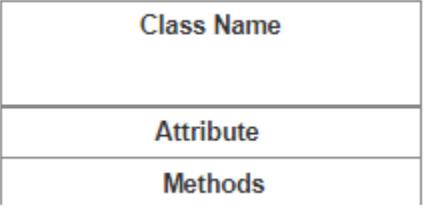
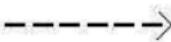
## ตารางที่ 3.12 เพิ่มเพลง

Use Case Title: เพิ่มเพลง	Use Case Id : 10
Primary Actor : ผู้ดูแลระบบ	
Main Flow : ผู้ดูแลระบบเพิ่มเพลง	

### 3.5 แผนภาพคลาส (Class Diagram)

Class Diagram คือแผนภาพที่ใช้แสดง Class และความสัมพันธ์ในแต่ต่าง ๆ (Relation) ระหว่าง Class เหล่านั้น สัญลักษณ์ที่ใช้การเขียน Class Diagram แสดงในตารางที่ 3.13

ตารางที่ 3.13 สัญลักษณ์ของ Class Diagram

สัญลักษณ์	การใช้งาน
 <p>Class Name Attribute Methods</p>	<p>1. Class Name คือ ชื่อของ Class</p> <p>2. Attributes คือ คุณลักษณะของ Class</p> <p>3. Operations หรือ Methods คือ กิจกรรมที่สามารถกระทำกับ Object นั้น ๆ ได้ สัญลักษณ์ที่ใช้แทนการเข้าถึงนั้นว่าเรียกว่า Visibility แบ่งออกได้เป็น 3 ประเภท Private แทนด้วย – หมายถึง Attribute หรือ 方法 ที่ไม่สามารถมองเห็นได้จากภายนอก Public แทนด้วย + หมายถึง Attribute หรือ 方法 ที่สามารถมองเห็นได้จากภายนอก และสามารถเข้าไปเปลี่ยนค่า อ่านค่าหรือเรียกใช้งาน Attribute หรือ 方法 ได้ Protected แทนด้วย # หมายถึงส่วนไว้สำหรับการทำ Inheritance โดยเฉพาะโดยปกติจะเป็นของ Super class เมื่อทำ inheritance แล้ว Attributes และ Operations เหล่านี้จะเป็นได้ทั้ง Private หรือ Protect ซึ่งขึ้นอยู่กับภาษาที่ใช้</p>
<p>Dependency</p> 	<p>Dependency หมายความว่า คลาสที่อยู่ผู้ดันลูก สามารถเรียกใช้คลาสที่อยู่ผู้ดันหัวลูกศร</p>

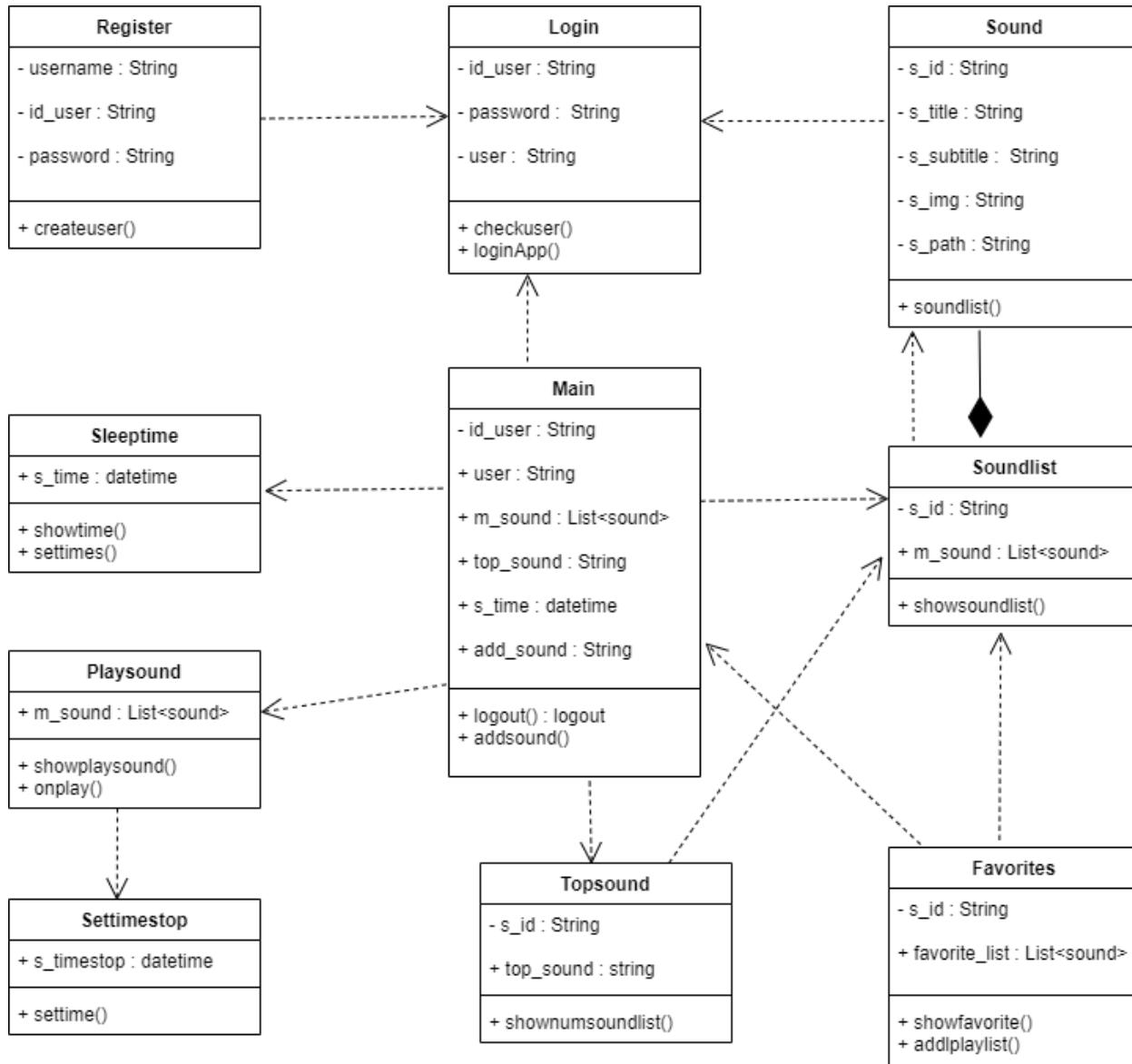
ตารางที่ 3.14 สัญลักษณ์ของ Class Diagram (ต่อ)

สัญลักษณ์	การใช้งาน
	เป็นความสัมพันธ์ Association แบบหนึ่ง (แบบพิเศษ) คือ Object หนึ่งมี Objects อื่นๆ เป็นส่วนประกอบอาจเรียกว่า whole-part relationship (ทั้งหมด-ส่วนประกอบ) ซึ่งมักจะใช้ค่าว่า consists of , contains , is part of เช่น Car มี Engine และ wheels เป็นส่วนประกอบ
	เป็นความสัมพันธ์ระหว่าง Object หรือ Class แบบขึ้นต่อ กันและ มีความเกี่ยวข้องกันเสมอ โดยจะมี Class ซึ่งเป็นองค์ประกอบของ Class อื่นที่ใหญ่กว่า เมื่อ Class ที่ใหญ่กว่าถูกทำลาย Class ที่เป็นองค์ประกอบก็จะถูกทำลายไปด้วย

Class Diagram แสดงความสัมพันธ์ในรูปแบบต่าง ๆ ระหว่างคลาสของของแต่ละระบบซึ่งจะแบ่งออกได้ทั้งหมด 10 Class Diagram ดังนี้

#### 1. Mobile Application Relax Sound

Class Diagram แสดงความสัมพันธ์ระหว่างคลาส แอปพลิเคชันเสียงเพื่อการผ่อนคลายอยู่ภายในไฟล์ภาพ 3.15



ภาพที่ 3.13 : Class diagram แอปพลิเคชันเสียงเพื่อการผ่อนคลาย

จากภาพที่ 3.13 สามารถอธิบายดังตารางต่อไปนี้

ตารางที่ 3.15 อธิบาย Class Diagram ของคลาสพื้นฐานแอปพลิเคชัน

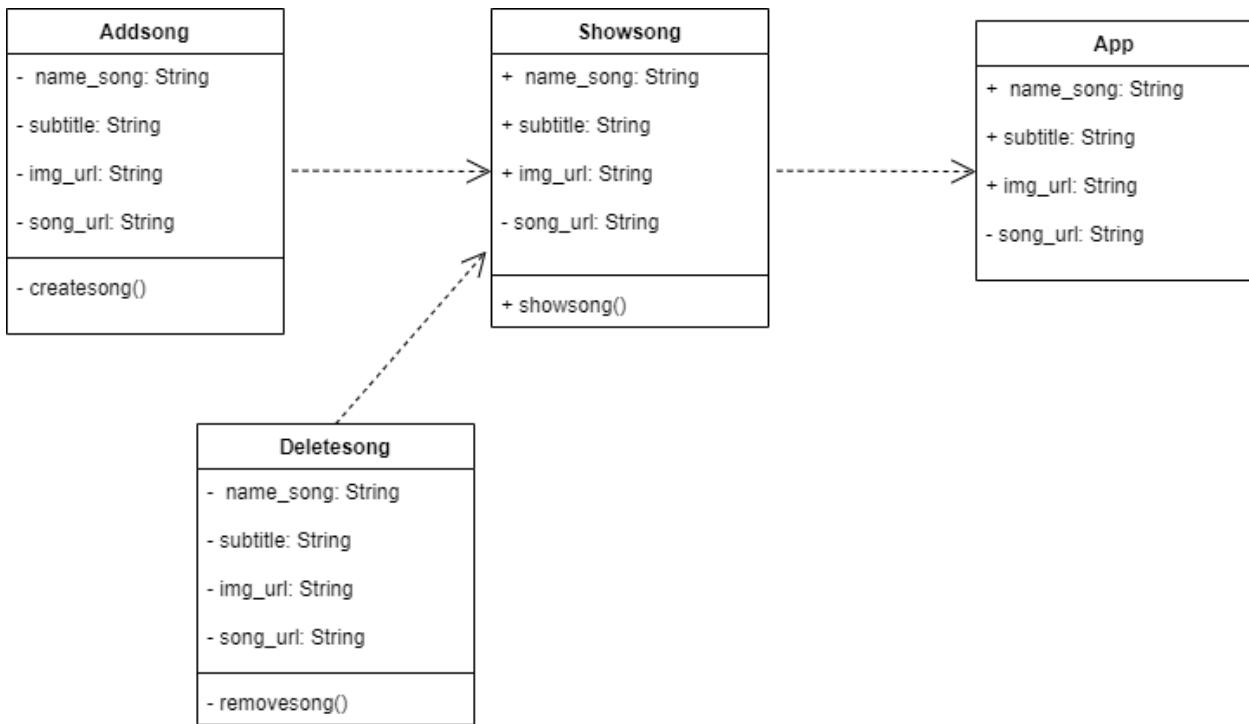
Class Diagram	คำอธิบาย
Register	เป็นคลาสที่ใช้เพื่อสมัครสมาชิกใหม่เข้าใช้งานแอปพลิเคชันครั้งแรกเท่านั้น

ตารางที่ 3.16 อธิบาย Class Diagram ของคลาสพื้นฐานแอปพลิเคชัน (ต่อ)

Class Diagram	คำอธิบาย
Login	เป็นคลาสที่ใช้เพื่อสมาชิกเข้าสู่ระบบแอปพลิเคชัน
Main	เป็นคลาสหลักในการทำแอปพลิเคชัน โดยแสดงข้อมูลส่วนต่างๆ ได้แก่ รายการเพลง Soundlist เพลงยอดฮิต Topsound เตือนเวลาอน Sleeptime และ เพิ่มเพลงเป็นรายโปรด Favorites
Sound	เป็นคลาสโมเดลเสียงเพลงในการสร้างลิสต์ คลาส Soundlist
Soundlist	เป็นคลาสสิตรายการเสียงเพลงที่ได้จากคลาส Sound
Favorites	เป็นคลาสที่เก็บเสียงเพลง รายการโปรดของสมาชิก โดยการเพิ่มที่ คลาส Main
Topsound	เป็นคลาสที่แสดงเพลงยอดฮิตที่มีฟังมากที่สุด โดยแสดงที่คลาส Main
Playsound	เป็นคลาஸสำหรับเล่นเพลง และตั้งเวลาปิดเสียง คลาส Settimestop
Settimestop	เป็นคลาสตั้งเวลาปิดเสียง
Sleeptime	เป็นคลาสตั้งเตือนเวลาเข้านอน โดยสามารถตั้งเวลาที่ คลาส Main

## 2. Web ส่วนเพิ่มเพลง

Class Diagram แสดงความสัมพันธ์คลาสวีบส่วนเพิ่มเพลงอธิบาย ได้ดังภาพ 3.14



ภาพที่ 3.14 : Class diagram เว็บเพิ่มเพลงสำหรับผู้ดูและระบบ

จากภาพที่ 3.14 สามารถอธิบายดังตารางต่อไปนี้

ตารางที่ 3.17 อธิบาย Class Diagram ของคลาสพื้นฐานแอปพลิเคชัน

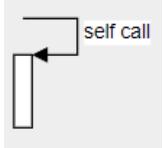
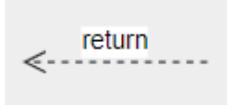
Class Diagram	คำอธิบาย
Addsong	เป็นคลาสที่ผู้ดูและระบบใช้สำหรับเพิ่มเพลง
Deletesong	เป็นคลาสที่ผู้ดูและระบบใช้สำหรับลบเพลง
ShowSong	เป็นคลาสแสดงเพลงที่เพิ่ม
App	เป็นคลาสสำหรับแสดงเพลงบนแอปพลิเคชัน

### 3.6 แผนภาพลำดับการทำงาน (Sequence Diagram)

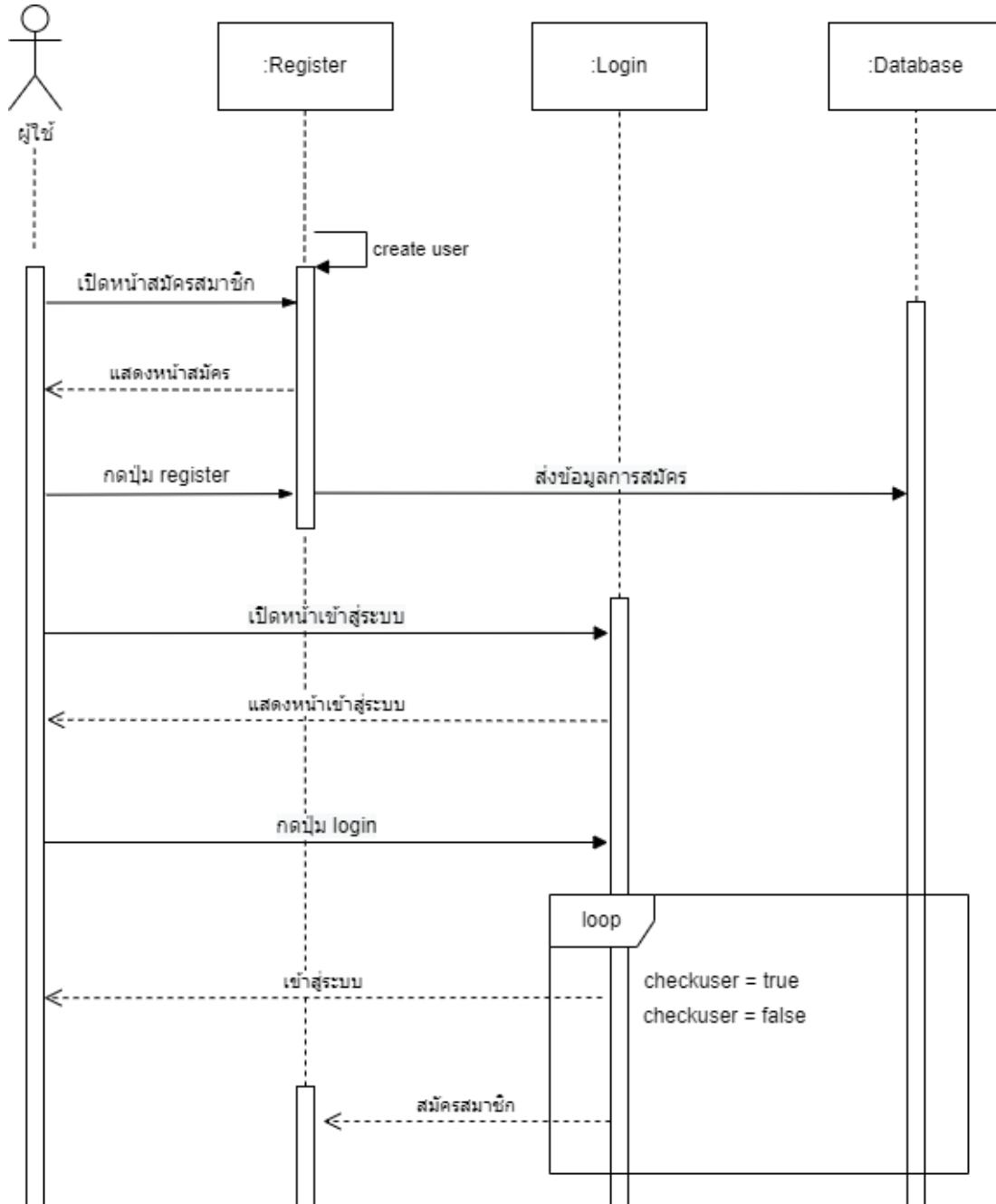
Sequence Diagram เป็นแผนผังการทำงานที่ประกอบไปด้วยคลาส (Class) หรือวัตถุ (Object) ส่งประทีซเพื่อแสดงลำดับเวลา และสืบสานที่ใช้เพื่อแสดงกิจกรรมที่เกิดขึ้นจากคลาสหรือวัตถุในแผนผังการทำงานภายใน Sequence Diagram จะใช้สีเหลืองแทนเสมอ บนคลาส และวัตถุโดยภายในจะมีชื่อของคลาสหรือวัตถุประกอบอยู่ในรูปแบบ {Object}: Class ส่วนประกอบการเขียนสัญลักษณ์ใน Sequence Diagram มีดังตาราง 3.12

ตารางที่ 3.18 สัญลักษณ์ Sequence Diagram

สัญลักษณ์	การใช้งาน
	ผู้ที่เกี่ยวข้องกับระบบ
	ออบเจกต์ที่ต้องทำหน้าที่ตอบสนองต่อ Actor
	สื้นแสดงชีวิตของออบเจกต์หรือคลาส
	จุดเริ่มต้นและจุดสิ้นสุดของแต่ละกิจกรรมในระหว่างที่มีชีวิตอยู่
	คำสั่งหรือฟังก์ชันที่ออบเจกต์หนึ่งส่งให้อีกออบเจกต์หนึ่ง ซึ่งสามารถส่งกลับได้ด้วย
	การประมวลผลและคืนค่าที่ได้รับไปอีกออบเจกต์เดียวกัน

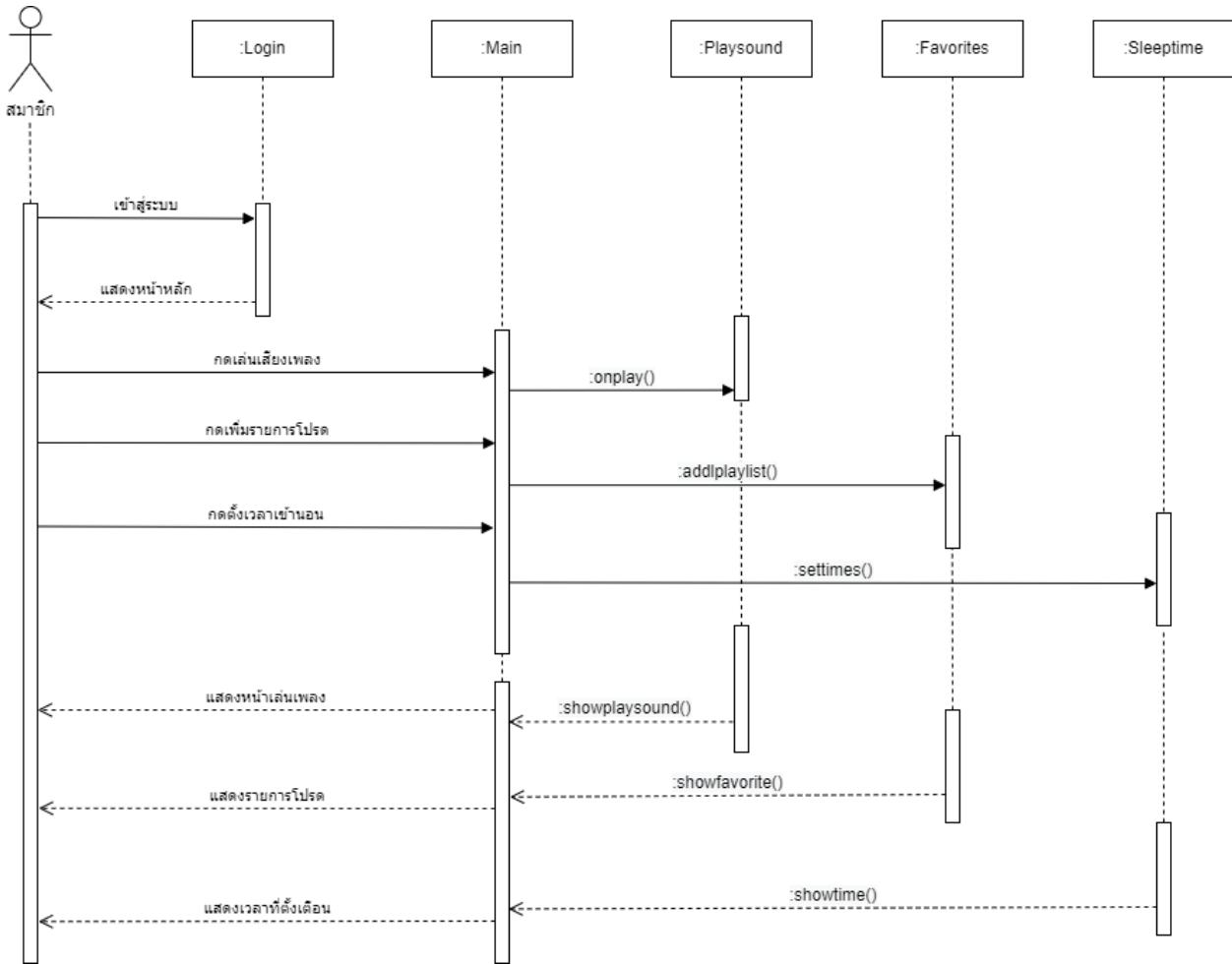
	พังก์ชันการทำงานในตัวเอง
	ส่งค่ากลับหลังทำงานเสร็จ
	สร้างลูป(loop)การทำงานของโปรแกรม

Sequence Diagram ที่ใช้อธิบายการทำงานแบบพอลิเคชันเสียงเพื่อการผ่อนคลายมีดังนี้



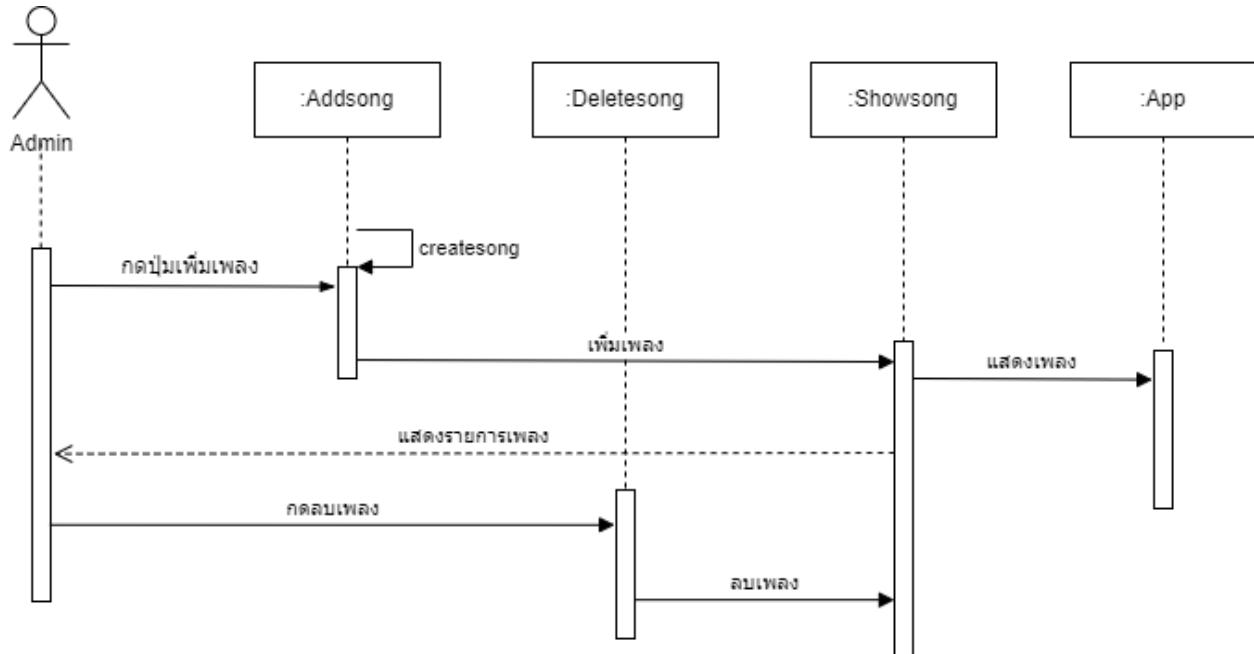
ภาพที่ 3.15 : Sequence Diagram เข้าสู่ระบบแอปพลิเคชัน

จากภาพที่ 3.15 เมื่อผู้ใช้เข้าหน้า Register ผู้ใช้ต้องกรอกข้อมูลเพื่อลงทะเบียนเมื่อผู้ใช้กรอกข้อมูลเสร็จ ข้อมูลจะถูกส่งไป Database เมื่อผู้ใช้เข้าหน้า Login ผู้ใช้ต้องกรอกข้อมูลที่ลงทะเบียนเพื่อเข้าสู่ระบบ แล้วกดปุ่ม โลγin ระบบจะเช็คเงื่อนไขถ้ารหัสผิดก็ต้องกรอกข้อมูลเข้าสู่ระบบใหม่หรือถ้าลีมรหัสกีสามารถลงลงทะเบียนใหม่ ถ้าผู้ใช้กรอกข้อมูลเข้าสู่ระบบถูกก็จะเข้าสู่ระบบได้



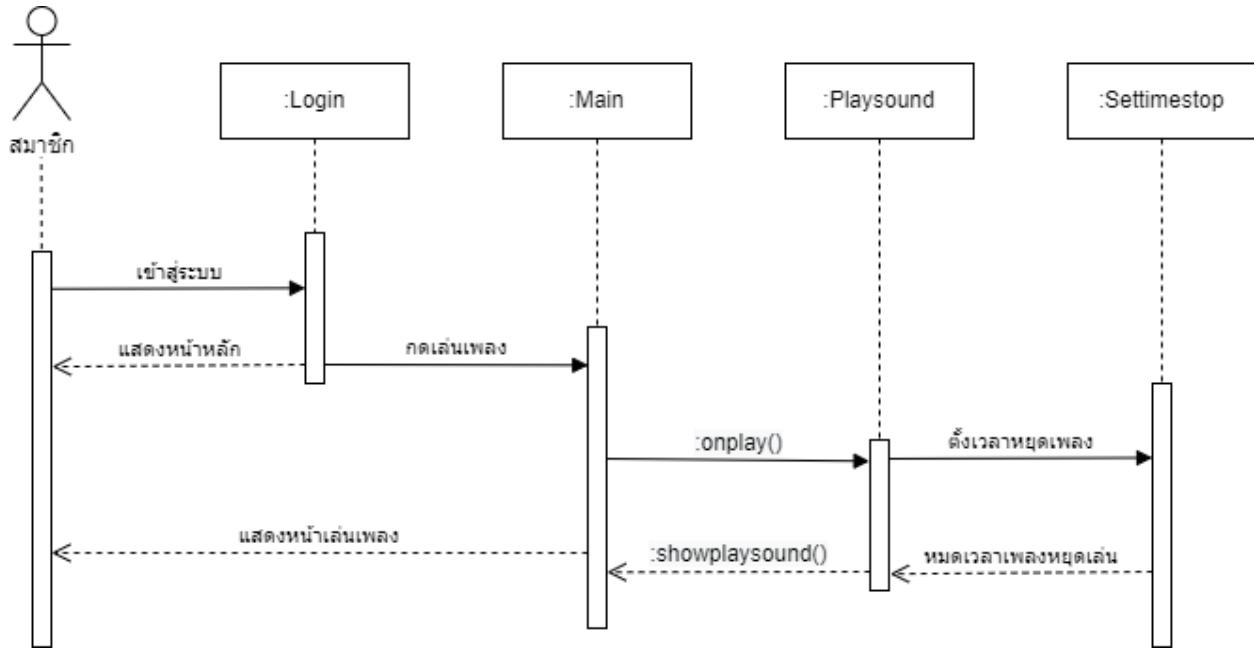
ภาพที่ 3.16 : Sequence Diagram แสดงหน้าหลักแอปพลิเคชัน

จากภาพที่ 3.16 เมื่อสมาชิก Login เข้าสู่ระบบ จะเข้าสู่หน้าหลัก สมาชิกสามารถดูรายการเสียงเพลง เมื่อกดเล่นเสียงเพลง ระบบจะแสดงหน้า Playsound เมื่อผู้ใช้กดเพิ่มรายการโปรดเสียงเพลงจะไปเก็บที่หน้า Favorites สามารถกดดูรายการเพลงโปรด เมื่อผู้ใช้กดตั้งเวลาเข้านอนระบบจะแสดงเวลาที่ตั้งไว้



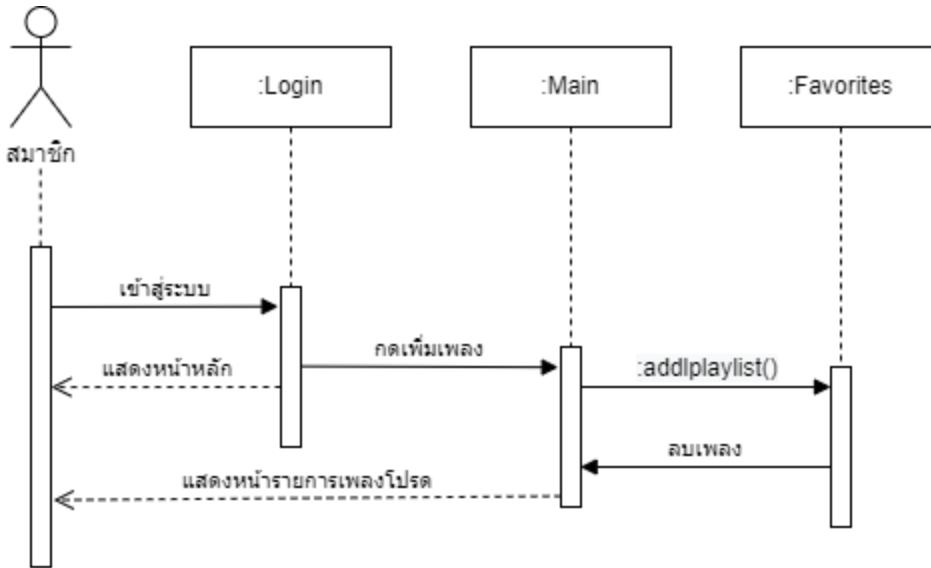
ภาพที่ 3.17 : Sequence Diagram แสดงส่วนผู้ดูแลระบบเพิ่มเพลงและลบเพลง

จากภาพที่ 3.17 เมื่อผู้ดูแลระบบเพิ่มเพลง Addsong เพลงที่ก็จะแสดงอยู่ที่หน้าเพลงที่เพิ่ม Showsong และแสดงบนแอปพลิเคชัน App เมื่อผู้ดูแลระบบลบเพลง เพลงที่แสดงที่หน้าแสดงเพลง Showsong และ บนหน้าแอปพลิเคชัน App เพลงนั้นก็จะโดนลบออก



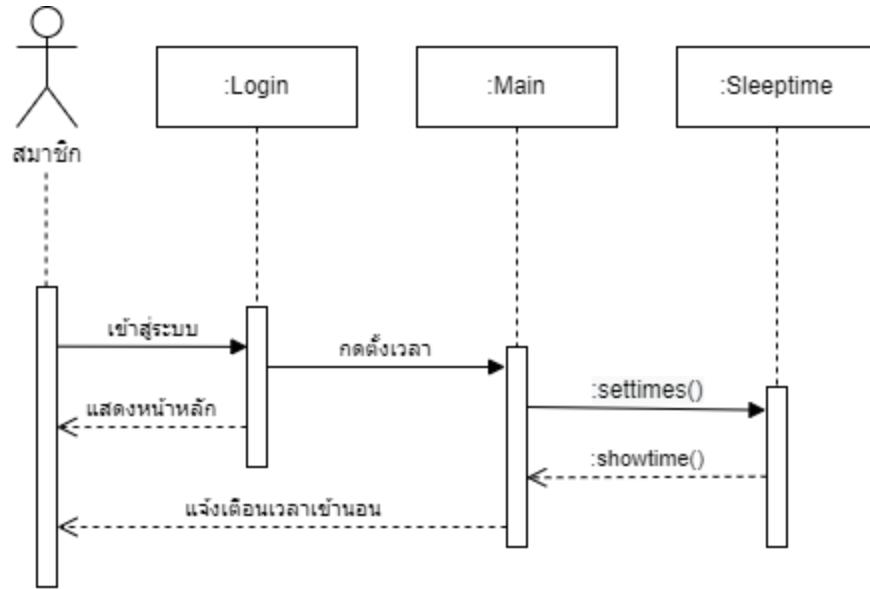
ภาพที่ 3.18 : Sequence Diagram แสดงส่วนเล่นเพลง

จากภาพที่ 3.18 เมื่อสมาชิก Login เข้าสู่ระบบ จะเข้าสู่หน้าหลัก เมื่อสมาชิกกดเล่นเพลงจะแสดงหน้าเล่นเพลง Playsound และสมาชิกสามารถตั้งเวลาหยุดเพลงได้ Settimestop



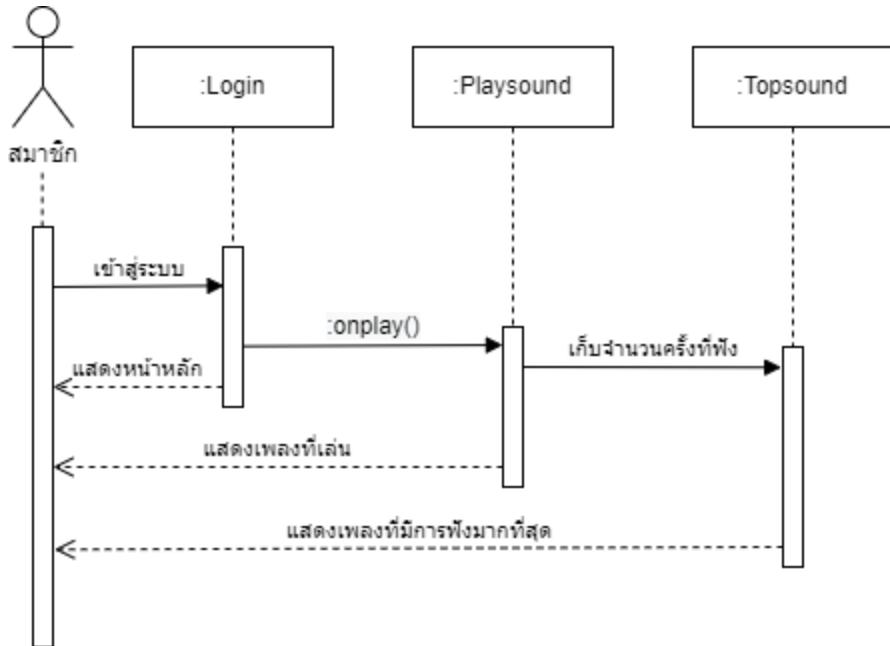
ภาพที่ 3.19 : Sequence Diagram แสดงส่วนเพิ่มและลบเพลงรายการโปรด

จากภาพที่ 3.19 เมื่อสมาชิก Login เข้าสู่ระบบ จะเข้าสู่หน้าหลัก สมาชิกสามารถกดเพิ่มเพลงที่ชื่อบนจากหน้าหลัก Main ไปยังหน้ารายการโปรด Favorites และสามารถลบเพลงจากหน้ารายการโปรด Favorites



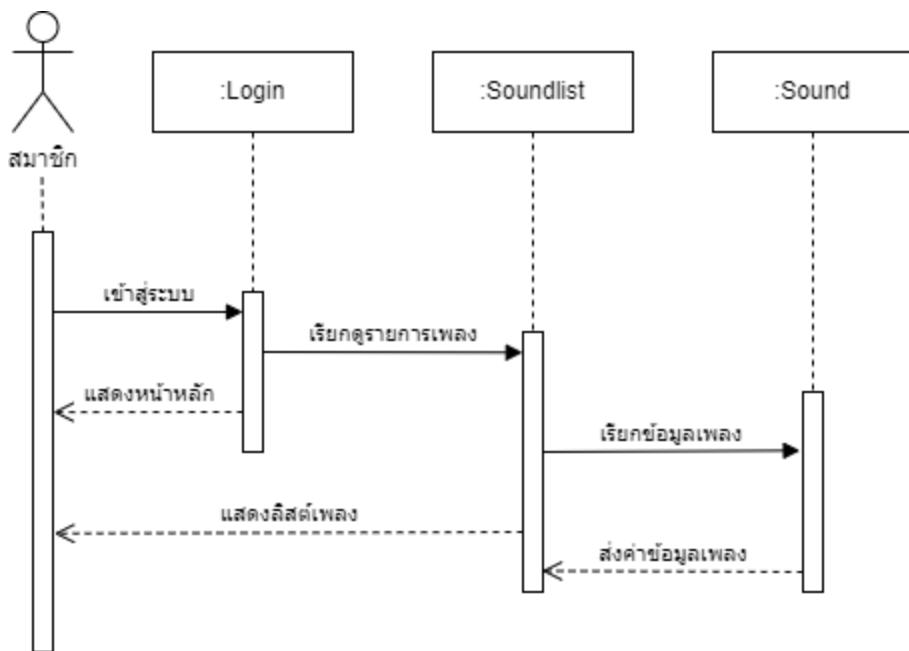
ภาพที่ 3.20 : Sequence Diagram แสดงส่วนตั้งเวลาเตือนเข้านอน

จากภาพที่ 3.20 เมื่อสมาชิก Login เข้าสู่ระบบ จะเข้าสู่หน้าหลัก สมาชิกสามารถกดตั้งเวลาจากหน้าหลัก Main และระบบจะทำการตั้งค่าเวลา Sleeptime และแจ้งเตือนเวลาที่กำหนด



ภาพที่ 3.21 : Sequence Diagram แสดงส่วนเพลงที่ชอบฟังมากที่สุด

จากภาพที่ 3.21 เมื่อสมาชิก Login เข้าสู่ระบบ จะเข้าสู่หน้าหลัก เมื่อสมาชิกกดเล่นเพลง Playsound ระบบจะเก็บจำนวนครั้งที่ฟังเพลงนั้นและจะแสดงเพลงที่มีการฟังมากที่สุด Topsound



ภาพที่ 3.22 : Sequence Diagram แสดงส่วนรายการเพลง

จากภาพที่ 3.22 เมื่อสมาชิก Login เข้าสู่ระบบ สมาชิกสามารถดูรายการเพลง Soundlist ระบบจะเรียกข้อมูลแต่ละเพลงมาจาก

Sound

## บทที่ 4

### การพัฒนาระบบ

หลังจากที่ได้เตรียมความพร้อมในการพัฒนาแอปพลิเคชันในด้านต่าง ๆ ไม่ว่าจะเป็นที่มาและความสำคัญของปัญหา เทคโนโลยีที่มีความเหมาะสมกับระบบ และการออกแบบระบบการทำงาน ในบทนี้จะการพูดถึงขั้นตอนการพัฒนาแอปพลิเคชัน

#### 4.1 การพัฒนาแอปพลิเคชัน

##### 4.1 การพัฒนาแอปพลิเคชัน

การพัฒนาแอปพลิเคชันโดยใช้ Ionic Framework ซึ่งเป็นเฟรมเวิร์กที่ใช้ง่ายสร้างครั้งเดียวสามารถใช้ได้หลายระบบปฏิบัติการ มีดังนี้

###### 4.1.1 การเชื่อมต่อ Firestore database

เชื่อมต่อเพื่อใช้บริการต่างๆของ ไฟร์เบส ดังนี้

```

1 apiKey: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
2 authDomain: "relax-app-e89f8.firebaseio.com",
3 projectId: "xxxxxxxxxxxxxx",
4 storageBucket: "relax-app-e89f8.appspot.com",
5 messagingSenderId: "xxxxxxxxxxxx",
6 appId: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
7 measurementId: "xxxxxxxxxxxx"

```

ภาพที่ 4.1 :ไฟล์ environment.ts

จากภาพที่ 4.1 โครงสร้างของไฟล์ environment.ts อยู่ภายในการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นหมายเลข apiKey

- บรรทัดที่ 2 เป็นหมายเลข authDomain
- บรรทัดที่ 3 เป็นชื่อ projectId
- บรรทัดที่ 4 เป็นค่า storageBucket สำหรับโครงการ
- บรรทัดที่ 5 เป็นค่า messagingSenderId สำหรับโครงการ
- บรรทัดที่ 6 เป็นหมายเลข appId
- บรรทัดที่ 7 เป็นหมายเลข measurementId

#### 4.1.2 การพัฒนาส่วนสมัครสมาชิกและเข้าสู่ระบบ

```

1 import { AngularFirestore } from '@angular/fire/compat/firestore';
2 import { Injectable } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root',
6 })
7
8 export class homes{
9
10   constructor(private db:AngularFirestore){}
11
12   add(tmpdoc: any){
13     return this.db.collection('acc').add(tmpdoc);
14   }
15
16   check(){
17     return this.db.collection('acc').snapshotChanges();
18   }
19
20
21   update(docId:any,tmpdoc:any){
22     return this.db.doc('acc/'+docId).update(tmpdoc);
23   }
24 }
```

ภาพที่ 4.2 : ไฟล์ homes.ts ส่วนเชื่อมบริการฐานข้อมูล

จากภาพที่ 4.2 โครงสร้างของ homets.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นการนำเข้าไลบรารีอ่านข้อมูลเบื้องต้นจากไฟร์เบส
- บรรทัดที่ 2-6 เป็นการนำเข้าไลบรารีคลาสร้องขอการพิงพาจากแหล่งภายนอก
- บรรทัดที่ 8-24 เป็น service ในการเชื่อมต่อ รับ-ส่งข้อมูล firebase
- บรรทัดที่ 10 เป็น constructor สำหรับสร้างตัวแปรเชื่อมฐานข้อมูล
- บรรทัดที่ 12-14 เป็นฟังก์ชันเพิ่มข้อมูลลงฐานข้อมูล
- บรรทัดที่ 16-18 เป็นฟังก์ชันเช็คเงื่อนไขข้อมูล
- บรรทัดที่ 21-23 เป็นฟังก์ชันอัปเดตข้อมูลลงฐานข้อมูล

```

1 constructor(public navCtrl: NavController,
2   public ounter : Router,
3   public alertCtrl: AlertController,
4   public toastCtrl: ToastController,
5   private homets: homets
6   ) { }
7 ngOnInit() {
8   this.homets.check().subscribe(data => {
9     this.list = data.map(e => {
10       return {
11         id: e.payload.doc.id,
12         isEdit: false,
13         ids: e.payload.doc.data()['id'].toString(),
14
15         pass: e.payload.doc.data()['pass'].toString(),
16         user: e.payload.doc.data()['user'].toString(),
17         img: e.payload.doc.data()['img'].toString(),
18       };
19
20     });
21
22   });
23
24 }
25

```

ภาพที่ 4.3 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ

จากภาพที่ 4.3 โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-6 เป็นการกำหนดตัวแปรจากไลบรารีที่นำเข้า
- บรรทัดที่ 8 ทำการเรียกใช้ homets ผ่าน check() จะได้ข้อมูลกลับมาในรูปแบบ Observable
- บรรทัดที่ 9 อ่านค่าผลลัพธ์จากข้อมูลที่ส่งออกมานา
- บรรทัดที่ 10-17 เป็นส่วนค่าข้อมูลสมาชิก

```

1 async login() {
2   let alert = this.alertCtrl.create({
3     header: 'Login',
4     inputs: [
5       {
6         name: 'lid',
7         placeholder: 'ID',
8       },
9       {
10        name: 'lpass',
11        placeholder: 'Password',
12        type: 'password'
13      }
14    ],
15    buttons: [
16      {
17        text: 'Cancel',
18        role: 'cancel',
19        handler: data => {
20        }
21      },
22      {
23        text: 'Login',
24        handler: data => {
25          for (let i=0; i< this.list.length; i++){
26            if(this.list[i].ids == data.lid && this.list[i].pass == data.lpass ){
27              this.getuser =this.list[i].user
28              this.getimg =this.list[i].img
29              this.obj = {
30                user: this.getuser,
31                id:this.getid,
32                pass:this.getpass,
33                img:this.getimg
34              };
35              console.log(this.getuser)
36              const dataString =JSON.stringify(this.obj)
37              this.rounter.navigate(['tabs/firstpage',dataString])
38            }
39          }
40        }
41      }
42    ]
43  });
44  (await alert).present();
45 }

```

ภาพที่ 4.4 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)

จากภาพที่ 4.4 โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 2 ทำการเรียกใช้ alertController ผ่าน create() จะทำการสร้าง dialog สำหรับกรอกข้อมูล
- บรรทัดที่ 3-12 เป็นส่วนของการกรอกข้อมูล id และกรอกรหัสผ่าน
- บรรทัดที่ 15-23 เป็นปุ่มยกเลิกและ login
- บรรทัดที่ 24-35 เป็นส่วนเงื่อนไขถ้าเลข id และรหัสผ่านตรงกันสามารถเข้าสู่ระบบได้
- บรรทัดที่ 37-38 เป็นส่วนส่งข้อมูลผู้ใช้แบบ Json ไปยังหน้าหลัก
- บรรทัดที่ 45 เป็นคำสั่งแสดง dialog

```

1 async register() {
2   let alert = this.alertCtrl.create({
3     header: 'Register',
4     inputs: [
5       {
6         name: 'ruser',
7         placeholder: 'Username'
8       },
9       {
10        name: 'rid',
11        placeholder: 'ID'
12      },
13      {
14        name: 'rpass',
15        placeholder: 'Password'
16      },
17    ],
18    buttons: [
19      {
20        text: 'Cancel',
21        role: 'cancel',
22        handler: data => {
23        }
24      },

```

ภาพที่ 4.5 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)

จากภาพที่ 4.5 โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ อยู่ในรูปการทำงานได้ดังนี้

- บรรทัดที่ 2 ทำการเรียกใช้ alertController ผ่าน create() จะทำการสร้าง dialog สำหรับกรอกข้อมูล
- บรรทัดที่ 3-15 เป็นส่วนของการกรอก ชื่อ,id และกรอกรหัสผ่าน
- บรรทัดที่ 18-24 เป็นปุ่มยกเลิก

```

1   {
2     text: 'Register',
3     handler: data => {
4       for (let i=0; i< this.list.length; i++){
5         if(this.list[i].ids != data.rid){
6           this.check=true
7           break
8         }
9       }
10      if(this.check==true){
11
12        let tmpobj = //db : inputform
13        {id: data.rid,
14         pass: data.rpass,
15         user: data.ruser,
16         img :"https://pic.onlinewebfonts.com/svg/img_24787.png",
17       };
18        this.homet.add(tmpobj);
19        this.openToast2()
20      }else{
21        this.openToast()
22      }
23    }//handler
24  }//update
25  ]
26 });
27 (await alert).present();
28 }

```

ภาพที่ 4.6 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)

จากภาพที่ 4.6 โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-9 เป็นส่วนเงื่อนไขถ้าข้อมูล id นั้นไม่ตรงกับที่มีอยู่สามารถใช้ id นั้นสมัครเข้าใช้งานได้
- บรรทัดที่ 10-20 ถ้าเงื่อนไขถูกต้องก็จะส่งค่าข้อมูลผู้ใช้ที่กรอกไปเก็บบนฐานข้อมูลและแสดงแจ้งเตือน Done
- บรรทัดที่ 20-22 ถ้าเงื่อนไขไม่ถูกจะแสดงแจ้งเตือน ID Password Can not use
- บรรทัดที่ 27 เป็นคำสั่งแสดง dialog

#### 4.1.3 การพัฒนาส่วนแสดงเพลง

```

1 songs:any = [
2   {
3     title:"Sea Wave",
4     subtitle:"sea-wave",
5     img:"/assets/img/wave.jpg",
6     path:"/assets/music/sea-waves.mp3",
7   },
8 relax :any = [
9   {
10    title:"Happy",
11    subtitle:"happy",
12    img:"/assets/img/happy.jpg",
13    path:"/assets/music/happy.mp3",
14  },
15 musical :any = [
16   {
17    title:"Piano",
18    subtitle:"piano",
19    img:"/assets/img/piano.jpg",
20    path:"/assets/music/piano.mp3",
21  },
22 sound:any = [];

```

ภาพที่ 4.7 : ตัวอย่างส่วนเก็บลิสต์เพลง

จากภาพที่ 4.7 โครงสร้างตัวอย่างส่วนเก็บลิสต์เพลง อยู่ในรูปแบบของภาษา JavaScript ดังนี้

- บรรทัดที่ 1-7 เป็นส่วนเก็บลิสต์ประเภทเพลงที่ 1
- บรรทัดที่ 8-14 เป็นส่วนเก็บลิสต์ประเภทเพลงที่ 2
- บรรทัดที่ 15-21 เป็นส่วนเก็บลิสต์ประเภทเพลงที่ 3
- บรรทัดที่ 22 เป็นส่วนเก็บลิสต์ประเภทเพลงที่ 4

#### 4.1.4 การพัฒนาส่วนໂຫດຂໍ້ອມຄາແສດງ

```
1 constructor(private backgroundMode:BackgroundMode,  
2   public actionSheetController:ActionSheetController,  
3   public actroute: ActivatedRoute,  
4   private Edit: edit,  
5   public alertCtrl: AlertController,  
6   public navCtrl : NavController,  
7   public rounter:Router,  
8   private http: HttpClient,  
9  
10  private https: HTTP,  
11  private localNotifications: LocalNotifications,  
12  private plt:Platform,  
13  private device: Device,  
14  private db:AngularFirestore,  
15  private homets: homets){
```

ກາພທີ 4.8 : ໂຄງສຮ້າງສ່ວນໂຫດຂໍ້ອມຄາແສດງ

ຈາກກາພທີ 4.8 ໂຄງສຮ້າງສ່ວນໂຫດຂໍ້ອມຄາແສດງ ອອິບາຍກາຣທໍາຈານໄດ້ຕັ້ງນີ້

- ບຣທັດທີ 1-15 ເປັນກາຣກຳທັດຕົວແປຣຈາກໄລບຣາຣີທີ່ນຳເຂົາ

```

16  this.homets.check().subscribe(data => {
17    this.list = data.map(e => {
18      return {
19        id: e.payload.doc.id,
20        isEdit: false,
21        ids: e.payload.doc.data()['id'].toString(),
22        pass: e.payload.doc.data()['pass'].toString(),
23        user: e.payload.doc.data()['user'].toString(),
24        img: e.payload.doc.data()['img'].toString(),
25        bird: e.payload.doc.data()['Bird'].toString(),
26        drippingwater: e.payload.doc.data()['DrippingWater'].toString(),
27        forest: e.payload.doc.data()['Forest'].toString(),
28        gentleoceanwaves: e.payload.doc.data()['GentleOceanWaves'].toString(),
29        higflowriver: e.payload.doc.data()['HigflowRiver'].toString(),
30        midnightstrom: e.payload.doc.data()['MidnightStrom'].toString(),
31        night: e.payload.doc.data()['Night'].toString(),
32        seawave: e.payload.doc.data()['SeaWave'].toString(),
33        surrealforest: e.payload.doc.data()['Surrealforest'].toString(),
34        waterfall: e.payload.doc.data()['Waterfall'].toString(),

```

ภาพที่ 4.9 : โครงสร้างส่วนโหลดข้อมูลมาแสดง (ต่อ)

จากภาพที่ 4.9 โครงสร้างส่วนโหลดข้อมูลมาแสดง อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 16 ทำการเรียกใช้ homets ผ่าน check() จะได้ข้อมูลกลับมาในรูปแบบ Observable
- บรรทัดที่ 17 อ่านค่าผลลัพธ์จากข้อมูลที่ส่งออกมา
- บรรทัดที่ 19-24 เป็นส่วนค่าข้อมูลสมาชิก
- บรรทัดที่ 25-34 เป็นส่วนค่าข้อมูลการฟังเพลงสมาชิก

```

35      happy: e.payload.doc.data()['Happy'].toString(),
36      relaxing: e.payload.doc.data()['Relaxing'].toString(),
37      hopeful: e.payload.doc.data()['Hopeful'].toString(),

```

```

38   powerfully: e.payload.doc.data()['Powerful'].toString(),
39   joyful: e.payload.doc.data()['Joyful'].toString(),
40   peaceful: e.payload.doc.data()['Peaceful'].toString(),
41   restless: e.payload.doc.data()['Restless'].toString(),
42   suspense: e.payload.doc.data()['Suspense'].toString(),
43   fanciful: e.payload.doc.data()['Fanciful'].toString(),
44   charming: e.payload.doc.data()['Charming'].toString(),
45
46   piano: e.payload.doc.data()['Piano'].toString(),
47   rnb: e.payload.doc.data()['RnB'].toString(),
48   classic: e.payload.doc.data()['Classic'].toString(),
49   jazz: e.payload.doc.data()['Jazz'].toString(),
50   synthpop: e.payload.doc.data()['Synthpop'].toString(),
51   indie: e.payload.doc.data()['Indie'].toString(),
52   blues: e.payload.doc.data()['Blues'].toString(),
53   electronic: e.payload.doc.data()['Electronic'].toString(),
54   hardrock: e.payload.doc.data()['Hardrock'].toString(),
55   hiphop: e.payload.doc.data()['Hiphop'].toString(),
56   time: e.payload.doc.data()['time'].toString(),

```

ภาพที่ 4.10 : โครงสร้างส่วนโหลดข้อมูลมาแสดง (ต่อ)

จากภาพที่ 4.10 โครงสร้างส่วนโหลดข้อมูลมาแสดง อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 35-56 เป็นส่วนข้อมูลการฟังเพลงสมาชิก

```

57   mbird: e.payload.doc.data()['m2'].toString(),
58   mdrippingwater: e.payload.doc.data()['m6'].toString(),
59   mforest: e.payload.doc.data()['m8'].toString(),
60   mgentleoceawaves: e.payload.doc.data()['m3'].toString(),
61   mhigflowriver: e.payload.doc.data()['m9'].toString(),
62   mmidnightstrom: e.payload.doc.data()['m4'].toString(),
63   mnight: e.payload.doc.data()['m5'].toString(),
64   mseawave: e.payload.doc.data()['m1'].toString(),
65   msurrealforest: e.payload.doc.data()['m10'].toString(),
66   mwaterfall: e.payload.doc.data()['m7'].toString(),
67
68   rhappy: e.payload.doc.data()['r1'].toString(),
69   rrelaxing: e.payload.doc.data()['r2'].toString(),
70   rhopeful: e.payload.doc.data()['r3'].toString(),
71   rpowerful: e.payload.doc.data()['r4'].toString(),

```

```

72 rjoyful: e.payload.doc.data()[r5'.toString()],
73 rpeaceful: e.payload.doc.data()[r6'.toString()],
74 rrestless: e.payload.doc.data()[r7'.toString()],
75 rsuspense: e.payload.doc.data()[r8'.toString()],
76 rfanciful: e.payload.doc.data()[r9'.toString()],
77 rcharming: e.payload.doc.data()[r10'.toString()],
78
79 spiano: e.payload.doc.data()[s1'.toString()],
81 srnb: e.payload.doc.data()[s2'.toString()],
82 sclassic: e.payload.doc.data()[s3'.toString()],
84 sjazz: e.payload.doc.data()[s4'.toString()],
85 ssynthpop: e.payload.doc.data()[s5'.toString()],
87 sindie: e.payload.doc.data()[s6'.toString()],
88 sblues: e.payload.doc.data()[s7'.toString()],
89 selectronic: e.payload.doc.data()[s8'.toString()],
90 shardrock: e.payload.doc.data()[s9'.toString()],
91 shiphop: e.payload.doc.data()[s10'.toString()],
92
93
94    };
95 });
96
97
98 console.log(this.list)
99
100 // this.checkbed()
101 this.count()
102 this.send()
103 });
104 }
```

ภาพที่ 4.11 : โครงสร้างส่วนเรียกใช้ฟังก์ชัน (ต่อ)

จากภาพที่ 4.11 โครงสร้างส่วนเรียกใช้ฟังก์ชัน อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 57-91 เป็นส่วนข้อมูลเพลงรายการโปรดสมาชิก
- บรรทัดที่ 101 เป็นส่วนเรียกใช้ฟังก์ชันนับจำนวนครั้งเพลงที่ฟัง
- บรรทัดที่ 102 เป็นส่วนเรียกใช้ฟังก์ชันส่งข้อมูลไปยังหน้าแท็บรายโปรด

#### 4.1.5 การพัฒนาส่วนแสดงเพลงที่ฟังบ่อย

```

1 count(){
2
3   for (let i=0; i< this.list.length; i++){
4     if(this.list[i].user == this.getuser ){
5       this.docId = this.list[i].id
6       for (let ii=0; ii< this.list.length; ii++){
7         if(this.docId == this.list[ii].id){
8           let xx = ii;
9
10
11   let arrayname = [
12   {val:Number(this.list[xx].bird), name:'bird' },
13   {val:Number(this.list[xx].drippingwater),name:'drippingwater'},
14
15   let arraynamesort =
16   arrayname.sort((b,a) => (a.val < b.val) ? -1 : ((b.val > a.val) ? 1 : 0))
17
18   for (let i=0; i< arraynamesort.length; i++){
19     if(arraynamesort[i].val !=0{
20       this.ax.push(arraynamesort[i].name)
21
21 let be=[

22   {name:'bird', title:"Bird", subtitle:"bird-voice", img:"/assets/img/bird.jpg", path:"/assets/music/bird-voice.mp3"},

23 if(this.timeout==0){
24   for (let i=0; i< this.ax.length; i++){
25     if(this.ax[i] == 'bird'){
26       this.ox .push({title:be[0].title, subtitle:be[0].subtitle, img:be[0].img, path:be[0].path })
27     }
28   }
29   else if(this.ax[i] == 'drippingwater'){
30     this.ox .push({title:be[1].title, subtitle:be[1].subtitle, img:be[1].img, path:be[1].path })
31   }

```

ภาพที่ 4.12 : โครงสร้างส่วนแสดงเพลงที่ฟังบ่อย

จากภาพที่ 4.12 โครงสร้างส่วนแสดงที่ฟังบ่อย อยู่ในไฟล์ main.js ดังนี้

- บรรทัดที่ 1-31 เป็นฟังก์ชันนับค่าจำนวนครั้งที่ฟังเพลงและทำการเรียงเพลงที่มีการฟังมากที่สุด

- บรรทัดที่ 3-8 เป็นส่วนเงื่อนไขหา id ของผู้ใช้ว่าเป็นของผู้ใช้คนไหน
- บรรทัดที่ 11-13 เป็นส่วนกำหนดเพลงที่สมาชิกคนนั้นฟังมีเพลงอะไรบ้าง
- บรรทัดที่ 15-16 เรียงเพลงที่สมาชิกฟังจากมากไปน้อย
- บรรทัดที่ 18-22 ถ้าค่าไม่ใช่ 0 ให้เก็บข้อมูลเพลงในตัวแปร ax
- บรรทัดที่ 23-31 เอาตัวแปร ax มาหาว่าคือเพลงอะไรบ้าง

```

1 for (let i=0; i< this.list.length; i++){
2   if(this.list[i].user == this.getuser ){

//this
3 this.docId = this.list[i].id
4 var t :number = this.list[i].bird

5 if (title =="Bird")
6 { console.log("Add",title)
7 t=t+1
8 }

9 }
10 let tmpobj = //db : inputform
11 {Bird: t,
12 };
13 return this.db.doc('acc/'+this.docId).update(tmpobj);
14
15 }
```

ภาพที่ 4.13 : ตัวอย่างโครงสร้างส่วนเก็บค่าจำนวนเพลง (ต่อ)

จากภาพที่ 4.13 ตัวอย่างโครงสร้างส่วนเก็บค่าจำนวนเพลง อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-2 for หาชื่อของ user
- บรรทัดที่ 3-4 เป็นส่วนเก็บค่าเพลงของ user

- บรรทัดที่ 5-7 ถ้าเล่นเพลงตรงกับเพลง bird ค่า t จะเพิ่มเป็นหนึ่ง
- บรรทัดที่ 10-13 เป็นส่วนอัปเดตค่าในฐานข้อมูลให้ส่งค่าไปที่ id ของ user

#### 4.1.6 การพัฒนาส่วนรับค่าข้อมูลสมาชิก

```

1 const dataRev = this.actroute.snapshot.paramMap.get('obj');
2 let getobj = JSON.parse(dataRev);
3 this.getuser = getobj['user'];
4 this.getid = getobj['id'];
5 this.getpass = getobj['pass'];
6 this.getimg = getobj['img'];

```

ภาพที่ 4.14 : โครงสร้างส่วนรับค่าข้อมูลสมาชิก

จากภาพที่ 4.14 โครงสร้างส่วนรับค่าข้อมูลสมาชิก อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-6 เป็นส่วนรับค่าข้อมูลผู้ใช้เมื่อเข้าสู่ระบบหน้าหลัก

#### 4.1.7 การพัฒนาส่วนเล่นเพลง

```

1 playSong(title,subTitle,img,song){
2   if (this.currSong != null) {
3     this.currSong.pause();
4   }
5
6   document.getElementById('fullPlayer').style.bottom = '0px';
7
8   this.currTitle = title;
9   this.currSubtitle = subTitle;
10  this.currImage = img;
11
12
13  this.currSong = new Audio(song);
14
15  this.currSong.play().then(() => {
16
17
18    this.durationText = this.sToTime(this.currSong.duration);
19
20    this.maxRangeValue = Number(this.currSong.duration.toFixed(2).toString().substring(0, 5));

```

ภาพที่ 4.15 : โครงสร้างส่วนเล่นเพลง

จากภาพที่ 4.15 โครงสร้างส่วนเล่นเพลง อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-4 เป็นส่วนเล่น-หยุดเพลง
- บรรทัดที่ 6 เป็นส่วนสำหรับเพลงแบบเต็มหน้าจอ
- บรรทัดที่ 8-10 เป็นส่วนรายละเอียดเพลงขณะนั้น
- บรรทัดที่ 13 เป็นส่วนเสียงเพลงปัจจุบัน
- บรรทัดที่ 18 เป็นส่วนระยะเวลาเพลงทั้งหมด
- บรรทัดที่ 20 เป็นส่วนตั้งค่าช่วงสูงสุดเวลาเพลง

```

21 var index = this.songs.findIndex(x => x.title == this.currTitle);
22
23
24 if ((index + 1) == this.songs.length) {
25   this.upNextImg = this.songs[0].img;
26   this.upNextTitle = this.songs[0].title;
27   this.upNextSubtitle = this.songs[0].subtitle;
28 }
29
30
31 else {
32   this.upNextImg = this.songs[index + 1].img;
33   this.upNextTitle = this.songs[index + 1].title;
34   this.upNextSubtitle = this.songs[index + 1].subtitle;
35 }
36
37 this.isPlaying = true;
38 })
39 this.currSong.addEventListener('timeupdate', () => {
40
41
42
43 if (!this.isTouched) {
44
45   this.currRangeTime = Number(this.currSong.currentTime.toFixed(2).toString().substring(0, 5));
46
47   this.currSecsText = this.sToTime(this.currSong.currentTime);
48
49   this.progress = (Math.floor(this.currSong.currentTime) / Math.floor(this.currSong.duration));
50
51
52   if (this.currSong.currentTime == this.currSong.duration) {
53     this.playNext();
54   }
55 }
56 })

```

ภาพที่ 4.16 : โครงสร้างส่วนเล่นเพลง (ต่อ)

จากภาพที่ 4.16 โครงสร้างส่วนเล่นเพลง อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 21-38 เป็นส่วนเล่นเพลงถัดไป

- บรรทัดที่ 39-49 เป็นส่วนอัปเดตเมื่อเลื่อนกรอเพลง
- บรรทัดที่ 52-53 เป็นส่วนเมื่อเพลงเล่นจบจะเล่นเพลงถัดไป

#### 4.1.8 การพัฒนาส่วนเพิ่มเพลงรายการโปรด

```

1 async addplaylist(title){

2   console.log("Add",title)
3 let tmpobj
4
5 if(title == "Sea Wave"){
6let tmpobj1 = //db : inputform
7 { m1 : 1
8 };
9 tmpobj = tmpobj1
10 this.oxx.push({title:"Sea Wave", subtitle:"sea-wave",img:"/assets/img/sea wave.jpg"})
11 }

12 return this.db.doc('acc/' + this.docId).update(tmpobj);
13
14 }

```

ภาพที่ 4.17 : ตัวอย่างโครงสร้างส่วนเพิ่มเพลงเป็นรายโปรด

จากภาพที่ 4.20 ตัวอย่างโครงสร้างส่วนเพิ่มเพลงเป็นรายการโปรด อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 3-11 ถ้าเพลงที่เลือกพังเป็นเพลง seawave ให้อัปเดตค่า m1 เป็น 1
- บรรทัดที่ 12 เป็นส่วนอัปเดตค่าในฐานข้อมูลให้ส่งค่าไปที่ id ของ user

#### 4.1.9 การพัฒนาส่วนตั้งเวลาหยุดเพลง

```

1 startTimer() {
2   var counter = setTimeout(() => {
3     var time = this.getTimerClock(this.remainingTime);
4     console.log(time);
5     if (this.remainingTime > 0) {
6       this.startTimer();
7       this.remainingTime--;
8     } else {
9       this.pause();
10      clearInterval(counter);
11    }
12  }, 1000);
13 }
14 }

15 getTimerClock(inputSeconds: number) {
16   var sec_num = parseInt(inputSeconds.toString(), 10);
17   this.remainingTime = sec_num; //Define variable
18   var hours = Math.floor(sec_num / 3600);
19   var minutes = Math.floor((sec_num - (hours * 3600)) / 60);
20   var seconds = sec_num - (hours * 3600) - (minutes * 60);
21   this.hoursString = (hours < 10) ? "0" + hours : hours.toString();
22   this.minutesString = (minutes < 10) ? "0" + minutes : minutes.toString();
23   this.secondsString = (seconds < 10) ? "0" + seconds : seconds.toString();
24   return this.hoursString + ":" + this.minutesString + ":" + this.secondsString;
25 }

```

ภาพที่ 4.18 : โครงสร้างส่วนตั้งเวลาหยุดเพลง

จากภาพที่ 4.18 โครงสร้างส่วนตั้งเวลาหยุดเพลง อยูใบายการทำงานได้ดังนี้

- บรรทัดที่ 1-25 เป็นฟังก์ชันตั้งเวลาหยุดเพลง

- บรรทัดที่ 1-13 เป็นส่วนนับเวลา

- บรรทัดที่ 15-25 เป็นส่วนคำนวณนับเวลาถอยหลัง

#### 4.2.0 การพัฒนาส่วนแจ้งเตือนเวลาบน

```
1 checkbed(){  
2     this.localNotifications.schedule({  
3         id: 1,  
4         title: 'Bedtime Reminder',  
5         text: 'Bedtime',  
6     });  
7 }
```

ภาพที่ 4.19 : โครงสร้างส่วนแจ้งเตือนเวลาเขียนบน

จากภาพที่ 4.19 โครงสร้างส่วนแจ้งเตือนเวลาเขียนบน อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-7 เป็นส่วนการแจ้งเตือน

```
1 ionViewDidEnter2() {
2     this.remainingTime2 = this.ax;
3     this.startTimer2();
4 }
5
6
7 startTimer2() {
8     var counter2 = setTimeout(() => {
9         var time2 = this.getTimerClock(this.remainingTime2);
10        console.log(time2);
11        if (this.remainingTime2 > 0) {
12            this.startTimer2();
13            this.remainingTime2--;
14        }
15        else {
16            console.log("หมดเวลา");
17            this.checkbed();
18            clearInterval(counter2);
19        }
20    }, 1000);
21 }
22 getTimerClock2(inputSeconds2: number) {
23     var sec_num2 = parseInt(inputSeconds2.toString(), 10);
24     this.remainingTime2 = sec_num2; //Define variable
25     var hours2 = Math.floor(sec_num2 / 3600);
26     var minutes2= Math.floor((sec_num2 - (hours2 * 3600)) / 60);
27     var seconds2 = sec_num2 - (hours2 * 3600) - (minutes2 * 60);
28     this.hoursString2 = (hours2 < 10) ? "0" + hours2 : hours2.toString();
29     this.minutesString2 = (minutes2 < 10) ? "0" + minutes2 : minutes2.toString();
30     this.secondsString2 = (seconds2 < 10) ? "0" + seconds2 : seconds2.toString();
31     return this.hoursString2 + ":" + this.minutesString2 + ":" + this.secondsString2;
32 }
```

```

33 loop(){
34 console.log(this.someString)
35 let date = new Date()
36 var sdate= String(date);
37 var splitted= sdate.split(" ", 5);
38 var time =splitted[4];
39 time = time[0]+time[1]+time[2]+time[3]+time[4]
40 let aa=((24-Number(time[0]+time[1]))*3600
41 )
42 let bb=((60-Number(time[3]+time[4]))*60
43 )
44
45 let cc=((24-Number(this.someString[11]+this.someString[12]))*3600
46 )
47 console.log(this.someString)
48 let dd=((60-Number(this.someString[14]+this.someString[15]))*60
49 )
50 console.log((aa+bb),(cc+dd))
51 this.xa=Number((aa+bb)-(cc+dd))
52
53
54 this.remainingTime2 = this.xa;
55 this.startTimer2();
56 }

```

ภาพที่ 4.20 : โครงสร้างส่วนแจ้งเตือนเวลาเข้าอน (ต่อ)

จากภาพที่ 4.20 โครงสร้างส่วนแจ้งเตือนเวลาเข้าอน อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นส่วนเริ่มนับเวลา
- บรรทัดที่ 3-4 เป็นส่วนเรียกใช้ฟังก์ชันนับเวลา
- บรรทัดที่ 7-20 เป็นส่วนนับเวลา
- บรรทัดที่ 22-32 เป็นส่วนคำนวณนับเวลาถอยหลัง
- บรรทัดที่ 33-56 เป็นส่วนคำนวณนับเวลาถอยหลังเป็นคูป

#### 4.2.1 การพัฒนาส่วนส่งข้อมูลไปอีกหน้า

```
1 send(){
2     let data1 = {"demo": this.docId,"fav":this.oxx,"sound":this.sound};
3     localStorage.setItem('myData',JSON.stringify(data1));
4     console.log(this.docId);
5 }
```

ภาพที่ 4.21 : โครงสร้างส่งข้อมูลไปอีกหน้า

จากภาพที่ 4.21 โครงสร้างส่วนส่งข้อมูลไปอีกหน้า อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-5 เป็นฟังก์ชันส่งค่าข้อมูลไปหน้ารายการโปรด

#### 4.2.2 การพัฒนาส่วนหน้าเก็บเพลงโปรด

```

1 this.reload()
2 }
3 reload(){
4   this.data1 = JSON.parse(localStorage.getItem('myData'));
5   console.log(this.data1);
6   this.docId= this.data1.demo
7   this.listfav=this.data1.fav
8   console.log(this.listfav);
9   this.db.collection('acc').snapshotChanges().subscribe(data => {
10     this.list = data.map(e => {
11       return {
12         id: e.payload.doc.id,
13       };
14     }
15   );
16   console.log(this.list)
17
18   for (let i=0; i< this.list.length; i++){
19
20     this.m1 = (this.list[i].bird)
21     if(this.m1>0){
22       this.user[1]=(this.sounds[1])
23     }
24   }
25
26 })};}

```

ภาพที่ 4.22 : ตัวอย่างโครงสร้างหน้าเก็บเพลงโปรด

จากภาพที่ 4.22 ตัวอย่างโครงสร้างหน้าเก็บเพลงโปรด อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-2 เป็นการเรียกใช้ฟังก์ชัน

- บรรทัดที่ 3-26 เป็นฟังก์ชันรับค่าข้อมูลที่ส่งมาและแสดงค่า

```

1 delete(i) {
2 // window.location.reload();
3 this.navCtrl.back()
4 console.log(i);
5 console.log(this.listfav[i].title);
6 let tmpobj
7
8 if(this.listfav[i].title ==="Sea Wave"){
9 let tmpobj1 = //db : inputform
10 { m1 : 0
11 };
12 tmpobj = tmpobj1
13
14 }
15 return this.db.doc('acc/' +this.docId).update(tmpobj);
16
17 }

```

ภาพที่ 4.23 : ตัวอย่างโครงสร้างหน้าเก็บเพลงโปรด

จากภาพที่ 4.23 ตัวอย่างโครงสร้างหน้าเก็บเพลงโปรด อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-17 เป็นฟังก์ชันลบเพลงออกจากรายการโปรด

```

1 ionViewWillEnter(){
2   this.data1 = JSON.parse(localStorage.getItem('myData'));
3   console.log(this.data1);
4   this.docId= this.data1.demo
5 }

```

ภาพที่ 4.24 : โครงสร้างหน้าเก็บเพลงโปรด

จากภาพที่ 4.24 โครงสร้างหน้าเก็บเพลงโปรด อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-5 เป็นฟังก์ชันรับค่าข้อมูลจากหน้าหลัก

#### 4.2.3 การพัฒนาส่วนผู้ดูแลระบบ

```

1 constructor(private DataService: DataService, private alertCtrl: AlertController,private modalCtrl: ModalController) {
2   this.dataService.getSongs().subscribe(res => {
3     console.log(res);
4     this.songs = res;
5   });
6 }
```

ภาพที่ 4.25 : โครงสร้างส่วนแสดงข้อมูล

จากภาพที่ 4.25 โครงสร้างส่วนแสดงข้อมูล อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นส่วนตัวแปรนำเข้าข้อมูล
- บรรทัดที่ 2-4 เป็นส่วนแสดงเพลง

```

1async deleteSong(){
2  await this.dataService.deleteSong(this.song);
3  this.modalCtrl.dismiss();
4 }
```

ภาพที่ 4.26 : โครงสร้างส่วนลบเพลง

จากภาพที่ 4.25 โครงสร้างส่วนลบเพลง อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-4 เป็นส่วนลบเพลง

```

1 async addSong() {
2   const alert = await this.alertCtrl.create({
3     header: 'Add Songs',
4     inputs: [
5       {
6         name: 'title',
7         placeholder: 'Name Song',
8         type: 'text'
9       },
10      {
11        name: 'subTitle',
12        placeholder: 'subtitle',
13        type: 'textarea'
14      },
15      {
16        name: 'img',
17        placeholder: 'img url',
18        type: 'url'
19      },
20      {
21        name: 'path',
22        placeholder: 'song url',
23        type: 'text'
24      }
25    ],
26    buttons: [
27      {
28        text: 'Cancel',
29        role: 'cancel'
30      }, {
31        text: 'Add',
32        handler: res => {
33          this.dataService.addSong({ title: res.title, subTitle: res.subTitle, img: res.img, path: res.path });
34        }
35      }
36    ]
37  });
38
39  await alert.present();
40 }

```

ภาพที่ 4.27 : โครงสร้างส่วนเพิ่มเพลง

จากภาพที่ 4.27 โครงสร้างส่วนเพิ่มเพลง อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 2 ทำการเรียกใช้ alertCtrl ผ่าน create() จะทำการสร้าง dialog สำหรับกรอกข้อมูล
- บรรทัดที่ 3-25 เป็นส่วนของกรอก ชื่อเพลง,ชื่อร้องเพลง,url รูป และ url เพลง
- บรรทัดที่ 26-29 เป็นปุ่มยกเลิก
- บรรทัดที่ 31-33 เป็นปุ่มเพิ่มเพลงและเรียกใช้ dataService ในการเพิ่มข้อมูลเพลง
- บรรทัดที่ 39 เป็นฟังก์ชันแสดง dialog

## บทที่ 5

### การทดสอบระบบ

การทดสอบการทำงานของระบบมีวัตถุประสงค์เพื่อตรวจสอบความถูกต้องของการทำงานแอปพลิเคชัน โดยทดสอบแบบ Black Box Testing นั้น จะจำลองระบบเป็นเหมือนกล่องดำ (Black box) โดยที่ไม่สนใจกระบวนการการทำงานว่ามีการทำงานของฟังก์ชันในระบบอย่างไร แต่จะตรวจสอบว่าเมื่อระบบทำงานสำเร็จผลลัพธ์ที่ได้ถูกต้องหรือไม่

#### 5.1 การทดสอบใช้งานแอปพลิเคชัน

##### 5.1.1 การทดสอบหน้าจอหลัก

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าจอหลัก	แสดงเพลงที่มีการพิมพ์มากที่สุด	ระบบแสดงเพลงที่มีการพิมพ์มากสุดแสดงขึ้นเป็นเพลงแรก
	เตือนเวลาอ่อน	ระบบจะเตือนเวลาเข้าบันนอนเมื่อถึงเวลาที่สามารถตั้งไว้

##### 5.1.2 การทดสอบหน้ารายการโปรด

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายการโปรด	แสดงเพลงเพิ่มที่เพิ่มรายการโปรด	ระบบแสดงเพลงที่เพิ่มเป็นรายการโปรด
	ลบเพลงรายการโปรด	เพลงที่ลบจากรายการโปรดจะไม่แสดง

### 5.1.3 การทดสอบส่วนเพิ่มเพลง

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าเพิ่มเพลง	แสดงเพลงที่เพิ่ม	ระบบแสดงเพลงที่เพิ่ม
	ลบเพลง	เพลงที่ลบจะไม่แสดงในระบบ

### 5.1.4 การทดสอบหน้าเล่นเพลง

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าเล่นเพลง	เล่นเพลงที่เลือก	ระบบเล่นตามเพลงที่เลือกได้ถูกต้อง
	ตั้งเวลาหยุดเพลง	เพลงหยุดตามเวลาที่ตั้งไว้

## บทที่ 6

### สรุปและข้อเสนอแนะ

โครงงานแอปพลิเคชันเสียงเพื่อการผ่อนคลายนี้ พบร่วมระบบมีส่วนที่ยังไม่สามารถทำงานได้ตามที่ออกแบบไว้ ในบทนี้ผู้พัฒนาจึงขอสรุปความสามารถของแอปพลิเคชัน ซึ่งจะปัญหาและอุปสรรค พร้อมเสนอแนวทางพัฒนาต่อ ตามลำดับ

#### 6.1 สรุปความสามารถแอปพลิเคชัน

##### 6.1.1 เว็บ

###### ผู้ดูแลระบบ

- เพิ่มเพลง
- เพิ่มรูป

##### 6.1.2 แอปพลิเคชัน

###### สมาชิก

- สามารถสมัครและเข้าสู่ระบบได้
- สามารถตั้งเวลาเข้านอน
- สามารถเพิ่มเสียงเป็นรายการโปรดได้
- สามารถลบเพลงที่เป็นรายการโปรดได้
- สามารถคุ้มเพลงขิดได้
- สามารถตั้งเวลาปิดเสียงได้

## 6.2 ปัญหาและอุปสรรคในการพัฒนา

- สร้างแอปพลิเคชัน ไม่สามารถสร้างแอปพลิเคชันโดยใช้ ionic capacitor ได้  
แนวทางในการแก้ไข : เปิดยังมาใช้ ionic cordova ในการสร้างแอปพลิเคชัน
- เวอร์ชันแอนดรอยด์ ไม่สามารถใช้แอนดรอยด์เวอร์ชัน 9 ในการสร้างแอปพลิเคชัน  
แนวทางในการแก้ไข : ใช้แอนดรอยด์เวอร์ชัน 10 ขึ้นไป

## 6.3 แนวทางพัฒนาต่อ

- เพิ่มเพลงให้หลากหลายแนว
- เพิ่มฟังก์ชันการเล่นในเบื้องหลัง

## บรรณานุกรม

- [1] N.Khodom (2018). มาเริ่มต้นสร้างแอพฯ แบบ Cross platform ด้วย Ionic Framework กันเถอะ [ออนไลน์]. สืบค้นเมื่อ 10 พฤษภาคม 2565. จาก <https://medium.com/artisan-digital-agency/มาเริ่มต้นสร้างแอพฯ-แบบ-cross-platform-ด้วย-ionic-framework-กันเถอะ-986c7d549780> .
- [2] luknam UIC (ม.ป.ป.). ระบบปฏิบัติการ ANDROID และ IOS [ออนไลน์]. สืบค้นเมื่อ 11 พฤษภาคม 2565. จาก <https://sites.google.com/site/luknamuic/home/rabb-ptibati-kar-1> .
- [3] Ponglang Petrung (2017). บทที่ 1 สถาปัตยกรรมของระบบแอนดรอยด์ ( โครงสร้างของระบบแอนดรอยด์) เป็งตัน [ออนไลน์]. สืบค้นเมื่อ 11 พฤษภาคม 2565. จาก <https://pongployappdev.medium.com/บทที่-1-สถาปัตยกรรมของระบบแอนดรอยด์-โครงสร้างของระบบแอนดรอยด์-เบ็งตัน-75481fcadbd8> .
- [4] nich (2020). Firebase คืออะไร [ออนไลน์]. สืบค้นเมื่อ 13 พฤษภาคม 2565. จาก <https://www.4xtreme.com/2020/11/20/firebase-คืออะไร/> .
- [5] marcuscode (2021). ทำความรู้จักกับ Node.js [ออนไลน์]. สืบค้นเมื่อ 13 พฤษภาคม 2565. จาก <http://marcuscode.com/tutorials/nodejs/introducing-nodejs> .
- [6] Mayon (2020). รีวิว Relax Melodies แอปฯ ช่วยคนนอนไม่หลับ ผ่อนคลายด้วยเสียง [ออนไลน์]. สืบค้นเมื่อ 27 พฤษภาคม 2565. จาก [https://intrend.trueid.net/article/รีวิว-relax-melodies-แอปฯ-ช่วยคนนอนไม่หลับ-ผ่อนคลายด้วยเสียง-trueidintrend\\_189029](https://intrend.trueid.net/article/รีวิว-relax-melodies-แอปฯ-ช่วยคนนอนไม่หลับ-ผ่อนคลายด้วยเสียง-trueidintrend_189029)
- [7] Craftsman Spirit (2019). Sleep Sounds - relaxing sounds [ออนไลน์]. สืบค้นเมื่อ 27 พฤษภาคม 2565. จาก <https://play.google.com/store/apps/details?id=relax.sleep.relaxation.sleepingsounds&hl=th&gl=US>
- [8] Maple Media (2017). Sleepo: เสียงผ่อนคลาย, นอนหลับ [ออนไลน์]. สืบค้นเมื่อ 27 พฤษภาคม 2565. จาก <https://play.google.com/store/apps/details?id=net.relaxio.sleepo> .
- [9] mikdroid (2013). ผ่อนคลายชนบท - เสียงธรรมชาติ [ออนไลน์]. สืบค้นเมื่อ 27 พฤษภาคม 2565. จาก <https://play.google.com/store/apps/details?id=it.mm.android.relaxcountryside> .

[10] Dream\_Studio (2015). เสียงธรรมชาติ [ออนไลน์]. สืบค้นเมื่อ 27 พฤษภาคม 2565. จาก <https://play.google.com/store/apps/details?id=com.dreamstudio.naturesounds> .

ภาคผนวก

## ภาคผนวก ก

### การติดตั้งเครื่องมือในการพัฒนาแอปพลิเคชัน

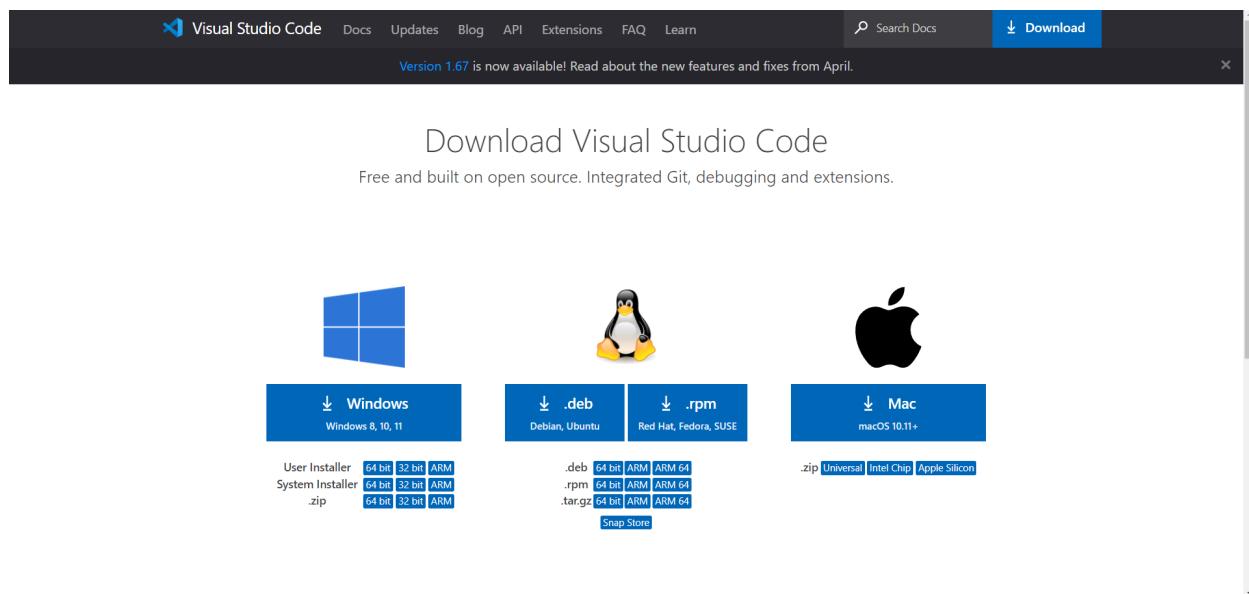
โปรแกรมในการพัฒนาแอปพลิเคชันมีดังต่อไปนี้

- การติดตั้ง Visual Studio Code
- การติดตั้ง Node.js
- การติดตั้ง Ionic Framework
- การติดตั้ง Android Studio

#### ก.1 การติดตั้ง Visual Studio Code

1. ดาวน์โหลด Visual Studio Code ได้ที่ <https://code.visualstudio.com/download> เลือกติดตั้งตามระบบปฏิบัติการ ดัง

ภาพ ก.1



ภาพที่ ก.1 : หน้าเว็บดาวน์โหลด Visual Studio Code

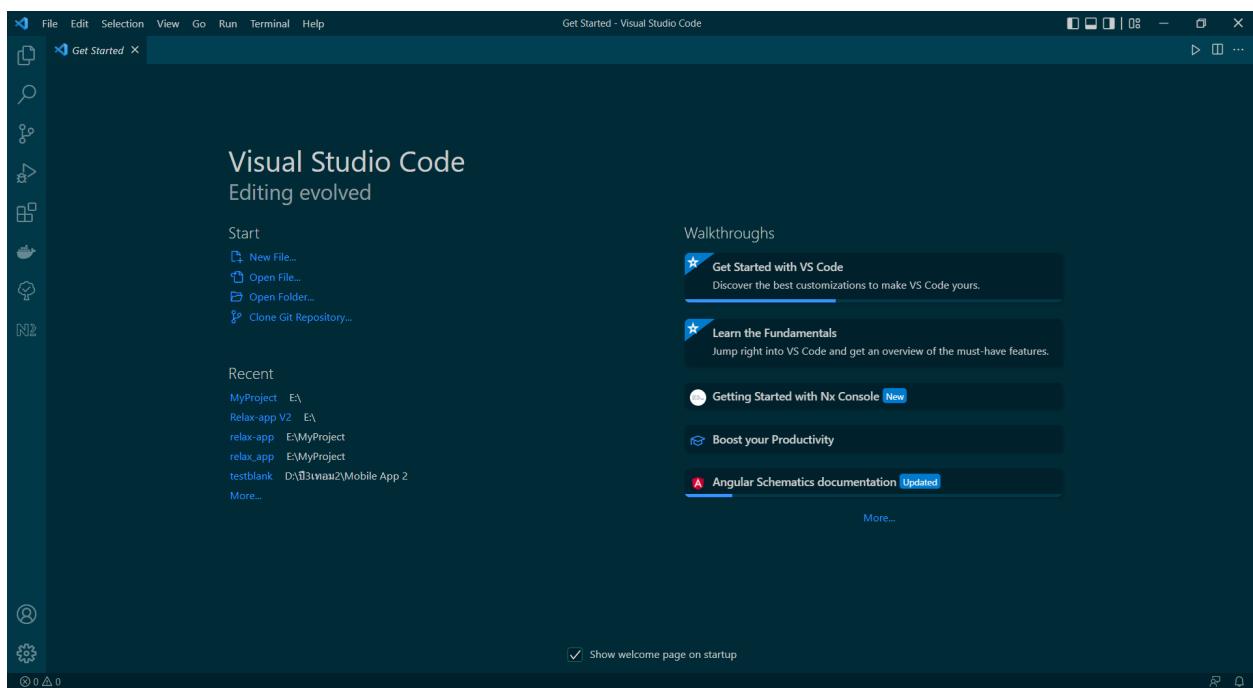
ที่มา: <https://code.visualstudio.com/download>

2. เปิดไฟล์ที่ดาวน์โหลดมา ดังภาพ ก.2



ภาพที่ ก.2 :ไฟล์โปรแกรม Visual Studio Code

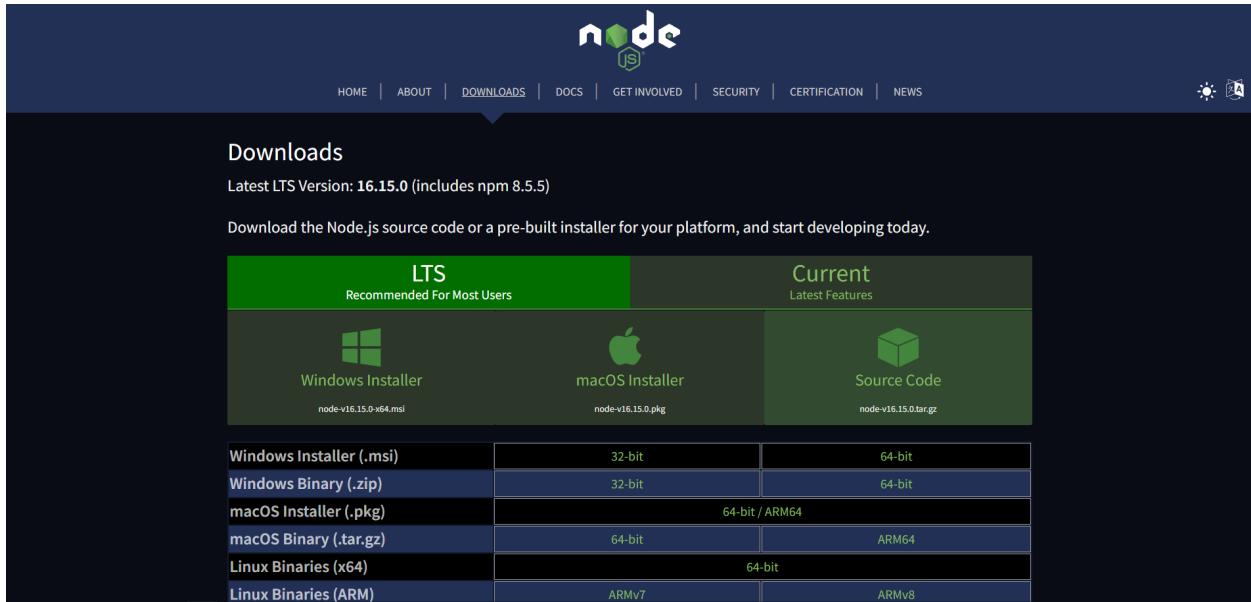
3. โปรแกรมพร้อมใช้งาน ดังภาพ ก.3



ภาพที่ ก.3 : โปรแกรม Visual Studio Code

## ก.2 การติดตั้ง Node.js

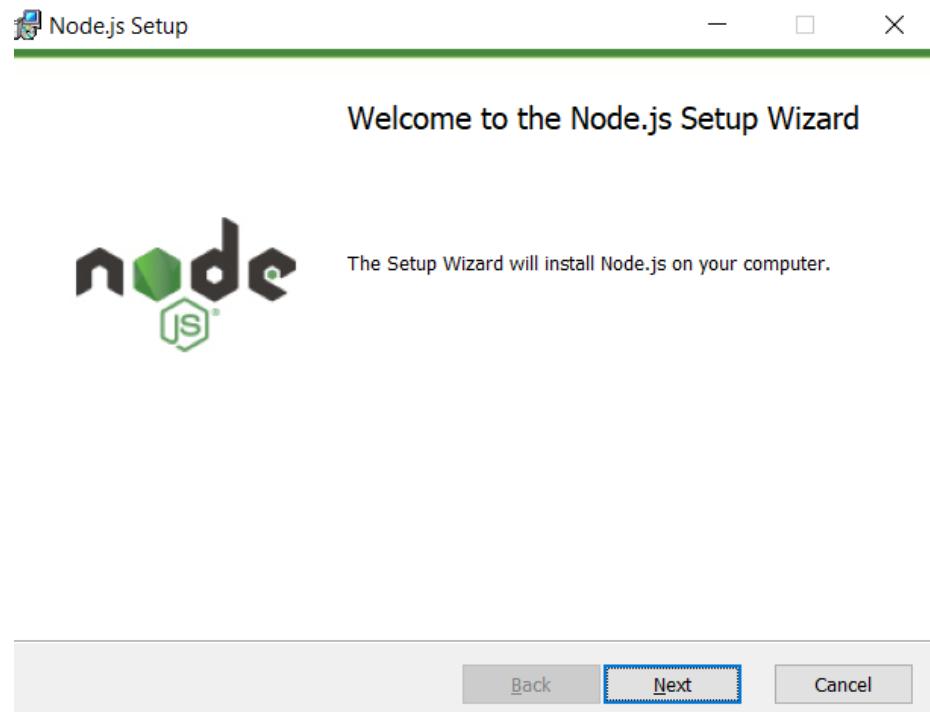
1. ดาวน์โหลด Node.js ได้ที่ <https://nodejs.org/en/download/> เลือกติดตั้งตามระบบปฏิบัติการ ดังภาพ ก.4



ภาพที่ ก.4 : หน้าเว็บดาวน์โหลด Node.js

ที่มา: <https://nodejs.org/en/download/>

2. กดติดตั้งไฟล์ที่โหลดมา ดังภาพ ก.5



ภาพที่ ก.5 : ติดตั้ง Node.js

3. เช็คว่าติดสำเร็จหรือไม่ โดยการเปิด Command Prompt พร้อมพิมคำสั่ง npm ดังภาพ ก.6

```
Usage: npm <command>
```

where <command> is one of:

```
access, adduser, audit, bin, bugs, c, cache, ci, cit,
clean-install, clean-install-test, completion, config,
create, ddp, dedupe, deprecate, dist-tag, docs, doctor,
edit, explore, fund, get, help, help-search, hook, i, init,
install, install-ci-test, install-test, it, link, list, ln,
login, logout, ls, org, outdated, owner, pack, ping, prefix,
profile, prune, publish, rb, rebuild, repo, restart, root,
run, run-script, s, se, search, set, shrinkwrap, star,
stars, start, stop, t, team, test, token, tst, un,
uninstall, unpublish, unstar, up, update, v, version, view,
whoami
```

```
npm <command> -h    quick help on <command>
npm -l              display full usage info
npm help <term>     search for help on <term>
npm help npm        involved overview
```

Specify configs in the ini-formatted file:

```
C:\Users\OMEN\.npmrc
```

or on the command line via: npm <command> --key value  
Config info can be viewed via: npm help config

```
npm@6.13.4 D:\Program\nodejs\node_modules\npm
```

### ก.3 การติดตั้ง Ionic Framework

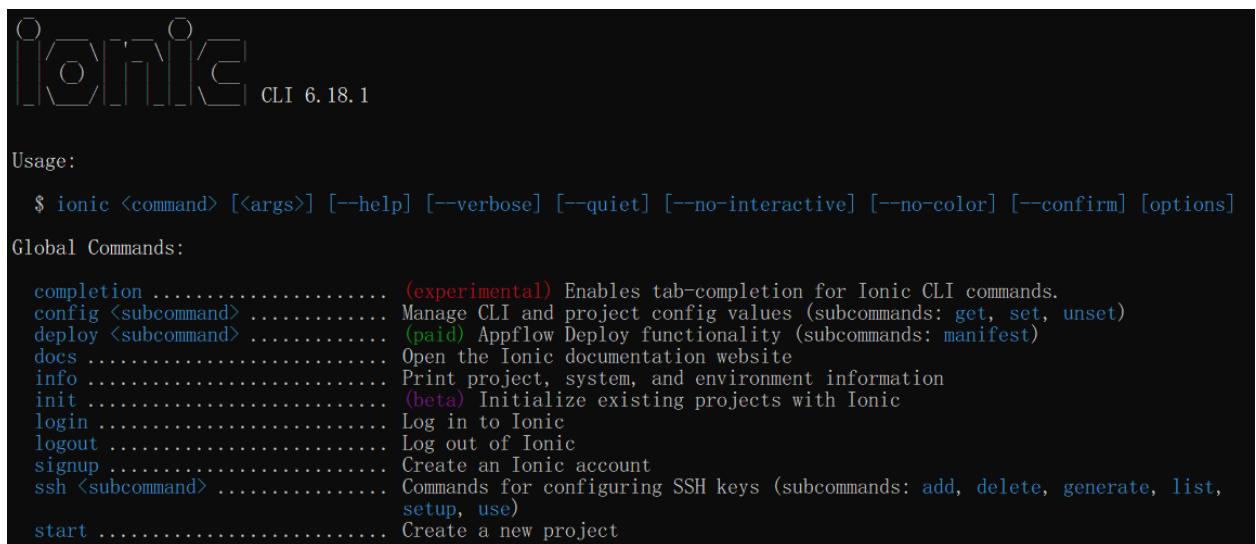
1. การติดตั้ง Ionic Framework สามารถติดตั้งโดยเปิด Command Prompt พร้อมพิมคำสั่ง `npm install -g @ionic/cli` ดัง

ภาพ ก.7

```
>npm install -g @ionic/cli
```

ภาพที่ ก.7 : คำสั่งติดตั้ง Ionic Framework

2. ผลการติดตั้ง Ionic Framework ดังภาพ ก.8



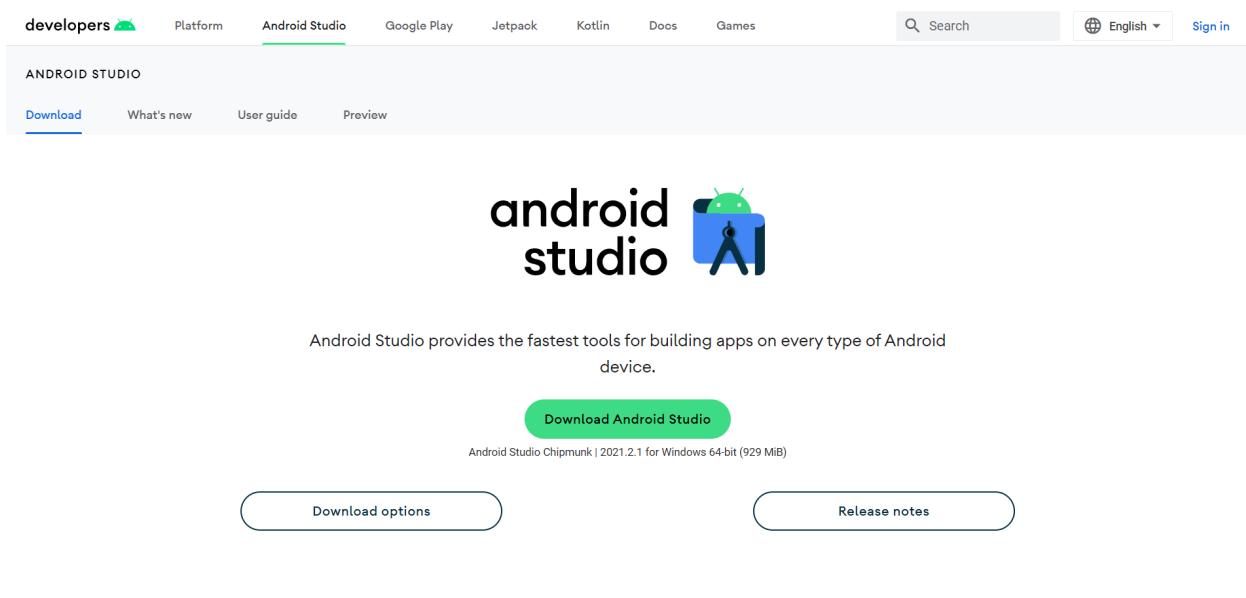
```
○○[○]CLI 6.18.1
Usage:
$ ionic <command> [<args>] [--help] [--verbose] [--quiet] [--no-interactive] [--no-color] [--confirm] [options]
Global Commands:
completion ..... (experimental) Enables tab-completion for Ionic CLI commands.
config <subcommand> ..... Manage CLI and project config values (subcommands: get, set, unset)
deploy <subcommand> ..... (paid) Appflow Deploy functionality (subcommands: manifest)
docs ..... Open the Ionic documentation website
info ..... Print project, system, and environment information
init ..... (beta) Initialize existing projects with Ionic
login ..... Log in to Ionic
logout ..... Log out of Ionic
signup ..... Create an Ionic account
ssh <subcommand> ..... Commands for configuring SSH keys (subcommands: add, delete, generate, list,
                      setup, use)
start ..... Create a new project
```

ภาพที่ ก.8 : ติดตั้ง Ionic Framework เสร็จสมบูรณ์

#### ก.4 การติดตั้ง Android Studio

##### 1. ดาวน์โหลด Android Studio ได้ที่

[https://developer.android.com/studio?gclid=EAIaIQobChMI3MzSkIrd9wIVUA4rCh1qdwLDEAAYASAAEglaR\\_D\\_BwE&gclsrc=aw.ds](https://developer.android.com/studio?gclid=EAIaIQobChMI3MzSkIrd9wIVUA4rCh1qdwLDEAAYASAAEglaR_D_BwE&gclsrc=aw.ds) ดูภาพ ก.9

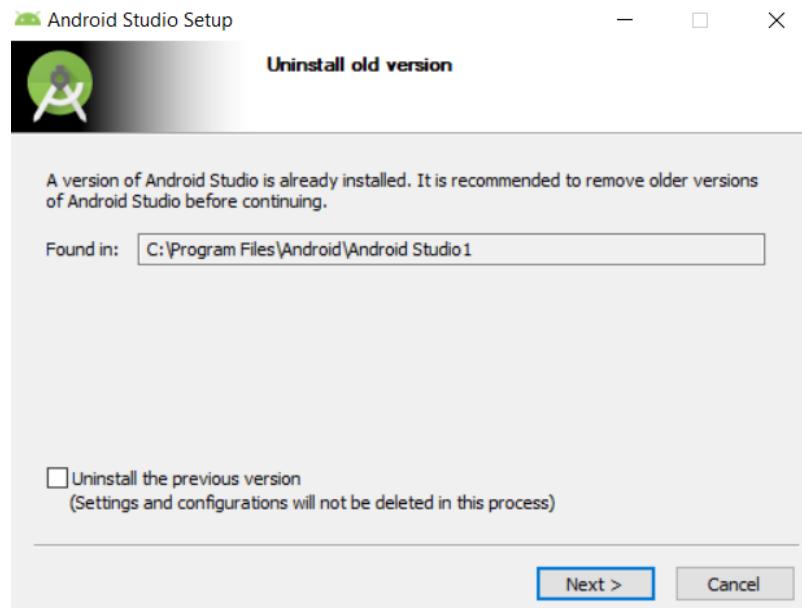


ภาพที่ ก.9 : หน้าเว็บดาวน์โหลด Android Studio

ที่มา:

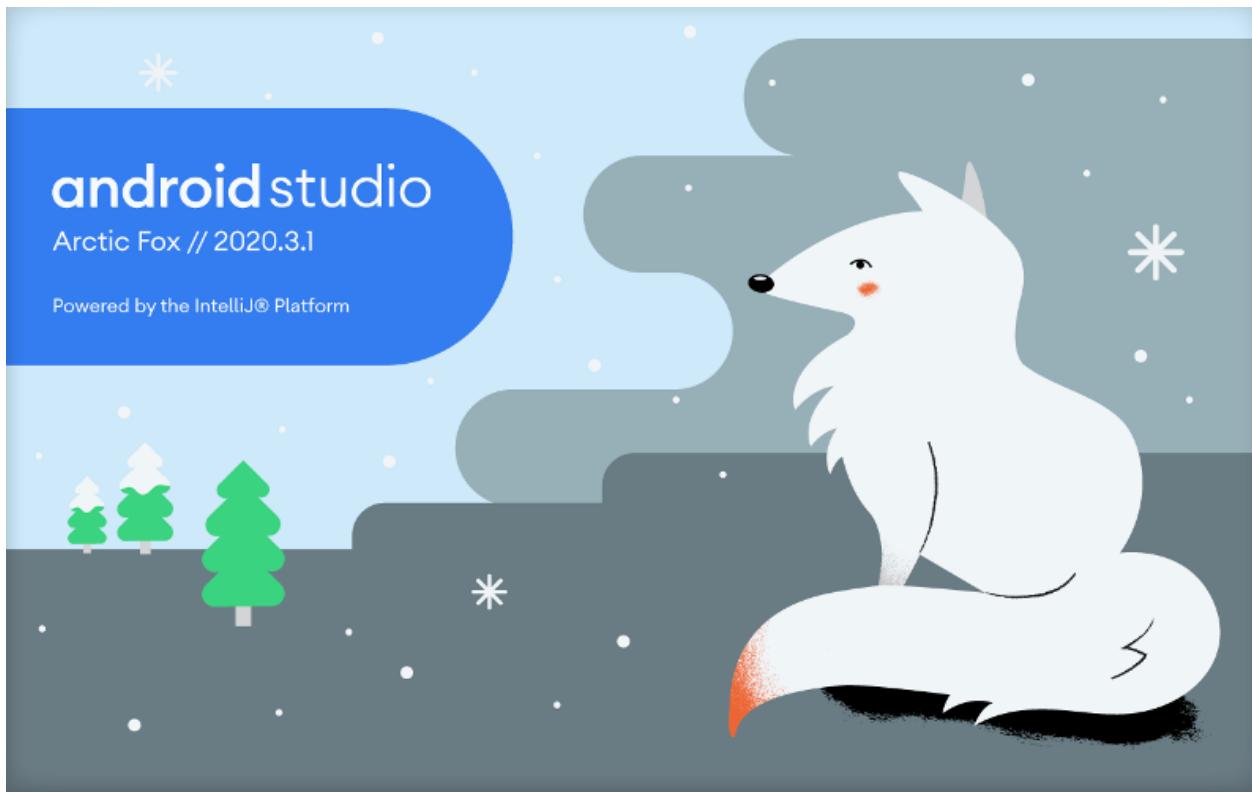
[https://developer.android.com/studio?gclid=EAIaIQobChMI3MzSkIrd9wIVUA4rCh1qdwLDEAAYASAAEgLDKPD\\_BwE&gclsrc=aw.ds](https://developer.android.com/studio?gclid=EAIaIQobChMI3MzSkIrd9wIVUA4rCh1qdwLDEAAYASAAEgLDKPD_BwE&gclsrc=aw.ds)

2. ติดตั้งไฟล์ Android Studio ที่ดาวน์โหลดมา ดังภาพ ก.10



ภาพที่ ก.10 : หน้าติดตั้ง Android Studio

3. เมื่อติดตั้งโปรแกรมแล้ว จะแสดงดังภาพ ก.11



ภาพที่ ก.11 : ติดตั้ง Android Studio เสร็จสมบูรณ์

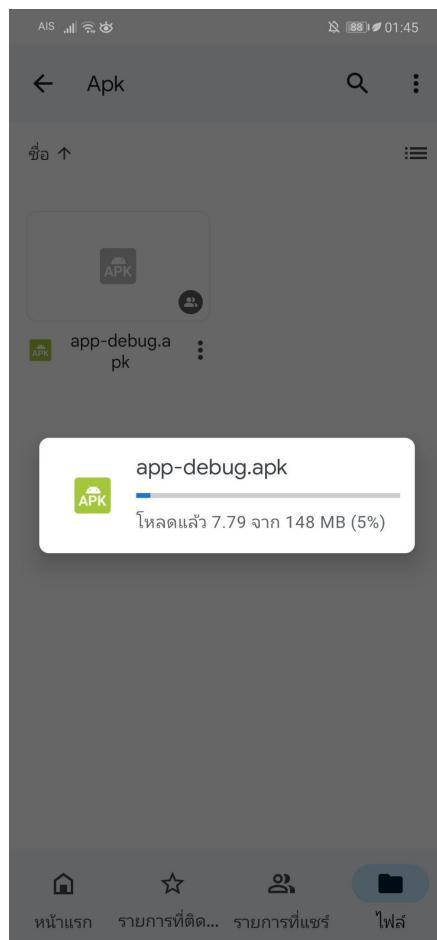
## ภาคผนวก ข

### คู่มือการติดตั้งแอปพลิเคชัน

การติดตั้งแอปพลิเคชัน Relax Sound มีขั้นตอนดังต่อไปนี้

1. ดาวน์โหลด Relax Sound ได้ที่

[https://drive.google.com/drive/u/0/folders/1ZeLnmyWYqlXRfzkucSNrQRdn-2A\\_jemR](https://drive.google.com/drive/u/0/folders/1ZeLnmyWYqlXRfzkucSNrQRdn-2A_jemR) ดูภาพที่ ข.1



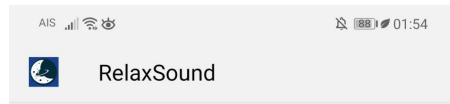
ภาพที่ ข.1 : หน้าดาวน์โหลดแอปพลิเคชัน Relax Sound

2. กดติดตั้งแอปพลิเคชัน ดังภาพที่ ข.2



ภาพที่ ข.2 : หน้าแสดงการติดตั้ง

3. กดเปิด เพื่อเริ่มใช้งานแอปพลิเคชัน ดังภาพที่ ข.3



ติดตั้งแอปพลิเคชันแล้ว



ภาพที่ ข.3 : หน้าแสดงการติดตั้งสำเร็จ

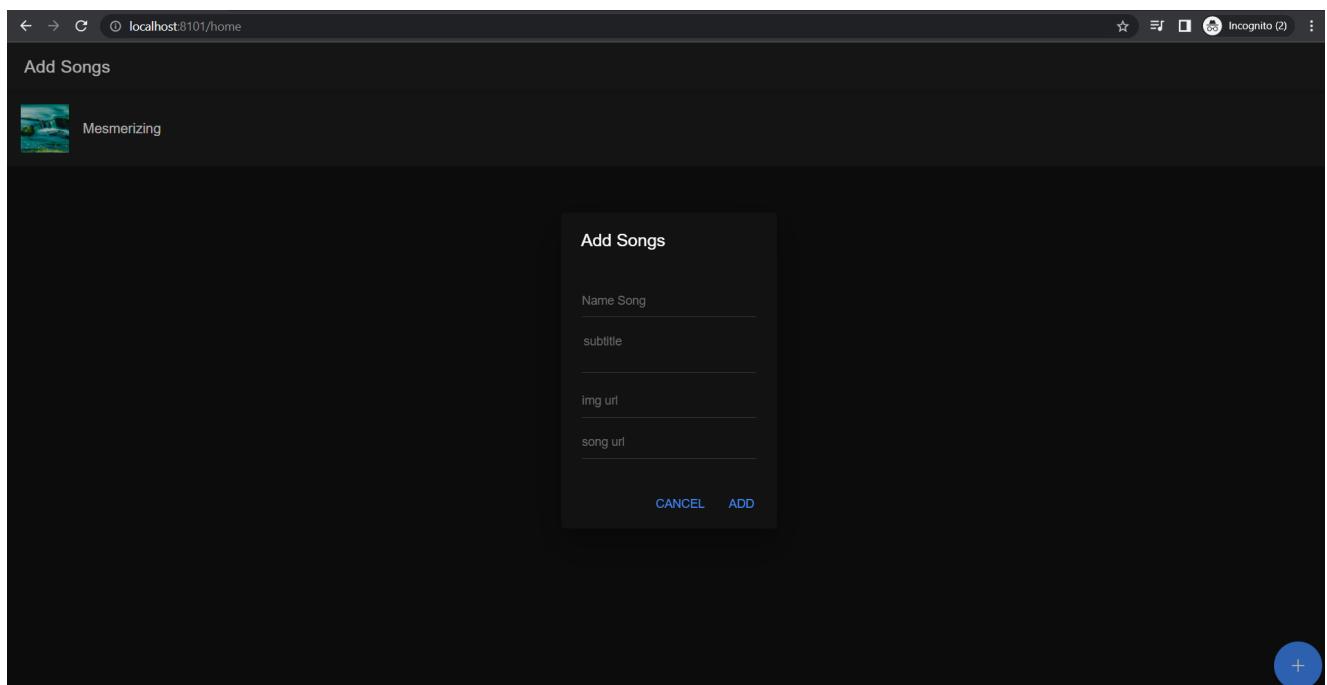
## ภาคผนวก ค

### คู่มือการใช้งานระบบ

คู่มือการใช้งานระบบแบ่งเป็น 2 ส่วนดังนี้

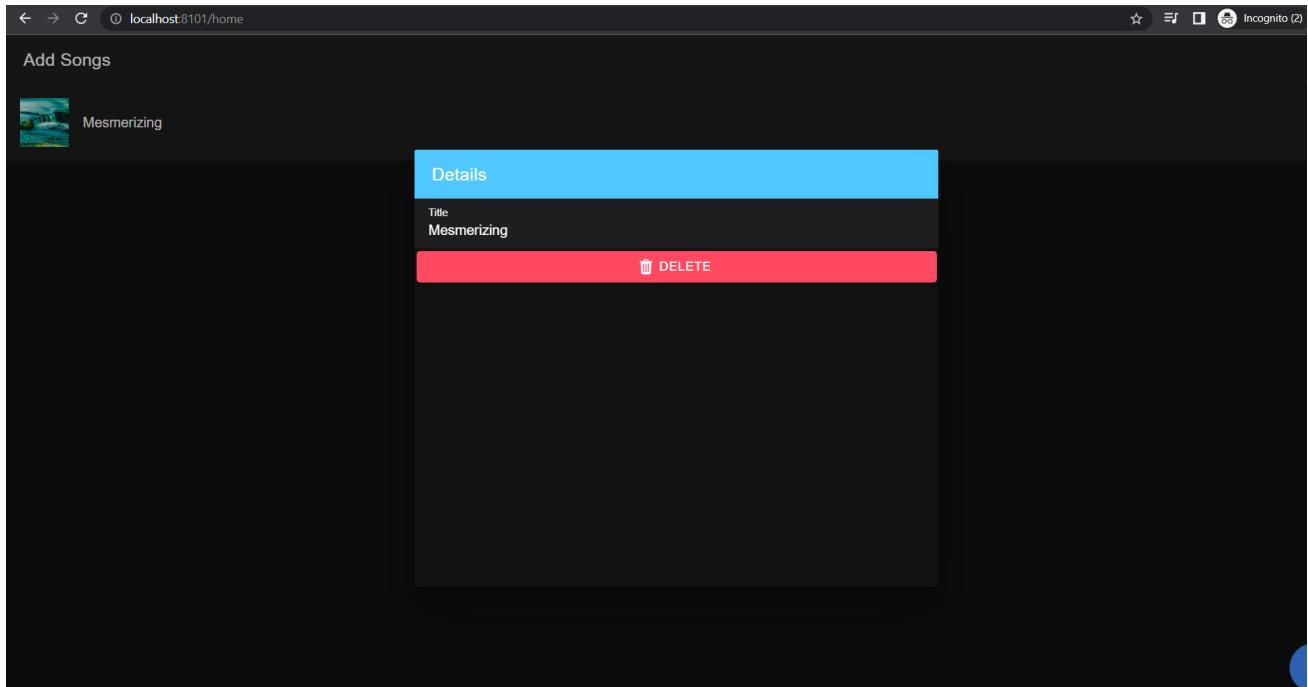
#### 1. ส่วนเว็บผู้ดูแลระบบ

- หน้าจอในหน้านี้ผู้ดูแลระบบสามารถเพิ่มเพลง โดยกรอก ชื่อเพลง ชื่อร้อง ที่อยู่รูป ที่อยู่เพลง ดังภาพที่ ค.1



ภาพที่ ค.1 : หน้าจอแสดงเพลงและเพิ่มเพลง

- ผู้ดูแลระบบสามารถลบเพลงที่เพิ่ม ดังภาพที่ ค.2



ภาพที่ ค.2 : หน้าจอส่วนลบเพลง

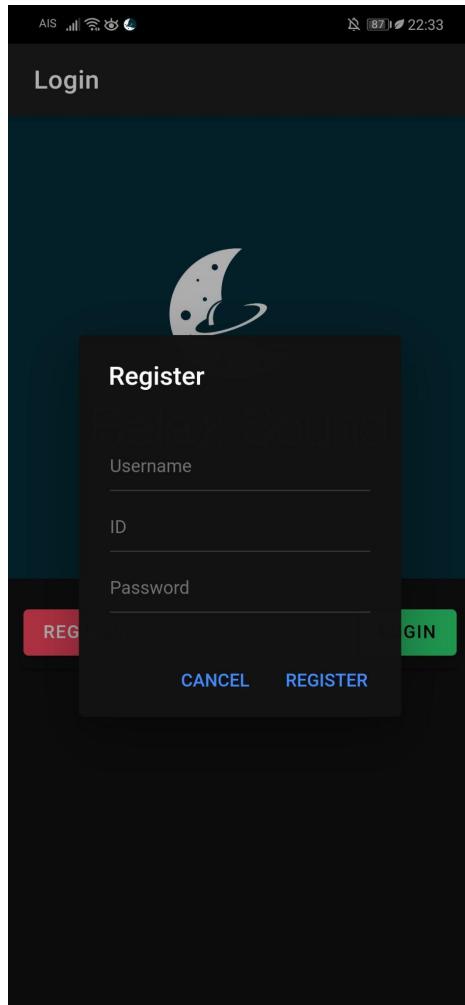
2. ส่วนหน้าจอป้องกันสำหรับสมาชิก

- หน้าจอต้อนรับเวลาเข้าแอปพลิเคชัน ดังภาพที่ ค.3



ภาพที่ ค.3 : หน้าจอต้อนรับ

- หน้าสมัครสมาชิก ดังภาพที่ ค.4

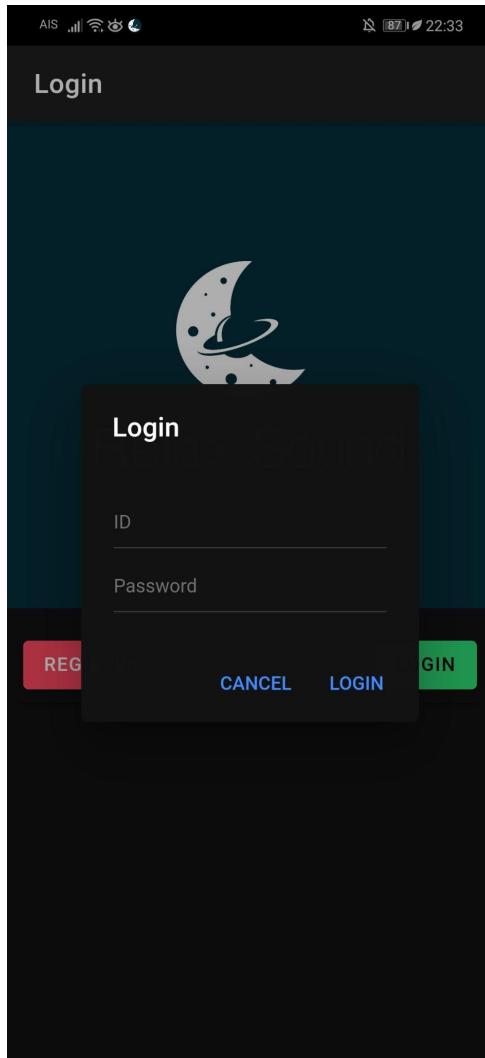


ภาพที่ ค.4 : หน้าสมัครสมาชิก

จากภาพที่ ค.4 สามารถอธิบายได้ดังนี้

- ผู้ใช้ที่สามารถกรอกข้อมูลสมัครสำหรับเข้าใช้งาน

- หน้าเข้าสู่ระบบ ดังภาพที่ ค.5

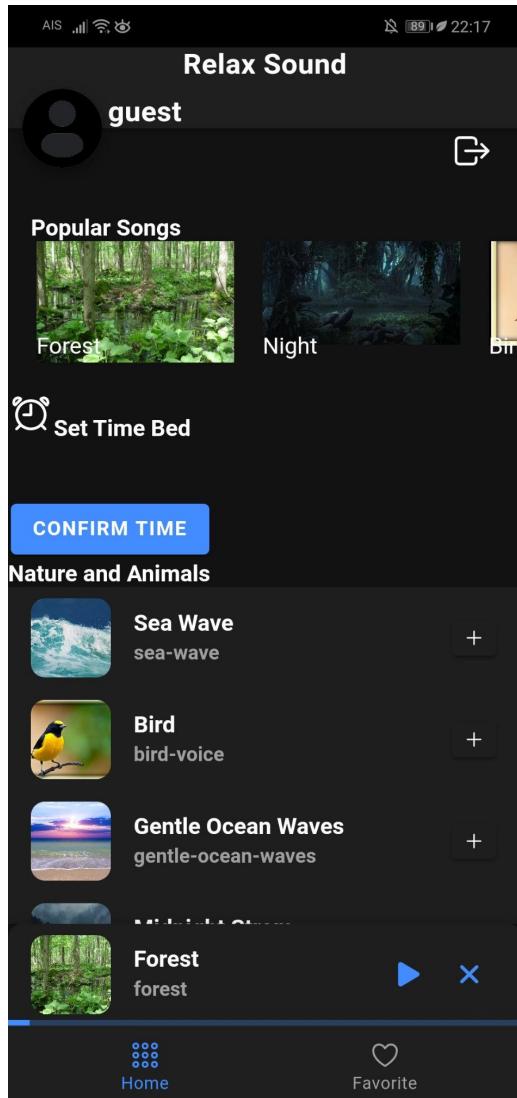


ภาพที่ ค.5 : หน้าเข้าสู่ระบบ

จากภาพที่ ค.5 สามารถอธิบายได้ดังนี้

- สามารถกรอกข้อมูลเข้าสู่ระบบ เมื่อสมัครสมาชิกแล้ว

- หน้าจอแสดงผลหลักสำหรับเล่นเพลง และแสดงเพลงที่ samaชิกชอบฟัง และอื่นๆ ดังภาพที่ ค.6

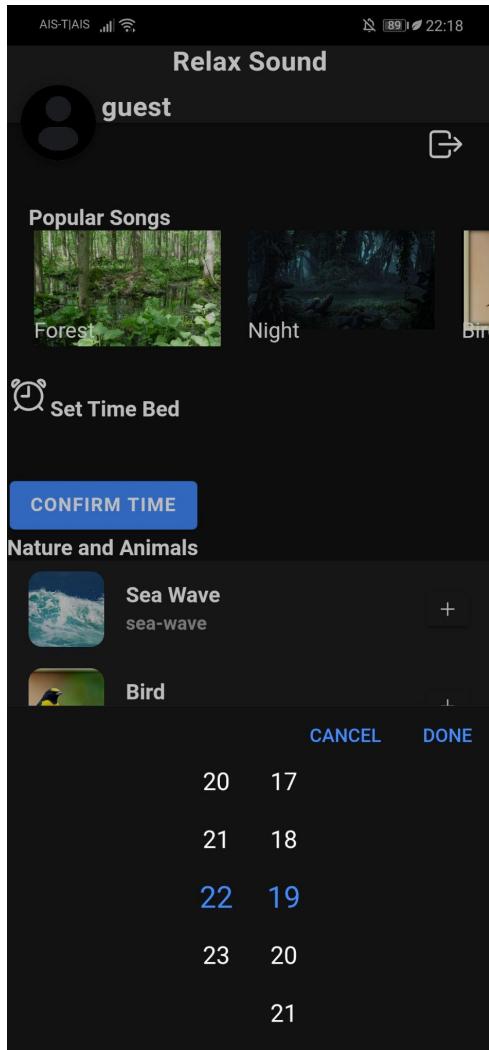


ภาพที่ ค.6 : หน้าจอหลัก

จากภาพที่ ค.6 สามารถอธิบายได้ดังนี้

- สามารถดูเพลงที่ชอบฟังมากที่สุด
- สามารถกดเลือกเล่นเพลง
- สามารถเพิ่มเพลงไปรายการโปรดโดยกดไอคอนเครื่องหมายบวก
- สามารถออกจากระบบโดยกดไอคอนรูปประตุมุมบนขวา
- สามารถกดไอคอนรูปหัวใจเพื่อถูกรายโปรด

- หน้าจอส่วนตั้งเวลาเข้านอน ดังภาพที่ ค.7

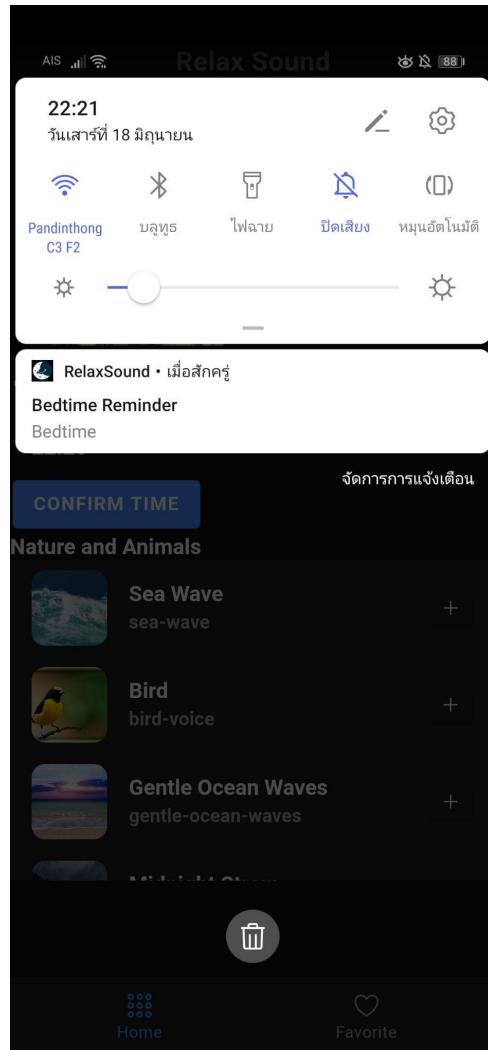


ภาพที่ ค.7 : หน้าจอส่วนตั้งเวลาเข้านอน

จากภาพที่ ค.7 สามารถอธิบายได้ดังนี้

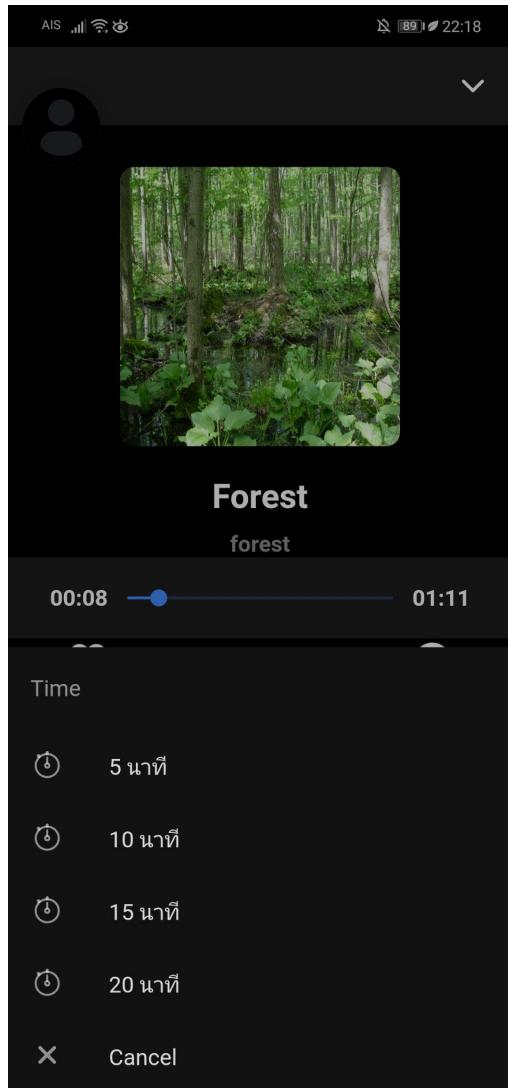
- สามารถตั้งเวลาสำหรับแจ้งเตือนการนอน โดยการกดไอคอนนาฬิกา เลือกเวลาที่ต้องการ กด Done และ กด CONFIRM TIME

- ส่วนแจ้งเตือนเวลาเข้านอน เมื่อถึงเวลาที่スマาร์ตโฟนจะแจ้งเตือน ดังภาพที่ ค.8



ภาพที่ ค.8 : ส่วนแจ้งเตือนเวลาเข้านอน

- หน้าจอส่วนตั้งเวลาหยุดเล่นเพลง ดังภาพที่ ค.9

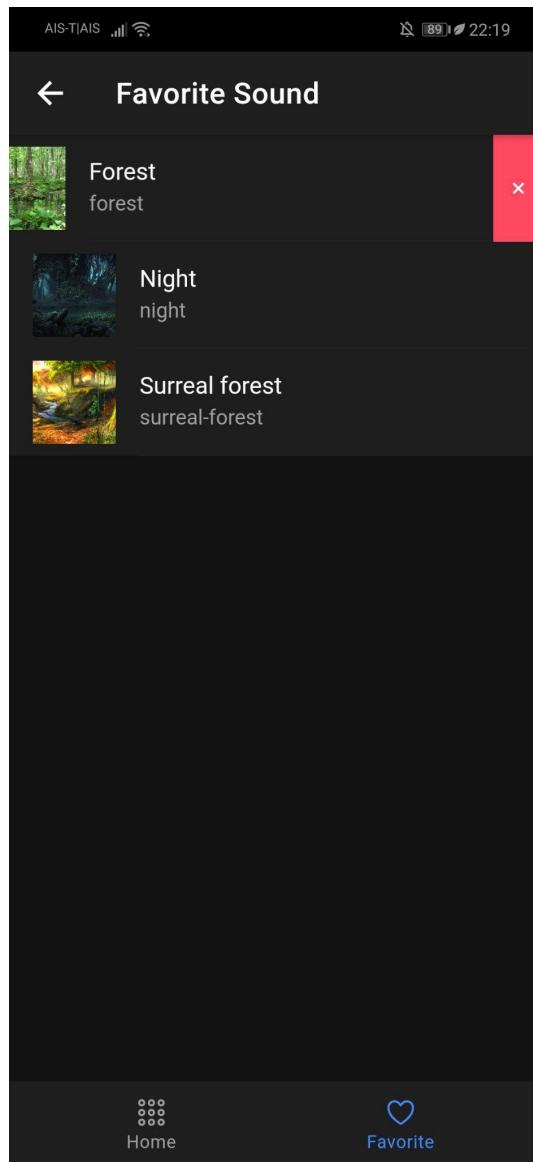


ภาพที่ ค.9 : หน้าจอตั้งเวลาหยุดเพลง

จากภาพที่ ค.9 สามารถอธิบายได้ดังนี้

- สามารถตั้งเวลาหยุดเพลงโดยกดรูปนาฬิกา และเลือกเวลาที่ต้องการหยุดเพลง

- หน้าจอส่วนรายการโปรด ดังภาพที่ ค.10



ภาพที่ ค.10 : หน้าจอรายการโปรด

จากภาพที่ ค.10 สามารถอธิบายได้ดังนี้

- สามารถดูเพลงที่เป็นรายการโปรด สามารถสไลด์ขวาเพื่อลบเพลงรายการโปรด

## ประวัติผู้เขียน

ชื่อ-สกุล นาย ศุภกิตติ์ วงศ์แก้ว

รหัสประจำตัวนักศึกษา 61114440720

วันเดือนปีเกิด 14 01 2543

ที่อยู่ 14 หมู่ 6 ต.ชุม อ.ศรีรัตน์ จ.ศรีสะเกษ

เบอร์โทรศัพท์ 093 973 8331

อีเมล supakit.wo.61@ubu.ac.th

ระดับมัธยมต้น โรงเรียนศรีรัตนวิทยา

ระดับมัธยมปลาย โรงเรียนศรีรัตนวิทยา

ระดับอุดมศึกษา ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์

สาขาวิชาการคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี