

## 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 หิรัญพงศ์สิน

ระดับการศึกษา : วิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

ปีการศึกษา : 2564

---

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา

ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

คณะกรรมการสอบประเมินความรู้โครงงานคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา

(ผศ.ดร. สุภาวดี หิรัญพงศ์สิน)

..... กรรมการ

(ดร. ทศพร อเลิร์ป)

..... กรรมการ

(อ. วานิช แห่งแกะ)

..... หัวหน้าภาควิชา

(ผศ.ดร.สุพจน์ สีบุตร)

ขันที่ ..../..../....

## กิตติกรรมประกาศ

โครงการเลี้ยงเพื่อการฝึกอบรมฯ สำหรับครุภัณฑ์ฯ ที่มีส่วนช่วยโครงการนี้

ขอขอบพระคุณอาจารย์ที่ปรึกษา พศ.ดร. สุวารดี หิรัญพงศ์สิน ที่เคยให้คำปรึกษาและช่วยแนะนำในการปรับแก้โครงการให้สมบูรณ์ ขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณบิดา มารดา ที่เคยให้กำลังใจ ความรักความห่วงใย ในการพัฒนาโครงการเสมอมา

นาย ศุภกิตติ์ วงศ์แก้ว

วันที่ 18 มิถุนายน 2565

โครงงาน : แอปพลิเคชันเสียงเพื่อการผ่อนคลาย

Relax Sound

โดย : นายศุภกิตติ์ วงศ์แก้ว

อาจารย์ที่ปรึกษา : ผศ.ดร. สุภารดี หรรษพงศ์สิน

ระดับการศึกษา : วิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

ปีการศึกษา : 2564

---

### บทคัดย่อ

ความเครียดสามารถส่งผลกระทบต่อมนุษย์ทุกเพศทุกวัย บทความนี้นำเสนอโดยแอปพลิเคชันเสียงเพื่อช่วยลด

ความเครียดจากการทำงานหรือการเรียน โดยใช้ออนิคเฟรมเวิร์คและฐานข้อมูลไฟล์เบส โดยผู้ใช้สามารถเข้าใช้งาน และ

ระบบจะให้ผู้ใช้เลือกประเภทเสียงที่ผู้ใช้สนใจ จากนั้นระบบจะแสดงเสียงเพลงที่เลือกมาจากผู้ใช้ในมีความสนใจในประเภทเสียง

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	15
บทนำ	15
1.1 ที่มาและเหตุผล	15
1.2 วัตถุประสงค์	15
1.3 ขอบเขตของโครงการ	15
1.4 ประโยชน์ที่ได้รับ	15
1.5 เครื่องมือที่ใช้ในการพัฒนา	16
1.5.1 ฮาร์ดแวร์	16
1.5.2 ซอฟต์แวร์ (Software)	16
1.5.3 แผนการดำเนินการ	17
บทที่ 2	18
ทฤษฎีที่เกี่ยวข้องและงานวิจัย	18
2.1 ความรู้พื้นฐาน Ionic Framework	18
2.2 ความรู้พื้นฐานเกี่ยวกับระบบปฏิบัติการแอนดรอยด์ (Android)	19
2.2.1 สถาปัตยกรรมของระบบแอนดรอยด์	20
2.3 ความรู้พื้นฐาน Firebase	22
2.3.1 Build Better Apps	23

2.3.2 Improve app quality	23
2.3.3 Grow your business	23
2.4 ความรู้พื้นฐาน Node.js	24
2.4.1 NPM Package management	26
2.6 แอปพลิเคชันที่เกี่ยวข้อง	26
บทที่ 3	33
การวิเคราะห์และออกแบบระบบ	33
3.1 โครงสร้างภาพรวมของระบบ (System Architecture)	34
3.2 การวิเคราะห์ความต้องการของระบบ (System Requirements)	35
3.2.1 Functional Requirements	35
3.2.2 Non-functional Requirements	35
3.3 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface Design)	35
3.3.1 หน้าสมัครสมาชิกและหน้าเข้าสู่ระบบ	36
3.3.2 หน้าหลักในการใช้งานแอปพลิเคชัน	37
3.3.3 หน้าแสดงรายการโปรด	41
3.3.4 ส่วนหน้าเว็บสำหรับผู้ดูแลระบบ	42
3.4 แผนภาพยุสเคส (Use Case Diagram)	45
3.5 แผนภาพ Flowchart	56
3.6 การออกแบบส่วนฐานข้อมูล (Database Design)	64
บทที่ 4	68
การพัฒนาระบบ	68
4.1 การพัฒนาแอปพลิเคชัน	68
4.1.1 การเชื่อมต่อ Firestore database	68
4.1.2 การพัฒนาส่วนสมัครสมาชิกและเข้าสู่ระบบ	69

4.1.3 การพัฒนาส่วนแสดงเพลง	75
4.1.4 การพัฒนาส่วนโหลดข้อมูลมาแสดง	76
4.1.5 การพัฒนาส่วนแสดงเพลงที่ฟังบ่อຍ	80
4.1.6 การพัฒนาส่วนรับค่าข้อมูลสมาชิก	82
4.1.7 การพัฒนาส่วนเล่นเพลง	83
4.1.8 การพัฒนาส่วนเพิ่มเพลงรายการโปรด	85
4.1.9 การพัฒนาส่วนตั้งเวลาหยุดเพลง	86
4.2.0 การพัฒนาส่วนแจ้งเตือนเวลาอน	87
4.2.1 การพัฒนาส่วนส่งข้อมูลไปอีกหน้า	89
4.2.2 การพัฒนาส่วนหน้าเก็บเพลงโปรด	90
4.2.3 การพัฒนาส่วนผู้ดูแลระบบ	92
บทที่ 5	95
การทดสอบระบบ	95
5.1 การทดสอบใช้งานแอปพลิเคชัน	95
5.1.1 การทดสอบหน้าจอหลัก	95
5.1.2 การทดสอบหน้ารายการโปรด	95
5.1.3 การทดสอบส่วนเพิ่มเพลงและลบเพลง	96
5.1.4 การทดสอบหน้าเล่นเพลง	96
5.1.5 การทดสอบหน้าสมัครสมาชิกเข้าใช้งานและเข้าสู่ระบบ	96
บทที่ 6	97
สรุปและข้อเสนอแนะ	97
6.1 สรุปความสามารถแอปพลิเคชัน	97
6.1.1 เว็บ	97
6.1.2 แอปพลิเคชัน	97

6.2 ปัญหาและอุปสรรคในการพัฒนา	98
6.3 แนวทางพัฒนาต่อ	98
บรรณานุกรม	99
ภาคผนวก	101
ภาคผนวก ก	102
การติดตั้งเครื่องมือในการพัฒนาแอปพลิเคชัน	102
ก.1 การติดตั้ง Visual Studio Code	102
ก.2 การติดตั้ง Node.js	104
ก.3 การติดตั้ง Ionic Framework	107
ก.4 การติดตั้ง Android Studio	108
ภาคผนวก ข	111
คู่มือการติดตั้งแอปพลิเคชัน	111
ภาคผนวก ค	114
คู่มือการใช้งานระบบ	114
ประวัติผู้เขียน	124

## สารบัญภาพ

ภาพที่ 2.1 หลักการทำงาน Ionic Framework	19
ภาพที่ 2.2 โครงสร้างของระบบแออนดรอยด์	20
ภาพที่ 2.3 ตัวอย่างการทำงาน Node.js ในการพัฒนาเว็บแอปพลิเคชัน	25
ภาพที่ 2.4 หน้าจอแอปพลิเคชัน Relax Melodies	27
ภาพที่ 2.5 หน้าจอแอปพลิเคชัน Sleep Sounds - relaxing sounds	28
ภาพที่ 2.6 หน้าจอแอปพลิเคชัน Sleepo: เสียงผ่อนคลาย, นอนหลับ	29
ภาพที่ 2.7 หน้าจอแอปพลิเคชัน ผ่อนคลายชนบท - เสียงธรรมชาติ	30

ภาพที่ 2.8 หน้าจอแอปพลิเคชัน เสียงธรรมชาติ	31
ภาพที่ 3.1: System architecture แอปพลิเคชันเสียงเพื่อการผ่อนคลาย	34
ภาพที่ 3.2: หน้าจอสมัครสมาชิก	36
ภาพที่ 3.3: หน้าจอเข้าสู่ระบบสมาชิก	37
ภาพที่ 3.4: หน้าแสดงรายการเพลง	38
ภาพที่ 3.5: หน้าเล่นเสียงเพลง	39
ภาพที่ 3.6: หน้าแสดงรายเพลงยอดฮิต	40
ภาพที่ 3.7: หน้าตั้งเวลาเข้านอน	40
ภาพที่ 3.8: หน้าแสดงรายการโปรด	41
ภาพที่ 3.9: หน้าแสดงเพลงที่เพิ่ม	42
ภาพที่ 3.10: ส่วนเพิ่มเพลง	43
ภาพที่ 3.11: ส่วนลบเพลง	44
ภาพที่ 3.12: ยูสเคสของแอปพลิเคชันเสียงเพื่อการผ่อนคลาย	46
ภาพที่ 3.13 : ยูสเคสส่วนเว็บแอปพลิเคชันเพิ่มและลบเพลงของผู้ดูแลระบบ	52
ภาพที่ 3.14 : Flowchart แสดงการทำงานฟังก์ชันสมัครสมาชิกและเข้าสู่ระบบ	58
ภาพที่ 3.15 : Flowchart แสดงการทำงานฟังก์ชันเล่นเพลง	59
ภาพที่ 3.16 : Flowchart แสดงการทำงานฟังก์ชันเพลนที่มีการฟังมากที่สุด	60
ภาพที่ 3.17 : Flowchart ฟังก์ชันตั้งเวลาเข้านอนและแจ้งเตือนเวลาเข้านอน	61
ภาพที่ 3.18 : Flowchart แสดงการทำงานฟังก์ชันเพิ่มเพลงและลบเพลงรายการโปรด	62
ภาพที่ 3.19 : Flowchart แสดงการทำงานฟังก์ชันเพิ่มและลบเพลงส่วนผู้ดูแลระบบ	63
ภาพที่ 3.20 : ส่วนเก็บข้อมูลสมาชิก	64
ภาพที่ 3.21 : ส่วนเก็บจำนวนครั้งเพลงที่ฟัง	65
ภาพที่ 3.22 : ส่วนเก็บเพลงรายการโปรด	65
ภาพที่ 3.23 : ส่วนเก็บข้อมูลสมาชิก	66

ภาพที่ 3.24 : ส่วนเก็บข้อมูลเพลงที่เพิ่ม	66
ภาพที่ 4.1 : ไฟล์ environment.ts	68
ภาพที่ 4.2 : ไฟล์ homes.ts ส่วนเชื่อมบริการฐานข้อมูล	69
ภาพที่ 4.3 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ	70
ภาพที่ 4.4 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)	72
ภาพที่ 4.5 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)	73
ภาพที่ 4.6 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)	74
ภาพที่ 4.7 : ตัวอย่างส่วนเก็บลิสต์เพลง	75
ภาพที่ 4.8 : โครงสร้างส่วนโหลดข้อมูลมาแสดง	76
ภาพที่ 4.9 : โครงสร้างส่วนโหลดข้อมูลมาแสดง (ต่อ)	77
ภาพที่ 4.10 : โครงสร้างส่วนโหลดข้อมูลมาแสดง (ต่อ)	78
ภาพที่ 4.11 : โครงสร้างส่วนเรียกใช้ฟังก์ชัน (ต่อ)	79
ภาพที่ 4.12 : โครงสร้างส่วนแสดงเพลงที่ฟังบ่อຍ	80
ภาพที่ 4.13 : ตัวอย่างโครงสร้างส่วนเก็บค่าจำนวนเพลง (ต่อ)	81
ภาพที่ 4.14 : โครงสร้างส่วนรับค่าข้อมูลสมาชิก	82
ภาพที่ 4.15 : โครงสร้างส่วนเล่นเพลง	83
ภาพที่ 4.16 : โครงสร้างส่วนเล่นเพลง (ต่อ)	84
ภาพที่ 4.17 : ตัวอย่างโครงสร้างส่วนเพิ่มเพลงเป็นรายโปรด	85
ภาพที่ 4.18 : โครงสร้างส่วนตั้งเวลาหยุดเพลง	86
ภาพที่ 4.19 : โครงสร้างส่วนแจ้งเตือนเวลาเข้านอน	87
ภาพที่ 4.20 : โครงสร้างส่วนแจ้งเตือนเวลาเข้านอน (ต่อ)	88
ภาพที่ 4.21 : โครงสร้างส่งข้อมูลไปอีกหน้า	89
ภาพที่ 4.22 : ตัวอย่างโครงสร้างหน้าเก็บเพลงโปรด	90
ภาพที่ 4.23 : ตัวอย่างโครงสร้างหน้าเก็บเพลงโปรด	91

ภาพที่ 4.24 : โครงสร้างหน้าเก็บเพลงโปรด	91
ภาพที่ 4.25 : โครงสร้างส่วนแสดงข้อมูล	92
ภาพที่ 4.26 : โครงสร้างส่วนลับเพลง	92
ภาพที่ 4.27 : โครงสร้างส่วนเพิ่มเพลง	93
ภาพที่ ก.1 : หน้าเว็บดาวน์โหลด Visual Studio Code	102
ภาพที่ ก.2 : ไฟล์โปรแกรม Visual Studio Code	103
ภาพที่ ก.3 : โปรแกรม Visual Studio Code	103
ภาพที่ ก.4 : หน้าเว็บดาวน์โหลด Node.js	104
ภาพที่ ก.5 : ติดตั้ง Node.js	105
ภาพที่ ก.6 : ติดตั้ง Node.js เสร็จสมบูรณ์	106
ภาพที่ ก.7 : คำสั่งติดตั้ง Ionic Framework	107
ภาพที่ ก.8 : ติดตั้ง Ionic Framework เสร็จสมบูรณ์	107
ภาพที่ ก.9 : หน้าเว็บดาวน์โหลด Android Studio	108
ภาพที่ ก.10 : หน้าติดตั้ง Android Studio	109
ภาพที่ ก.11 : ติดตั้ง Android Studio เสร็จสมบูรณ์	110
ภาพที่ ข.1 : หน้าดาวน์โหลดแอปพลิเคชัน Relax Sound	111
ภาพที่ ข.2 : หน้าแสดงการติดตั้ง	112
ภาพที่ ข.3 : หน้าแสดงการติดตั้งสำเร็จ	113
ภาพที่ ค.1 : หน้าจอแสดงเพลงและเพิ่มเพลง	114
ภาพที่ ค.2 : หน้าจอส่วนลับเพลง	115
ภาพที่ ค.3 : หน้าจอต้อนรับ	116
ภาพที่ ค.4 : หน้าสมัครสมาชิก	117
ภาพที่ ค.5 : หน้าเข้าสู่ระบบ	118
ภาพที่ ค.6 : หน้าจอหลัก	119

ภาพที่ ค.7 : หน้าจอส่วนตั้งเวลาเข้าอน	120
ภาพที่ ค.8 : ส่วนแจ้งเตือนเวลาเข้าอน	121
ภาพที่ ค.9 : หน้าจอตั้งเวลาหยุดเพลง	122
ภาพที่ ค.10 : หน้าจอรายการโปรด	123

## สารบัญตาราง

ตารางที่ 1.1 ขั้นตอนการดำเนินการ	17
ตารางที่ 3.1 สัญลักษณ์ Use Case Diagram	45
ตารางที่ 3.2 อธิบาย Use case หน้าที่ระบบแอป ในภาพที่ 3.12	47
ตารางที่ 3.3 Use Case สมัครเข้าใช้งาน	48
ตารางที่ 3.4 Use Case Login เข้าใช้งาน	48
ตารางที่ 3.5 เลือกเสียงที่ต้องการฟัง	49
ตารางที่ 3.6 เดือนเวลาเข้าอน	49
ตารางที่ 3.7 เพิ่มเสียงที่ชอบเป็นรายการโปรด	49
ตารางที่ 3.8 ดูเพลงที่ฟังมากที่สุด	50
ตารางที่ 3.9 ตั้งเวลาปิดเสียง	50
ตารางที่ 3.10 ลบเสียงรายการโปรด	51
ตารางที่ 3.11 เดือนเวลาเข้าอน	51
ตารางที่ 3.12 อธิบาย Use case หน้าที่ระบบเว็บแอป ในภาพที่ 3.13	53
ตารางที่ 3.13 เพิ่มเพลง	53
ตารางที่ 3.14 ลบเพลง	54
ตารางที่ 3.15 แสดงเพลง	54

ตารางที่ 3.16 กรอกชื่อเพลง	54
ตารางที่ 3.17 กรอกชื่อร่องเพลง	55
ตารางที่ 3.18 กรอกบารูป	55
ตารางที่ 3.19 กรอกบลเพลง	55
ตารางที่ 3.20 สัญลักษณ์ของ Flowchart	56
ตารางที่ 3.21 สัญลักษณ์ของ Flowchart (ต่อ)	57

## บทที่ 1

### บทนำ

#### 1.1 ที่มาและเหตุผล

ปัจจุบันผู้คนเผชิญกับโรคไวรัสโคโรนา COVID-19 ทำให้มีความเครียดจากการทำงานออนไลน์หรือเรียนออนไลน์ ซึ่งเสียงเพลงได้รับความนิยมมาอย่างยาวนานโดยมีความเชื่อกันว่าเสียงเพลงจะช่วยให้คนฟังผ่อนคลาย หรือลดความเครียด และยังช่วยให้หอรูมิโนที่เกี่ยวกับความเครียดลดลงมากกว่าใช้ยาลดความเครียด และยังช่วยลดภาวะซึมเศร้า ทั้งยังช่วยปรับให้อารมณ์สงบขึ้นทำให้นอนหลับง่าย เนี่ยกได้ว่าเสียงเพลงเปรียบเสมือนยาที่มีผลกับจิตใจ ช่วยกระตุ้นสมอง ปรับระดับคอร์ติซอลฮอร์โมน (Cortisol) ที่เกี่ยวกับความเครียดให้คนฟังรู้สึกผ่อนคลายได้ เสียงเพลงที่ใช้ในการผ่อนคลายนั้นไม่ได้ถูกจำกัดว่าจะต้องเพลงบรรเลง หรือเสียงธรรมชาติเท่านั้น เพลงทุกประเภทไม่ว่าจะเป็นคลาสสิก แร็ป และอีกหลากหลายแนว สามารถช่วยผ่อนคลายได้ ขึ้นอยู่กับความชอบของผู้ใช้ฟังประเภทไหนแล้วผู้ใช้มีความชอบมากกว่า จึงเป็นที่มาในการพัฒนาโมบายแอปพลิเคชันในบทความนี้นำเสนอโมบายแอปพลิเคชันเสียงเพื่อช่วยลดความเครียดจากการทำงานหรือการเรียน โดยใช้โอโซนิกเฟรมเวิร์ค (Ionic Framework) และฐานข้อมูลไฟล์เบส (Firebase) โดยผู้ใช้สมัครสมาชิกเข้าใช้งาน ซึ่งเมื่อเข้าใช้งานผู้ใช้สามารถเพิ่มและลบเพลงที่เป็นรายการโปรด ตั้งเวลาปิดเสียง ตั้งเวลาเข้านอน ดูเพลงที่ฟังมากที่สุด และเลือกฟังเสียงได้หลากหลาย

#### 1.2 วัตถุประสงค์

##### 1.2.1 เพื่อพัฒนาโมบายแอปพลิเคชันเสียงเพลง

#### 1.3 ขอบเขตของโครงงาน

1. แอปพลิเคชันใช้งานได้บนสมาร์ทโฟนที่ใช้ระบบปฏิบัติการ Android เท่านั้น
2. ผู้ใช้ต้องเข้าสู่ระบบก่อนเข้าใช้งานแอปพลิเคชัน
3. มีเสียงให้เลือกทั้ง 4 ประเภท ได้แก่ 1.ธรรมชาติ 2.ผ่อนคลาย 3.เสียงเพลงและเสียงเครื่องดนตรี และ 4.เสียงจังหวะเพลง

#### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อช่วยเพิ่มสมาธิเมื่อเวลาทำงานหรือเวลาเรียน
2. เพื่อช่วยให้เกิดการผ่อนคลาย
3. เพื่อช่วยให้ความสงบกับผู้ใช้ที่ใช้งานบนสมาร์ทโฟน

## 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 ในการพัฒนาแอปพลิเคชันแอนดรอยด์

#### 3. Firebase เป็นฐานข้อมูล

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

#### 5. Node.js ใช้เป็นชุดคำสั่งในการติดต่อไลบรารี

### 1.5.3 แผนการดำเนินการ

แผนการดำเนินการมี 8 ขั้นตอนดังนี้

ตารางที่ 1.1: ขั้นตอนการดำเนินการ

2564				2565					
ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.
1.ศึกษาความเป็นไปได้	↔								
2.เสนอหัวข้อโครงการ	↔								
3.ศึกษาค้นคว้าข้อมูล	↔	↔							
4.ศึกษาการใช้เฟรมเวิร์ก		↔	↔						
5.วิเคราะห์และออกแบบ		↔	↔						
6.เขียนโปรแกรม			↔			↔			
7.ทดสอบและแก้ปัญหา						↔	↔		
8.จัดทำเอกสาร							↔	↔	

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้องและงานวิจัย

บทนี้จะเป็นรายละเอียดเกี่ยวกับทฤษฎีและแอปพลิเคชันที่เกี่ยวข้องกับการพัฒนาแอปพลิเคชันในครั้งนี้ ประกอบด้วยหัวข้อหลัก หัวข้อมด 5 ข้อดังนี้

2.1 ความรู้พื้นฐาน Ionic Framework

2.2 ความรู้พื้นฐานเกี่ยวกับระบบปฏิบัติการแอนดรอยด์ (Android)

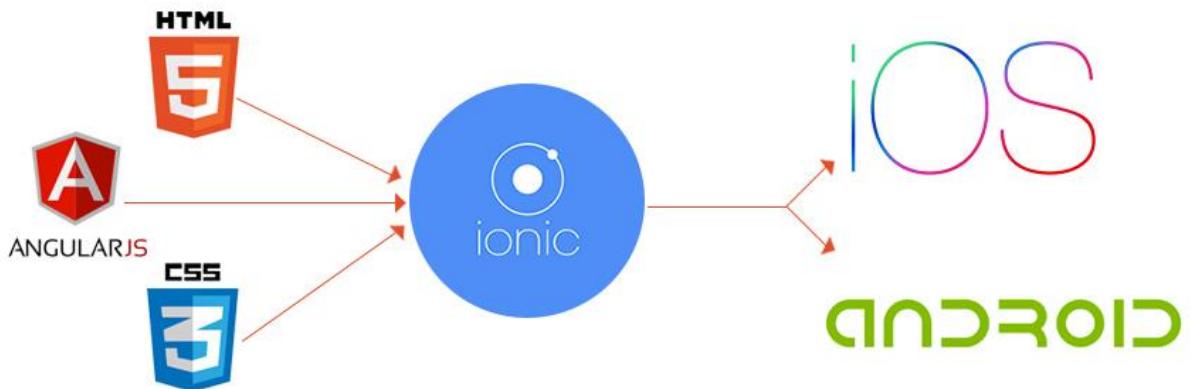
2.3 ความรู้พื้นฐาน Firebase

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>

การเริ่มต้นใช้งาน ionic โดยสามารถใช้คำสั่งในการสร้างโปรเจกต์ [13]

ionic start [ชื่อโปรเจกต์] [เทมเพลต]

เทมเพลตเริ่มต้นอาจมาจากเทมเพลตที่มีชื่อใน Github, Codepen หรือ Local

ไดเรกทอรี เทมเพลตเริ่มต้นคือสิ่งที่คล้ายเป็นไดเรกทอรี www ภายในโปรเจกต์ Cordova

ชื่อเทมเพลต starters

- tabs (Default)
- sidemenu
- blank

## 2.2 ความรู้พื้นฐานเกี่ยวกับระบบปฏิบัติการแอนดรอยด์ (Android)

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

### 2.2.1 สถาปัตยกรรมของระบบแอนดรอยด์

ระบบ Android [3] มีรากฐานมาจากระบบ Linux แต่ละแอปพลิเคชันรวมถึงแอปพลิเคชันที่พัฒนาขึ้นจะทำงานบนกระบวนการของตัวเอง โดยที่แอปพลิเคชันจะอยู่ส่วนขั้นบนสุด (Application) และทำการเรียกใช้งานส่วนสนับสนุน ขั้นล่างๆ ลงไปจนถึงขั้นล่างสุด คือ Linux Kernel ซึ่งเป็นส่วนติดต่อกับอุปกรณ์

ภาษาหลักที่ใช้พัฒนา App คือ ภาษา JAVA หากเคยศึกษาเกี่ยวกับสามารถเข้าใจได้ไม่ยาก โดยเฉพาะหากเคยศึกษาการเขียนโปรแกรมเชิงวัตถุ (OOP) อุปกรณ์ ซึ่งการทำงานของระบบแอนดรอยด์แบ่งการทำงานเป็นชั้น 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 แต่ละชั้นมีความเกี่ยวเนื่องกัน ขั้นบนสุดจะเป็นชั้นแอปพลิเคชันที่ติดกับผู้ใช้งาน จากนั้นจะเป็นชั้นอื่น ๆ ถัดลงมาที่ทำงานในลักษณะต่าง ๆ ออกแบบมาให้สามารถอธิบายแต่ละชั้นได้ดังนี้

### 1. ชั้นแอปพลิเคชัน (Application )

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

### 2. ชั้นแอปพลิเคชันเฟรมเวิร์ค (Application Framework Layer)

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

- View System เป็นส่วนที่ใช้ในการควบคุมการทำงานทางงานสาหรับการสร้างแอปพลิเคชัน 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 ที่ไม่ส่วนของโค้ดโปรแกรม เช่น รูปภาพ, localized strings, layout ซึ่งจะอยู่ในไดร์รีกทอย res/
- Notification Manager เป็นส่วนที่ควบคุมอีเวนต์ (Event) ต่างๆ ที่แสดงบนแถบสถานะ (Status bar) เช่น ในการรีเซ็ตตั้งเวลา
- Activity Manager เป็นส่วนควบคุม Life Cycle ของแอปพลิเคชัน

### 3. ชั้นไลบรารี (Library)

Android ได้รวบรวมกลุ่มของไลบรารีต่างๆ ที่สำคัญและมีความจำเป็นเอาไว้มาก many เพื่ออำนวยความสะดวกให้กับนักพัฒนาและง่ายต่อการพัฒนาโปรแกรม โดยตัวอย่างของไลบรารีที่สำคัญ เช่น

- 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 ในชื่อ Java จำเป็นต้องเรียกใช้แอปพลิเคชันในตัวเองได้ โดยจะต้องเรียกใช้แอปพลิเคชันในชื่อที่สูงกว่าเท่านั้นจึงจะสามารถเรียกใช้ได้ นอกจากนี้ในชื่อ Java ยังแบ่งเป็นชั้นอยู่ที่เรียกว่า Android Runtime ซึ่งจะประกอบไปด้วย 2 ส่วน คือ Dalvik VM (Virtual Machine) และ Core Java Library

- Dalvik VM (Virtual Machine) ส่วนนี้ถูกเขียนด้วยภาษา Java เพื่อใช้เฉพาะการใช้งานในอุปกรณ์เคลื่อนที่ Dalvik VM จะแตกต่างจาก 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. ชั้นลีนักซ์เครื่องเนล (Linux Kernel)

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

ส่วนของ Linux Kernel นี้จะทำหน้าที่เป็น Hardware Abstraction Layer กล่าวคือเป็นตัวกลางระหว่าง Hardware และ Software ที่อยู่ติดชั้นไป และทำหน้าที่บริหารจัดการทรัพยากรต่าง ๆ ของเครื่อง เช่น การจัดการหน่วยความจำ การจัดการprocess ฯลฯ ผู้ผลิตอุปกรณ์สามารถ “พอร์ต” แอนดรอยด์ให้ไปรันบนชาร์ดแวร์แบบต่างๆ ได้โดยเปลี่ยนแปลงในส่วนของ Linux Kernel

ผู้ใช้มือถือหรืออุปกรณ์แอนดรอยด์จะไม่เห็นว่ามีลีนักซ์อยู่ในเครื่อง และแอปพลิเคชันที่เราพัฒนาไว้จะไม่ได้เรียกไปยังลีนักซ์โดยตรง แต่ในฐานะนักพัฒนาเราจำเป็นต้องรับรู้ว่ามีลีนักซ์อยู่ เนื่องจากโปรแกรม utility บางตัวที่ Android SDK เตรียมมาให้จะติดต่อกับลีนักซ์ในอุปกรณ์แอนดรอยด์ เช่น โปรแกรม adb ที่ช่วยให้เราเรียกใช้คำสั่งเพื่อสำรวจระบบไฟล์ของเครื่อง ดูprocessที่รันอยู่ในขณะนั้นและอื่น ๆ ได้

#### 2.3 ความรู้พื้นฐาน Firebase

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

Firebase [11] มีบริการให้ใช้หลายอย่าง สามารถแบ่งเป็นหมวดหมู่ดังนี้

### 2.3.1 Build Better Apps

- Cloud Firestore – จัดเก็บและซิงค์ข้อมูลระหว่างผู้ใช้และอุปกรณ์โดยใช้ฐานข้อมูล NoSQL ที่ทดสอบคลาวด์ Cloud Firestore ให้การเชื่อมต่อในรูปแบบสดและการสนับสนุนของไฟล์พร้อมกับการสืบค้นข้อมูลที่มีประสิทธิภาพ
- Authentication – จัดการผู้ใช้งานด้วยวิธีที่ง่ายและปลอดภัย Firebase Auth มีหลายวิธีในการตรวจสอบสิทธิ์รวมถึงอีเมลและรหัสผ่านผู้ให้บริการบุคคลที่สาม เช่น Google หรือ Facebook และใช้ระบบบัญชีที่มีอยู่โดยตรง สร้างอินเทอร์เฟซหรือใช้ประโยชน์จากโกล็อกอินชอร์ส UI ที่ปรับแต่งได้
- Hosting – ลดความซับซ้อนของเว็บไซต์ด้วยเครื่องมือที่สร้างขึ้นเฉพาะสำหรับเว็บแอปสมัยใหม่ เมื่ออัปโหลดเนื้อหาเว็บ เว็บเหล่านั้นจะส่งไปยัง CDN ทั่วโลกโดยอัตโนมัติและมอบใบอนุญาต SSL ฟรีเพื่อให้ผู้ใช้ได้รับความปลอดภัย
- Realtime Database – Realtime Database คือฐานข้อมูลดังเดิมของ Firebase เป็นโคลัชันที่มีประสิทธิภาพและมีเวลาแฝงต่ำสำหรับแอปบนอุปกรณ์เคลื่อนที่ ที่ต้องการสถานะการซิงค์ระหว่างคลื่อนต์แบบเรียลไทม์

### 2.3.2 Improve app quality

- Crashlytics – ลดเวลาในการแก้ไขปัญหาด้วยการเปลี่ยนข้อขัดข้องให้เป็นรายการปัญหาที่จัดการได้ รับข้อมูลเชิงลึกที่ชัดเจนและนำไปปฏิบัติได้ทันทีที่ต้องจัดการก่อนโดยเห็นผลกระทบของผู้ใช้ในแดชบอร์ด Crashlytics การแจ้งเตือนแบบเรียลไทม์จะช่วยให้มีความเสถียรแม่นยำและเดินทาง Crashlytics เป็นตัวรายงานข้อขัดข้องหลักของ Firebase
- Performance Monitoring – วินิจฉัยปัญหาประสิทธิภาพของแอปที่เกิดขึ้นบนอุปกรณ์ของผู้ใช้ ใช้การติดตามเพื่อตรวจสอบประสิทธิภาพของบางส่วนของแอปและคุณภาพของแอปในคอนโซล Firebase อุปกรณ์เหล่านี้จะเริ่มต้นของแอปและตรวจสอบคำขอ HTTP โดยไม่ต้องเขียนโค้ดใด ๆ
- Test Lab – เรียกใช้การทดสอบอัตโนมัติและกำหนดเองสำหรับแอปบนอุปกรณ์เสมือนและจริงที่ทดสอบโดย Google ใช้ Firebase Test Lab ตลอดจนการพัฒนาเพื่อค้นหาจุดบกพร่องและความไม่สอดคล้อง

### 2.3.3 Grow your business

- Google Analytics – วิเคราะห์คุณลักษณะและพฤติกรรมของผู้ใช้ในแดชบอร์ดเดียว รับข้อมูลเชิงลึกแบบเรียลไทม์จากการรายงาน หรือส่งออกข้อมูลเหตุการณ์ดิบไปยัง Google BigQuery สำหรับการวิเคราะห์ที่กำหนดเอง

- Remote Config – กำหนดวิธีการแสดงผลแอปสำหรับผู้ใช้แต่ละคน เปลี่ยนรูปลักษณ์เปิดตัวฟีเจอร์ทีละน้อยเรียกใช้การทดสอบ A / B ส่งมอบเนื้อหาที่กำหนดเองให้กับผู้ใช้งานรายหรือทำการอัปเดตอื่น ๆ โดยไม่ต้องปรับใช้เวอร์ชันใหม่ทั้งหมดนี้ทำได้จากคอนโซล Firebase ตรวจสอบผลกระทบของการเปลี่ยนแปลงและทำการปรับเปลี่ยนในเวลาไม่กี่นาที
- Cloud Messaging – ส่งข้อความและการแจ้งเตือนไปยังผู้ใช้ข้ามแพลตฟอร์มทั้ง Android, iOS และเว็บได้ฟรี สามารถส่งข้อความไปยังอุปกรณ์เดียวกันคู่มุมอุปกรณ์หรือหัวข้อเฉพาะหรือกลุ่มผู้ใช้ Firebase Cloud Messaging (FCM) ปรับขนาดเป็นแบบที่ใหญ่ที่สุดโดยส่งข้อความหลายแ-snล้านข้อความต่อวัน

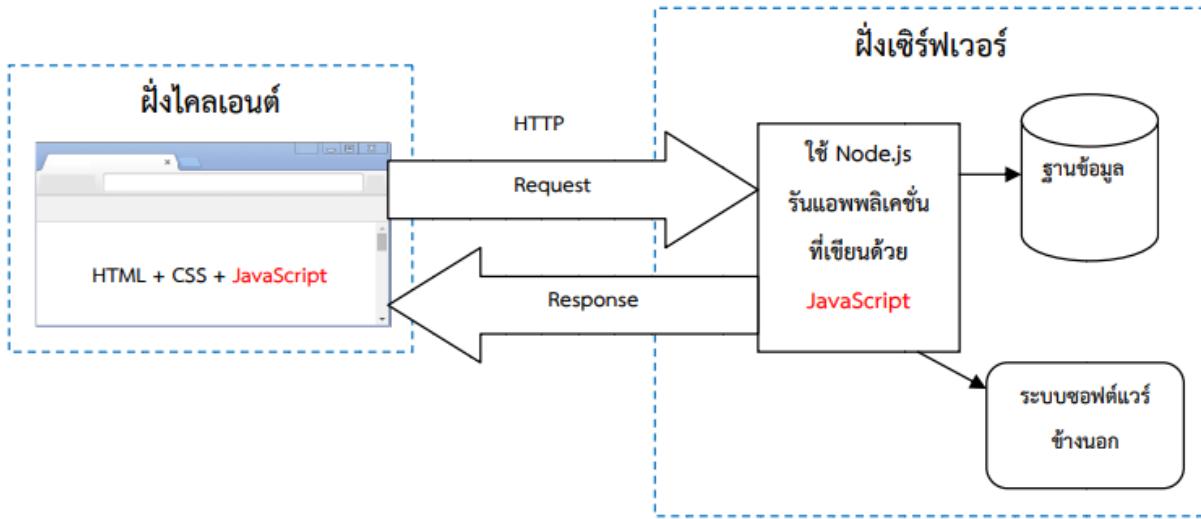
## 2.4 ความรู้พื้นฐาน Node.js

Node.js คือ [5] สภาพแวดล้อมการทำงานของภาษา JavaScript นอกเว็บเบราว์เซอร์ที่ทำงานด้วย V8 engine นั่นหมายความว่า เราสามารถใช้ Node.js ในการพัฒนาแอ�� พลิเคชันแบบ Command line และพลิกเคชัน Desktop หรือแม้แต่เว็บเชิร์ฟเวอร์ได้ 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 ยังมีชุดคำสั่งในการติดตั้งไลบรารีดังนี้

การใช้งาน Node.js [12]

ส่วนใหญ่จะนิยมนำไปพัฒนาเว็บแอปพลิเคชัน โดยให้ทำงานอยู่ฝั่งเชิร์ฟเวอร์ ดังภาพที่ 2.3



ภาพที่ 2.3 ตัวอย่างการทำงาน Node.js ในการพัฒนาเว็บแอปพลิเคชัน

ที่มา: <https://www.patanasongsivilai.com/blog/download-ebook-free-javascript/#>

ดังภาพในผู้ใช้ เว็บเบราว์เซอร์ จะต้องใช้ 3 ภาษาโปรแกรมในการพัฒนาหน้าเว็บไซต์ ซึ่ง 3 ภาษาโปรแกรม ซึ่งเป็นภาษาหลักในผู้ใช้หน้าเว็บ

โดยจะประกอบด้วย

1. ภาษา HTML (ปัจจุบันเป็นเวอร์ชัน HTML5) ใช้สำหรับแสดงหน้าเว็บไซต์
2. ภาษา CSS (ปัจจุบันเป็นเวอร์ชัน CSS3) ใช้สำหรับตกแต่งหน้าเว็บ
3. จาลสคริปต์ (ปัจจุบันเป็นเวอร์ชัน ES6) ใช้ตรวจสอบการแสดงค่าต่าง ๆ และทำให้เว็บมีความยืดหยุ่น

ซึ่งสามารถสรุปลักษณะที่น่าสนใจบางส่วนของ Node.js ได้ดังนี้ [14]

- Node.js สร้างโดย Ryan Dahl ในปี ค.ศ. 2009
- Node.js จะทำงานเป็นอิสระเหมือนกับภาษาคอมพิวเตอร์ทั่วไป โดยไม่ขึ้นกับเว็บเบราว์เซอร์ นั่นคือสามารถสร้างไฟล์ที่มีเฉพาะโค้ดจาลสคริปต์แล้วรันด้วย Node.js ได้เลย โดยไม่จำกัดว่าจะต้องเขียนไว้ในเว็บเพจแล้วเปิดด้วยเว็บเบราว์เซอร์เท่านั้น

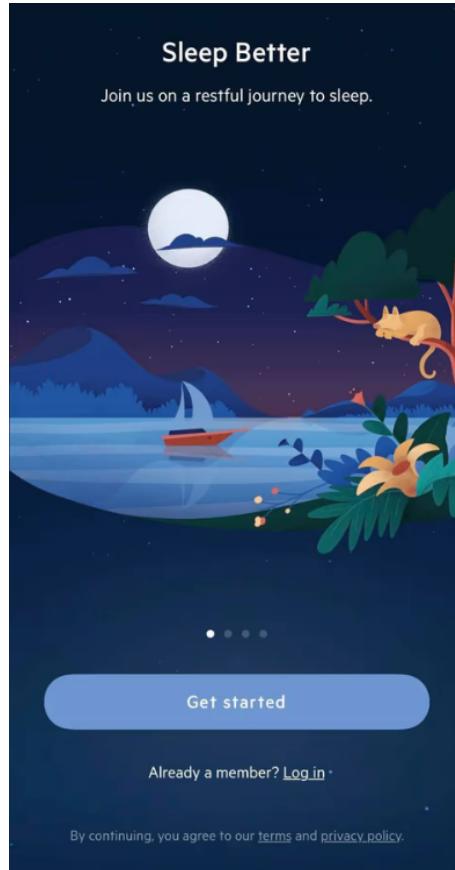
- Node.js ถูกสร้างขึ้นมาเพื่อทำงานทางผู้เซิร์ฟเวอร์ (Server) เป็นหลัก คล้ายกับ PHP, Django (ใช้ภาษา Python) หรืออื่น ๆ แต่การใช้ Node.js มีข้อได้เปรียบคือ สามารถควบคุมการทำงานทั้งทางผู้ใช้เบราว์เซอร์ (หรือ Frontend) และผู้เซิร์ฟเวอร์ (หรือ Backend) โดยใช้ภาษาสคริปต์เดียว
- Node.js มี Community ที่ใหญ่ที่สุดในปัจจุบัน และจำนวนโมดูล (Module) จากนักพัฒนาที่สร้างขึ้นมาสนับสนุนการทำงานของ Node.js มีมากที่สุด เช่น กัน เมื่อเทียบกับภาษาอื่น ๆ

#### 2.4.1 NPM Package management

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

#### 2.6 แอปพลิเคชันที่เกี่ยวข้อง

Relax Melodies [6] เป็นแอปพลิเคชันช่วยให้ผ่อนคลายด้วยเสียงต่าง ๆ ที่มีให้เลือกพังมามาก ทำให้นอนหลับได้เจ้ายิ่งขึ้น เหมาะสมสำหรับคนนอนไม่หลับ นอกจากตัวแอปพลิเคชัน จะมีเสียงต่าง ๆ ที่ช่วยให้หลับง่ายแล้วนั้น ยังสามารถสร้างเสียงขึ้นมาเองได้



ภาพที่ 2.4 หน้าจอแอปพลิเคชัน Relax Melodies

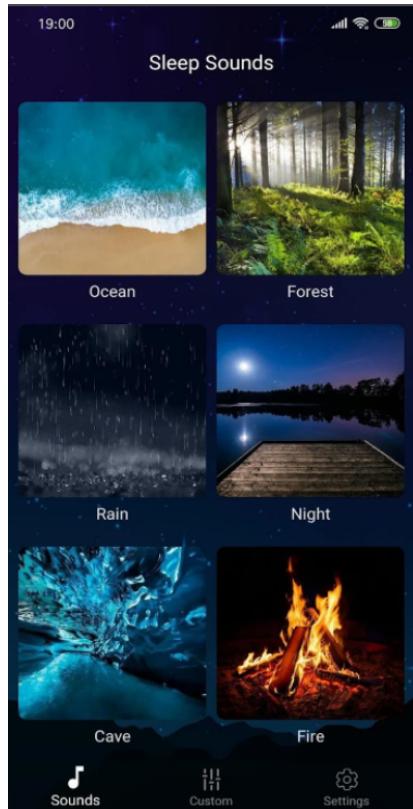
ที่มา:

[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
- แจ้งเตือนเวลาอนของเราได้
- สามารถตั้งเวลาปิดการเล่นเสียงได้
- สามารถตั้งเวลาปลุกได้

Sleep Sounds - relaxing sounds [7] แอปพลิเคชันสามารถช่วยให้ สามารถผ่อนคลายและนอนหลับไปกับเสียงการนอน เสียงธรรมชาติ (Nature sounds) เสียงฝน (Rain sounds) เสียงการทำสมาธิ (Meditation sounds) และเสียงสีขาว (White noise)



ภาพที่ 2.5 หน้าจอแอปพลิเคชัน Sleep Sounds - relaxing sounds

ที่มา:

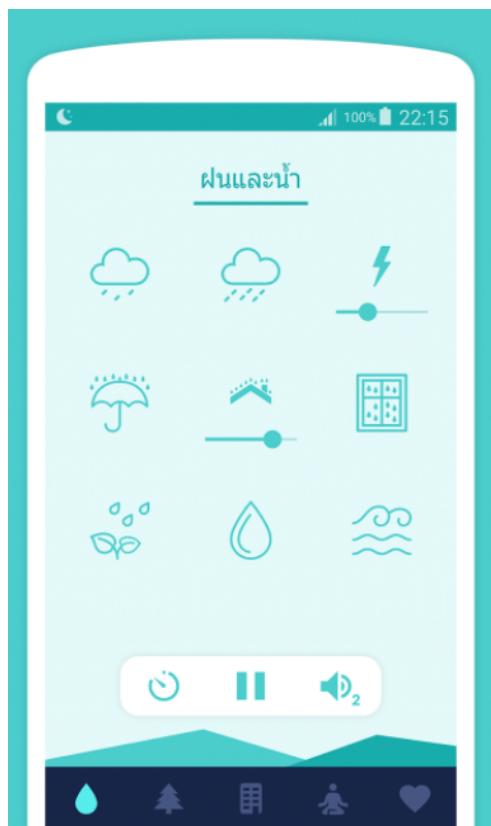
<https://play-lh.googleusercontent.com/4aO1zotY0u5VfVN4FsaAAHOfND91FtpBMfqqisaW2AfFXtk3jLF35iTuwzbPKYGqnS3=w2560-h1440-rw>

คุณสมบัติการทำงานของแอปพลิเคชัน Sleep Sounds - relaxing sounds มีดังนี้

- ผู้ใช้สามารถสร้างมิกซ์เสียงเพื่อผ่อนคลายและนอนหลับได้
- ปรับระดับเสียงสำหรับแต่ละเสียงเมื่อผู้ใช้สร้างมิกซ์เสียง
- เล่นเสียงในพื้นหลัง
- ตั้งเวลาปิดเครื่องเพื่อหยุดเสียงโดยอัตโนมัติ
- เป็นตัวช่วยในการทำสมาธิ
- ไม่ต้องใช้เครือข่าย

- การออกแบบที่สวยงามและเรียบง่าย
- เสียงผ่อนคลายคุณภาพสูง

Sleepo: เสียงผ่อนคลาย, นอนหลับ [8] มีคอลเลกชันของเสียง HD มากมายที่สามารถนำมาผสมผสานให้เป็นบรรยากาศผ่อนคลายอันสมบูรณ์แบบได้ สามารถเลือกฟังเสียงที่แตกต่างกันของฝน เสียงธรรมชาติ (Nature sounds) เสียงเมือง (City sounds) เสียงสีขาว (white noise) หรือเครื่องดนตรี (Musical instruments) บันทึกมิกซ์เสียงที่ชื่นชอบ สามารถตั้งเวลาเครื่องจับเวลาเพื่อหยุดการทำงานของแอปแบบอัตโนมัติ Sleepo และสามารถใช้งานโดยไม่ต้องต่อ กับอินเทอร์เน็ต



ภาพที่ 2.6 หน้าจอแอปพลิเคชัน Sleepo: เสียงผ่อนคลาย, นอนหลับ

ที่มา:

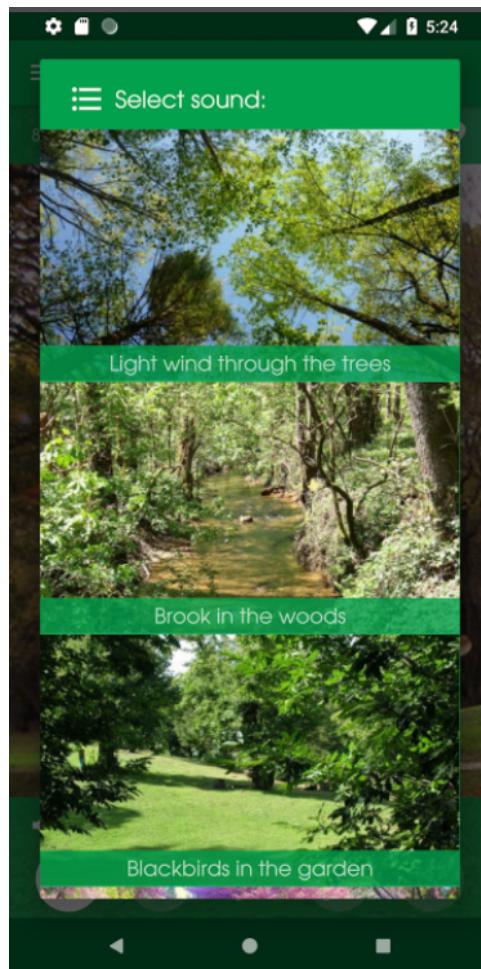
[https://play-lh.googleusercontent.com/qpun2ElwLiu3fqA2qhCbut31erWHn9yuV1NlM3ed4RRFTnPZGd-XTxJhpORwnPl\\_u\\_g=w2560-h1440-rw](https://play-lh.googleusercontent.com/qpun2ElwLiu3fqA2qhCbut31erWHn9yuV1NlM3ed4RRFTnPZGd-XTxJhpORwnPl_u_g=w2560-h1440-rw)

คุณสมบัติการทำงานของแอปพลิเคชัน Sleepo: เสียงผ่อนคลาย, นอนหลับ มีดังนี้

- ผสมผสานเสียง

- เครื่องจำเวลาแบบกำหนดเองได้
- เสียงคุณภาพสูง
- สร้างรายการโปรด
- ทำงานแบบอัตโนมัติ

ผ่อนคลายชนบท - เสียงธรรมชาติ [9] สามารถตั้งเวลาและใช้งานแอปพลิเคชันในพื้นหลังหรือปิดหน้าจอ ที่เหมาะสมสำหรับการผ่อนคลายหลังจากวันที่เครียด ช่วยในการการทำสมาธิและการนอนหลับ



ภาพที่ 2.7 หน้าจอแอปพลิเคชัน ผ่อนคลายชนบท - เสียงธรรมชาติ

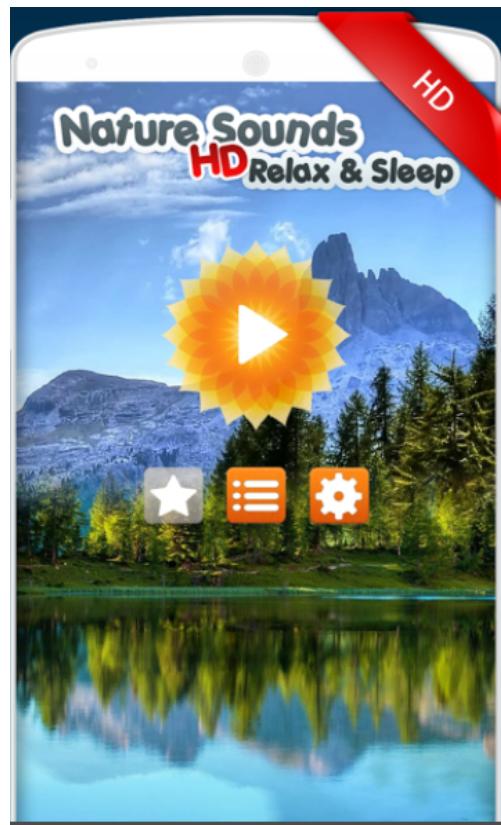
ที่มา:

[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] แอปพลิเคชันสำหรับการพักผ่อนนอนหลับนั้นสามารถให้ความสงบและช่วยปัญหาการนอนไม่หลับ แอปพลิเคชันสามารถทำงานในพื้นหลังหรือเมื่อมีการปิดหน้าจอโดยการเล่นเสียงของธรรมชาติ มีเสียงผ่อนคลายหลากหลายเสียง เช่น เสียงน้ำ (Sound of water) เสียงป่า (Sound of forest) เสียงฝน (Sound of rain) เสียงทะเล (sound of sea) และ เสียงสีขาว (White noise)



ภาพที่ 2.8 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 เสียงธรรมชาติ มีดังนี้

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

แอปพลิเคชัน Relax Sound ที่ผู้พัฒนา พัฒนาขึ้นนั้นมีความแตกต่างจากแอปพลิเคชันที่เกี่ยวข้องคือ สามารถดูเพลงที่มีการฟังมากที่สุดได้

## บทที่ 3

### การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และออกแบบระบบ แบ่งออกเป็น 5 ขั้นตอนเพื่อให้เห็นการดำเนินงานอย่างมีระบบ ในหัวข้อแรกจะนำเสนอภาพรวมของระบบก่อนจะนำเสนอเอกสารแสดงความต้องการของระบบซึ่งจะทำให้เห็นที่มาของหน้าต่าง ๆ ส่วนหัวข้อที่สามจะแสดง แผนภาพการการทำงานของระบบโดยใช้ UML diagram ซึ่งประกอบไปด้วย แผนภาพที่แสดง การทำงานของผู้ใช้ระบบ แผนภาพที่ใช้แสดงคลาส และ การเกิดปฏิสัมพันธ์ระหว่างคอมเพิล์เตอร์ เพื่อแสดงรายละเอียดของระบบก่อนนำไปเขียนคำสั่งด้วยภาษาโปรแกรมในบทต่อไป

3.1 โครงสร้างภาพรวมของระบบ (System Architecture) เป็นการออกแบบภาพรวมและเทคโนโลยีของระบบ

3.2 การวิเคราะห์ความต้องการของระบบ (System Requirements) คือ ความต้องการหรือสิ่งที่ระบบควรจะทำ หรือหน้าที่หลักของระบบที่จะต้องทำ

3.3 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface Design)

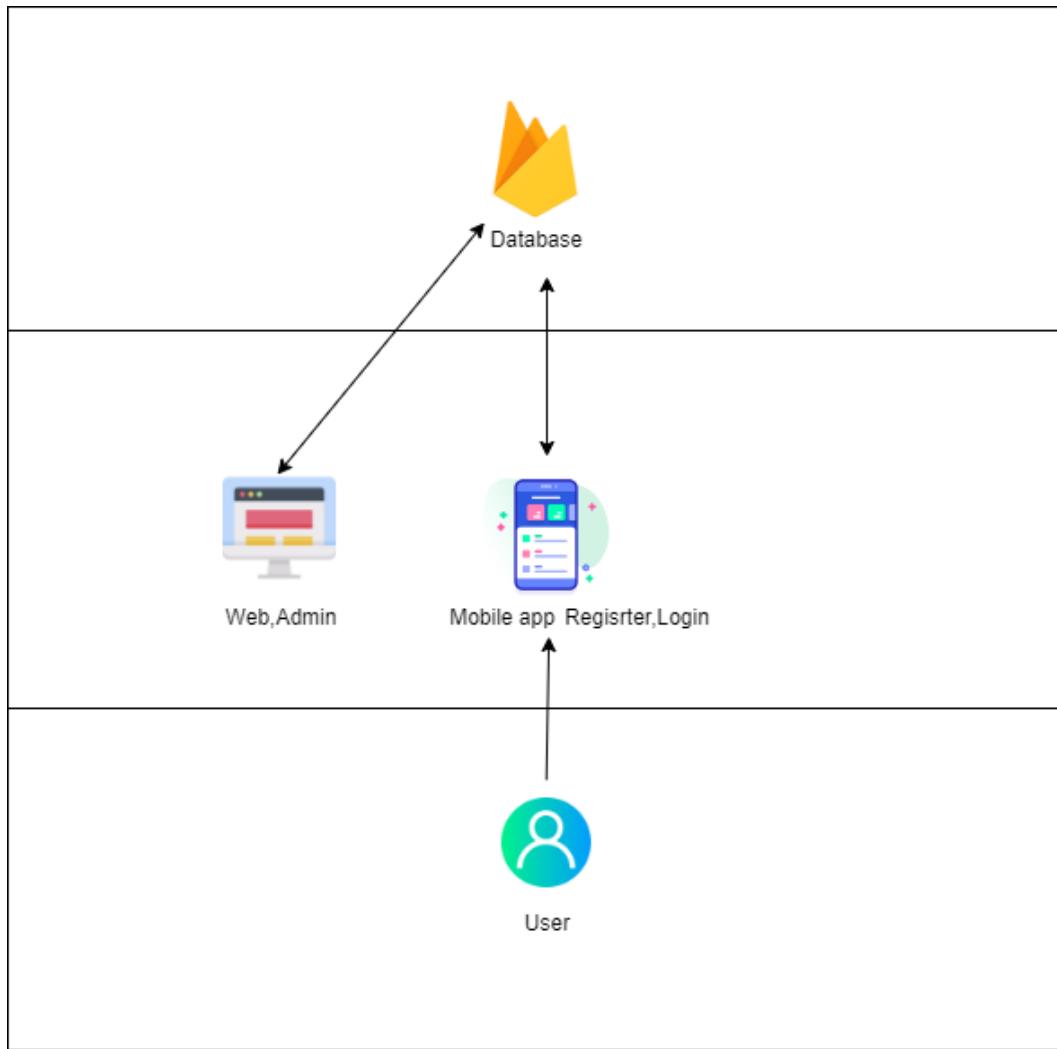
3.4 แผนภาพยุสเคส (Use Case Diagram) เป็นแผนภาพที่ใช้แสดงให้ทราบว่าระบบทำงานหรือมีหน้าที่ใดบ้าง

3.5 แผนภาพ Flowchart เป็นแผนภาพที่ใช้แสดงการทำงานของระบบ

3.6 การออกแบบฐานข้อมูล (Database Design)

### 3.1 โครงสร้างภาพรวมของระบบ (System Architecture)

การออกแบบ System architecture แสดงภาพรวมและเทคโนโลยีของแอปพลิเคชันสี่ยงเพื่อการผ่อนคลาย มีรายละเอียดดังภาพที่ 3.1



ภาพที่ 3.1: System architecture แอปพลิเคชันสี่ยงเพื่อการผ่อนคลาย

จากภาพที่ 3.1 สามารถอธิบายโครงสร้างระบบดังนี้

1. Firebase ใช้เป็นฐานข้อมูลในการเก็บข้อมูลผู้ใช้โดยผู้ใช้สมัครสมาชิกผ่าน Mobile App
2. Mobile App สำหรับผู้ใช้ที่สมัครสมาชิกเข้าใช้งานแอปข้อมูลต่าง ๆ ของผู้ใช้จะเก็บบน Firebase
3. User สำหรับผู้ใช้ที่ยังไม่เป็นสมาชิกต้องสมัครเข้าใช้งานผ่าน Mobile App

4. Web สำหรับผู้ดูแลระบบเพิ่มเพลงไว้ที่ฐานข้อมูลและแสดงเพลงที่ผู้ดูแลระบบเพิ่มบนโมบายแอปพลิเคชัน

### 3.2 การวิเคราะห์ความต้องการของระบบ (System Requirements)

#### 3.2.1 Functional Requirements

##### 1. ผู้ดูแลระบบ

- สามารถเพิ่มเพลงได้
- สามารถลบเพลงได้

##### 2. สมาชิก

- สามารถสมัครสมาชิกและเข้าสู่ระบบได้
- สามารถตั้งเวลาเข้านอนได้
- เตือนเวลาเข้านอน
- สามารถเพิ่มเพลงเป็นรายการโปรดได้
- สามารถลบเพลงที่เป็นรายการโปรดได้
- สามารถดูเพลงที่พึ่งมากที่สุดได้
- สามารถตั้งเวลาหยุดเพลงได้

#### 3.2.2 Non-functional Requirements

- พื้นแบบอฟฟ์ไลน์ได้เมื่อเข้าสู่ระบบ

### 3.3 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface Design)

การออกแบบหน้าจอผู้ใช้งานแอปพลิเคชัน แบ่งเป็น 4 ส่วนดังนี้

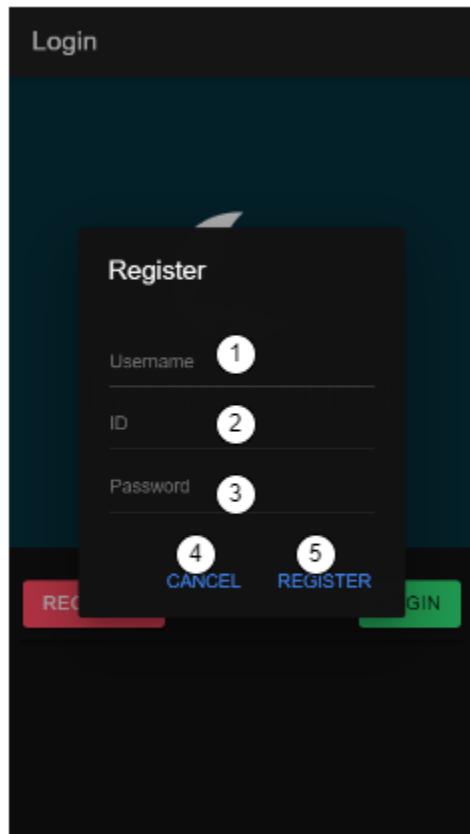
- หน้าสมัครสมาชิกและหน้าเข้าสู่ระบบ
- หน้าหลักในการใช้งานแอปพลิเคชัน
- หน้าแสดงรายการโปรด

- ส่วนหน้าเว็บสำหรับผู้ดูแลระบบ

### 3.3.1 หน้าสมัครสมาชิกและหน้าเข้าสู่ระบบ

หน้าสมัครสมาชิกและเข้าสู่ระบบมีดังนี้

#### 1. หน้าจอสมัครสมาชิก มีดังนี้

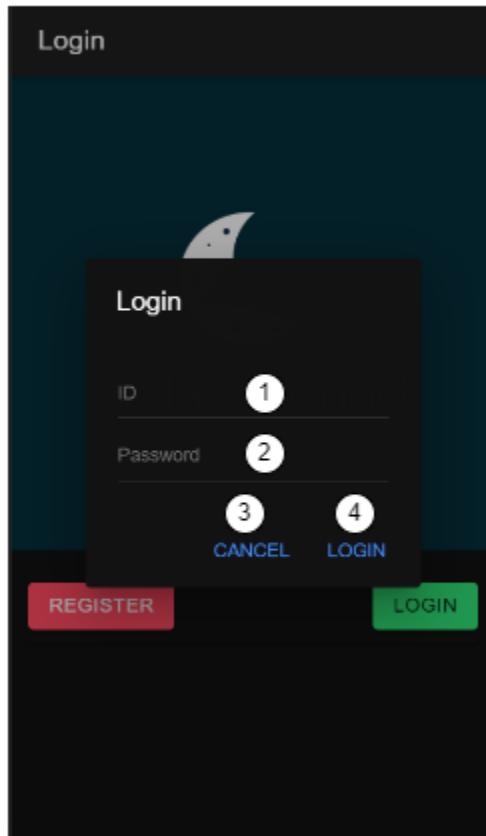


ภาพที่ 3.2: หน้าจอสมัครสมาชิก

จากภาพที่ 3.2 หน้าจอสมัครสมาชิก มี 5 ส่วนดังนี้

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

2. หน้าจอเข้าสู่ระบบสมาชิก มีดังนี้



ภาพที่ 3.3: หน้าจอเข้าสู่ระบบสมาชิก

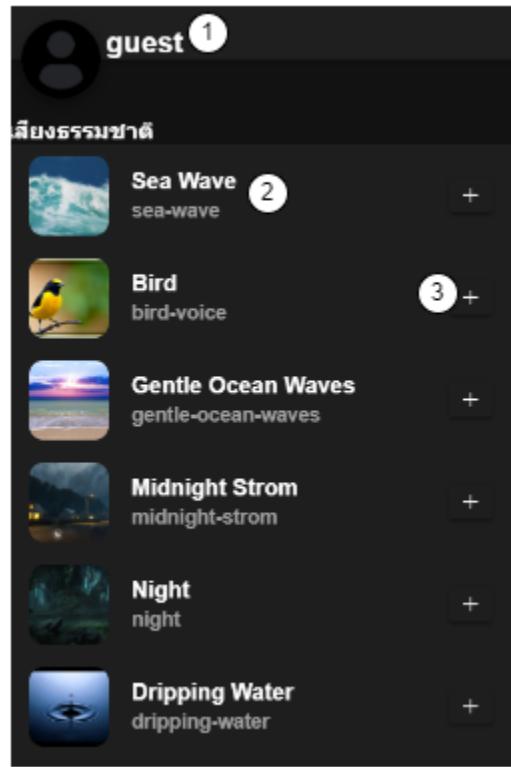
จากภาพที่ 3.3 หน้าจอเข้าสู่ระบบสมาชิก มี 4 ส่วนดังนี้

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

### 3.3.2 หน้าหลักในการใช้งานแอปพลิเคชัน

หน้าหลักที่ให้สมาชิกใช้งานแอปพลิเคชัน มีดังนี้

### 1. หน้าแสดงรายการเพลง

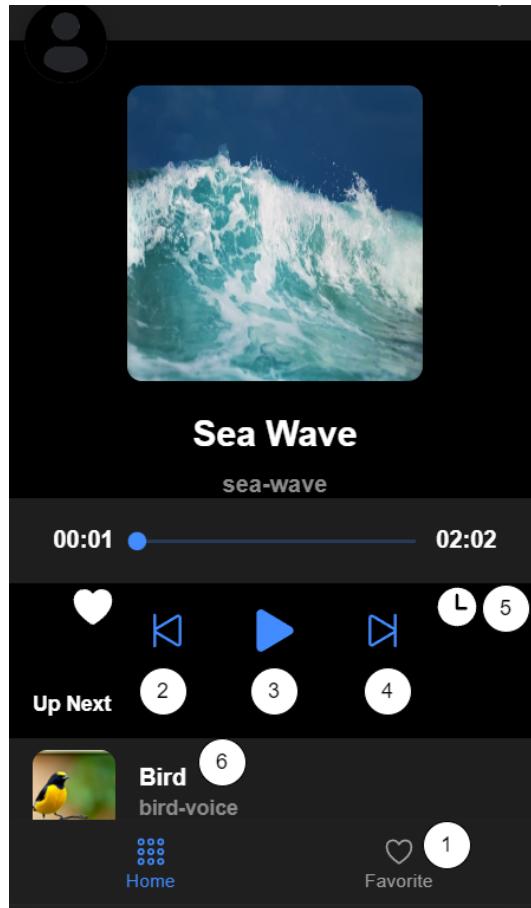


ภาพที่ 3.4: หน้าแสดงรายการเพลง

จากภาพที่ 3.4 หน้าแสดงรายการเพลง มี 3 ส่วนดังนี้

- ส่วนที่ 1 ชื่อผู้ใช้
- ส่วนที่ 2 รายการเพลง
- ส่วนที่ 3 ปุ่มเพิ่มเพลงเป็นรายการโปรด

## 2. หน้าเล่นเสียงเพลง

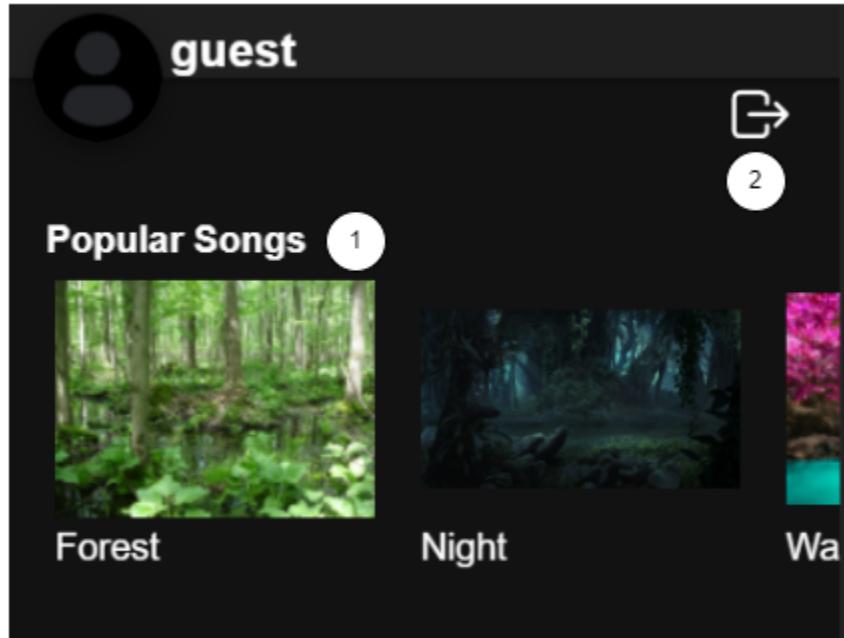


ภาพที่ 3.5: หน้าเล่นเสียงเพลง

จากภาพที่ 3.5 หน้าแสดงรายการเพลง มี 6 ส่วนดังนี้

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

### 3. หน้าแสดงรายเพลงยอดฮิต

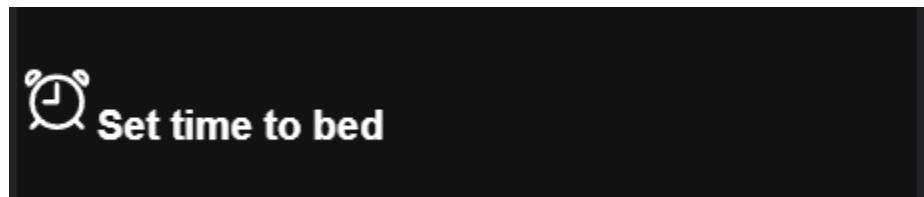


ภาพที่ 3.6: หน้าแสดงรายเพลงยอดฮิต

จากภาพที่ 3.6 หน้าแสดงรายการเพลง มี 2 ส่วนดังนี้

- ส่วนที่ 1 ส่วนแสดงเพลงที่มีการฟังมากที่สุด
- ส่วนที่ 2 ปุ่มออกจากระบบ

### 4. หน้าตั้งเวลาเข้านอน



ภาพที่ 3.7: หน้าตั้งเวลาเข้านอน

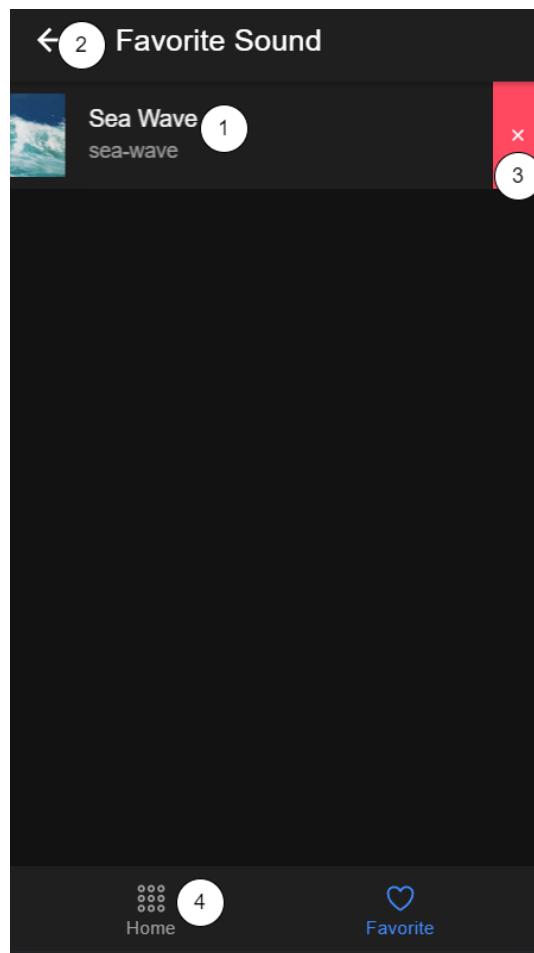
จากภาพที่ 3.7 หน้าแสดงรายการเพลง มีดังนี้

- ตั้งเวลาเข้านอน

### 3.3.3 หน้าแสดงรายการโปรด

หน้าแสดงเพลงรายโปรด มีดังนี้

#### 1. หน้ารายการโปรด



ภาพที่ 3.8: หน้าแสดงรายการโปรด

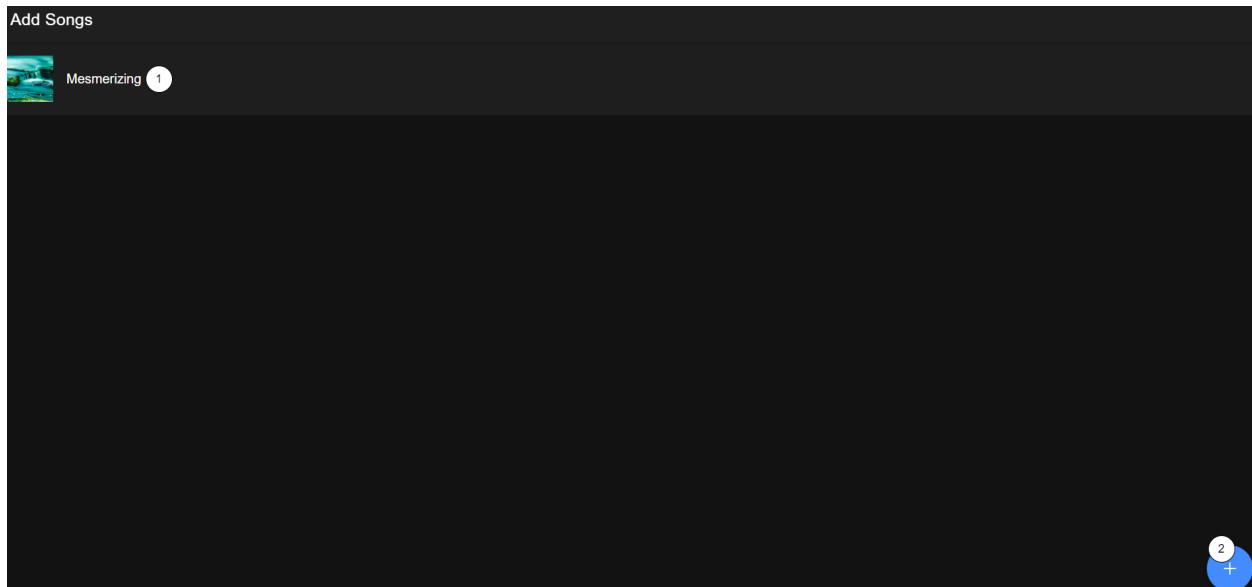
จากภาพที่ 3.8 หน้าแสดงรายการโปรด มีดังนี้

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

### 3.3.4 ส่วนหน้าเว็บสำหรับผู้ดูและระบบ

ส่วนหน้าเว็บผู้ดูและระบบ มีดังนี้

#### 1. หน้าแสดงเพลงที่เพิ่ม

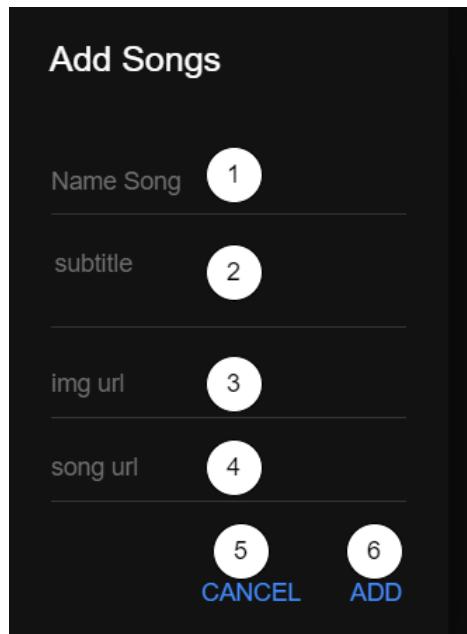


ภาพที่ 3.9: หน้าแสดงเพลงที่เพิ่ม

จากภาพที่ 3.9 หน้าแสดงรายการเพลงที่เพิ่ม มีดังนี้

- ส่วนที่ 1 รายเพลงที่เพิ่ม
- ส่วนที่ 2 ปุ่มเพิ่มเพลง

## 2. ส่วนเพิ่มเพลง

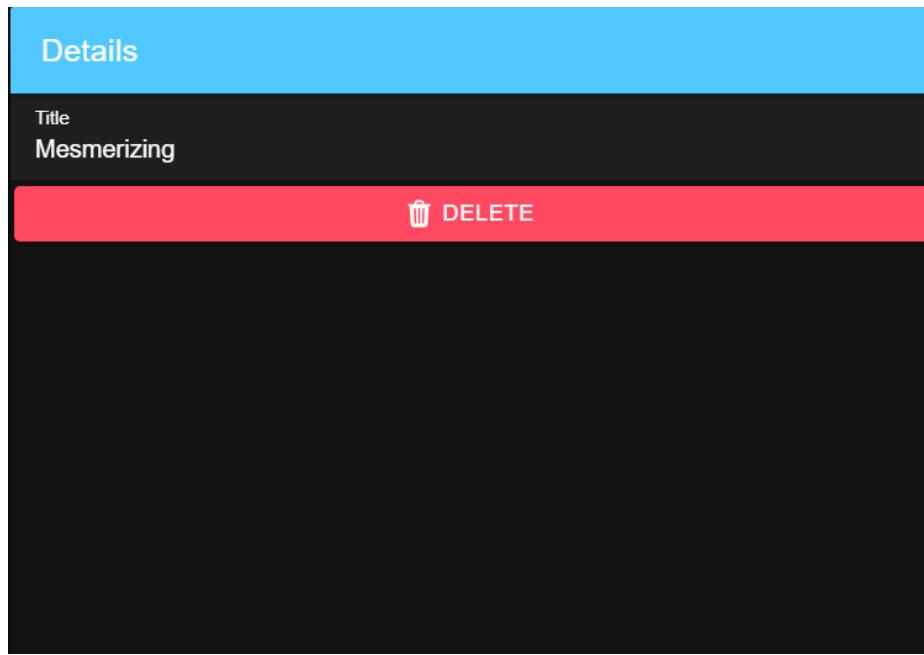


ภาพที่ 3.10: ส่วนเพิ่มเพลง

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

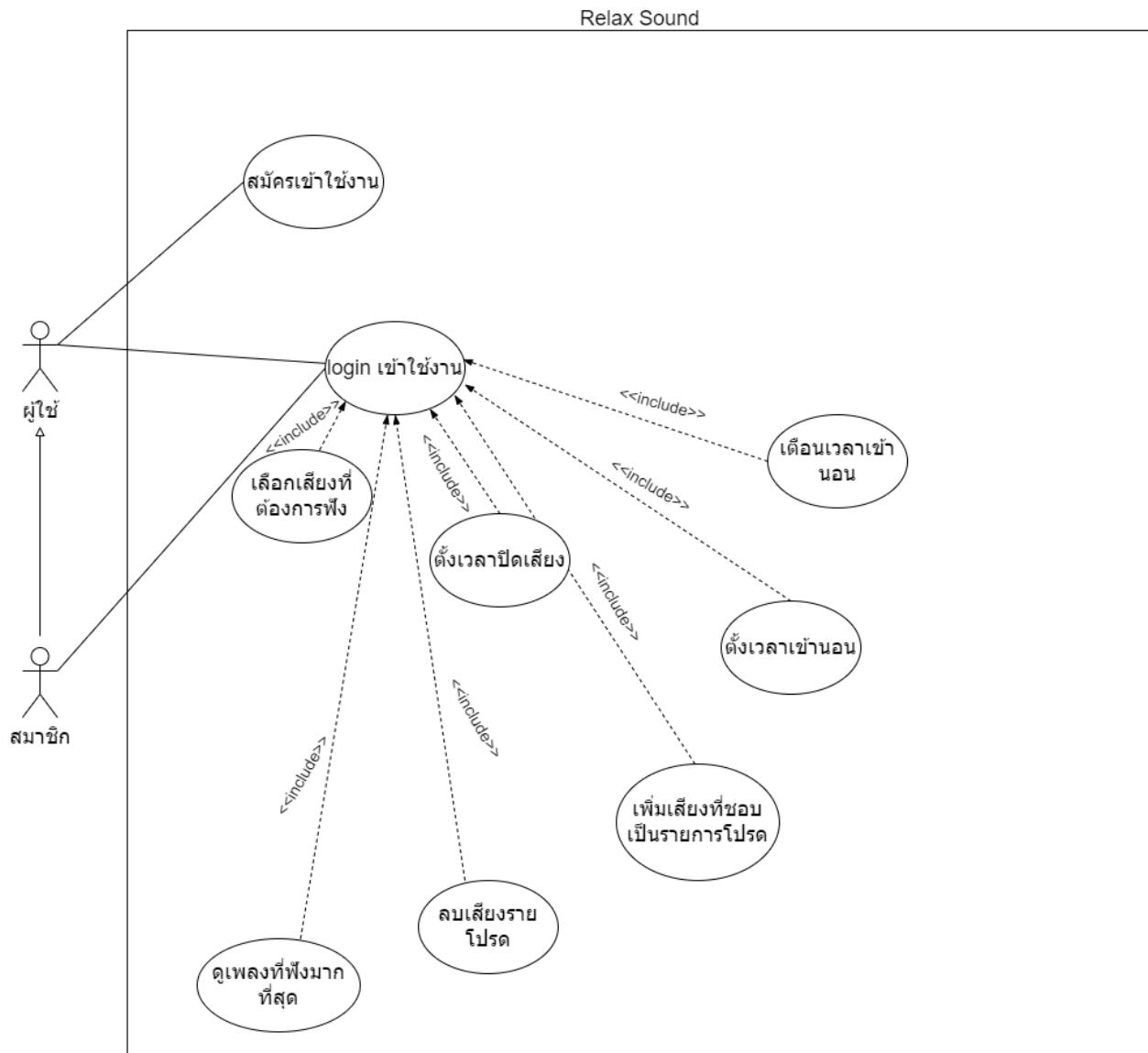
- ส่วนที่ 1 แสดงชื่อเพลงที่ลับ
- ส่วนที่ 2 ปุ่มลับเพลง

### 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
 Actor	ผู้ที่กระทำการกับระบบ อาจเป็นผู้ที่ทำการส่งข้อมูล, รับข้อมูล หรือແລກเบլี่ยนข้อมูลกับระบบนั้นๆ
— — — — —	เส้นเชื่อมระหว่าง Actor กับ Use case
<b>System Name</b>	เส้นแบ่งขอบเขตระหว่างระบบกับผู้กระทำการต่อระบบ
— · · · <<include>> — — →	กิจกรรมใน Use Case หนึ่ง อาจจะถูกผนวกเข้าไปรวมกับกิจกรรมของอีก Use Caseหนึ่ง
— · · · <<extend>> — — →	Use Case หนึ่งทำหน้าที่ตามปกติ เมื่อเกิดเหตุการณ์ผิดปกติขึ้นจะต้องทำหน้าที่พิเศษเพิ่ม
— — — — →	สืบทอดคุณสมบัติจาก Superclass ไปยัง Subclass



ภาพที่ 3.12: ยูสเคสของแอปพลิเคชันเสียงเพื่อการผ่อนคลาย

ตารางที่ 3.2 อธิบาย Use case หน้าที่ระบบแอป ในภาพที่ 3.12

Use Case	คำอธิบาย
สมัครเข้าใช้งาน	สำหรับผู้ใช้จะเข้าใช้งานแอปพลิเคชันต้องสมัครเข้าใช้งาน
Login เข้าใช้งาน	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน
เลือกเสียงที่ต้องการฟัง	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถเลือกเสียงที่ต้องการฟัง
ตั้งเวลาอ่าน	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถตั้งเวลาอ่าน สำหรับการเตือนเวลาเข้าอน
เพิ่มเสียงที่ชอบเป็นรายการโปรด	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถเพิ่มเสียงที่ชอบเป็นรายการโปรด
ดูเพลงที่ฟังมากที่สุด	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถดูเพลงที่มีการฟังมากที่สุด
ตั้งเวลาปิดเสียง	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถตั้งเวลาปิดเสียงได้
ลบเสียงรายการโปรด	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน สามารถลบเสียงที่เป็นรายการโปรดได้
เตือนเวลาเข้าอน	สำหรับสมาชิกเข้าใช้งานแอปพลิเคชัน เมื่อสมาชิกตั้งเวลาเข้าอนระบบจะแจ้งเตือนเวลาเข้าอนเมื่อถึงเวลาที่สมาชิกตั้งไว้

ตารางที่ 3.3 Use Case สมัครเข้าใช้งาน

Use Case Title: สมัครเข้าใช้งาน	Use Case Id : 1
Primary Actor : ผู้ใช้	
Preconditions : เข้าสู่ระบบ	
Main Flow : ส่วนสมัครเข้าใช้งาน เมื่อผู้ใช้เข้าใช้งานเมื่อใช้งานแอปพลิเคชันครั้งแรก ผู้ใช้ต้องสมัครเป็นสมาชิก เพื่อเข้าสู่ระบบ	

ตารางที่ 3.4 Use Case Login เข้าใช้งาน

Use Case Title: Login เข้าใช้งาน	Use Case Id : 2
Primary Actor : สมาชิก	
Stakeholder Actor : ผู้ใช้	
Preconditions : เข้าสู่ระบบ	
Main Flow : ส่วนLogin เข้าใช้งาน มีกี่ลุ่มบุคคลเกี่ยวข้อง 2 กลุ่ม ได้แก่ ผู้ใช้ และสมาชิก โดยเมื่อผู้ใช้สมัครสมาชิกสำเร็จ จะสามารถทำการลงทะเบียนเข้าสู่ระบบเข้าใช้งานได้	

ตารางที่ 3.5 เลือกเสียงที่ต้องการฟัง

Use Case Title: เลือกเสียงที่ต้องการฟัง	Use Case Id : 3
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : ส่วนเลือกเสียงที่ต้องการฟัง เมื่อสมาชิกLogin เข้าใช้งาน สมาชิกเลือกฟังเสียงที่ต้องการฟัง และปิดเครื่องจะแสดงหน้า เล่นเพลงให้สมาชิก	

ตารางที่ 3.6 ตั้งเวลาเข้านอน

Use Case Title: ตั้งเวลาเข้านอน	Use Case Id : 4
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : ส่วนตั้งเวลาเข้านอน เมื่อสมาชิกLogin เข้าใช้งาน สมาชิกเลือกตั้งเวลานอน	

ตารางที่ 3.7 เพิ่มเสียงที่ชอบเป็นรายการโปรด

Use Case Title: เพิ่มเสียงที่ชอบเป็นรายการโปรด	Use Case Id : 5
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
Main Flow : ส่วนเพิ่มเสียงที่ชอบเป็นรายการโปรด เมื่อสมาชิก Login เข้าใช้งานสมาชิกเพิ่มเสียงที่ชอบเป็นรายการโปรด ระบบจะ เพิ่มเพลงที่สมาชิกชอบไปที่หน้ารายการโปรด	

### ตารางที่ 3.8 ดูเพลงที่ฟังมากที่สุด

Use Case Title: ดูเพลงที่ฟังมากที่สุด	Use Case Id : 6
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
<p>Main Flow : ส่วนดูเพลงที่ฟังมากที่สุด เมื่อสมาชิก Login เข้าใช้งาน เมื่อสมาชิกขอบฟังเพลงใหม่บ่อย ๆ ระบบจะเก็บจำนวนครั้งเพลง เพลงที่มีการฟังปอยจะแสดงให้สมาชิกเห็นเป็นเพลงแรก</p>	

### ตารางที่ 3.9 ตั้งเวลาปิดเสียง

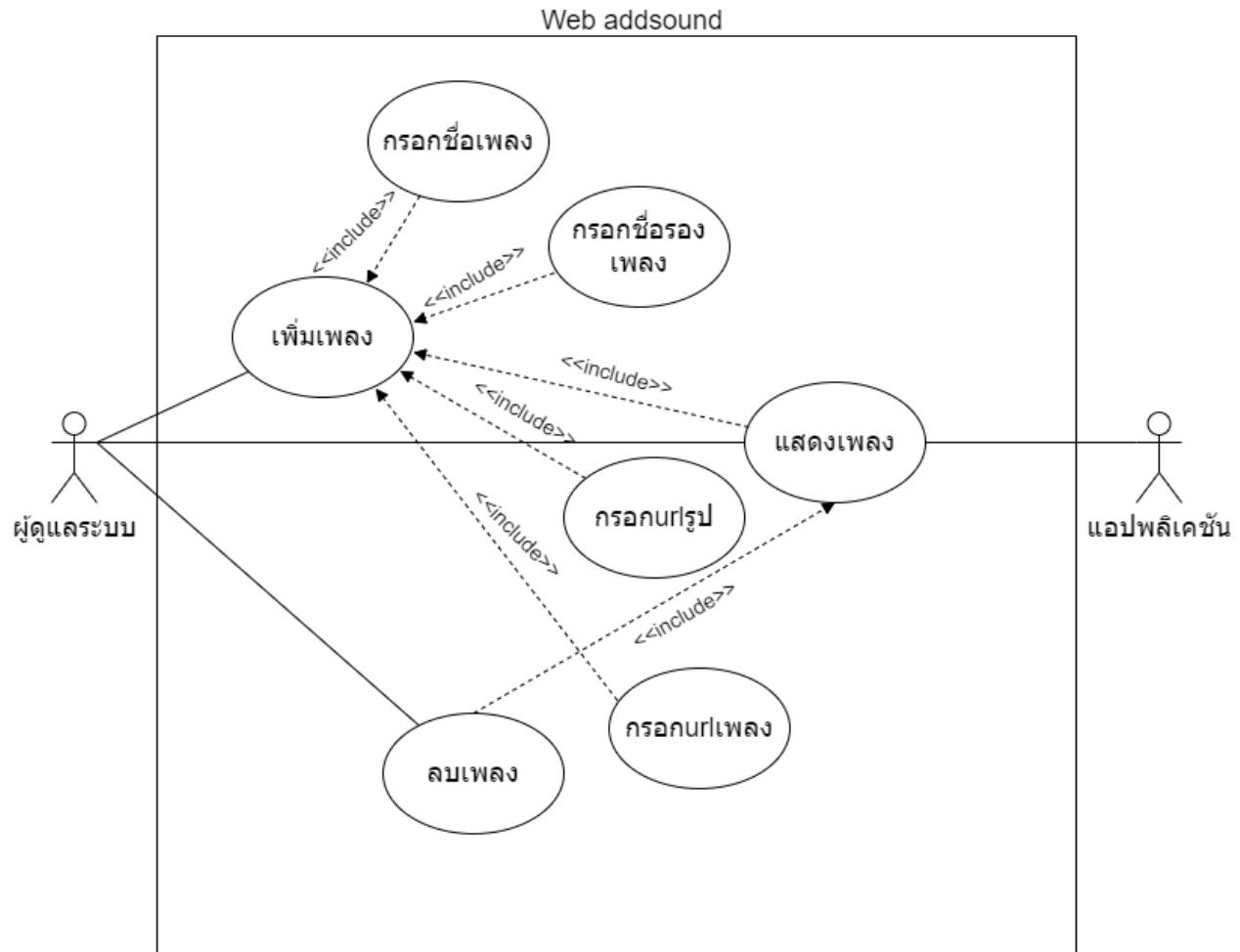
Use Case Title: ตั้งเวลาปิดเสียง	Use Case Id : 7
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
<p>Main Flow : ส่วนตั้งเวลาปิดเสียง เมื่อสมาชิก Login เข้าใช้งาน เมื่อสมาชิกกดลุ่นเพลง และทำการตั้งเวลาหยุดเพลง เมื่อหมดเวลาตามที่สมาชิกตั้งไว้เพลงจะหยุด</p>	

ตารางที่ 3.10 ลบเสียงรายการโปรด

Use Case Title: ลบเสียงรายการโปรด	Use Case Id : 8
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
<p>Main Flow : ส่วนลบเสียงรายการโปรด เมื่อสมาชิกLogin เข้าใช้งาน เมื่อสมาชิกต้องการลบเพลงรายโปรด สมาชิกต้องไปที่หน้ารายโปรดแล้วทำการเลือกเพลงที่ต้องการลบ</p>	

ตารางที่ 3.11 เตือนเวลาเข้านอน

Use Case Title: เตือนเวลาเข้านอน	Use Case Id : 9
Primary Actor : สมาชิก	
Preconditions : เข้าสู่ระบบ	
<p>Main Flow : ส่วนเตือนเวลาเข้านอน เมื่อสมาชิกLogin เข้าใช้งาน เมื่อถึงเวลาที่สมาชิกตั้งเวลาเข้านอนไว้ แอปพลิเคชันจะทำการแจ้งเตือนเข้านอน</p>	



ภาพที่ 3.13: ยูสเคสส่วนเรื๊บแอปพลิเคชันเพิ่มและลบเพลงของผู้ดูแลระบบ

ตารางที่ 3.12 อธิบาย Use case หน้าที่ระบบเว็บแอป ในภาพที่ 3.13

Use Case	คำอธิบาย
เพิ่มเพลง	สำหรับผู้ดูและระบบสามารถเพิ่มเพลงได้โดยสามารถกรอกชื่อเพลง ชื่อร้องเพลง url รูป url เพลง
ลบเพลง	สำหรับผู้ดูและระบบสามารถลบเพลงได้ โดยกดเพลงที่ต้องการลบและทำการลบเพลง
แสดงเพลง	เมื่อผู้ดูและระบบเพิ่มเพลง เพลงที่เพิ่มที่จะแสดงที่หน้าเว็บ และแสดงบนแอปพลิเคชัน
กรอกชื่อเพลง	สำหรับผู้ดูและระบบสามารถกรอกชื่อเพลงที่ต้องการเพิ่ม
กรอกชื่อร้องเพลง	สำหรับผู้ดูและระบบสามารถกรอกชื่อร้องเพลงที่ต้องการเพิ่ม
กรอกurlรูป	สำหรับผู้ดูและระบบสามารถกรอกurlรูปที่ต้องการเพิ่ม
กรอกurlเพลง	สำหรับผู้ดูและระบบสามารถกรอกurlเพลงที่ต้องการเพิ่ม

ตารางที่ 3.13 เพิ่มเพลง

Use Case Title: เพิ่มเพลง	Use Case Id : 1
Primary Actor : ผู้ดูและระบบ	
Main Flow : ส่วนเพิ่มเพลง ผู้ดูและระบบจะเพิ่มเพลงโดยการกรอก ชื่อเพลง ได้แก่ ชื่อร้องเพลง url รูป และurl เพลง เมื่อกรอกชื่อเพลง ครบถ้วนแล้วเพิ่ม เพลงที่เพิ่มจะแสดงบนหน้าเว็บของผู้ดูและ ระบบ	

ตารางที่ 3.14 ลับเพลง

Use Case Title: ลับเพลง	Use Case Id : 2
Primary Actor : ผู้ดูแลระบบ	
<p>Main Flow : ส่วนลับเพลง ผู้ดูแลระบบจะลบเพลงโดยเมื่อเพลง นั้นถูกเพิ่มขึ้นมาและ แสดงอยู่บนเว็บผู้ดูแลระบบ โดยจะลบ เพลงนั้นด้วยการเลือกเพลงที่ต้องการลบ และทำการลบเพลง เมื่อ เพลงนั้นโดนลบ เพลงนั้นจะไม่แสดงบนเว็บและแอปพลิเคชัน</p>	

ตารางที่ 3.15 แสดงเพลง

Use Case Title: แสดงเพลง	Use Case Id : 3
Primary Actor : ผู้ดูแลระบบ	
Stakeholder Actor : แอปพลิเคชัน	
<p>Main Flow : ส่วนแสดงเพลงบนเว็บแอปพลิเคชันและแสดงบนแอป พลิเคชันมีบุคคลที่เกี่ยวข้อง 2 กลุ่มได้แก่ ผู้ดูแลระบบ และแอป พลิเคชัน โดยเพลงจะแสดงก็ต่อเมื่อผู้ดูแลระบบ ทำการเพิ่มเพลง เพลงที่เพิ่มก็จะแสดงบนเว็บผู้ดูแลระบบ และแสดงบนแอปพลิเคชัน</p>	

ตารางที่ 3.16 กรอกชื่อเพลง

Use Case Title: กรอกชื่อเพลง	Use Case Id : 4
Primary Actor : ผู้ดูแลระบบ	
<p>Main Flow : ส่วนกรอกชื่อเพลง โดยผู้ดูแลระบบจะทำการหา เพลง เมื่อหาเพลงได้ จะทำการกรอกชื่อเพลงในช่องกรอกที่แสดง เป็น dialog ขึ้นมา</p>	

ตารางที่ 3.17 กรอกชื่อรองเพลง

Use Case Title: กรอกชื่อเพลง	Use Case Id : 5
Primary Actor : ผู้ดูแลระบบ	
<p>Main Flow : ส่วนกรอกชื่อรองเพลง โดยผู้ดูแลระบบจะทำการหาเพลง เมื่อหาเพลงได้ จะทำการกรอกชื่อรองเพลงเป็นตัวพิมพ์เล็ก ในช่องกรอกที่แสดงเป็น dialog ขึ้นมา</p>	

ตารางที่ 3.18 กรอกบารูป

Use Case Title: กรอกบารูป	Use Case Id : 6
Primary Actor : ผู้ดูแลระบบ	
<p>Main Flow : ส่วนกรอกบารูป โดยผู้ดูแลระบบจะทำการหารูป เมื่อหารูปได้ จะทำการคัดลอกที่อยู่รูปแล้วนำมารอกบารูปในช่องกรอกที่แสดงเป็น dialog ขึ้นมา</p>	

ตารางที่ 3.19 กรอกบารีเพลง

Use Case Title: กรอกบารีเพลง	Use Case Id : 7
Primary Actor : ผู้ดูแลระบบ	
<p>Main Flow : ส่วนกรอกบารีเพลง โดยผู้ดูแลระบบจะทำการหาเพลง เมื่อหาเพลงได้ จะทำการคัดลอกที่อยู่เพลงแล้วนำมารอกบารีเพลงในช่องกรอกที่แสดงเป็น dialog ขึ้นมา</p>	

### 3.5 แผนภาพ Flowchart

Flowchart คือ แผนภาพที่ใช้แสดงการทำงานของระบบ โดยใช้สัญลักษณ์แสดงขั้นตอนและลักษณะการทำงานต่าง ๆ สัญลักษณ์ที่ใช้ในการเขียน Flowchart แสดงในตารางที่ 3.20

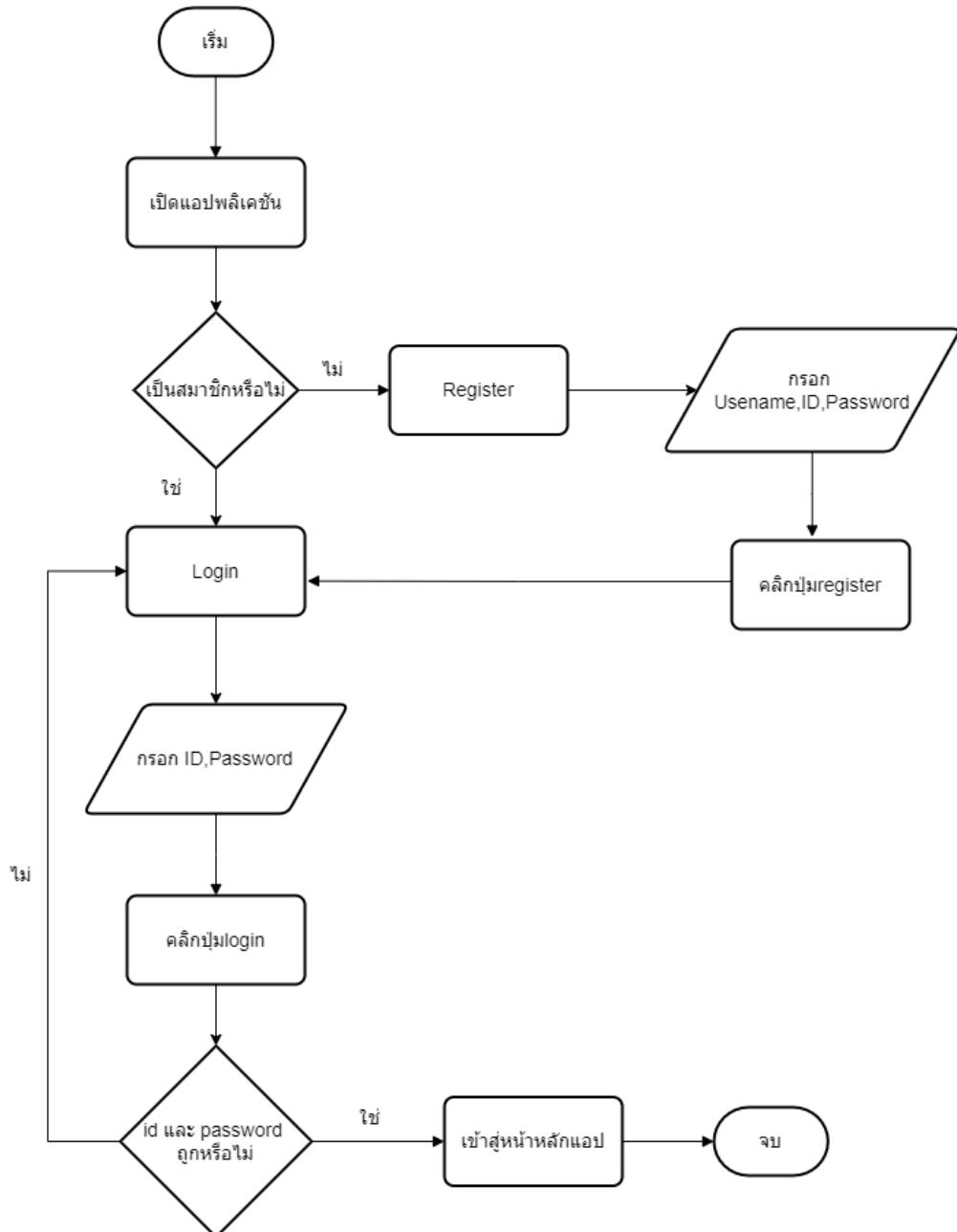
ตารางที่ 3.20 สัญลักษณ์ของ Flowchart

สัญลักษณ์	คำอธิบาย
	แสดงจุดเริ่มต้น และจุดสิ้นสุดการทำงาน
	ใช้แสดงการปฏิบัติงาน
	ใช้แสดงผลและรับค่า
	ใช้เป็นการตรวจสอบเงื่อนไข
	ใช้แสดงทิศทางขั้นตอนการทำงาน

ตารางที่ 3.21 สัญลักษณ์ของ Flowchart (ต่อ)

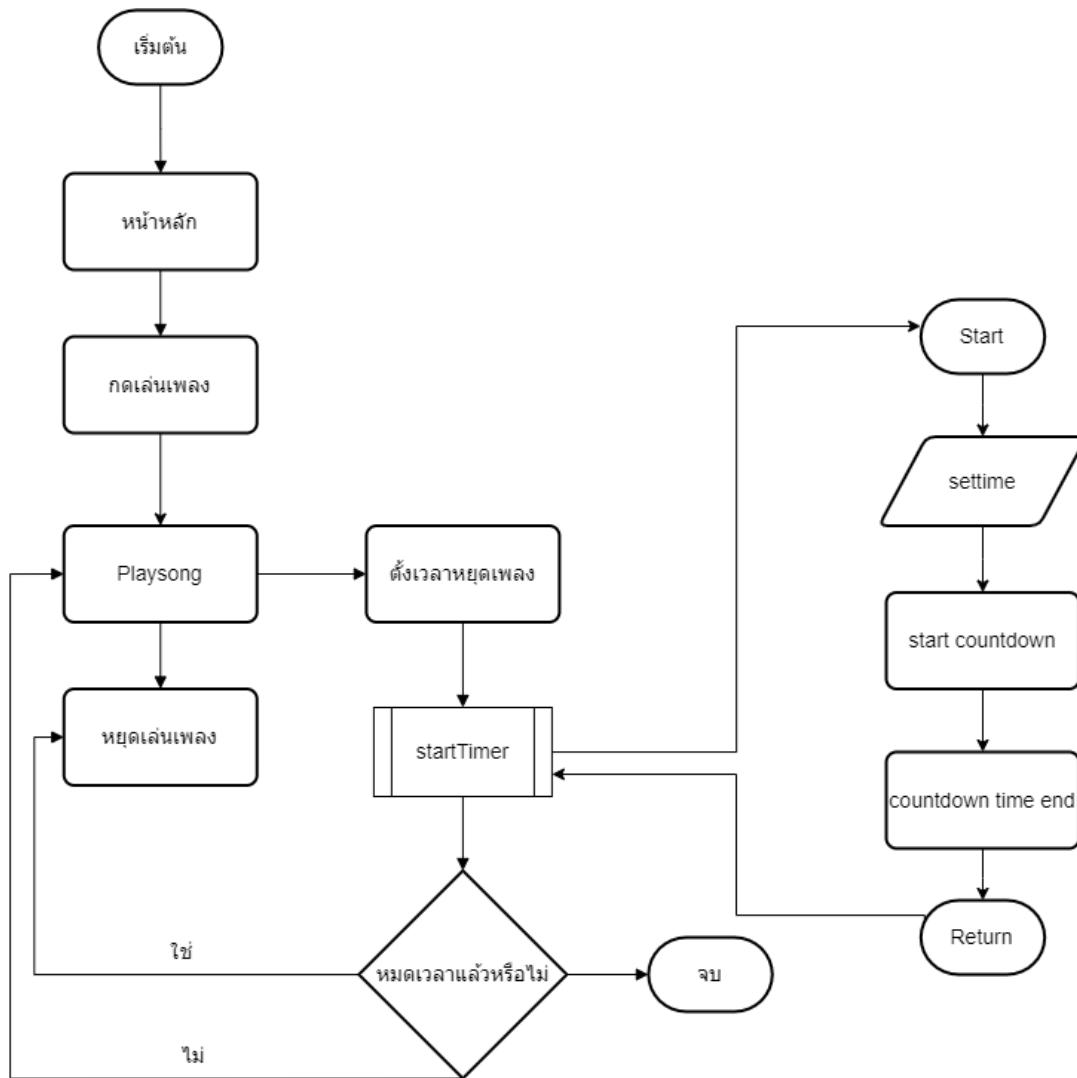
สัญลักษณ์	คำอธิบาย
	ใช้เป็นจุดเชื่อมต่อไปยังขั้นตอนต่าง ๆ
	ใช้เป็นจุดเชื่อมต่อไปยังหน้าใหม่
	ใช้แสดงผลลัพธ์ในขณะประมวลผล เช่น การแสดงผลลัพธ์ออกทางหน้าจอ
	การเรียกใช้ฟังก์ชันย่อยภายใน
	การเรียกใช้ฟังก์ชันย่อยภายนอก

Flowchart แสดงการทำงานแต่ละฟังก์ชันของแอปพลิเคชัน และเว็บแอปพลิเคชันมี ดังนี้



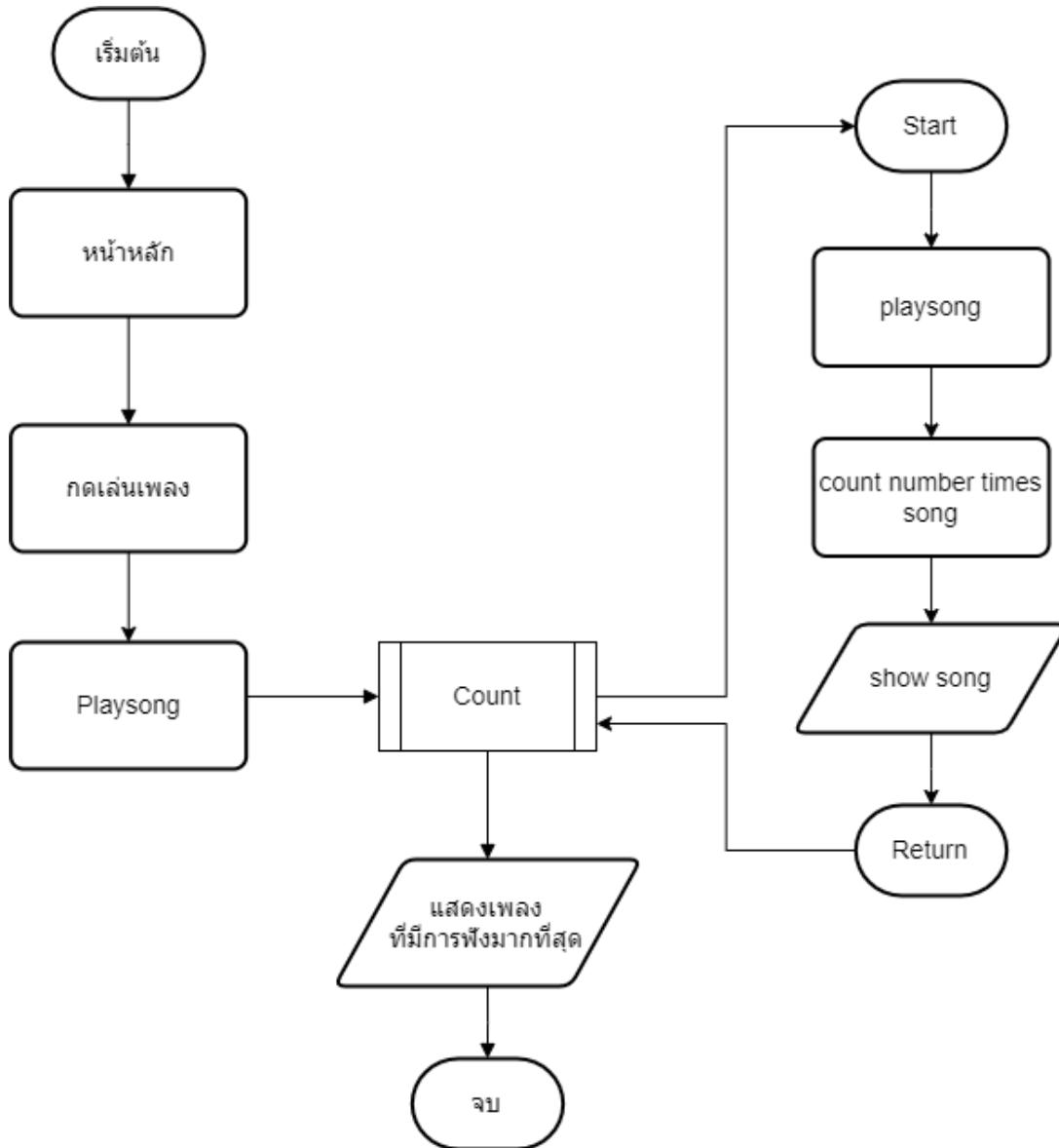
ภาพที่ 3.14 : Flowchart แสดงการทำงานฟังก์ชันสมัครสมาชิกและเข้าสู่ระบบ

จากภาพที่ 3.14 เมื่อผู้ใช้เข้าใช้งานแอปพลิเคชัน กรณีแรกถ้ายังไม่เป็นสมาชิกสามารถเลือก Register และทำการกรอกข้อมูล Username ID และ Password และทำการคลิกปุ่มregister เพื่อยืนยันการสมัครและสามารถทำ Login เข้าสู่ระบบได้ กรณีที่สองเมื่อเป็นสมาชิกสามารถทำ Login เข้าสู่ระบบโดยทำการกรอก ID และ Password เมื่อทำการคลิกปุ่มlogin ระบบจะตรวจสอบ id และ password ถูกต้องหรือไม่ ถ้าไม่จะต้อง Login อีกครั้ง ถ้าใช่ จะเข้าสู่หน้าหลักแอปพลิเคชัน



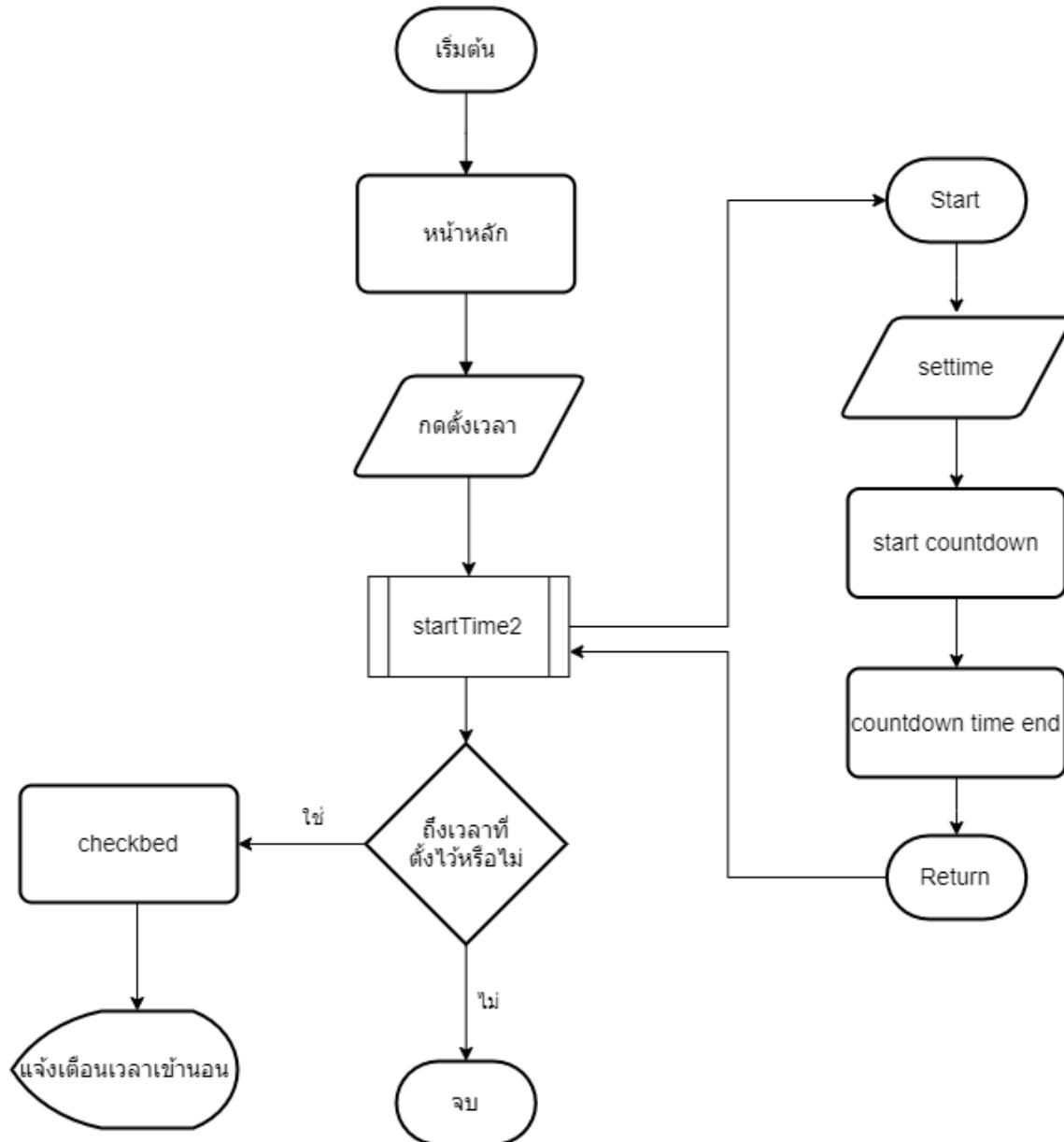
ภาพที่ 3.15 : Flowchart แสดงการทำงานพัฟก์ชันเล่นเพลง

จากภาพที่ 3.15 เมื่อสมาชิกอยู่ที่หน้าหลักแอปพลิเคชัน สมาชิกกดเล่นเพลง พังก์ชัน Playsong จะเริ่มทำงาน เมื่อสมาชิกต้องการตั้งเวลาหยุดเพลง พังก์ชัน StartTime จะเริ่มทำงาน โดยจะเรียกใช้งานพังก์ชันย่อยโดย เมื่อเริ่มจะต้องตั้งเวลาเมื่อตั้งเวลา จะเริ่มนับเวลาโดยหลัง และเมื่อหมดเวลาบัญชีอยหลัง จะคืนค่าไปให้พังก์ชัน StartTime เมื่อเวลาหมดจะทำการหยุดเพลงถ้าเวลาไป่หมดจะเล่นต่อ



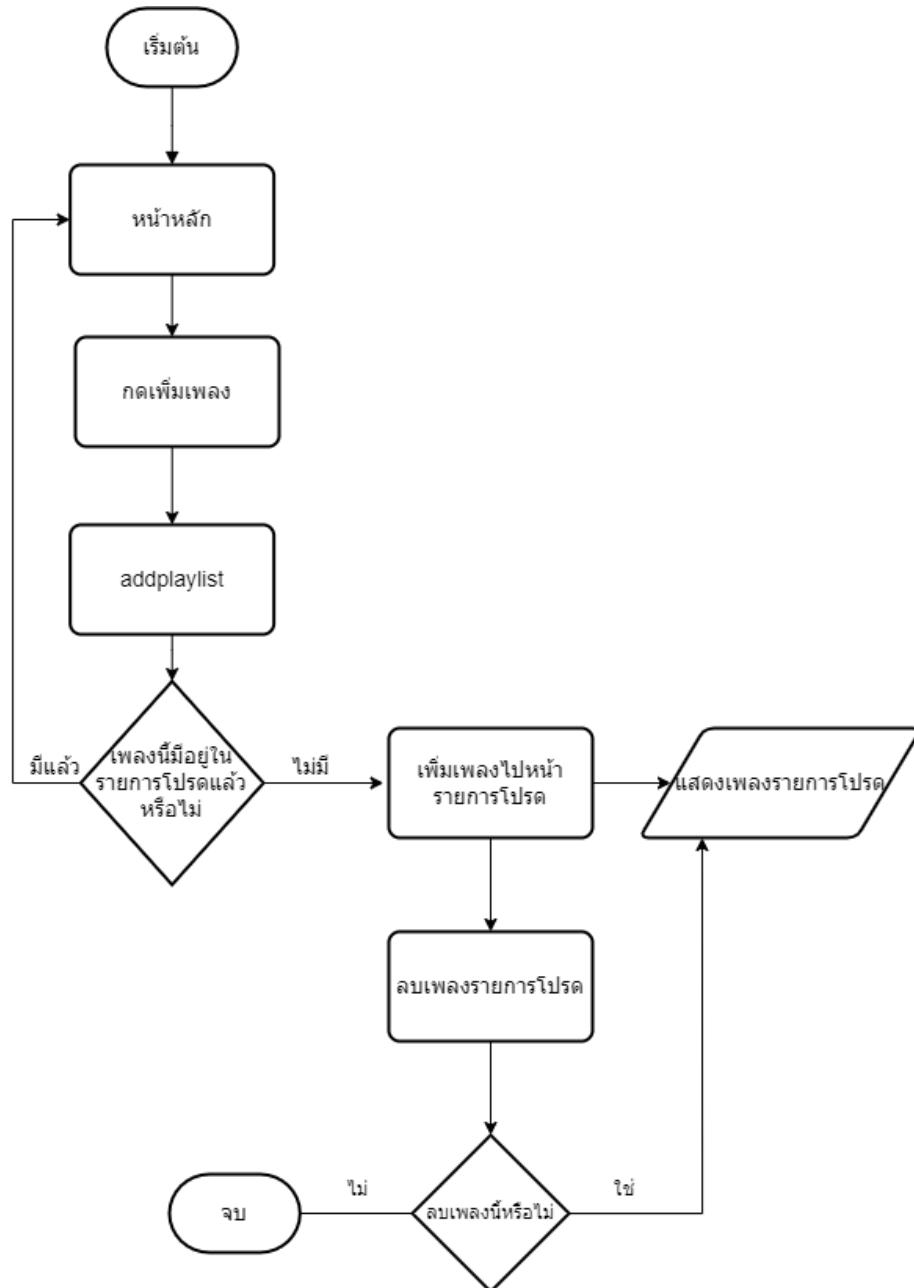
ภาพที่ 3.16 : Flowchart แสดงการทำงานฟังก์ชันเพลงที่มีการฟังมากที่สุด

จากภาพที่ 3.16 เมื่อ sama ซึ่กอยู่ที่หน้าหลักแอปพลิเคชัน สามารถกดเล่นเพลง ฟังก์ชัน Playsong จะเริ่มทำงาน และฟังก์ชัน Count จะเริ่มทำงาน โดยจะเรียกใช้งานฟังก์ชันย่อยโดย เมื่อทำการเล่นเพลง จะนับจำนวนครั้งเพลง และแสดงเพลง จะคืนค่าไปให้ฟังก์ชัน Playsong และจะแสดงเพลงที่มีการฟังมากที่สุด



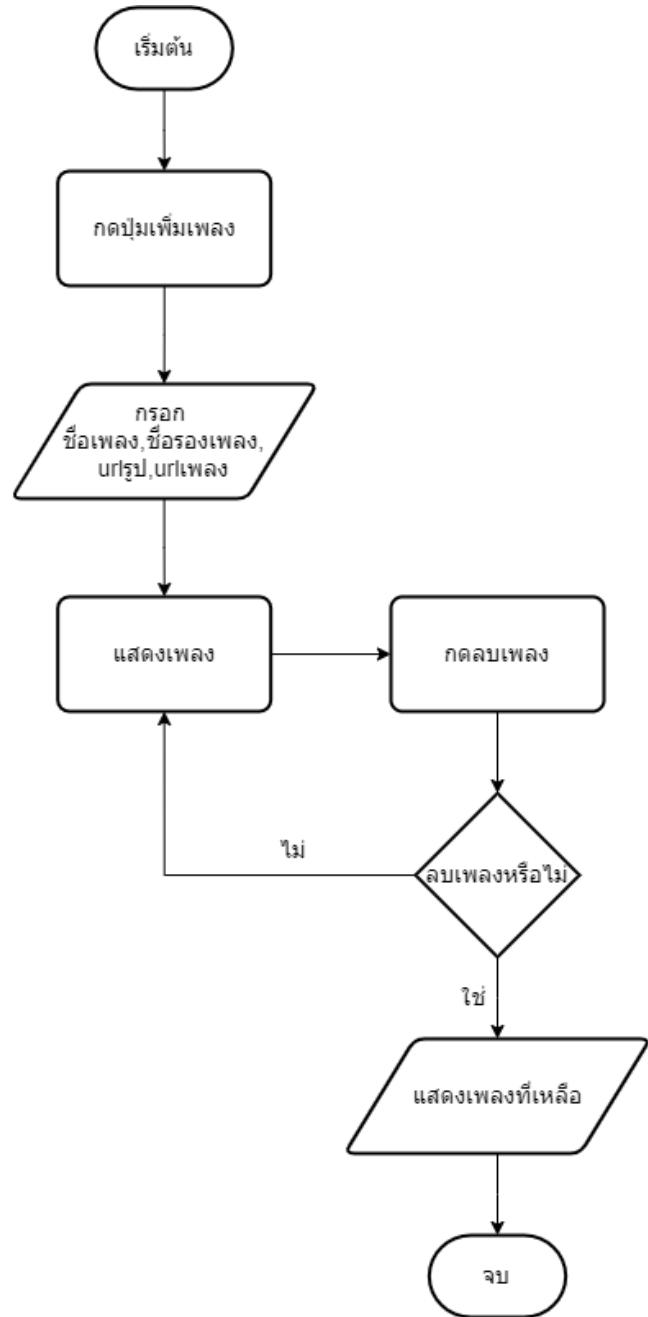
ภาพที่ 3.17 : Flowchart พังก์ชันตั้งเวลาเข้านอนและแจ้งเตือนเวลาเข้านอน

จากภาพที่ 3.17 เมื่อスマาร์ทโฟนที่หน้าหลักแอปพลิเคชัน สามารถกดตั้งเวลา พังก์ชันStartTime2 จะเริ่มทำงาน โดยจะเรียกใช้งาน พังก์ชันย่อยโดย เมื่อเริ่มจะต้องตั้งเวลาเมื่อตั้งเวลา จะเริ่มนับเวลาโดยหลัง และเมื่อหมดเวลาหนึ่งนาทีจะคืนค่าไปให้พังก์ชันStartTime2 เมื่อถึงเวลาที่ตั้งไว้จะทำการเรียกใช้พังก์ชัน checkbed และจะทำการแจ้งเตือนเวลาเข้านอน ถ้ายังไม่ถึงเวลาที่ตั้งไว้จะไม่มีการแจ้งเตือน



ภาพที่ 3.18 : Flowchart แสดงการทำงานฟังก์ชันเพิ่มเพลงและลบเพลงรายการโปรด

จากภาพที่ 3.18 เมื่อสมาชิกอยู่ที่หน้าหลักแอปพลิเคชัน สมาชิกกดเพิ่มเพลง ฟังก์ชัน addplaylist จะเริ่มทำงาน เงื่อนไขเพลงนี้มีอยู่ในรายการโปรดแล้วหรือไม่ ถ้ามีแล้วจะไม่ทำการเพิ่มเพลงนี้ ถ้าไม่มีจะทำการเพิ่มเพลงไปแสดงหน้ารายการโปรด เมื่อต้องการลบเพลงรายการโปรด ถ้าไม่ จบการทำงาน ถ้าใช่จะแสดงเพลงที่ไม่ได้ลบที่หน้ารายการโปรด



ภาพที่ 3.19 : Flowchart แสดงการทำงานฟังก์ชันเพิ่มและลบเพลงส่วนผู้ดูแลระบบ

จากภาพที่ 3.19 เมื่อผู้ดูแลระบบกดปุ่มเพิ่มเพลง แล้วทำการกรอกชื่อเพลง ชื่อร้องเพลง บารูป บราเพลง เมื่อทำการเพิ่มเพลงจะแสดงเพลงที่เพิ่ม เมื่อกดลบเพลง เงื่อนไขลบเพลงหรือไม่ ถ้าไม่จะแสดงเพลง ถ้าใช่เพลงจะถูกลบ และแสดงเพลงที่เหลือ

### 3.6 การออกแบบส่วนฐานข้อมูล (Database Design)

การออกแบบฐานข้อมูล มีดังนี้

Collection	Document ID	Fields
acc	addsongs	
acc	dBbpGAFFoOB0BsmCt9e	<ul style="list-style-type: none"> <li>Bird: 0</li> <li>Blues: 0</li> <li>Charming: 0</li> <li>Classic: 0</li> <li>DrippingWater: 3</li> <li>Electronic: 0</li> <li>Fanciful: 0</li> <li>Forest: 4</li> <li>GentleOceanWaves: 0</li> <li>Happy: 0</li> <li>Hardrock: 0</li> </ul>

ภาพที่ 3.20 : ส่วนเก็บข้อมูลสมาชิก

จากภาพที่ 3.20 ส่วนเก็บข้อมูลสมาชิก มีดังนี้

- ส่วนที่ 1 collection ใช้เก็บข้อมูลเอกสาร
- ส่วนที่ 2 document ใช้เก็บข้อมูลเป็น object
- ส่วนที่ 3 field เก็บข้อมูลแบบ key และ value

1 Bird: 0 2

Blues: 0

Charming: 0

Classic: 0

DrippingWater: 3

ภาพที่ 3.21 : ส่วนเก็บจำนวนครั้งเพลงที่ฟัง

จากภาพที่ 3.21 ส่วนเก็บจำนวนครั้งเพลง มีดังนี้

- ส่วนที่ 1 ชื่อของ key ที่จะเป็นชื่อเพลงแต่ละเพลงในการเก็บค่า
- ส่วนที่ 2 ค่าข้อมูลเป็น Number

m1: 1

m10: 0

m2: 0

m3: 0

m4: 0

m5: 1

ภาพที่ 3.22 : ส่วนเก็บเพลงรายการโปรด

จากภาพที่ 3.22 ส่วนเก็บเพลงรายการโปรด มีดังนี้

- ส่วนเก็บเพลงรายโปรดถ้า key นั้นมีค่าเป็น 0 เพลงนั้นไม่ได้อยู่ในรายการโปรด ถ้า key นั้นมีค่าเป็น 1 เพลงนั้นเป็นเพลงที่อยู่ในรายการโปรด

```
id: "111"
pass: "123456"
user: "test"
```

ภาพที่ 3.23 : ส่วนเก็บข้อมูลสมาชิก

จากภาพที่ 3.23 ส่วนเก็บข้อมูลสมาชิก มีดังนี้

- เป็นส่วนเก็บข้อมูล id password และชื่อ user ของสมาชิก

relax-app-e89f8	addsongs	vVu9jQ9yV0jUbx
+ Start collection	+ Add document	+ Start collection
acc	e3odgChwiwy8XFJI1D16	
addsongs >	sINfsdQ8Fj7qwe600GZW	+ Add field
	vVu9jQ9yV0jUbx60NQw >	<img: "https:="" "slowly"="" 03="" 07="" 2022="" <="" audio="" audio_48408439d="" cdn.pixabay.com="" download="" encrypted-tbn0.gstatic.com="" filename='mindfulness-relaxation-amp-meditation-music-22174.mp3"' images?q='tbn:ANd9GcTyKvmpZnCuG_gh7jWgF-tR_1vvR9VT3hNvPQ&amp;usqp=CAU"' path:="" subtitle:="" td="" title:=""> </img:>

ภาพที่ 3.24 : ส่วนเก็บข้อมูลเพลงที่เพิ่ม

จากภาพที่ 3.24 ส่วนเก็บข้อมูลเพลงที่เพิ่ม มีดังนี้

- Key ชื่อ img เก็บข้อมูล url รูป

- Key ชื่อ path เก็บชื่อ拿出 url เพลง
- Key ชื่อ subTitle เก็บชื่อร้องเพลง
- Key ชื่อ title เก็บชื่อเพลง

## 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 homes.ts อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นการนำเข้าไลบรารีอ่านข้อมูลเบื้องต้นจากไฟร์เบส

- บรรทัดที่ 2-6 เป็นการนำเข้าไลบรารีคลาสสร้องขอการพิ่งจากแหล่งภายนอก

- 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 อ่านค่าผลลัพธ์จากข้อมูลที่ส่งออกมา
- 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            }
38          }
39        }
40      }
41    ]
42  }
43}

```

```

38         this.router.navigate(['tabs/firstpage'],dataString)
39     }
40   }
41 }
42 ]
43 });
44 (await alert).present();
45 }
46 }
```

ภาพที่ 4.4 : โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ (ต่อ)

จากภาพที่ 4.4 โครงสร้างส่วนสมัครสมาชิกและเข้าสู่ระบบ อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 2 ทำการเรียกใช้ alertController ผ่าน create() จะทำการสร้าง dialog สำหรับกรอกข้อมูล
- บรรทัดที่ 3-12 เป็นส่วนของกรอกข้อมูล 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 อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 2 ทำการเรียกใช้ alertCtrl ผ่าน create() จะทำการสร้าง dialog สำหรับกรอกข้อมูล
- บรรทัดที่ 3-15 เป็นส่วนของกรอก 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.hometts.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 นั้นไม่ตรงกันกับที่มีอยู่สามารถใช้ 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 อธิบายการทำงานได้ดังนี้

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

#### 4.1.4 การพัฒนาส่วนໂຫລດຂໍ້ອມລາມາແສດງ

```
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 : ໂຄງຮຽງສ່ວນໂຫລດຂໍ້ອມລາມາແສດງ

ຈາກກາພທີ 4.8 ໂຄງຮຽງສ່ວນໂຫລດຂໍ້ອມລາມາແສດງ ອົບຍາກຮ່າງການໄດ້ດັ່ງນີ້

- ບຣທັດທີ 1-15 ເປັນການກຳນົດຕັ້ງແປປຈາກໄລບຣາຣີທີ່ນຳເຂົາ

```

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 : โครงสร้างส่วนໂ Holden ข้อมูลมาแสดง (ต่อ)

จากภาพที่ 4.9 โครงสร้างส่วนໂ Holden ข้อมูลมาแสดง 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 : โครงสร้างส่วนໂ Holden ข้อมูลมาแสดง (ต่อ)

จากภาพที่ 4.10 โครงสร้างส่วนໂ Holden ข้อมูลมาแสดง อยู่ภายในการทำงานไปได้ดังนี้

- บรรทัดที่ 35-56 เป็นส่วนข้อมูลการพิ่งเพลงสมาชิก

```

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   mwatervalley: 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(),
80   srnb: e.payload.doc.data()['s2'].toString(),
81   sclassic: e.payload.doc.data()['s3'].toString(),
82   sjazz: e.payload.doc.data()['s4'].toString(),
83   ssynthpop: e.payload.doc.data()['s5'].toString(),
84   sindie: e.payload.doc.data()['s6'].toString(),
85   sblues: e.payload.doc.data()['s7'].toString(),
86   selectronic: e.payload.doc.data()['s8'].toString(),
87   shardrock: e.payload.doc.data()['s9'].toString(),
88   shiphop: e.payload.doc.data()['s10'].toString(),
89
90
91
92
93
94   };
95 });
96 console.log(this.list)
97 this.count()
98 this.send()
99 });
100 }

```

ภาพที่ 4.11 : โครงสร้างส่วนเรียกใช้พังก์ชัน (ต่อ)

จากภาพที่ 4.11 โครงสร้างส่วนเรียกใช้พังก์ชัน อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 57-91 เป็นส่วนข้อมูลเพลงรายการโปรดสมาชิก

- บรรทัดที่ 97 เป็นส่วนเรียกใช้ฟังก์ชันนับจำนวนครั้งเพลงที่พัง
- บรรทัดที่ 98 เป็นส่วนเรียกใช้ฟังก์ชันส่งข้อมูลไปยังหน้าแท็บรายโปรด

#### 4.1.5 การพัฒนาส่วนแสดงเพลงที่พังบ่อย

```

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 : โครงสร้างส่วนแสดงเพลงที่พังบ่อย

จากภาพที่ 4.12 โครงสร้างส่วนแสดงที่พังบ่อย อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-31 เป็นฟังก์ชันนับค่าจำนวนครั้งที่ฟังเพลงและทำการเรียงเพลงที่มีการฟังมากที่สุด
- บรรทัดที่ 3-8 เป็นส่วนเงื่อนไขหา id ของผู้ใช้ว่าเป็นของผู้ใช้คนไหน
- บรรทัดที่ 11-13 เป็นส่วนกำหนดเพลงที่สมาชิกคนนั้นฟังมีเพลงอะไรบ้าง
- บรรทัดที่ 15-16 เรียงเพลงที่สมาชิกฟังจากมาเกิลเบน้อย
- 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 หรือรายการทำงานได้ดังนี้

- 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 อยู่ในไฟล์ main.js ดังนี้

- 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 อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-25 เป็นฟังก์ชันตั้งเวลาหยุดเพลง
- บรรทัดที่ 1-13 เป็นส่วนนับเวลา
- บรรทัดที่ 15-25 เป็นส่วนคำนวณนับเวลาถอยหลัง

#### 4.2.0 การพัฒนาส่วนแจ้งเตือนเวลานอน

```

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

```

ภาพที่ 4.19 : โครงสร้างส่วนแจ้งเตือนเวลาเข้านอน

จากภาพที่ 4.19 โครงสร้างส่วนแจ้งเตือนเวลาเข้านอน อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-7 เป็นส่วนการแจ้งเตือน

```

1 ionViewDidEnter2() {
3 this.remainingTime2 = this.ax;
4 this.startTimer2();
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 : โครงสร้างส่วนแจ้งเตือนเวลาเข้านอน (ต่อ)

จากภาพที่ 4.20 โครงสร้างส่วนแจ้งเตือนเวลาเข้าอน อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นส่วนเริ่มนับเวลา
- บรรทัดที่ 3-4 เป็นส่วนเรียกใช้ฟังก์ชันนับเวลา
- บรรทัดที่ 7-20 เป็นส่วนนับเวลา
- บรรทัดที่ 22-32 เป็นส่วนคำนวณนับเวลาโดยหลัง
- บรรทัดที่ 33-56 เป็นส่วนคำนวณนับเวลาโดยหลังเป็นลูป

#### 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 เป็นฟังก์ชันลับเพลงออกจากรายการโปรด

```

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 alertController: 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 เป็นส่วนแสดงผล

```

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,ชื่อร่องเพลง, 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 การทดสอบส่วนเพิ่มเพลงและลบเพลง

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าเพิ่มเพลง	แสดงเพลงที่เพิ่ม	ระบบแสดงเพลงที่เพิ่ม
	ลบเพลง	เพลงที่ลบจะไม่แสดงในระบบ

### 5.1.4 การทดสอบหน้าเล่นเพลง

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าเล่นเพลง	เล่นเพลงที่เลือก	ระบบเล่นตามเพลงที่เลือกได้ถูกต้อง
	ตั้งเวลาหยุดเพลง	เพลงหยุดตามเวลาที่ตั้งไว้

### 5.1.5 การทดสอบหน้าสมัครสมาชิกเข้าใช้งานและเข้าสู่ระบบ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าสมัครสมาชิกเข้าใช้งานและเข้าสู่ระบบ	สมัครสมาชิก	สามารถสมัครสมาชิกสำหรับเข้าสู่ระบบได้
	เข้าสู่ระบบ	สามารถเข้าสู่ระบบได้

## บทที่ 6

### สรุปและข้อเสนอแนะ

โครงงานแอปพลิเคชันเสียงเพื่อการผ่อนคลายนี้ พบร่วมระบบมีส่วนที่ยังไม่สามารถทำงานได้ตามที่ออกแบบไว้ ในบทนี้ผู้พัฒนาจึงขอสรุปความสามารถของแอปพลิเคชัน ซึ่งจะปัญหาและอุปสรรค พร้อมเสนอแนวทางพัฒนาต่อ ตามลำดับ

#### 6.1 สรุปความสามารถแอปพลิเคชัน

##### 6.1.1 เว็บ

###### ผู้ดูแลระบบ

- สามารถเพิ่มเพลงได้
- สามารถลบเพลงได้

##### 6.1.2 แอปพลิเคชัน

###### สมาชิก

- สามารถสมัครและเข้าสู่ระบบได้
- สามารถตั้งเวลาเข้านอน
- เตือนเวลาเข้านอน
- สามารถเพิ่มเสียงเป็นรายการโปรดได้
- สามารถลบเพลงที่เป็นรายการโปรดได้
- สามารถดูเพลงชิดได้
- สามารถตั้งเวลาปิดเสียงได้

## 6.2 ปัญหาและอุปสรรคในการพัฒนา

- สร้างแอปพลิเคชัน ไม่สามารถสร้างแอปพลิเคชันโดยใช้ ionic capacitor ได้ เนื่องจากบางไฟบรารีใน ionic capacitor ใหม่กว่า ionic cordova ซึ่งแอนดรอยด์ยังไม่รองรับบางไฟบรารี

แนวทางในการแก้ไข : เปลี่ยนมาใช้ ionic cordova ในการสร้างแอปพลิเคชัน

- เวอร์ชันแอนดรอยด์ ไม่สามารถใช้แอนดรอยด์เวอร์ชัน 9 ใน การสร้างแอปพลิเคชัน เนื่องจากแอนดรอยด์เวอร์ชัน 9 ไม่รองรับกราฟิกแสดงผลของ ionic cordova

แนวทางในการแก้ไข : เปลี่ยนมาใช้แอนดรอยด์เวอร์ชัน 10 ขึ้นไป

## 6.3 แนวทางพัฒนาต่อ

- เพิ่มเพลงให้หลากหลายแนว เช่น เพิ่มแนวเพลงแอ็ฟรีบีท ที่เป็นแนวเพลงฮิปฮอปสม RnB ที่ให้ความสนุกสนาน

- เพิ่มฟังก์ชันการเล่นในเบื้องหลัง โดยเมื่อออกจากแอปพลิเคชันจะสามารถ กดเล่นเพลง กดหยุดเพลง กดฟังเพลงถัดไป กดเล่นเพลงก่อนหน้า โดยแสดงที่แท็บการแจ้งเตือนของสมาร์ทโฟน และ เล่นเพลงขณะปิดหน้าจอโดยสามารถ กดเล่นเพลง กดหยุดเพลง กดฟังเพลงถัดไป กดเล่นเพลงก่อนหน้า โดยแสดงที่หน้าจอล็อคของสมาร์ทโฟน

## 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 Petrungr (2017). บทที่ 1 สถาปัตยกรรมของระบบแอนดรอยด์ ( โครงสร้างของระบบแอนดรอยด์) เป็งตัน [ออนไลน์]. สืบค้นเมื่อ 11 พฤษภาคม 2565. จาก <https://pongployappdev.medium.com/บทที่-1-สถาปัตยกรรมของระบบแอนดรอยด์-โครงสร้าง-ของระบบแอนดรอยด์-เบี้งตัน-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 ช่วยนอนไม่หลับ ผ่อนคลายด้วยเสียง [ออนไลน์]. สืบค้นเมื่อ 27 พฤษภาคม 2565. จาก [https://intrend.trueid.net/article/รีวิว-relax-melodies-แอปฯ-ช่วยนอนไม่หลับ-ผ่อนคลายด้วยเสียง-\\_trueidintrend\\_189029](https://intrend.trueid.net/article/รีวิว-relax-melodies-แอปฯ-ช่วยนอนไม่หลับ-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>.

[11] เอกนรินทร์ คำคุณ, คุณสมบัติของ Firebase, หน้า 101.

[12] จตุรพัชร์ พัฒนทรงศิวีล, Node.js คืออะไร, หน้า 7.

[13] A-Droid Tech Andlan Pinto, Start Building Apps in Ionic, p. 55.

[14] บัญชา ประสีลະเตสัง, เกี่ยวกับ Node.js และการติดตั้ง, (กรุงเทพฯ: ชีเอ็ดยูเคชั่น, 2564),

หน้า 18.

[15] ดร.ปกรณ์ ลีสุทธิพรชัย, ถ้าจะนะของแอปพลิเคชัน, (กรุงเทพฯ: สำนักพิมพ์มหาวิทยาลัยธรรมศาสตร์, 2561), หน้า 14.

ภาคผนวก

## ภาคผนวก ก

### การติดตั้งเครื่องมือในการพัฒนาแอปพลิเคชัน

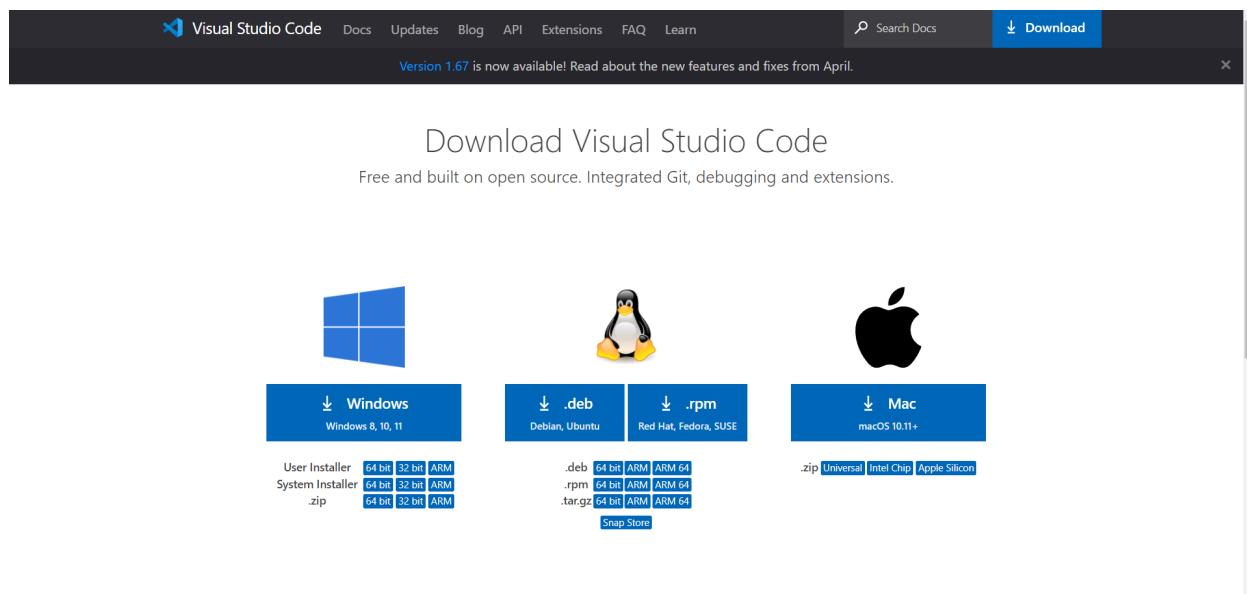
โปรแกรมในการพัฒนาแอปพลิเคชันมีดังต่อไปนี้

- 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

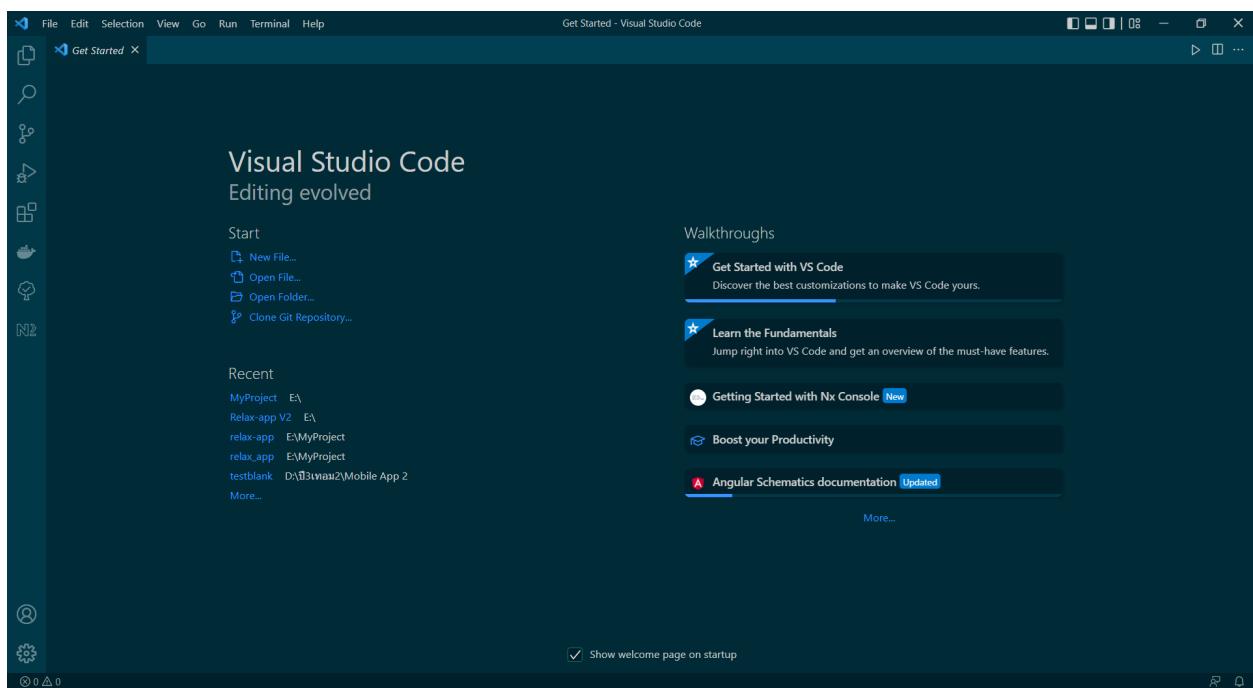
ที่มา: <https://code.visualstudio.com/download>

2. เปิดไฟล์ที่ดาวน์โหลดมา ดังภาพ ก.2



ภาพที่ ก.2 :ไฟล์โปรแกรม Visual Studio Code

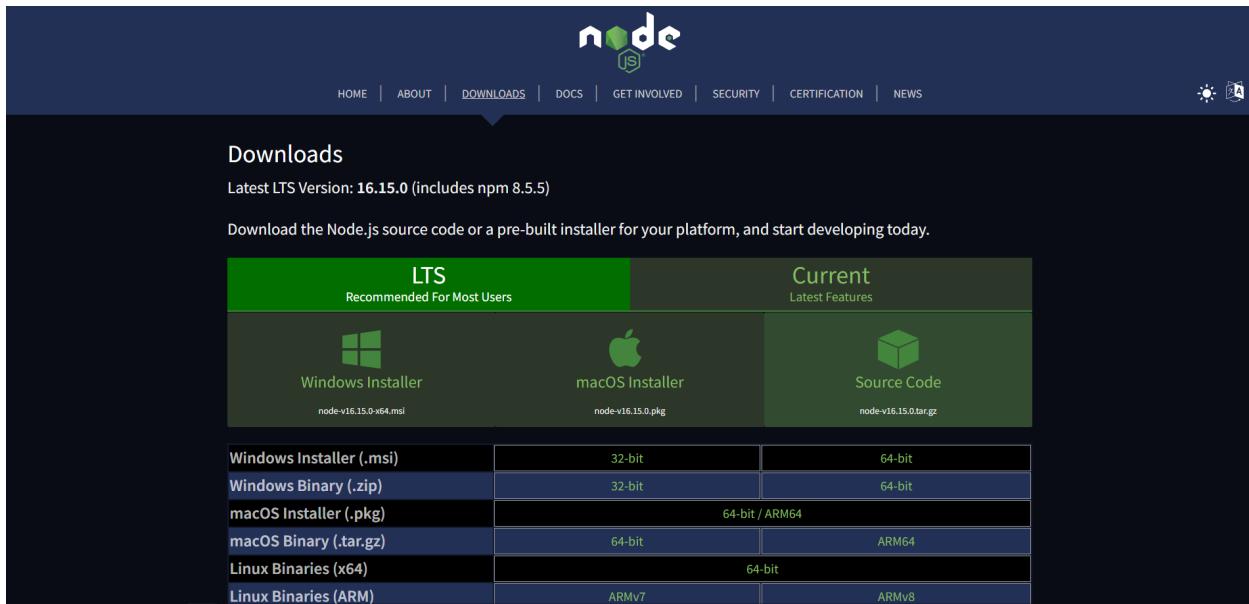
3. โปรแกรมพร้อมใช้งาน ดังภาพ ก.3



ภาพที่ ก.3 : โปรแกรม Visual Studio Code

## ก.2 การติดตั้ง Node.js

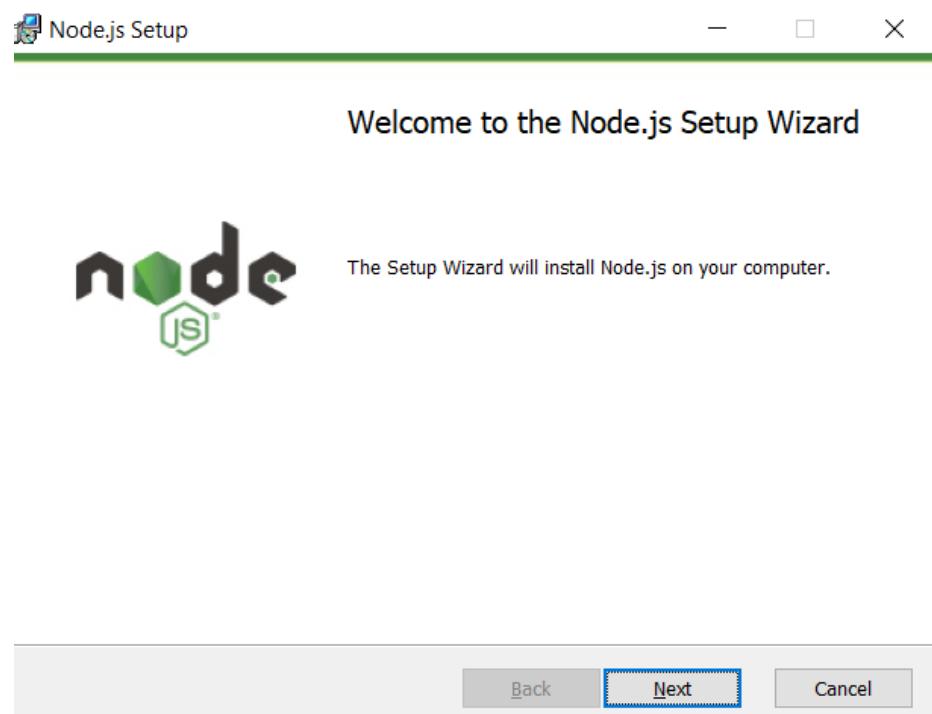
1. ดาวน์โหลด Node.js ได้ที่ <https://nodejs.org/en/download/> เลือกติดตั้งตามระบบปฏิบัติการ ดังภาพ ก.4



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

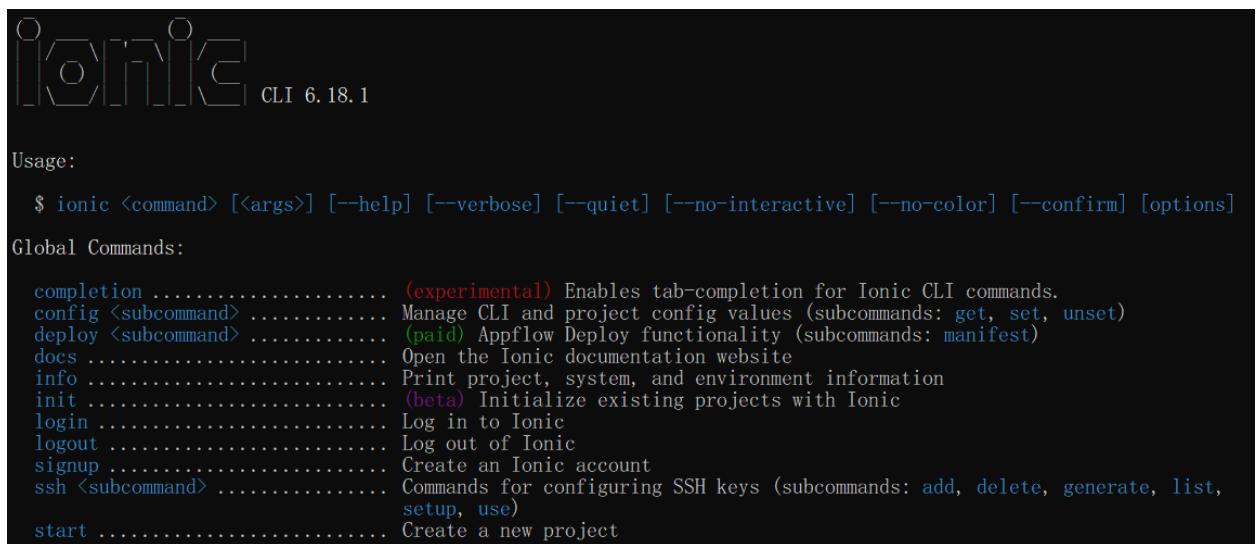
1. การติดตั้ง Ionic Framework สามารถติดตั้งโดยเปิด Command Prompt พร้อมพิมคำสั่ง `npm install -g @ionic/cli` ดัง

ภาพ ก.7

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

ภาพที่ ก.7 : คำสั่งติดตั้ง Ionic Framework

2. ผลการติดตั้ง Ionic Framework ดังภาพ ก.8



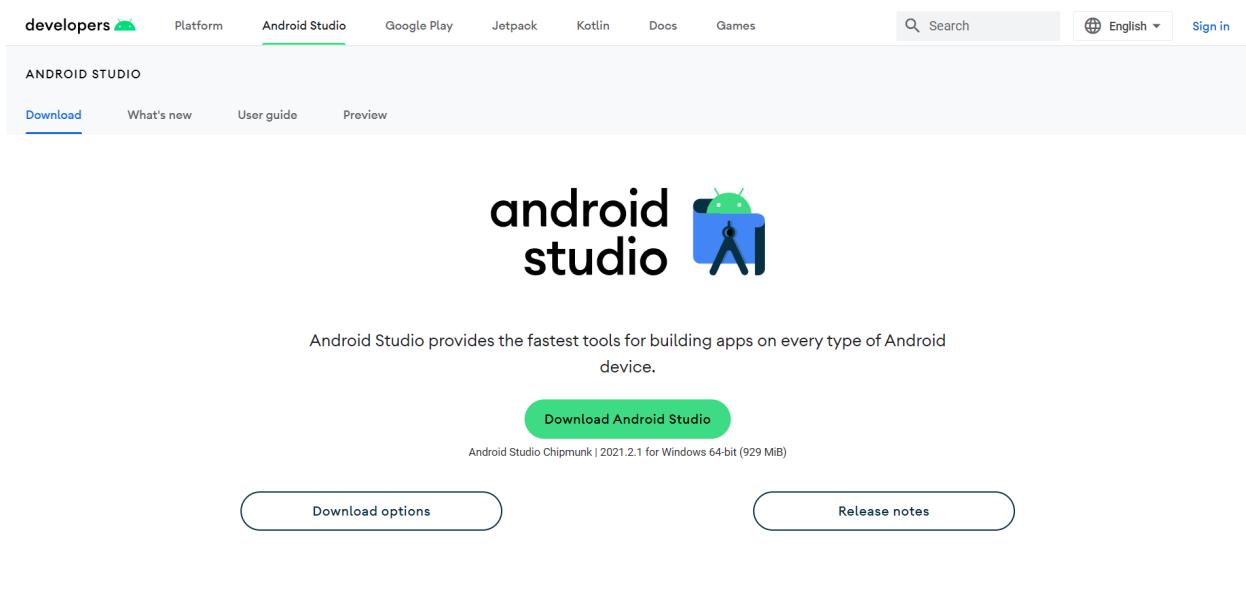
```
○○[○]CLI 6.18.1
Usage:
$ ionic <command> [<args>] [--help] [--verbose] [--quiet] [--no-interactive] [--no-color] [--confirm] [options]
Global Commands:
completion ..... (experimental) Enables tab-completion for Ionic CLI commands.
config <subcommand> ..... Manage CLI and project config values (subcommands: 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=EAIaIQobChMl3MzSkIrd9wIVUA4rCh1qdwLDEAAYASAAEglaR\\_D\\_BwE&gclsrc=aw.ds](https://developer.android.com/studio?gclid=EAIaIQobChMl3MzSkIrd9wIVUA4rCh1qdwLDEAAYASAAEglaR_D_BwE&gclsrc=aw.ds) ดูภาพ ก.9

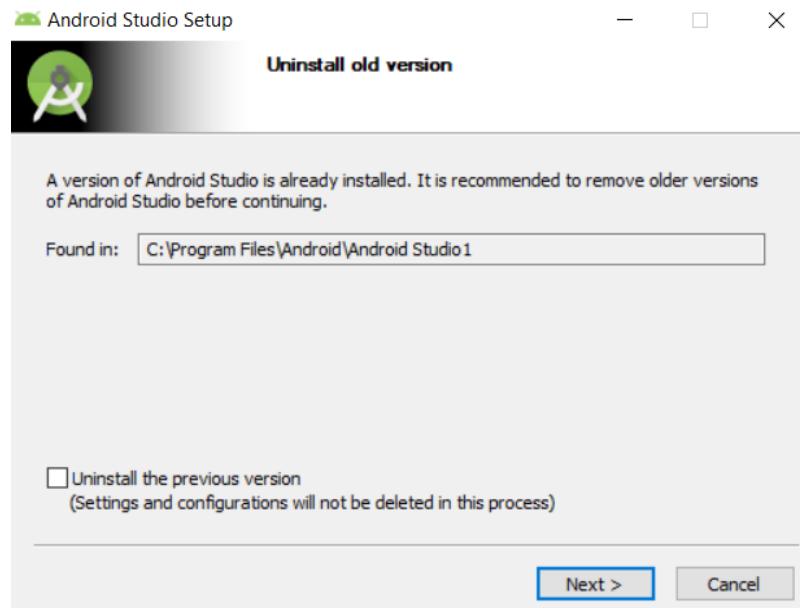


ภาพที่ ก.9 : หน้าเว็บดาวน์โหลด Android Studio

ที่มา:

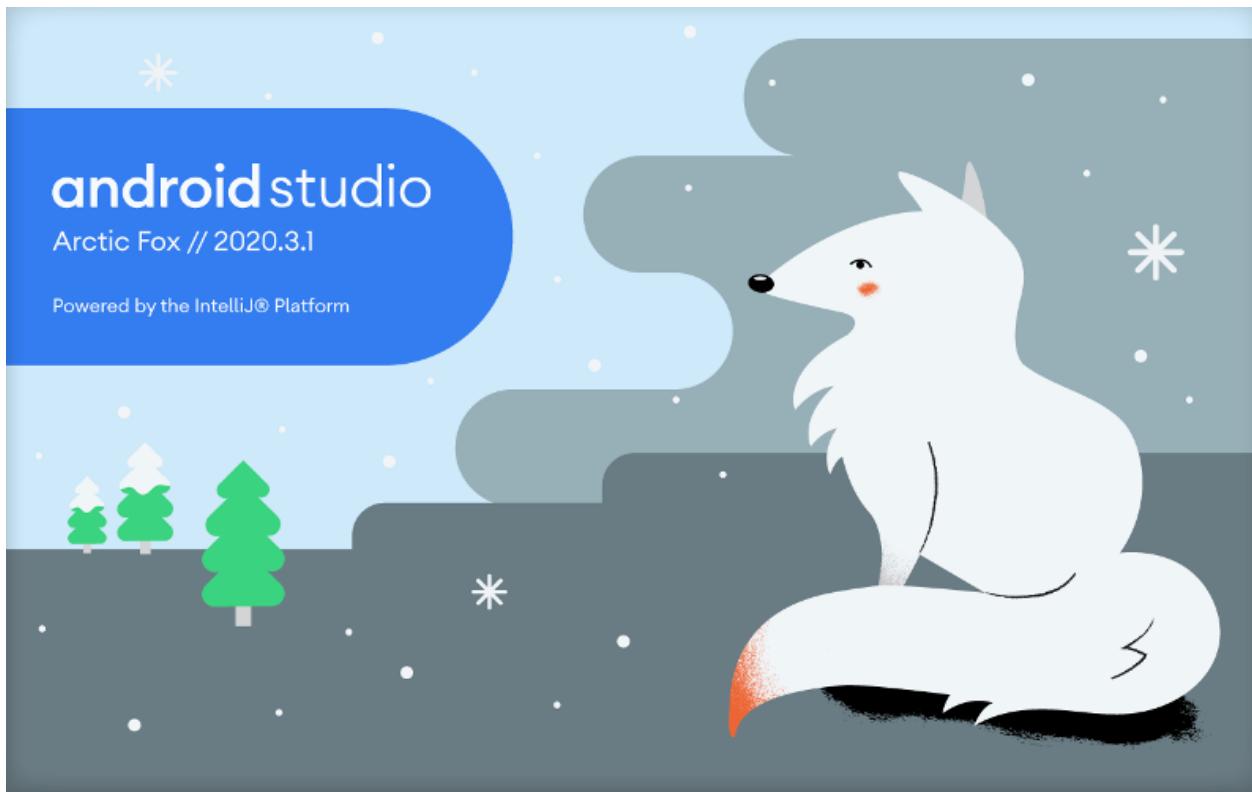
[https://developer.android.com/studio?gclid=EAIaIQobChMl3MzSkIrd9wIVUA4rCh1qdwLDEAAYASAAEgLDKPD\\_BwE&gclsrc=aw.ds](https://developer.android.com/studio?gclid=EAIaIQobChMl3MzSkIrd9wIVUA4rCh1qdwLDEAAYASAAEgLDKPD_BwE&gclsrc=aw.ds)

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

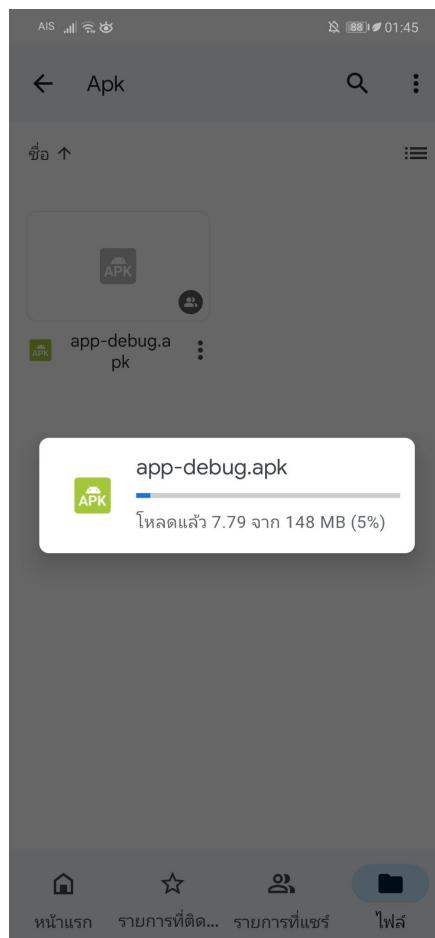
## ภาคผนวก ข

### 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



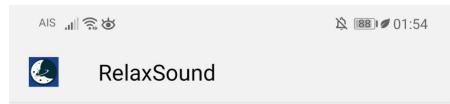
ภาพที่ ข.1 : หน้าดาวน์โหลดแอปพลิเคชัน Relax Sound

2. กดติดตั้งแอปพลิเคชัน ดังภาพที่ ข.2



ภาพที่ ข.2 : หน้าแสดงการติดตั้ง

3. กดเปิด เพื่อเริ่มใช้งานแอปพลิเคชัน ดังภาพที่ ข.3



ติดตั้งแอปพลิเคชันแล้ว

เสร็จสิ้น

เปิด

ภาพที่ ข.3 : หน้าแสดงการติดตั้งสำเร็จ

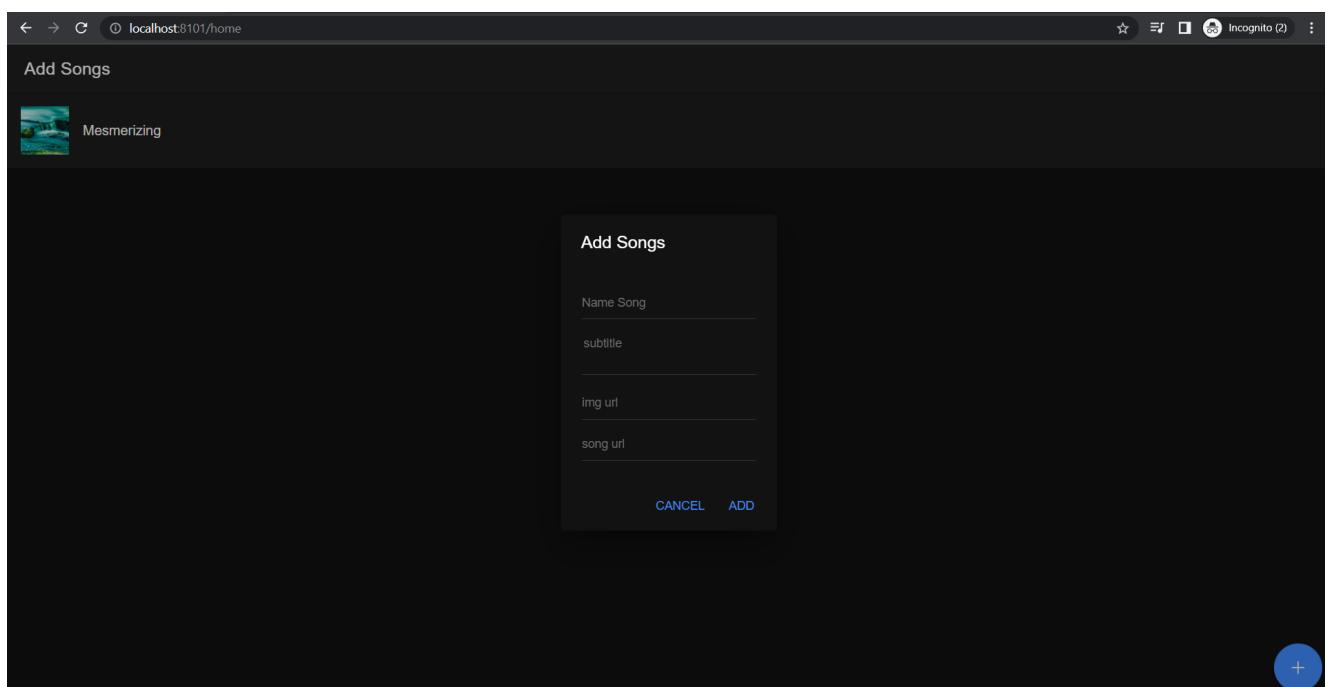
## ການພັບປຸງ ດ

### ຄູ່ມືອກາຮົາໃຊ້ຈານຮະບບ

ຄູ່ມືອກາຮົາໃຊ້ຈານຮະບບແປ່ງເປັນ 2 ສ່ວນດັ່ງນີ້

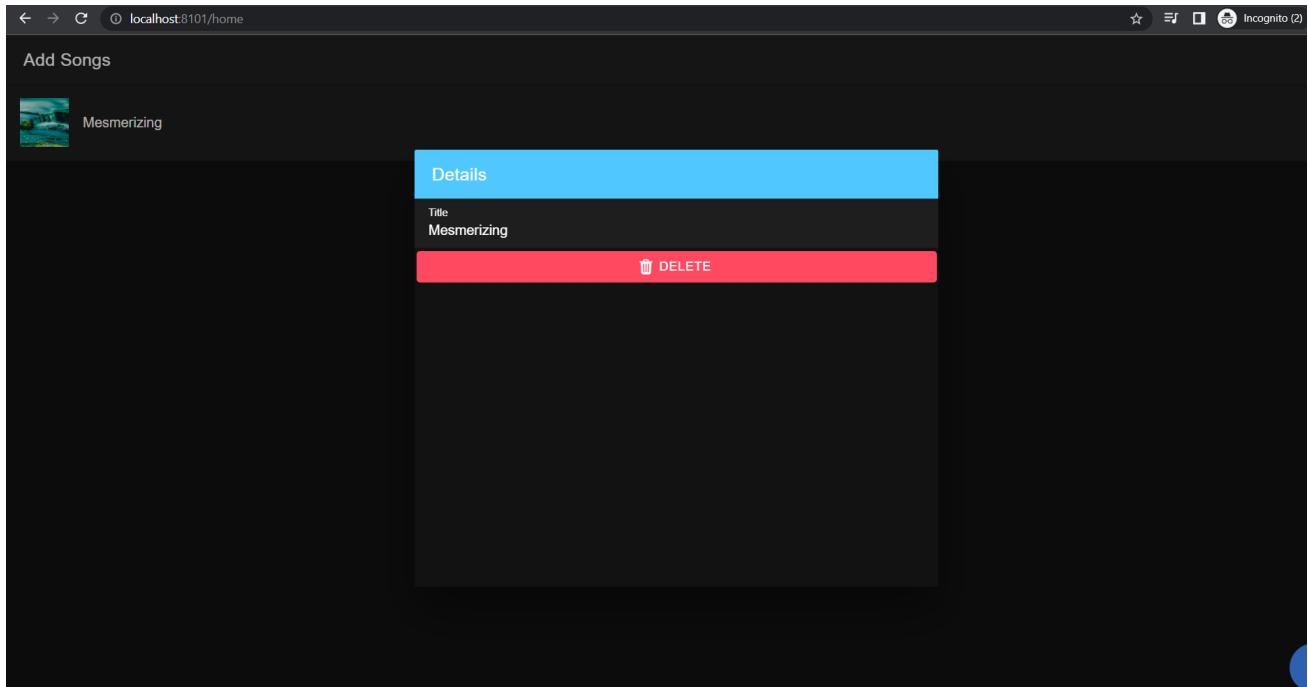
#### 1. ສ່ວນເວັບຜູ້ດູແລຮະບບ

- ໜ້າຈອໃນໜ້ານີ້ຜູ້ດູແລຮະບບສາມາດພິມເພັນ ໂດຍກຣອກ ຂຶ່ອເພັນ ຂໍ້ອຮອງ ທີ່ອຢູ່ຮູບ ທີ່ອຢູ່ເພັນ ດັ່ງການທີ່ ດ.1



ການທີ່ ດ.1 : ໜ້າຈອແສດງເພັນ ແລະ ພິມເພັນ

- ผู้ดูแลระบบสามารถลบเพลงที่เพิ่ม ดังภาพที่ ค.2



ภาพที่ ค.2 : หน้าจอส่วนลบเพลง

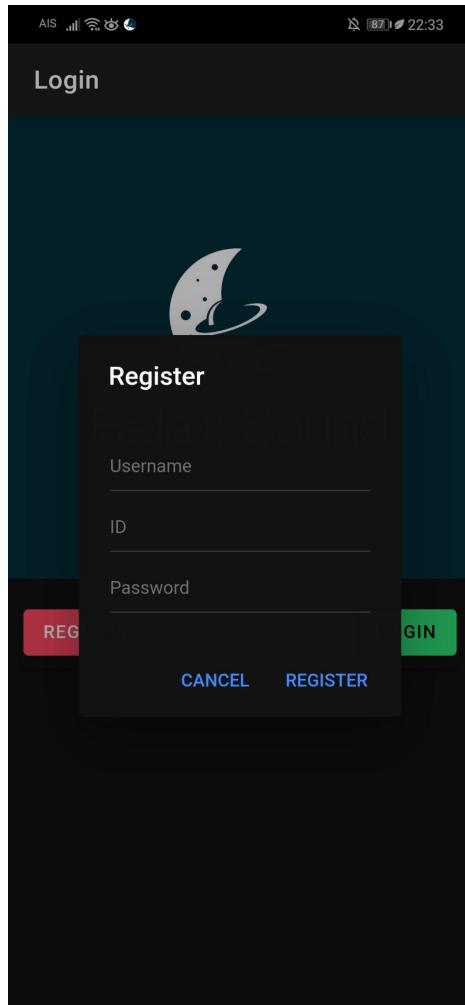
2. ส่วนหน้าจอป้องกันสำหรับสมาชิก

- หน้าจอต้อนรับเวลาเข้าแอปพลิเคชัน ดังภาพที่ ค.3



ภาพที่ ค.3 : หน้าจอต้อนรับ

- หน้าสมัครสมาชิก ดังภาพที่ ค.4

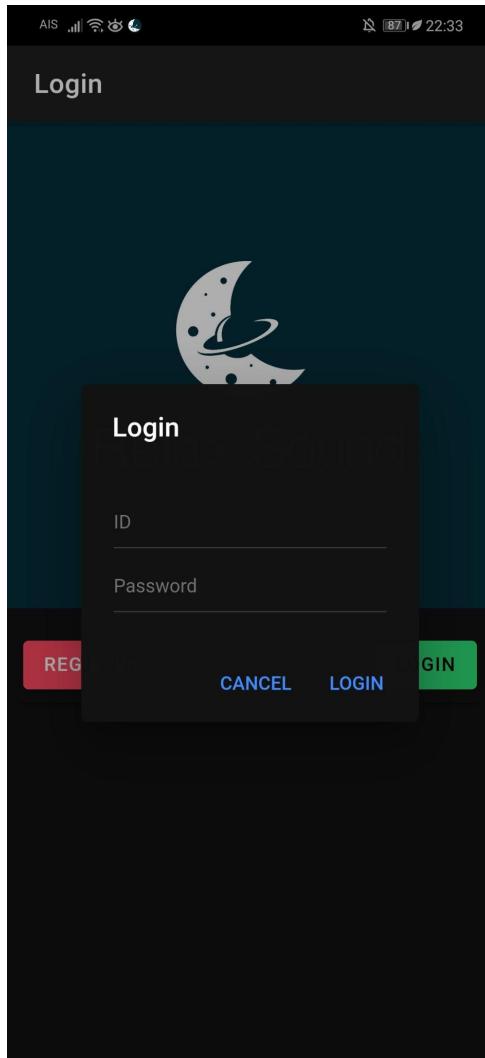


ภาพที่ ค.4 : หน้าสมัครสมาชิก

จากภาพที่ ค.4 สามารถอธิบายได้ดังนี้

- ผู้ใช้ที่สามารถกรอกข้อมูลสมัครสำหรับเข้าใช้งาน

- หน้าเข้าสู่ระบบ ดังภาพที่ ค.5

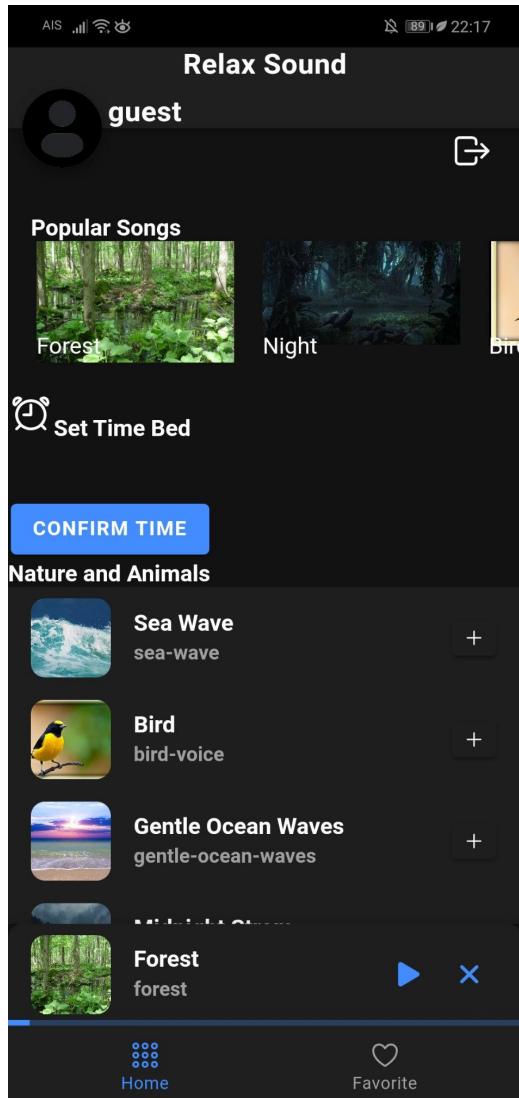


ภาพที่ ค.5 : หน้าเข้าสู่ระบบ

จากภาพที่ ค.5 สามารถอธิบายได้ดังนี้

- สามารถกรอกข้อมูลเข้าสู่ระบบ เมื่อสมัครสมาชิกแล้ว

- หน้าจอแสดงผลหลักสำหรับเล่นเพลง และแสดงเพลงที่ samaชิกชอบฟัง และอื่นๆ ดังภาพที่ ค.6

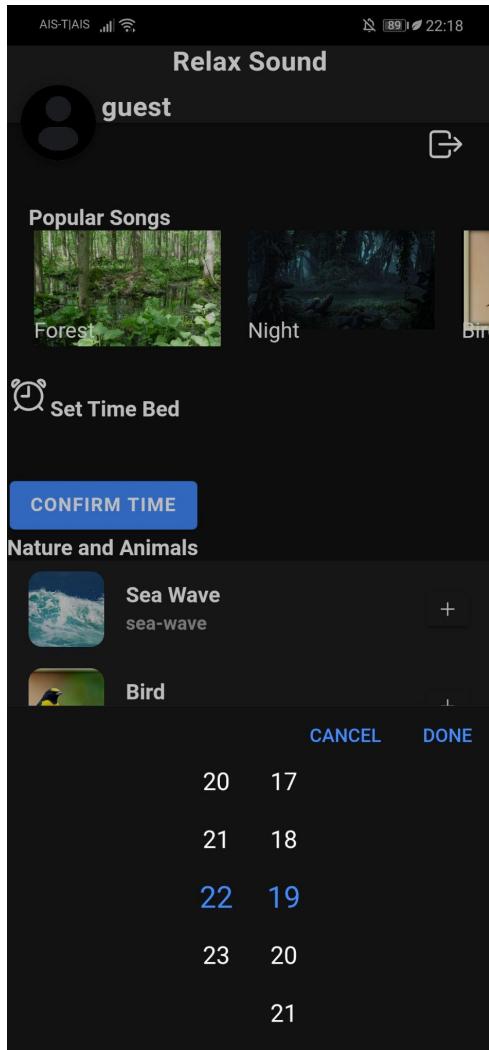


ภาพที่ ค.6 : หน้าจอหลัก

จากภาพที่ ค.6 สามารถอธิบายได้ดังนี้

- สามารถดูเพลงที่ชอบฟังมากที่สุด
- สามารถกดเลือกเล่นเพลง
- สามารถเพิ่มเพลงไปรายการโปรดโดยกดไอคอนเครื่องหมายบวก
- สามารถออกจากระบบโดยกดไอคอนรูปประตุมุมบนขวา
- สามารถกดไอคอนรูปหัวใจเพื่อถูрайโปรด

- หน้าจอส่วนตั้งเวลาเข้านอน ดังภาพที่ ค.7

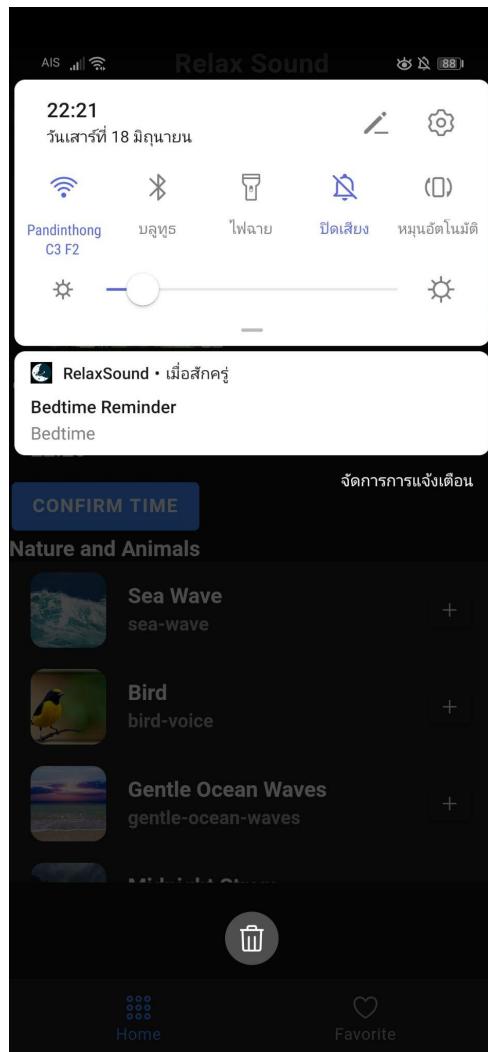


ภาพที่ ค.7 : หน้าจอส่วนตั้งเวลาเข้านอน

จากภาพที่ ค.7 สามารถอธิบายได้ดังนี้

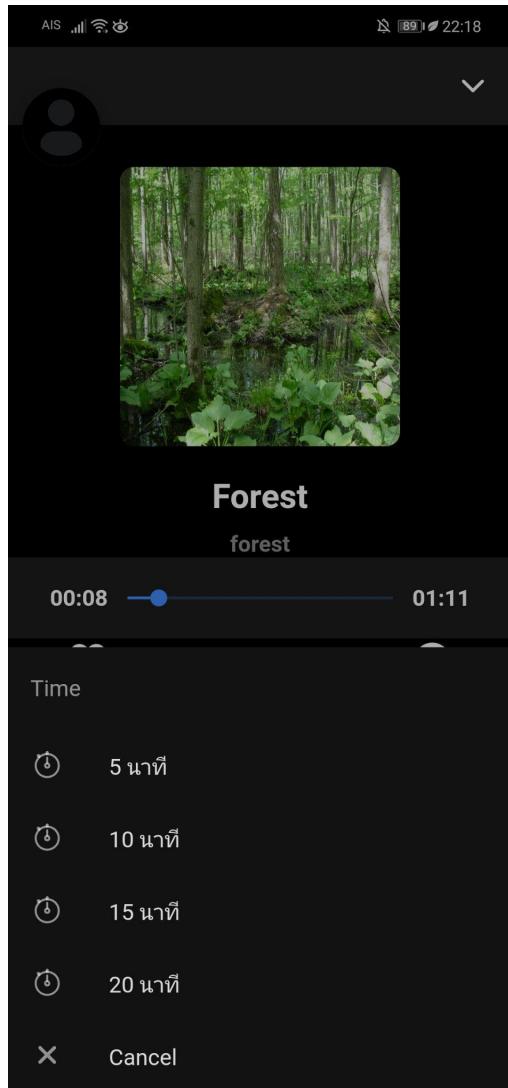
- สามารถตั้งเวลาสำหรับแจ้งเตือนการนอน โดยการกดไอคอนนาฬิกา เลือกเวลาที่ต้องการ กด Done และ กด CONFIRM TIME

- ส่วนแจ้งเตือนเวลาเข้านอน เมื่อถึงเวลาที่スマาร์ตโฟนจะแจ้งเตือน ดังภาพที่ ค.8



ภาพที่ ค.8 : ส่วนแจ้งเตือนเวลาเข้านอน

- หน้าจอส่วนตั้งเวลาหยุดเล่นเพลง ดังภาพที่ ค.9

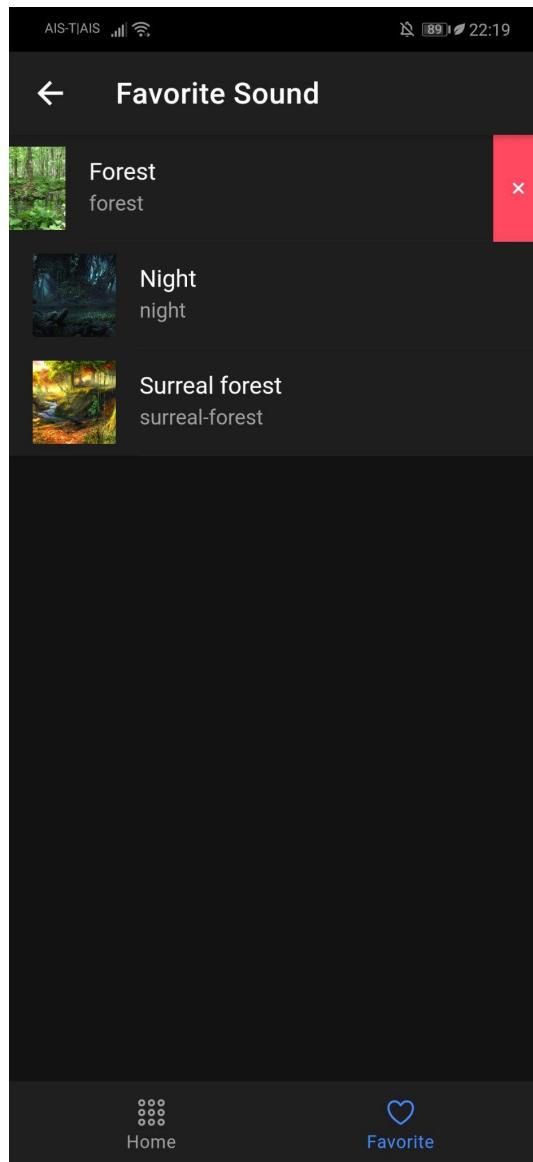


ภาพที่ ค.9 : หน้าจอตั้งเวลาหยุดเพลง

จากภาพที่ ค.9 สามารถอธิบายได้ดังนี้

- สามารถตั้งเวลาหยุดเพลงโดยกดรูปนาฬิกา และเลือกเวลาที่ต้องการหยุดเพลง

- หน้าจอส่วนรายการโปรด ดังภาพที่ ค.10



ภาพที่ ค.10 : หน้าจอรายการโปรด

จากภาพที่ ค.10 สามารถอธิบายได้ดังนี้

- สามารถคุ้มเพลงที่เป็นรายการโปรด สามารถใส่ลิสต์ขวาน้ำเพื่อลบเพลงรายการโปรด

## 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