



ระบบจัดการงานแสดงของชุมนุมดนตรี คณะวิทยาศาสตร์และเทคโนโลยี

โดย

นาย ณกalin บุณยกะลิน
นาย สันติภพ สโมสร

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์
ปีการศึกษา 2567
ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

ระบบจัดการงานแสดงของชุมชนดนตรี คณะวิทยาศาสตร์และเทคโนโลยี

โดย

นาย ณกalin บุณยกะลิน

นาย สันติภพ สโมสร

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

SCI BAND MUSIC PERFORMANCE MANAGEMENT SYSTEM

BY

MR. NAGALIN PUNYAKALIN

MR. SANTIPHAP SAMOSORN

**A FINAL-YEAR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE**

COMPUTER SCIENCE

FACULTY OF SCIENCE AND TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2024

COPYRIGHT OF THAMMASAT UNIVERSITY

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

นาย ณกalin บุณยกะลิน
นาย สันติภาพ สมสทร

เรื่อง

ระบบจัดการงานแสดงของชุมชนดนตรี คณะวิทยาศาสตร์และเทคโนโลยี

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
เมื่อ วันที่ 30 พฤษภาคม พ.ศ. 2568

อาจารย์ที่ปรึกษา

กรรมการสอบโครงการพิเศษ

กรรมการสอบโครงการพิเศษ

(ผศ. ดร. ทรงศักดิ์ รองวิริยะพานิช)

(ผศ. ดร. เด่นดวง ประดับสุวรรณ)

(ผศ. ดร. อรจิรา สิทธิชักดี)

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

นาย ณัฐิน บุณยกะลิน
นาย สันติภาพ สโนมร
เรื่อง

ระบบจัดการงานแสดงของชุมชนดินทรี คณะวิทยาศาสตร์และเทคโนโลยี

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
เมื่อ วันที่ 30 พฤษภาคม พ.ศ. 2568

อาจารย์ที่ปรึกษา

กรรมการสอบโครงการพิเศษ

กรรมการสอบโครงการพิเศษ

(ผศ. ดร. ทรงศักดิ์ รองวิริยะพาณิช)

(ผศ. ดร. เด่นดวง ประดับสุวรรณ)

(ผศ. ดร. อรจิรา สิทธิศักดิ์)

หัวข้อโครงการพิเศษ	ระบบจัดการงานแสดงของชุมชนตรี คณะวิทยาศาสตร์และเทคโนโลยี
ชื่อผู้เขียน	นาย ณกลิน บุณยกะลิน
ชื่อผู้เขียน	นาย สันติภาพ สมสระ
ชื่อปริญญา	วิทยาศาสตรบัณฑิต สาขาวิชาบริการคอมพิวเตอร์
สาขาวิชา/คณะ/มหาวิทยาลัย	สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์
อาจารย์ที่ปรึกษาโครงการพิเศษ	ผศ. ดร. ทรงศักดิ์ รองวิริยะพาณิช

ปีการศึกษา 2567

บทคัดย่อ

โครงการนี้เป็นการพัฒนาระบบการจัดการงานแสดงดนตรีสำหรับชุมชน SCI BAND คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต โดยมีปัญหาหลักจากการสื่อสารที่ไม่เป็นระเบียบในกลุ่มไลน์จำนวนมาก และการสลับตัวของสมาชิก (Performer) ระหว่างการแสดงที่ทำให้เกิดความสับสนในการจำลำดับการแสดง จึงได้พัฒนาระบบขึ้นเป็นโมบายแอปพลิเคชันด้วย React Native, NestJS, PostgreSQL ซึ่งสามารถจัดการข้อมูลสมาชิก, ลำดับการแสดง, คิวเพลง, และการแจ้งเตือนผ่าน Discord ในวันแสดงจริง

การพัฒนาระบบนี้มุ่งเน้นการจัดการลำดับการแสดงให้มีความเป็นระเบียบ โดยใช้ระบบแจ้งเตือนและการจัดเก็บข้อมูลที่เป็นระบบ ทำให้สมาชิกและทีมงานสามารถติดตามความคืบหน้าของงานได้แบบเรียลไทม์ นอกจากนี้ระบบยังสามารถอ่านข้อมูลจาก Google Forms ที่ใช้สำหรับการสมัครเข้าชุมชนผ่านทาง Google Sheets เพื่อเป็นการจำกัดการเข้าถึงระบบให้มีเฉพาะสมาชิกชุมชน เท่านั้นที่สามารถเข้าใช้งานได้ พร้อมทั้งมีฟีเจอร์ในการปิดบัญชีผู้ใช้งานเมื่อสมาชิกชุมชนจบการศึกษา

ระบบนี้ช่วยเพิ่มประสิทธิภาพในการจัดการงานแสดงดนตรีของ SCI BAND ทำให้การแสดงดนตรีมีความราบรื่นและมีความเป็นมืออาชีพมากขึ้น ลดความสับสนที่เกิดจากการสื่อสารที่ไม่เป็นระบบ และเพิ่มความพึงพอใจให้แก่ผู้เกี่ยวข้อง ทั้งสมาชิก, ผู้จัดงาน, และผู้ชม โดยระบบนี้ยังส่งเสริมการทำงานร่วมกันและความร่วมมือที่ดีในทีมงานภายในชุมชน

คำสำคัญ: SCI BAND, งานแสดงดนตรี, ระบบจัดการงานแสดงดนตรี, การสื่อสาร

(2)

Thesis Title	SCI BAND MUSIC PERFORMANCE MANAGEMENT SYSTEMS
Author	Mr. Nagalin Punyakalin
Author	MR. Santiphap Samosorn
Degree	Bachelor of Science
Major Field/Faculty/University	Computer Science Faculty of Science and Technology Thammasat University
Project Advisor	Assistant Professor Dr. Songsakdi Rongviriyapanish
Academic Years	2024

ABSTRACT

This project involves the development of a music performance management system for the **SCI BAND** club under the Faculty of Science and Technology, Thammasat University, Rangsit Campus. The primary issues addressed include disorganized communication within multiple LINE chat groups and confusion caused by last-minute performer swaps during live events, which often disrupted the order of performances.

To solve these problems, a mobile application was developed using **React Native**, **NestJS**, and **PostgreSQL**. The system facilitates the management of member information, performance order, song queues, and real-time notifications via Discord on the event day.

The system emphasizes organized performance scheduling and structured data management, enabling members and organizers to monitor event progress in real-time. It also supports integration with **Google Sheets**, which stores data from Google Forms used for club registration, allowing the system to restrict access exclusively to active SCI BAND members. Furthermore, the system includes functionality to deactivate accounts for members who have graduated.

This system enhances the overall efficiency of SCI BAND's event management, contributing to smoother and more professional musical performances. It reduces confusion stemming from poor communication, improves stakeholder satisfaction—including members, event organizers, and audiences—and fosters effective collaboration within the club's team.

Keywords: SCI BAND, music performance, music management system, communication

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงด้วยความร่วมมือและการสนับสนุนจากหลายฝ่าย ข้าพเจ้าขอแสดงความขอบคุณจากใจจริงท่อผู้ที่มีส่วนสำคัญในการช่วยเหลือให้โครงการนี้ประสบความสำเร็จ ก่อนอื่น ข้าพเจ้าขอขอบพระคุณอาจารย์ที่ปรึกษา พศ. ดร.ทรงศักดิ์ รองวิริยะพาณิช ซึ่งได้ให้คำแนะนำอย่างใกล้ชิด ตั้งแต่ขั้นตอนการวางแผน การพัฒนางาน ไปจนถึงการแก้ไขปัญหาต่าง ๆ ท่านได้มอบความรู้และประสบการณ์อันทรงคุณค่า รวมถึงการให้กำลังใจที่มีความสำคัญอย่างยิ่งต่อความสำเร็จของโครงการนี้

ข้าพเจ้าขอขอบคุณสมาชิกชุมนุม SCI BAND ทุกคนที่ให้การสนับสนุนอย่างเต็มที่ใน การร่วมให้ข้อมูล ความคิดเห็น และข้อเสนอแนะต่าง ๆ การช่วยเหลือจากทุกคนทำให้ข้าพเจ้าเข้าใจถึงปัญหาและความต้องการที่แท้จริงของผู้ใช้โครงการนี้ และสามารถพัฒนางานให้ตอบสนองต่อความต้องการได้อย่างเหมาะสม

นอกจากนี้ ข้าพเจ้ายังรู้สึกซาบซึ้งต่อเพื่อนร่วมชั้นเรียนและเพื่อนร่วมงานทุกคนที่ได้ร่วมให้คำปรึกษาและกำลังใจตลอดระยะเวลาการดำเนินโครงการ ความร่วมมือและการสนับสนุนจากเพื่อนเหล่านี้ทำให้ข้าพเจ้าสามารถเชื่อมั่นในความสามารถของทุกคน ได้อย่างมั่นใจ

สุดท้ายนี้ ข้าพเจ้าขอขอบคุณครอบครัวที่เคยอยู่เคียงข้างและให้กำลังใจมาโดยตลอด ความรักและการสนับสนุนจากครอบครัวเป็นพลังสำคัญที่ช่วยผลักดันให้ข้าพเจ้าก้าวข้ามทุกอุปสรรคและทำให้โครงการนี้สำเร็จ

โครงการนี้ไม่เพียงเป็นผลลัพธ์ของการทำงานหนัก แต่ยังสะท้อนถึงความร่วมมือและน้ำใจจากผู้คนรอบข้าง ข้าพเจ้าหวังเป็นอย่างยิ่งว่าโครงการนี้จะมีประโยชน์ต่อชุมนุม SCI BAND และจะสามารถนำไปพัฒนาต่ออยู่ต่อไปในอนาคต

นาย ณกalin บุณยกะลิน

นาย สันติภาพ สโนสร

สารบัญ

บทคัดย่อ	1
ABSTRACT	3
กิตติกรรมประกาศ	4
สารบัญ	5
สารบัญตาราง	8
สารบัญภาพ	9
รายการสัญลักษณ์และคำย่อ	13
บทที่ 1 บพนฯ	1
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ของโครงการ	3
1.5 ข้อจำกัดของโครงการ	3
บทที่ 2 วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	5
2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง	5
2.1.1 ทฤษฎีการสื่อสารในองค์กร	5
2.1.2 Discord	6
2.1.3 Discord API	6
2.1.4 Discord bot	7
2.1.5 React Native	7

2.1.6	NestJS.....	8
2.1.7	PostgreSQL.....	10
2.1.8	Prisma.....	10
2.2	แอปพลิเคชันต้นแบบ	12
บทที่ 3	วิธีการวิจัย	14
3.1	ภาพรวมของโครงงาน	14
3.1.2	ER diagram สำหรับฐานข้อมูล	34
3.2	การวิเคราะห์ขอบเขตและความต้องการของระบบ.....	35
3.2.2	คำอธิบายแผนภาพกรณีใช้งาน (Use case Description).....	36
3.3	การออกแบบการทำงานของระบบ	39
3.3.1	คำอธิบาย use case ของระบบ (Use case Specification).....	39
3.4	ประเด็นที่น่าสนใจและสิ่งที่ท้าทาย.....	85
3.4.1	การจัดการบทบาทภายในระบบ	85
3.4.2	การเชื่อมต่อ กับ Discord	85
	3.4.2.1 การแจ้งเตือนไปยัง Discord.....	85
	3.4.2.2 การจัดการผู้ใช้ใน Discord.....	85
3.5	ผลลัพธ์ที่คาดหวัง	85
3.5.1	การจัดงานแสดงเป็นไปได้อย่างราบรื่น	85
3.5.2	การสื่อสารและการเก็บข้อมูลที่ชัดเจนยิ่งขึ้น	86
3.6	ระบบต้นแบบและผลลัพธ์เบื้องต้น	87
บทที่ 4	ทรัพยากรและแผนการดำเนินงาน	115
4.1	การจัดเตรียมฮาร์ดแวร์และซอฟต์แวร์	115
4.1.1	ฮาร์ดแวร์ที่ใช้ในโครงงาน	115

4.1.1.1	คอมพิวเตอร์ส่วนบุคคล	115
4.1.1.2	เครื่องมือเสริม.....	115
4.1.2	ซอฟต์แวร์ที่ใช้ในโครงงาน.....	115
4.1.2.1	ระบบปฏิบัติการ (Operating System).....	115
4.1.3	ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ	115
4.2	แผนการดำเนินงาน	116
4.3	วิธีการประเมินผล.....	117
บทที่ 5	120
5.1	ผลการประเมินผล	120
5.2	แนวคิดการต่อยอดและปรับปรุงระบบ.....	126
รายการอ้างอิง.....		128

สารบัญตาราง

3.2.2.1	ตารางรายละเอียดกรณีการใช้งานระบบทั้งหมด	36
3.3.1.1	รายละเอียดกรณีการใช้งานเข้าสู่ระบบ	39
3.3.1.2	รายละเอียดกรณีดูรายละเอียดงานแสดงดนตรี	40
3.3.1.3	รายละเอียดกรณีสร้างงานแสดงดนตรี.....	42
3.3.1.4	รายละเอียดกรณีแก้ไขงานแสดงดนตรี	45
3.3.1.5	รายละเอียดกรณีลบงานแสดงดนตรี.....	47
3.3.1.6	รายละเอียดกรณีดู queue เพลง	49
3.3.1.7	รายละเอียดกรณีเพิ่มเพลงลงใน queue	52
3.3.1.8	รายละเอียดกรณีแก้ไขเพลงใน Queue.....	55
3.3.1.9	รายละเอียดกรณีย้าย Queue เพลง.....	58
3.3.1.10	รายละเอียดกรณีลบเพลงใน Queue	61
3.3.1.11	รายละเอียดกรณีดูรายละเอียดเพลง	63
3.3.1.12	รายละเอียดกรณี Assign เพลงให้ Performer	67
3.3.1.13	รายละเอียดกรณี Run event.....	70
3.3.1.14	รายละเอียดกรณีเลือกเพลงถัดไป.....	72
3.3.1.15	รายละเอียดกรณีแจ้งเตือน Queue ให้ Performer	74
3.3.1.16	รายละเอียดกรณีเคลียร์ User ออกจากระบบ.....	77
3.3.1.17	รายละเอียดกรณีเพิ่มบัญชีผู้ใช้เข้าสู่ระบบ	79
3.3.1.18	รายละเอียดกรณีดู Profile.....	80
3.3.1.19	รายละเอียดกรณี Logout	81
3.3.1.20	รายละเอียดกรณีโอนย้ายสิทธิ์ system admin.....	82
4.2.1	ตารางการดำเนินงานที่ผ่านมาจนถึงปัจจุบัน	116
4.2.2	ตารางแผนการดำเนินงานในอนาคต.....	117

สารบัญภาพ

3.1.1 ภาพสถาปัตยกรรมระบบ (Architecture Diagram)	14
3.1.1.1 โครงสร้าง folders frontend	16
3.1.1.2 คำอธิบาย file ต่าง ๆ สำหรับ frontend	17
3.1.1.3 API spec สำหรับงานแสดงดนตรี	19
(1) API spec สำหรับดึงข้อมูลงานแสดงดนตรีทั้งหมด	19
(2) API spec สำหรับสร้างงานแสดงดนตรี	20
(3) API spec สำหรับดึงข้อมูลของงานแสดงดนตรีนั้น ๆ ด้วย event ID	21
(4) API spec สำหรับอัปเดตรายละเอียดงานแสดงดนตรี	21
(5) API spec สำหรับลบงานแสดงดนตรี	22
(6) API spec สำหรับดึงข้อมูลเพลงปัจจุบันของงานแสดงดนตรีนั้น ๆ	22
(7) API spec สำหรับอัปเดตเพลงปัจจุบันของงานแสดงดนตรีนั้น ๆ	23
(8) API spec สำหรับอัปเดตสถานะงานแสดงดนตรีนั้น ๆ เป็น COMPLETED หลัง จบงานแสดงดนตรีแล้ว	23
(9) API spec สำหรับอัปเดตสถานะงานแสดงดนตรีนั้น ๆ เป็น ONGOING เมื่อถึง เวลาเริ่มแสดงดนตรี	24
3.1.1.4 API spec สำหรับการจัดการเพลง	25
(1) API spec สำหรับดึงข้อมูลเพลงทั้งหมดของงานแสดงดนตรีนั้น ๆ	25
(2) API spec สำหรับสร้างเพลงในงานแสดงดนตรีนั้น ๆ	26
(3) API spec สำหรับดึงข้อมูลรายละเอียดเพลงของงานแสดงดนตรีนั้น ๆ	26
(4) API spec สำหรับอัปเดตรายละเอียดเพลงของงานแสดงดนตรีนั้น ๆ	27
(5) API spec สำหรับลบเพลงของงานแสดงดนตรีนั้น ๆ	27
(6) API spec สำหรับการสลับลำดับคิวเพลงในงานแสดงดนตรีนั้น ๆ	28
(7) API spec สำหรับส่งแจ้งเตือนไปยัง performer ผ่าน Discord bot	28

3.1.1.5 API spec สำหรับการจัดการ User.....	29
(1) API spec สำหรับดึงข้อมูล User ทั้งหมด.....	29
3.1.1.6 API spec สำหรับการจัดการ performer.....	29
(1) API spec สำหรับดึงข้อมูล performer ที่เล่นในเพลนน์ ๆ ตามตำแหน่ง.....	30
(2) API spec สำหรับดึงข้อมูล performer ที่ยังไม่ถูกเลือกให้เล่นเพลนน์ ๆ ตามตำแหน่ง.....	30
(3) API spec สำหรับ assign performer ไปยังเพลนน์ ๆ	31
(4) API spec สำหรับ unassign performer ออกจากเพลนน์ ๆ	31
3.1.1.7 API spec สำหรับ system admin	32
(1) API spec สำหรับเพิ่มบัญชีผู้ใช้ใหม่	32
(2) API spec สำหรับโอนย้ายสิทธิ์ system admin ไปยังผู้ใช้อื่น ๆ	32
(3) API spec สำหรับบิดบัญชีผู้ใช้ที่จบการศึกษาแล้ว	33
3.2.1 แผนภาพกรณีศึกษา (Use case diagram).....	35
3.2.1.1 แสดงแผนภาพกรณีศึกษา (Use case diagram)	35
3.6.1 หน้าเข้าสู่ระบบ	87
3.6.2 หน้า Main menu สำหรับ Performer	88
3.6.3 หน้า Event สำหรับ Performer & Staff & System admin	90
3.6.4 หน้ารายละเอียด Event สำหรับ Performer & Staff & System admin	90
3.6.5 หน้ารายชื่อเพลนของ Performer.....	91
3.6.6 หน้ารายชื่อเพลนของ Performer.....	92
3.6.7 หน้ารายละเอียดเพลนของ Performer & Staff & System admin	93
3.6.8 หน้ารายชื่อนักร้องนักดนตรี Performer & Staff & System admin.....	94
3.6.9 หน้า My Event ของ Performer	95
3.6.10 หน้า Profile สำหรับ User ทุกประเภท	96

3.6.11	หน้า Main Menu ของ System admin.....	97
3.6.12	หน้า Transfer system admin privileges ของ System admin	99
3.6.13	หน้า account config ของ System admin	100
3.6.14	หน้ารวม Event สำหรับ Backstage.....	101
3.6.15	หน้าฟอร์มสร้าง Event ของ Backstage.....	102
3.6.16	หน้ารายละเอียด Event ของ Backstage	103
3.6.17	หน้าแก้ไข Event ของ Backstage	104
3.6.18	หน้ารายชื่อเพลงของ Backstage	105
3.6.19	หน้าฟอร์มการสร้างเพลงของ Backstage	106
3.6.20	หน้ารายละเอียดแต่ละเพลงของ Backstage.....	107
3.6.21	หน้า Assign Performer ของ Backstage.....	108
3.6.22	หน้าแก้ไขเพลง ของ Backstage	109
3.6.23	หน้า Run event ของ Backstage.....	110
3.6.24	หน้า Run event ของ Performer & Staff & System admin.....	111
3.6.25	หน้าแจ้งเตือน Performer.....	112
3.6.26	หน้ายกเลิกการแสดง ของ Backstage	113
3.6.27	หน้าลับเพลงใน queue.....	114
5.1.1	ผลการประเมินความสะอาดในการใช้งาน	121
5.1.2	ผลการประเมินความรวดเร็วของระบบ.....	122
5.1.3	ผลการประเมินความสะอาดสบายนในการเข้าถึงข้อมูลที่ต้องการ	122
5.1.4	ผลการประเมินความสะอาดสบายนในการสร้าง, แก้ไข, ลงงานแสดงดนตรี	123
5.1.5	ผลการประเมินการสร้างและแก้ไข รายละเอียดเพลง.....	123
5.1.6	ผลการประเมินความสะอาดสบายนในการสร้าง Lineup Performer	124
5.1.7	ผลการประเมินความสะอาดสบายนในการจัดการสมาชิกชุมชนในระบบ	124

5.1.8	ผลการประเมินความสอดคล้องของภาระทางกายภาพจัดการสมาชิกภายใน Discord server ...	125
5.1.9	ผลการประเมินข้อเสนอแนะเพิ่มเติม	125
5.1.10	จำนวนผู้กรอกแบบฟอร์มและฝ่ายในชุมชน	125

รายการสัญลักษณ์และคำย่อ

สัญลักษณ์/คำย่อ	คำเต็ม/คำจำกัดความ
SCI BAND	ชุมนุมดนตรีคณาจารย์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต
Performer	นักร้องหรือนักดนตรีทั้งหมดภายใน SCI BAND
Staff	ฝ่ายอำนวยความสะดวกของ SCI BAND ฝ่าย Staff ของ SCI BAND มีหน้าที่ดูแลและ จัดการงานแสดงดนตรี
อ้างอิงเพลง	ลิงก์ YouTube ไปยังเพลงนั้น ๆ ใช้สำหรับ อ้างอิงถึงเวอร์ชันของเพลงที่ต้องการใช้แสดง

(14)

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ชุมนุม SCI BAND คือชุมนุมดนตรีของคณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต ที่ก่อตั้งขึ้นเพื่อเป็นพื้นที่แสดงความสามารถด้านดนตรีของนักศึกษาภายในคณะเพื่อส่งเสริมความคิดสร้างสรรค์และการทำงานเป็นทีมโดยภายในชุมนุมนี้มีโครงการสร้างตั้งนี้

1. Staff คือฝ่ายที่ดูแลอำนวยความสะดวกและดำเนินงานต่าง ๆ ภายในชุมนุมโดยแบ่งฝ่ายได้ดังนี้ ฝ่ายเอกสาร, ฝ่ายสถานที่, ฝ่ายพัสดุ, ฝ่ายสวัสดิการ, ฝ่ายเทคนิค, ฝ่ายประชาสัมพันธ์, ฝ่าย Backstage และฝ่ายประสานงาน
2. Performer คือสมาชิกที่เป็นนักร้องหรือนักดนตรีโดยแบ่งสมาชิกได้เป็นดังนี้ นักร้อง, มือกีตาร์, มือเบส, มือกลอง, มือคีย์บอร์ด, มือ percussion และอื่น ๆ โดยในแต่ละฝ่ายนั้นจะสื่อสารและมีกลุ่มไลน์ในฝ่ายของตัวเองรวมไปถึงกลุ่มไลน์ใหญ่

และภายในชุมนุมนี้มีขั้นตอนการดำเนินงานในการจัดงานดนตรีดังนี้

1. เมื่อมีงานดนตรีเข้ามาฝ่าย Staff จะเปิดรับ Performer ผ่านทางกลุ่มไลน์ใหญ่
2. Performer ที่ต้องการร่วมแสดงทำการลงทะเบียนในกลุ่ม
3. Backstage ทำการสร้างกลุ่มไลน์สำหรับงานนั้น ๆ และมีการนัดหมายจัดการประชุมเพื่อพูดคุยเรื่องเพลงที่ต้องการเล่นและแบ่งบทบาทหน้าที่ในแต่ละเพลงในงานนั้น ๆ
4. Backstage ให้ Performer กรอกวันและเวลาไว้เพื่อนำข้อมูลไปจัดเวลาซ้อม
5. Performer ทำการซ้อมตามตารางที่กำหนด
6. Staff ฝ่ายต่าง ๆ ทำหน้าที่ที่เกี่ยวข้องจนถึงวันแสดงจริง

โดยจากขั้นตอนการดำเนินงานที่กล่าวมานั้นทำให้พบกับปัญหาดังนี้

- จำนวนสมาชิก Performer ที่มีปริมาณมากจึงต้องมีการสลับตัวกันในแต่ละเพลงอยู่ตลอดเวลาคนนั้นทำให้เกิดความสับสนถึงลำดับการแสดงของตนเองในบางครั้ง ทำให้การแสดงนั้นไม่เป็นไปอย่างราบรื่นเท่าที่ควร
- จำนวน LINE groups มีจำนวนมากทำให้เกิดความสับสนและข้อมูลสำคัญบางอย่างอาจตกหล่นหรือหมดอายุได้ รวมถึงการที่ Staff ต้องคอยตรวจสอบว่า สมาชิกใหม่ของ SCI BAND แต่ละคนนั้นได้เข้าร่วมกลุ่มไลน์ครบถ้วนแล้วหรือไม่

1.2 วัตถุประสงค์

โครงการนี้มีเป้าหมายเพื่อแก้ปัญหาและพัฒนาประสิทธิภาพการทำงานของชุมชน SCI BAND ทั้งด้านการสื่อสารและการจัดการสมาชิก Performer เพื่อให้การแสดงงานดนตรีเป็นไปได้อย่างราบรื่น สร้างความประทับใจให้แก่คุณดู เพื่อให้บรรลุเป้าหมายดังกล่าวจึงกำหนดวัตถุประสงค์ของโครงการต่อไปนี้

- พัฒนาระบบจัดการสมาชิกและลำดับการแสดง เพื่อจัดการข้อมูล Performer ได้อย่างมีระบบ ลดความสับสนในการสลับตัวระหว่างการแสดง และทำให้ทุกคนทราบถึงลำดับการแสดงได้อย่างชัดเจน
- ปรับปรุงช่องทางการสื่อสารภายในชุมชน เพื่อลดความซับซ้อนของการสื่อสารที่เกิดจากการใช้กลุ่มไลน์จำนวนมาก ป้องกันการตกหล่นของข้อมูลสำคัญ และส่งเสริมการประสานงานที่มีประสิทธิภาพ
- เสริมประสิทธิภาพการแสดงดนตรี เพื่อให้การแสดงดนตรีของชุมชน SCI BAND ดำเนินไปอย่างราบรื่น ลดปัญหาที่เกิดจากความไม่พร้อมหรือการจัดการที่ขาดความแม่นยำ พัฒนาระบบทดตามงานดนตรี เพื่อให้ Staff และ Performer สามารถติดตามความคืบหน้าและรายละเอียดของงานดนตรีได้แบบเรียลไทม์ ลดความเสี่ยงในการพลาดข้อมูลสำคัญ

1.3 ขอบเขตของโครงการ

ขอบเขตของโครงการนี้ครอบคลุมการพัฒนาระบบในรูปแบบโมบายแอปพลิเคชัน ด้วย React Native เพื่อใช้ในการจัดการงานแสดงดนตรีซึ่งรวมไปถึงการสร้างงานแสดงดนตรี การสร้างคิวเพลง การอุบหมายเพลงให้กับ Performer โดยมีระบบแจ้งเตือนไปยังบัญชี Discord ในรั้ว

งานจริงเพื่อเป็นการย้ำเตือน Performer ถึงลำดับการแสดงของตนเอง โดย Discord เป็นแพลตฟอร์มสำหรับการสื่อสารผ่านทางข้อความและเสียง มีระบบการจัดการข้อความและการจัดการสิทธิ์การเข้าถึงช่องทางสื่อสารแต่ละช่องได้อย่างอิสระมีประสิทธิภาพเจิงสามารถช่วยแก้ปัญหาอยู่ LINE ที่เยอะและจะจัดกระจาดซึ่งทำให้ข้อมูลสำคัญอาจถูกหละเลยได้ และผู้ใช้สามารถดูรายละเอียดงานแสดงต่าง ๆ เช่นวันเวลาในการแสดง, Dress code รวมไปถึงรายละเอียดในแต่ละเพลง เช่น มี Performer คนไหนร่วมเล่นเพลงในเพลงนั้น ๆ บ้าง, ลำดับของเพลง, คีย์เพลง, อังกฤษเพลง โดยมีการจัดเก็บข้อมูลต่าง ๆ ด้วย PostgreSQL ซึ่งเป็นฐานข้อมูลเชิงสัมพันธ์ และประมวลผลข้อมูลต่าง ๆ ด้วย NestJS ซึ่งเป็น Framework สำหรับสร้าง server-side applications และระบบสามารถอ่านข้อมูล Google Forms ซึ่งใช้สำหรับการสมัครเข้าชมนิทรรศการ SCI BAND ผ่านทาง Google Sheets เข้าสู่ฐานข้อมูล PostgreSQL เพื่อเป็นข้อมูลบัญชีผู้ใช้สำหรับใช้งานระบบและยังเป็นการจำกัดการเข้าใช้งานระบบให้มีแค่สมาชิกชุมชนสามารถใช้งานได้เท่านั้น รวมไปถึงหากสมาชิกชุมชนได้จบการศึกษาแล้วนั้นตัวระบบมีไฟเขียวในการปิดบัญชีผู้ใช้งานส่งผลให้สมาชิกคนนั้น ๆ ไม่สามารถเข้าใช้งานระบบได้อีกต่อไป

1.4 ประโยชน์ของโครงงาน

1. SCI BAND ได้ระบบการจัดการงานแสดงที่มีประสิทธิภาพ ทำให้การแสดงเป็นไปได้อย่างราบรื่น มีการติดตามอย่างต่อเนื่อง
2. Performer และ Staff สามารถดูคิวการแสดงได้เพื่อเป็นการเตรียมตัวล่วงหน้า ก่อนถึงวันจริง
3. Performer สามารถดูข้อมูลสำคัญที่จำเป็นต่อการฝึกซ้อมได้ เช่น คีย์เพลง, Performer คนอื่นที่ร่วมเล่นในเพลงนั้น ๆ เพื่อเป็นการปรับตัวเข้าหากันและกัน
4. คนจ้างงานมีความมั่นใจและพึงพอใจในการจ้างงานชุมชน SCI BAND ว่างานแสดงดูน่าสนใจ จะอกรายงานต่อไป
5. คนดูรู้สึกสนุกและมีอารมณ์ร่วมไปกับงานมากยิ่งขึ้นเนื่องจากงานแสดงเป็นไปได้อย่างราบรื่น ส่งผลให้ SCI BAND มีชื่อเสียงมากขึ้น

1.5 ข้อจำกัดของโครงงาน

1. ผู้ใช้งานจะต้องมีบัญชี Discord เพื่อใช้เข้าสู่ระบบ

2. ระบบรองรับเพียงขนาดหน้าจอ 412×915 dp ซึ่งเป็นขนาดมาตรฐานสำหรับมือถือทั่วไป

บทที่ 2

วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง

2.1.1 ทฤษฎีการสื่อสารในองค์กร

Shannon & Weaver. (1949) การสื่อสารที่มีประสิทธิภาพเป็นหัวใจสำคัญของ ชุมนุมที่มีสมาชิกจำนวนมาก โดยมีองค์ประกอบหลักสำคัญ ได้แก่

1. Source (ผู้ส่งสาร): บุคคลหรือหน่วยงานที่เป็นต้นกำเนิดของข้อมูล เช่น ผู้บริหารหรือ Staff ของ SCI BAND
2. Message (ข้อความ): เนื้อหาของข้อมูลที่ต้องการสื่อ เช่น ตารางกิจกรรม หรือข้อกำหนด สำหรับ Performer
3. Channel (ช่องทาง): สื่อหรือแพลตฟอร์มที่ใช้ในการส่งข้อมูล เช่น Discord, Line, หรือการ ประชุมแบบตัวต่อตัว
4. Receiver (ผู้รับสาร): บุคคลหรือกลุ่มที่เป็นเป้าหมายในการรับข้อมูล เช่น สมาชิก Performer หรือ Staff
5. Noise (สัญญาณรบกวน): ปัจจัยที่ขัดขวางหรือบิดเบือนข้อมูล เช่น ข้อความที่ไม่ซัดเจน ข้อมูลสูญหาย การสื่อสารที่ซับซ้อนเกินไป
6. Feedback (ผลสะท้อนกลับ): การตอบสนองของผู้รับสารต่อผู้ส่ง ซึ่งเป็นตัววัดความสำเร็จ ของกระบวนการสื่อสาร

Shannon and Weaver's Model of Communication (1949) อธิบายว่าปัญหา ในกระบวนการสื่อสาร เช่น ความซับซ้อนหรือข้อมูลสูญหาย (Noise) อาจทำให้ประสิทธิภาพของ องค์กรลดลง ซึ่งจากการเก็บข้อมูลใน SCI BAND ณ ปัจจุบันพบถึงปัญหาด้าน Noise และ Channel เป็นอย่างมาก ซึ่งจากการสื่อสารผ่านไลน์นั้นอาจทำให้ข้อมูลสำคัญบางอย่างตกหล่นจากการเลื่อนของข้อความ หรือหมดอายุได้ เช่น ข้อมูลกำหนดการของงานแสดงนั้น ๆ หรือไฟล์เสียงที่ได้มีการบันทึกไว้ในแต่ละ ครั้งหลังการซ้อมซึ่งใช้ในการให้ Performer ทราบและซ้อมในสิ่งที่ยังทำไม่ดีในครั้งนั้น ๆ รวมไปถึง กลุ่มไลน์ที่มีจำนวนเยอะซึ่งสร้างความสับสนและทำให้เกิดการละเลยข้อมูลได้ในบางครั้ง ในโภเจคนี้

จึงได้มีการทดลองย้ายมาใช้ Discord ซึ่งเป็นแพลตฟอร์มสำหรับสื่อสารผ่านข้อความและเสียงที่มีประสิทธิภาพมากกว่า เพื่อเป็นการแก้ปัญหา noise และ channel ที่เกิดขึ้นจากการใช้ LINE

2.1.2 Discord

Discord เป็นซอฟต์แวร์ที่ถูกออกแบบมาเพื่อให้ผู้ใช้งานสามารถสื่อสารกันได้ผ่านทางข้อความและเสียง โดยได้รับความนิยมในกลุ่มผู้ใช้งานทั่วไปและกลุ่มชุมชนออนไลน์ที่ต้องการสื่อสารและทำงานร่วมกันอย่างมีประสิทธิภาพ โดย Discord มีระบบจัดการเซิร์ฟเวอร์ที่ช่วยให้ผู้ดูแลสามารถสร้างกลุ่มย่อย (Channels) สำหรับการพูดคุยในหัวข้อต่าง ๆ และยังสามารถกำหนดสิทธิ์การเข้าถึงในแต่ละช่อง (Channel) เพื่อให้เฉพาะกลุ่มผู้ใช้งานที่กำหนดสามารถเข้าถึงข้อมูลและการสนทนาก็ได้ ซึ่งสามารถนำมาประยุกต์ใช้เพื่อปรับปรุงประสิทธิภาพการสื่อสารของ SCI BAND ได้ดียิ่งขึ้นโดยการแบ่งช่อง Channels ตามสิทธิ์ของผู้ใช้ทำให้มีผู้ใช้ได้ทำการเข้าร่วมเซิร์ฟเวอร์และระบบได้มอบสิทธิ์ให้กับผู้ใช้แล้วนั้น ผู้ใช้จะถูกเพิ่มไปยัง Channels ที่เกี่ยวข้องได้โดยอัตโนมัติ ลดปัญหาลึงกering ที่ Staff ไม่แน่ใจว่าสมาชิกใหม่ของ SCI BAND นั้นได้ทำการเข้าร่วมกลุ่มออนไลน์ครบถ้วนแล้วหรือไม่ และ Discord นั้นสามารถเก็บไฟล์ได้โดยไม่มีวันหมดอายุ และการตึงข้อความหรือไฟล์สำคัญที่สามารถทำได้ง่าย รวมถึง Discord นั้นมีส่วนของ API และ Discord bot ซึ่งสามารถใช้ในการเชื่อมต่อกับซอฟต์แวร์อื่นๆ ทำให้มีประสิทธิภาพในการสื่อสารที่มากกว่าใน SCI BAND ภายใต้การดำเนินการทดลองใช้ Discord เป็นช่องทางสื่อสารหลักสำหรับ SCI BAND ภายในโปรเจคนี้

2.1.3 Discord API

Discord API เป็นเครื่องมือที่ช่วยให้นักพัฒนาสามารถเชื่อมต่อและนำข้อมูลจาก Discord มาใช้ในแอปพลิเคชันต่าง ๆ ได้ โดยในโปรเจคนี้มีการใช้งานส่วนของ Authentication API ซึ่งช่วยในการยืนยันตัวตนของผู้ใช้งานเพื่อเข้าสู่ระบบ เมื่อผู้ใช้ทำการลงชื่อเข้าใช้ผ่าน Discord API ระบบจะสามารถดึงข้อมูลที่จำเป็นเพื่อยืนยันตัวตนและช่วยให้โมบายแอปพลิเคชันนี้เชื่อมต่อกับ Discord ได้

2.1.4 Discord bot

Discord bot เป็นโปรแกรมอัตโนมัติที่ใช้บนแพลตฟอร์ม Discord ซึ่งสามารถทำหน้าที่ต่าง ๆ ตามที่ได้รับการออกแบบและการกำหนดสิทธิ์มา เช่น

- การจัดการสิทธิ์การเข้าถึงช่องทาง (Channel)
- การส่งข้อความอัตโนมัติถึงผู้ใช้ที่อยู่ภายใต้ชื่อในเซิร์ฟเวอร์เดียวกัน
- การจัดการเพิ่มลบสมาชิกในเซิร์ฟเวอร์และ channel
- การกำหนดบทบาทให้กับผู้ใช้ภายใต้ชื่อในเซิร์ฟเวอร์เพื่อใช้ในการกำหนดสิทธิ์การเข้าถึง channel หรือการจัดการเซิร์ฟเวอร์
- การดำเนินการต่าง ๆ ตามคำสั่งที่ผู้ใช้ให้ไว้ โดยบอทสามารถตั้งค่าการทำงานได้หลากหลาย เช่น การตรวจสอบข้อความ, การเล่นเพลง, การแจ้งเตือนข่าวสาร, และอื่น ๆ อีกมากมาย

โดยในปัจจุบันมีการใช้ Discord bot ใน การกำหนดบทบาทให้กับผู้ใช้เพื่อให้ผู้ใช้ถูกเพิ่มไปยัง Channel ที่ตนเองเกี่ยวข้องโดยอัตโนมัติ โดยจะเกิดขึ้นหลังจาก System admin ได้ทำการเพิ่มบัญชีผู้ใช้เข้าสู่ระบบแล้ว และใช้ Discord bot ในการส่งการแจ้งเตือนไปยังกล่องข้อความของผู้ใช้โดยตรงเมื่อใกล้ถึงวันกำหนดการแสดงแล้ว ทำให้ช่วยลดความผิดพลาดในกระบวนการแสดง

2.1.5 React Native

React Native คือ Framework สำหรับพัฒนา User interface ของโมบายแอปพลิเคชัน เขียนด้วยภาษา JavaScript หรือ TypeScript โดย React Native เป็น Single codebase framework ซึ่งช่วยให้ผู้พัฒนาสามารถเขียนโค้ดเพียงครั้งเดียวและนำไปใช้ได้ทั้งระบบ Android และ iOS รวมไปถึงเว็บไซต์

React Native ถูกพัฒนาโดยบริษัท Meta โดยมีจุดประสงค์หลักคือการลดความซับซ้อนในการพัฒนาโมบายแอปพลิเคชัน และเพิ่มความเร็วในการพัฒนา โดยมีคุณสมบัติเด่นดังนี้

1. Reusable component

React Native มีชุดของ Components ที่สามารถใช้งานซ้ำได้ เช่น View, Text, Image รวมไปถึง Reusable component ที่ผู้พัฒนาสร้างขึ้นเองได้ เช่น Button, Input field ที่ถูกออกแบบเฉพาะสำหรับแอปพลิเคชันนั้น ๆ ซึ่งทำให้ช่วยลดการเขียนโค้ดที่ซ้ำซ้อน ส่งผลให้การพัฒนาสะดวกรวดเร็ว และมีประสิทธิภาพยิ่งขึ้น

2. ประสิทธิภาพที่ดี

React Native ใช้ Native Modules และ Native Components ในการประมวลผล ซึ่งทำให้แอปพลิเคชันที่พัฒนาขึ้นมีประสิทธิภาพสูงใกล้เคียงกับแอปพลิเคชันแบบ Native

3. เรียนรู้ได้ง่าย

React Native นั้นใช้พื้นฐานจาก React.js และมี syntax ที่เข้าใจง่าย ซึ่งตัวผู้จัดทำมีความคุ้นเคยและเคยใช้ในการทำโปรเจคมาก่อนทำให้สามารถเรียนรู้และสร้าง User interface ได้รวดเร็ว

4. สนับสนุนโดย Meta

การที่ React Native ถูกพัฒนาโดย Meta และมีการใช้งานจริงในแอปพลิเคชันขนาดใหญ่ เช่น Facebook และ Instagram ช่วยให้มั่นใจในความเสถียรและประสิทธิภาพของ Framework นอกจากนี้ React Native ยังมีชุมชนผู้พัฒนาขนาดใหญ่ที่คอยให้คำแนะนำ และอัปเดตฟีเจอร์ใหม่ ๆ อย่างต่อเนื่อง การสนับสนุนนี้ยังรวมถึงการแก้ไขข้อผิดพลาด การปรับปรุงประสิทธิภาพ ซึ่งช่วยให้ผู้พัฒนามั่นใจในระยะยาวว่า Framework นี้จะคงมีความทันสมัยและพร้อมใช้งานกับเทคโนโลยีใหม่ ๆ

ด้วยคุณสมบัติเด่นดังกล่าว นั่นจึงมีการกำหนดให้ใช้ React Native ด้วยภาษา TypeScript สำหรับการพัฒนา User interface ในโปรเจคนี้

React Native. (n.d.). Official React Native documentation. <https://reactnative.dev>

2.1.6 NestJS

NestJS คือ Framework สำหรับการพัฒนา Server-side applications ที่ได้รับความนิยมในปัจจุบัน ถูกออกแบบมาเพื่อรองรับแอปพลิเคชันขนาดใหญ่ โดยพัฒนาบนพื้นฐานของ Express.js ซึ่งเป็น Framework สำหรับการพัฒนาระบบฝั่งหลังบ้าน นอกจากนี้ NestJS ยังถูกพัฒนาขึ้นเพื่อช่วยให้การเขียนโค้ดมีความเป็นระบบและมีโครงสร้างที่ชัดเจนมากกว่า Express.js ซึ่งส่งผลให้การพัฒนามีประสิทธิภาพสูงขึ้น โดย NestJS มีคุณสมบัติเด่นดังนี้

1. Modular Design

NestJS ใช้แนวคิดโมดูล (Module) เพื่อจัดระเบียบโค้ดในโปรเจค ช่วยให้สามารถแบ่งฟังก์ชันการทำงานออกเป็นส่วนย่อยที่ชัดเจน ตัวอย่างเช่น Authentication Module, User Module, หรือ Payment Module ทำให้ช่วยจัดกลุ่มระบบต่าง ๆ เช่น การจัดงานแสดง การจัดการคิวเพลง รวมไปถึงการแยกบทบาทของผู้ใช้งานได้มีประสิทธิภาพ

2. Command line interface

NestJS มี Command line interface ที่ช่วยสร้าง Boilerplate code ในแต่ละ Module ได้อย่างรวดเร็ว ซึ่งช่วยประหยัดเวลาในการเขียนโค้ดเดิมซ้ำ ๆ ในแต่ละ Module

3. สนับสนุน TypeScript อย่างสมบูรณ์

NestJS ถูกพัฒนาด้วย TypeScript ทำให้ได้รับประโยชน์จาก Type safety และการตรวจสอบข้อผิดพลาดตั้งแต่ขั้นตอนการเขียนโค้ด ช่วยให้การพัฒนามีความรัดกุมมากยิ่งขึ้น

4. การผสานรวมกับ Third-Party Libraries

NestJS สามารถทำงานร่วมกับ Library ต่าง ๆ ได้ง่าย เช่น Prisma ซึ่งเป็น ORM สำหรับการจัดการฐานข้อมูล

5. สนับสนุน WebSocket และ Real-Time Applications

NestJS มี Built-in support สำหรับ WebSocket ซึ่งหมายความว่ามีความสามารถในการทำงานแบบเรียลไทม์ เช่น การรับคิวเพลงขณะการแสดงซึ่งต้องมีการอัปเดตคิวเพลงไปยังผู้ใช้งานทุกคน

ด้วยคุณสมบัติเด่นดังกล่าวขึ้นจึงมีการกำหนดให้ใช้ NestJS สำหรับการพัฒนาระบบหลังบ้านในโปรเจคนี้เพื่อให้การพัฒนาระบบมีประสิทธิภาพมากที่สุด

NestJS. (n.d.). Official NestJS documentation. <https://docs.nestjs.com>

2.1.7 PostgreSQL

PostgreSQL คือระบบจัดการฐานข้อมูล หรือ Database Management System ซึ่งสามารถเก็บได้ทั้งข้อมูลเชิงสัมพันธ์และข้อมูลแบบไม่สัมพันธ์ โดย PostgreSQL คือ Open source ระดับธุรกิจที่มีความปลอดภัยและแม่นยำ มีความน่าเชื่อถือสูงทำให้เป็นทางเลือกที่นิยมในปัจจุบัน จึงมีความเหมาะสมกับองค์กรหรือธุรกิจขนาดเล็กที่มีความต้องการพัฒนาระบบโดยมีงบประมาณที่จำกัด ซึ่งตรงกับลักษณะของ SCI BAND ซึ่งเป็นองค์กรขนาดเล็กที่มีงบจำกัด โดยในโปรเจคนี้จะมีการเก็บข้อมูลต่าง ๆ เช่น ข้อมูลงานแสดง ข้อมูลเพลง ข้อมูลผู้ใช้ ข้อมูล lineup Performer ในแต่ละเพลง โดย lineup performer หมายถึงรายชื่อผู้เล่นในแต่ละเพลงโดยระบุว่า ใครรับหน้าที่เป็นนักร้อง มือกีตาร์ มือกลอง ฯลฯ ในเพลงนั้น ๆ ซึ่งข้อมูลแต่ละส่วนมีโครงสร้างข้อมูลที่ค่อนข้างตายตัว ไม่มีการเปลี่ยนแปลง จึงทำให้ PostgreSQL เป็นตัวเลือกที่เหมาะสมสำหรับการเก็บข้อมูลในรูปแบบเชิงสัมพันธ์ในโปรเจคนี้

2.1.8 Prisma

Prisma คือ library สำหรับการทำ ORM (Object relational mapping) ซึ่งเป็นเทคนิคหรือแนวคิดในการแปลงข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปแบบ Object oriented และแปลงข้อมูลในรูปแบบ Object oriented ให้อยู่ในรูปแบบข้อมูลเชิงสัมพันธ์ ซึ่งช่วยให้ผู้พัฒนาสามารถทำงานกับฐานข้อมูลในรูปแบบที่เข้าใจง่ายขึ้น สามารถจัดการตาราง และการเชื่อมโยงข้อมูล ได้ในลักษณะของวัตถุ (Objects) โดยไม่ต้องใช้ SQL ที่ซับซ้อน และ Prisma นั้นยังเป็นที่นิยมในการใช้งานโดยเฉพาะในโปรเจคขนาดใหญ่ที่มีโครงสร้างฐานข้อมูลเชิงสัมพันธ์ที่มีความซับซ้อน โดย Prisma มีคุณสมบัติเด่นดังนี้

1. การเขียนโค้ดที่สะอาดและปลอดภัย

Prisma ช่วยให้การเขียนโค้ดง่ายขึ้น โดยไม่จำเป็นต้องเขียน SQL query โดยตรง ซึ่งช่วยลดความซับซ้อนในการติดต่อกับฐานข้อมูล และช่วยให้โค้ดดูสะอาดรวมไปถึง Prisma ยังมีการป้องกันปัญหาความปลอดภัย เช่น SQL Injection โดยใช้ Parameterized queries หรือการแยกข้อมูลจากคำสั่ง SQL อย่างชัดเจน ตัวอย่างเช่น เมื่อรับข้อมูลจากผู้ใช้งาน Prisma จะจัดการข้อมูลเหล่านี้ให้อยู่ในรูปแบบที่ปลอดภัยก่อนส่งคำสั่งไปยังฐานข้อมูล ซึ่งช่วยให้มั่นใจได้ว่าข้อมูลที่รับเข้ามาจะไม่ถูกใช้ในลักษณะที่เป็นภัยคุกคามต่อระบบ

2. การตรวจสอบข้อผิดพลาดที่ดีขึ้น

Prisma ใช้ TypeScript ซึ่งช่วยให้สามารถตรวจสอบข้อผิดพลาดได้ในระหว่างการพัฒนา ทำให้การจัดการกับข้อมูลและความสัมพันธ์ในฐานข้อมูลมีความถูกต้องและแม่นยำ

3. การจัดการฐานข้อมูลที่มีประสิทธิภาพ

Prisma สามารถทำงานร่วมกับฐานข้อมูลได้อย่างมีประสิทธิภาพ โดยสามารถจัดการการเข้ามายังข้อมูลและทำการ Query ได้อย่างรวดเร็ว และสามารถปรับแต่งการทำงานของฐานข้อมูลให้เหมาะสมกับความต้องการของแอปพลิเคชันได้อย่างยืดหยุ่น

4. การใช้งานที่ง่ายสำหรับนักพัฒนา

Prisma ช่วยให้การจัดการข้อมูลในฐานข้อมูลสะดวกขึ้นสำหรับนักพัฒนา โดยไม่ต้องกังวลเกี่ยวกับรายละเอียดการเข้ามายังฐานข้อมูลหรือการทำ Query ที่ซับซ้อน

นอกจากคุณสมบัติหลักที่กล่าวมาแล้ว Prisma ยังมีเครื่องมือสำคัญที่ช่วยสนับสนุนการพัฒนาและการจัดการฐานข้อมูลให้มีประสิทธิภาพมากยิ่งขึ้น ได้แก่ Prisma Client, Prisma Migrate และ Prisma Studio ซึ่งแต่ละเครื่องมือมีบทบาทที่แตกต่างกันและช่วยให้นักพัฒนาสามารถทำงานกับฐานข้อมูลได้อย่างสะดวกและเป็นระบบมากขึ้น ดังนี้

1. Prisma Client

Prisma Client เป็น client library ที่ใช้สำหรับเข้าถึงฐานข้อมูลผ่านโค้ด โดยนักพัฒนาสามารถใช้ Prisma Client เพื่อทำการ query, สร้าง, แก้ไข, หรือlobข้อมูลจากฐานข้อมูลได้ในลักษณะของการเรียกใช้เมธอดในภาษา TypeScript หรือ JavaScript แทนการเขียน SQL โดยตรง ซึ่งช่วยให้โค้ดอ่านง่าย ปลอดภัย และลดโอกาสในการเกิดข้อผิดพลาด โดย Prisma Client จะถูกสร้างขึ้นตามโมเดลที่นักพัฒนากำหนดใน schema ของ Prisma ซึ่งหมายความว่าเมธอดและชนิดข้อมูลที่ใช้ในโค้ดจะสอดคล้องกับโครงสร้างของฐานข้อมูลอย่างสมบูรณ์แบบ

2. Prisma Migrate

Prisma Migrate เป็นเครื่องมือสำหรับจัดการการเปลี่ยนแปลงของโครงสร้างฐานข้อมูล (Database schema migrations) โดยช่วยให้สามารถเพิ่ม แก้ไข หรือลบตารางและ

คอลัมน์ในฐานข้อมูลได้อย่างปลอดภัยและเป็นระบบ โดย Prisma Migrate จะสร้างไฟล์ migration อัตโนมัติจากการเปลี่ยนแปลงในไฟล์ schema ซึ่งสามารถนำไปใช้ในการควบคุมเวอร์ชันของฐานข้อมูล และทำให้ทีมพัฒนาทำงานร่วมกันได้อย่างราบรื่น อีกทั้งยังสามารถย้อนกลับการเปลี่ยนแปลงได้ในกรณีที่เกิดปัญหาในการปรับ schema

3. Prisma Studio

Prisma Studio เป็นเครื่องมือแบบกราฟิก (GUI) ที่ช่วยให้นักพัฒนาสามารถดูและจัดการข้อมูลภายในฐานข้อมูลได้อย่างสะดวกผ่านทางเบราว์เซอร์ โดยไม่ต้องเขียนโค้ดหรือ SQL ได้ ๆ ซึ่งเหมาะสมสำหรับการดูตัวอย่างข้อมูล ทดสอบการเปลี่ยนแปลง หรือตรวจสอบความถูกต้องของข้อมูลระหว่างการพัฒนา Prisma Studio ช่วยให้เห็นข้อมูลได้ชัดเจนและสามารถแก้ไขได้แบบเรียลไทม์ ทำให้ขั้นตอนการพัฒนาและดีบักเป็นไปอย่างราบรื่นและแม่นยำมากขึ้น

ด้วยคุณสมบัติเด่นที่กล่าวมา จึงกำหนดให้มีการใช้งาน Prisma ในการจัดการฐานข้อมูลเพื่อเพิ่มประสิทธิภาพ และความรวดเร็วในการพัฒนาและความปลอดภัยในการเข้าถึงฐานข้อมูล

Prisma. (n.d.). Official Prisma documentation. <https://www.prisma.io/docs>

2.2 แอปพลิเคชันต้นแบบ

2.2.1 Band Mule

โมบายแอปพลิเคชันสำหรับการจัดการวงโดยมีผู้ใช้สามารถเข้าสู่ระบบและทำการสร้างวง ดนตรีพร้อมกับเชิญสมาชิกเข้าสู่วง และภายในวงมีระบบในการแข่ง, การลงตารางปฏิทิน, การลงตารางว่างของสมาชิก, การสร้าง set lists สำหรับเพลง, การสร้างเพลง หลังจากการได้ทดลองใช้งานแอปพลิเคชัน Band Mule เพื่อศึกษารูปแบบและฟีเจอร์ต่าง ๆ ที่เกี่ยวข้องกับการจัดการวงดนตรี หลังจากได้ทดลองใช้งานแอปพลิเคชัน Band Mule เพื่อศึกษารูปแบบและฟีเจอร์ต่าง ๆ ที่เกี่ยวข้อง กับการจัดการวงดนตรีพบว่าแอปมีความสามารถพื้นฐานในการจัดการวงได้ดี อย่างไรก็ตาม Band Mule ยังขาดฟีเจอร์ที่เกี่ยวข้องกับการจัดการงานแสดงดนตรี ไม่สามารถแก้ไข set lists กับรายละเอียดเพลงได้ และไม่มีรายละเอียดของแต่ละเพลงในคิว

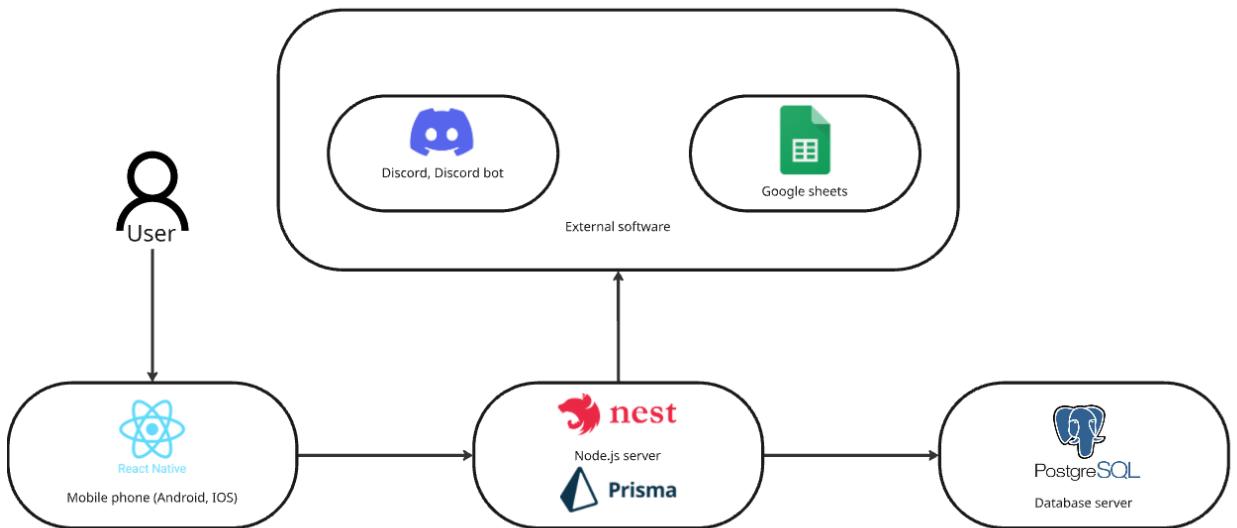
จากข้อจำกัดดังกล่าวจึงได้มีการพัฒนาระบบของตนเองให้สามารถจัดการระบบงานแสดงดนตรีได้โดยสามารถระบุรายละเอียดของงานแสดงดนตรี เช่น สถานที่ วันที่ เวลา รวมถึงสามารถกำหนดลำดับเพลงที่ใช้ได้ อีกทั้งยังเพิ่มความสามารถในการแสดง รายละเอียดของแต่ละเพลง เช่น นักดนตรีที่เล่นในเพลงนั้น เพื่อให้การจัดการวางแผนดนตรีเป็นระบบมากยิ่งขึ้น และตอบโจทย์วงดนตรีที่มีการเปลี่ยนแปลงผู้เล่นในแต่ละเพลงหรือมีความซับซ้อนในลำดับการแสดง

บทที่ 3

วิธีการวิจัย

3.1 ภาพรวมของโครงงาน

3.1.1 ภาพสถาปัตยกรรมระบบ (Architecture Diagram)



ระบบจัดการงานแสดงของชุมชนดนตรี SCI BAND ผ่านโมบายแอปพลิเคชัน! เป็นระบบที่ถูกออกแบบมาเพื่อเพิ่มความสะดวกในการจัดการงานแสดงของชุมชนดนตรี SCI BAND โดยแบ่งผู้ใช้งานออกเป็น 4 บทบาทหลัก ได้แก่

1. Performer – นักร้องและนักดนตรีที่มีส่วนร่วมในแต่ละเพลงของงานแสดง
2. Backstage – ทีมงานเบื้องหลังที่ดูแลการจัดการและการเตรียมความพร้อมของงาน
3. Staff – ทีมงานสนับสนุนที่มีหน้าที่เฉพาะกิจ
4. System Admin – ผู้ดูแลระบบที่รับผิดชอบการจัดการข้อมูลผู้ใช้งานและสิทธิ์ในการใช้งาน

การเข้าใช้งานระบบ

ระบบสามารถจำกัดการเข้าใช้งานให้เฉพาะสมาชิก SCI BAND ที่ยังไม่จบการศึกษา โดยดึงข้อมูลจาก Google Sheets ซึ่งเป็นแหล่งเก็บข้อมูลการสมัครจาก Google Forms เข้าสู่ฐานข้อมูล PostgreSQL โดยพิจารณ์จำกัดสิทธิ์เฉพาะ System admin เท่านั้น

ใน Google Sheets จะมีข้อมูลสำคัญ เช่น ชื่อ นามสกุล ตำแหน่ง และชื่อ Discord ID ซึ่งระบบจะใช้เพื่อตรวจสอบตัวตนเมื่อผู้ใช้ล็อกอินผ่านบัญชี Discord หากพบว่ามีข้อมูลในฐานข้อมูล ระบบจะอนุญาตให้เข้าสู่ระบบพร้อมสิทธิตามบทบาทของตน

หลังนำเข้าข้อมูลแล้ว ระบบจะเชื่อมต่อกับ Discord bot เพื่อเพิ่มผู้ใช้เข้าสู่ Discord channels ที่เกี่ยวข้องโดยอัตโนมัติ ลดภาระของ Staff ในการตรวจสอบว่าผู้ใช้ได้เข้าร่วมทุกกลุ่มหรือยัง

หากผู้ใช้จบการศึกษา System admin สามารถปิดบัญชีได้ ซึ่งจะทำให้ผู้ใช้ไม่สามารถเข้าสู่ระบบได้อีก และถูกลบออกจาก Discord server โดยอัตโนมัติ นอกจากนี้ System admin ยังสามารถโอนสิทธิ์ให้ผู้อื่นได้ในกรณีที่ต้องจบการศึกษาหรือหมดภาระ

การจัดการงานแสดงดนตรี

หลังจากเข้าสู่ระบบแล้วนั้น Backstage จะสามารถสร้าง, แก้ไข และลบงานแสดงดนตรีได้ โดยการสร้างงานดนตรีนั้น Backstage จะทำการกรอกชื่อ วันเวลา และรายละเอียดเพิ่มเติมต่าง ๆ ลงไป หลังกดยืนยันจะเป็นการบันทึกข้อมูลงานแสดงดนตรีเข้าสู่ระบบ ทำให้ผู้ใช้งานทุกประเภทสามารถเข้ามาดูรายละเอียดของงานแสดงดนตรีได้

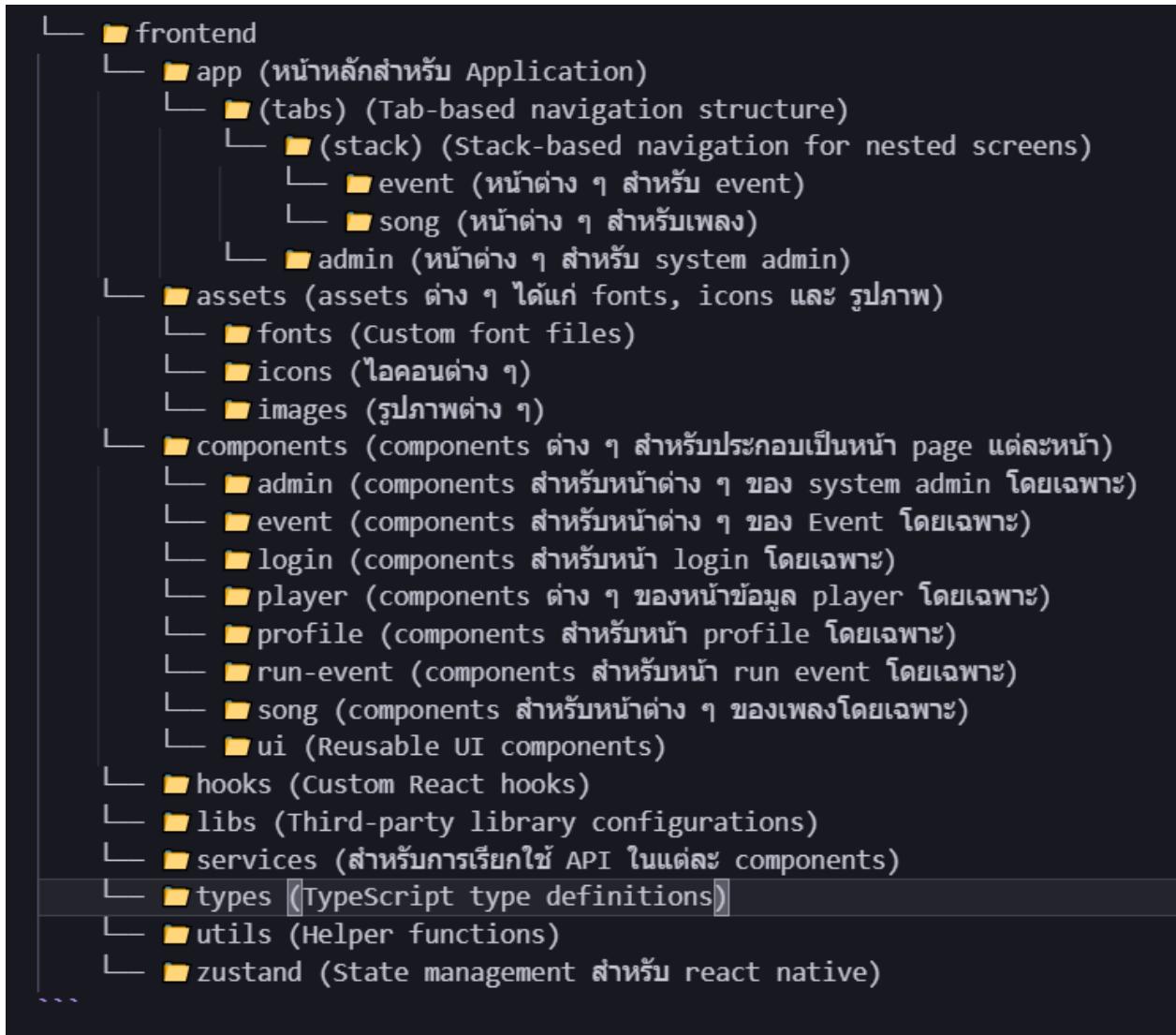
การสร้างคิวเพลง

เมื่อสร้างงานดนตรีเสร็จแล้วนั้น Backstage สามารถสร้างคิวเพลงภายในงานดนตรีนั้น ๆ ได้ โดยใส่รายละเอียดต่าง ๆ เช่น ชื่อเพลง คิวเพลง อ้างอิงเพลง (Reference) และสามารถสลับลำดับของเพลงได้อย่างอิสระ รวมไปถึงสามารถมอบหมายเพลงให้กับ Performer ที่มีส่วนเกี่ยวข้องกับเพลงนั้น ๆ โดยการเลือกจำนวน Performer ที่ต้องการในแต่ละตำแหน่งและทำการ Assign Performer ตามจำนวนที่ระบุไว้เพื่อเป็นข้อมูลให้แก่ชุมนุมและระบบสำหรับทำการแจ้งเตือนไปยัง Performer ในวันแสดงจริงได้

การแจ้งเตือน

ในเวลาแสดงจริงนั้น Backstage สามารถกดแจ้งเตือนไปยัง Performer ในเพลงคัดไปได้โดยตรงผ่านทางระบบ โดยข้อความแจ้งเตือนนั้นจะถูกส่งไปยังกล่องข้อความ Discord ของแต่ละ Performer โดยตรง โดย Backstage นั้นสามารถเลือกใส่ข้อความแจ้งเตือนเองได้ตามความเหมาะสม

3.1.1.1 โครงสร้าง folders frontend



3.1.1.2 คำอธิบาย file ต่าง ๆ สำหรับ frontend

```

└─ frontend
    └─ app
        └─ _layout.tsx (root layout file สำหรับ application)
        └─ (tabs)
            └─ _layout.tsx (layout file สำหรับ tab bars)
            └─ (stack)
                └─ _layout.tsx (layout file สำหรับ folder event และ song)
                    └─ event
                        └─ _layout.tsx (layout file สำหรับหน้า event)
                        └─ detail.tsx (หน้ารายละเอียด Event)
                        └─ index.tsx (หน้า Event ทั้งหมด)
                        └─ run.tsx (หน้า run event)
                    └─ song
                        └─ _layout.tsx (layout file สำหรับหน้า song)
                        └─ assign-player.tsx (หน้าสำหรับ assing player และดูรายละเอียด player)
                        └─ create.tsx (หน้าสำหรับสร้างเพลง)
                        └─ detail.tsx (หน้ารายละเอียดเพลง)
                        └─ index.tsx (หน้าคิวเพลงสำหรับ Event นั้น ๆ)
                └─ admin
                    └─ account-config.tsx (หน้า account config สำหรับ System admin)
                    └─ transfer-privilege.tsx (หน้าโอนย้ายสิทธิ์ System admin)
                └─ main-menu.tsx (หน้า Main menu)
                └─ profile.tsx (หน้า Profile)
                └─ index.tsx (entry file)
    └─ components
        └─ admin
            └─ activate-account.tsx (components สำหรับเปิดบัญชีใหม่)
            └─ deactivate-account.tsx (components สำหรับปิดบัญชี)
            └─ tranfer-admin.tsx (components สำหรับโอนย้ายสิทธิ์ System admin)
            └─ use-activate-account.ts (hook สำหรับเปิดบัญชี)
            └─ use-deactivate-account.ts (hook สำหรับปิดบัญชี)
            └─ use-transfer-admin.ts (hook สำหรับโอนย้ายสิทธิ์ System admin)
        └─ event
            └─ card.tsx (file card แต่ละ event ในหน้า event ทั้งหมด)
            └─ event-action-buttons.tsx (ปุ่มดูรายละเอียด event และไป run event)
            └─ form.tsx (form สำหรับสร้างหรืออัปเดต event)
            └─ modal.tsx (modal สำหรับโชว์ฟอร์มสร้าง event)
            └─ use-event-form.ts (hook สำหรับสร้างหรืออัปเดต event)
        └─ login
            └─ login.tsx (component สำหรับหน้า login)
            └─ use-login.ts (hook สำหรับ login)
        └─ player
            └─ assigned-player-list.tsx (component สำหรับหน้า assign player)
            └─ unassigned-player-list.tsx (component สำหรับหน้า unassign player)
            └─ use-assign-player.ts (hook สำหรับ assign player)
            └─ use-unassign-player.ts (hook สำหรับ unassign player)
        └─ profile
            └─ profile.tsx (component สำหรับหน้า profile)
        └─ run-event
            └─ header.tsx (component สำหรับ header หน้า run event)
            └─ noti-modal.tsx (modal component สำหรับแจ้งเตือน player )
            └─ song-queue-run.tsx (component สำหรับ queue เพลง)
            └─ use-noti.ts (hook สำหรับการแจ้งเตือน player)
        └─ song
            └─ card.tsx (file card แต่ละเพลงในหน้าคิวเพลง)
            └─ form.tsx (component สร้างและอัปเดตเพลง)
            └─ total-player-input.tsx (component สำหรับกรอกจำนวน player ที่ต้องการในแต่ละเพลง)
            └─ use-song-form.ts (hook สร้างและอัปเดตเพลง)
        └─ ui
            └─ background.tsx (component สำหรับ background ใน application)
            └─ button.tsx (component button)
            └─ form-controller.tsx (component form controller สำหรับ form)
            └─ main-menu-card.tsx (card component สำหรับ main menu )
            └─ text-input.tsx (component สำหรับ text input)

```

```
└── hooks
    └── use-socket-query.ts (hook สำหรับการเชื่อม socket ไปยัง backend)
    └── use-theme.ts (hook สำหรับเก็บค่าสีต่าง ๆ ใน application)
└── libs
    └── axios.ts (file config axios สำหรับการเรียก API)
└── services
    └── event.service.ts (file สำหรับเรียก API ที่เกี่ยวข้องกับ event)
    └── song.service..ts (file สำหรับเรียก API ที่เกี่ยวข้องกับเพลง)
    └── user.service.ts (file สำหรับเรียก API ที่เกี่ยวข้องกับ user)
└── types
    └── event.d.ts (file types ที่เกี่ยวข้องกับ event)
    └── song.d.ts (file types ที่เกี่ยวข้องกับเพลง)
    └── user.d.ts (file types ที่เกี่ยวข้องกับ user)
└── utils
    └── check-user-role.ts (help function สำหรับการเช็ค role ของ user)
└── zustand
    └── store.ts (global state management สำหรับเก็บข้อมูล eventId, songId และ player type)
```

3.1.1.3 API spec สำหรับงานแสดงดนตรี

Event

GET	/events	Get all the events	
POST	/events	Create a new event	
GET	/events/{eventId}	Get one event detail by eventId	
PUT	/events/{eventId}	Update event details	
DELETE	/events/{eventId}	Delete an event and its associated songs	
GET	/events/{eventId}/current-song	Get current song of running event	
PUT	/events/{eventId}/current-song	Update current song of running event	
PATCH	/events/{eventId}/end	End an event and mark it as completed	
PATCH	/events/{eventId}/start	Start an event, mark status as ONGOING and mark the first song as current song	

(1) API spec สำหรับดึงข้อมูลงานแสดงดนตรีทั้งหมด

GET /events Get all the events

Parameters

No parameters

Responses

Code	Description	Links
200	Retrieved all events successfully	No links
	Media type application/json Controls Accept header.	
	Example Value Schema	
	[{ "eventId": "f81ddfae-7dec-11d0-a765-00a0c91c6bf6", "eventName": "กุลลักษณ์", "eventDate": "2025-06-07T15:00:00.00Z", "endTime": "2025-06-07T15:04:03.00Z", "status": "UPCOMING" }]	
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
	Media type application/json Controls Accept header.	
	Example Value Schema	
	{ "message": "Missing authorization header, invalid access token or your discord account does not have access to this application" }	

(2) API spec สำหรับสร้างงานแสดงดนตรี

POST /events Create a new event

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "eventName": "งานฉลอง",
  "eventDate": "2025-06-07T15:04:25.279Z",
  "startTime": "2025-06-07T15:04:25.279Z",
  "endTime": "2025-06-07T15:04:25.279Z",
  "dressCode": "quintessential",
  "additionalDetails": "ตั๋วเข้าชมฟรี 14.00"
}
```

Responses

Code	Description	Links
201	Create a new event successfully	No links
400	Invalid payload	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - Backstage access required	No links
409	Event with this name already exists	No links

(3) API spec สำหรับดึงข้อมูลของงานแสดงดนตรีนั้น ๆ ด้วย event ID

GET /events/{eventId} Get one event detail by eventId

Parameters

Name	Description
eventId <small>* required</small>	eventId

Responses

Code	Description	Links
200	Retrieved one event detail by eventId successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links

Example Value | Schema

```
{
  "eventId": "f81d4fae-7dec-11db-a765-00a0c91e6bf6",
  "eventName": "คุณลุง",
  "eventDate": "2025-06-07T15:06:28.066Z",
  "startTime": "2025-06-07T15:06:28.066Z",
  "endTime": "2025-06-07T15:06:28.066Z",
  "dressCode": "casual",
  "status": "UPCOMING",
  "additionalDetails": "เริ่มเวลา 14.00 น."
}
```

Example Value | Schema

```
{
  "message": "Missing authorization header, invalid access token or your discord account does not have access to this application"
}
```

(4) API spec สำหรับอัปเดตรายละเอียดงานแสดงดนตรี

PUT /events/{eventId} Update event details

Parameters

Name	Description
eventId <small>* required</small>	eventId

Request body required

Example Value | Schema

```
{
  "eventName": "คุณลุง",
  "eventDate": "2025-06-07T15:06:54.012Z",
  "startTime": "2025-06-07T15:06:54.012Z",
  "endTime": "2025-06-07T15:06:54.012Z",
  "dressCode": "casual",
  "additionalDetails": "เริ่มเวลา 14.00"
}
```

Responses

Code	Description	Links
200	Update event detail by eventId successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - Backstage access required	No links

Example Value | Schema

```
{
  "message": "Missing authorization header, invalid access token or your discord account does not have access to this application"
}
```

Example Value | Schema

```
{
  "message": "Backstage access required"
}
```

(5) API spec สำหรับลบงานแสดงดนตรี

DELETE /events/{eventId} Delete an event and its associated songs

Parameters

Name	Description
eventId <small>* required</small>	eventId

Responses

Code	Description	Links
200	Delete event and its songs by eventId successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - Backstage access required	No links

Example Value | Schema

```
{
  "message": "Missing authorization header, invalid access token or your discord account does not have access to this application"
}
```

Example Value | Schema

```
{
  "message": "Backstage access required"
}
```

(6) API spec สำหรับดึงข้อมูลเพลงปัจจุบันของงานแสดงดนตรีนั้น ๆ

GET /events/{eventId}/current-song Get current song of running event

Parameters

Name	Description
eventId <small>* required</small>	eventId

Responses

Code	Description	Links
200	Retrieved current song of the running event successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links

Example Value | Schema

```
{
  "songId": "23836c79-9e48-4e1b-babf-443fef4c6412",
  "songName": "attention"
}
```

Example Value | Schema

```
{
  "message": "Missing authorization header, invalid access token or your discord account does not have access to this application"
}
```

(7) API spec สำหรับอัปเดตเพลงปัจจุบันของงานแสดงดนตรีนั้น ๆ

The screenshot shows the API specification for a **PUT** request to `/events/{eventId}/current-song`. The operation is described as "Update current song of running event".

Parameters:

Name	Description
eventId * required	eventid

Responses:

Code	Description	Links
200	Update current song of running event successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - Backstage access required	No links

Example values for the 200 response:

```
{
  "songId": "3383dc79-9e48-4e1b-babf-443fe4c6d412",
  "songName": "attention"
}
```

Example values for the 401 response:

```
{
  "message": "Missing authorization header, invalid access token or your discord account does not have access to this application"
}
```

Example values for the 403 response:

```
{
  "message": "Backstage access required"
}
```

(8) API spec สำหรับอัปเดตสถานะงานแสดงดนตรีนั้น ๆ เป็น COMPLETED หลังจบงานแสดงดนตรีแล้ว

The screenshot shows the API specification for a **PATCH** request to `/events/{eventId}/end`. The operation is described as "End an event and mark it as completed".

Parameters:

Name	Description
eventId * required	eventid

Responses:

Code	Description	Links
200	Update event status to COMPLETED after the event end	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - Backstage access required	No links

Example values for the 200 response:

```
{
  "message": "Event status updated to COMPLETED"
}
```

Example values for the 401 response:

```
{
  "message": "Missing authorization header, invalid access token or your discord account does not have access to this application"
}
```

Example values for the 403 response:

```
{
  "message": "Backstage access required"
}
```

(9) API spec สำหรับอัปเดตสถานะงานแสดงดนตรีนั้น ๆ เป็น ONGOING เมื่อถึงเวลาเริ่มแสดงดนตรี

PATCH /events/{eventId}/start Start an event, mark status as ONGOING and mark the first song as current song

Parameters

Name	Description
eventId <small>* required</small>	eventId
string	
(path)	

Responses

Code	Description	Links
200	Update event status to ONGOING	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
	Media type application/json	
	Example Value Schema	
	{ "message": "Missing authorization header, invalid access token or your discord account does not have access to this application" }	
403	Forbidden - Backstage access required	No links
	Media type application/json	
	Example Value Schema	
	{ "message": "Backstage access required" }	

3.1.1.4 API spec สำหรับการจัดการเพลง

Song

Method	Path	Description	Lock
GET	/events/{eventId}/songs	Get all songs in an specific event by eventId	🔒
POST	/events/{eventId}/songs	Create a new song in the event queue	🔓
GET	/events/{eventId}/songs/{songId}	Get one song details by songId	🔒
PUT	/events/{eventId}/songs/{songId}	Update song details	🔓
DELETE	/events/{eventId}/songs/{songId}	Delete a song from the event	🔒
PATCH	/events/{eventId}/songs/reorder	Reorder songs in the event queue	🔓
POST	/events/{eventId}/songs/{songId}/notification	Send notification about a song via Discord	🔓

(1) API spec สำหรับดึงข้อมูลเพลงทั้งหมดของงานแสดงดนตรีนั้น ๆ

GET /events/{eventId}/songs Get all songs in an specific event by eventId

Parameters

Name	Description
eventId * required	string (path)

Responses

Code	Description	Links
200	Get a song queue by eventId	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links

Code 200: Get a song queue by eventId

Media type: application/json

Example Value | Schema

```
[{"isAssigned": true, "songId": "2b232072-6b3c-4044-98e8-6c6655a2ff91", "songName": "attention", "songOrder": 1, "songKey": "C"}]
```

Code 401: Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions

Media type: application/json

Example Value | Schema

```
{"message": "Missing authorization header, invalid access token or your discord account does not have access to this application"}
```

(2) API spec สำหรับสร้างเพลงในงานแสดงดนตรีนั้น ๆ

The screenshot shows the API specification for the endpoint `/events/{eventId}/songs`. The method is `POST`. The description is "Create a new song in the event queue".

Parameters

Name	Description
<code>eventId</code> * required	eventid string (path)

Request body required

Media type: application/json

Example Value | Schema

```
{
  "songName": "Kittation",
  "songDescription": "za data แมวแมว",
  "songKey": "C",
  "songReference": "https://www.youtube.com/watch?v=zgLUxIhrzQ",
  "totalVocalist": "1",
  "totalDrummer": "1",
  "totalBassist": "1",
  "totalKeyboardist": "1",
  "totalExtra": "1",
  "totalPercussionist": "1"
}
```

Responses

Code	Description	Links
201	Create a new song successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
	Media type: application/json	
	Example Value Schema	
	{ "message": "Missing authorization header, invalid access token or your discord account does not have access to this application" }	
403	Forbidden - Backstage access required	No links
	Media type: application/json	
	Example Value Schema	
	{ "message": "Backstage access required" }	

(3) API spec สำหรับดึงข้อมูลรายละเอียดเพลงของงานแสดงดนตรีนั้น ๆ

The screenshot shows the API specification for the endpoint `/events/{eventId}/songs/{songId}`. The method is `GET`. The description is "Get one song details by songId".

Parameters

Name	Description
<code>eventId</code> * required	eventid string (path)
<code>songId</code> * required	songId string (path)

Responses

Code	Description	Links
200	Get a song detail by songId	No links
	Media type: application/json	
	Controls Accept header.	
	Example Value Schema	
	{ "songOrder": 1, "eventId": "2b232072-6b3c-4044-98e8-6c6655a2ff91", "songId": "3c232072-6b3c-4044-98e8-6c6655a2ff92", "songName": "Attention", "songDescription": "za data ลิว", "songKey": "C", "songReference": "https://youtube.com/watch?v=example", "totalVocalist": "2", "totalDrummer": "1", "totalBassist": "1", "totalKeyboardist": "1", "totalExtra": "1", "totalPercussionist": "1", "totalBassist": "1" }	
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
	Media type: application/json	
	Example Value Schema	
	{ "message": "Missing authorization header, invalid access token or your discord account does not have access to this application" }	

(4) API spec สำหรับอัปเดตรายละเอียดเพลงของงานแสดงดนตรีนั้น ๆ

The screenshot shows the API specification for the `PUT /events/{eventId}/songs/{songId}` endpoint. The endpoint is described as "Update song details".

Parameters:

Name	Description
songId * required	string (path)

Request body required

Media type: application/json

Example Value | Schema

```
{
  "songName": "attention",
  "songDescription": "so data wanna",
  "songKey": "C",
  "songRefLink": "https://www.youtube.com/watch?v=2gLUXhrz7Q",
  "totalGuitarist": "1",
  "totalDrummer": "1",
  "totalBassist": "1",
  "totalPercussionist": "1",
  "totalExtra": "1",
  "totalPercussionist": "1"
}
```

Responses:

Code	Description	Links
200	Update song detail successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - Backstage access required	No links

(5) API spec สำหรับลบเพลงของงานแสดงดนตรีนั้น ๆ

The screenshot shows the API specification for the `DELETE /events/{eventId}/songs/{songId}` endpoint. The endpoint is described as "Delete a song from the event".

Parameters:

Name	Description
eventId * required	string (path)
songId * required	string (path)

Responses:

Code	Description	Links
200	Delete a song by songId successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - Backstage access required	No links

(6) API spec สำหรับการสลับลำดับคิวเพลงในงานแสดงตอนทรีนนิ่ง ๆ

PATCH /events/{eventId}/songs/reorder Reorder songs in the event queue

Parameters

Name	Description
eventId <small>required</small>	eventid

Request body required

application/json

Example Value | Schema

```
{
  "songorder": [
    {
      "songId": "123e4567-e89b-12d3-a456-426614174000",
      "songorder": 1
    },
    {
      "songId": "123e4567-e89b-12d3-a456-426614174001",
      "songorder": 2
    }
  ]
}
```

Responses

Code	Description	Links
200	Reorder two songs successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - Backstage access required	No links

(7) API spec สำหรับส่งแจ้งเตือนไปยัง performer ผ่าน Discord bot

POST /events/{eventId}/songs/{songId}/notification Send notification about a song via Discord

Parameters

Name	Description
songId <small>required</small>	songId

Responses

Code	Description	Links
201	successfully send notification to discord dm via discord bot	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - Backstage access required	No links

3.1.1.5 API spec สำหรับการจัดการ User

User

GET /user Get all users

(1) API spec สำหรับดึงข้อมูล User ทั้งหมด

The screenshot shows the API specification for the User endpoint. It includes a header bar with 'User' and a URL 'GET /user Get all users'. Below this is a 'Parameters' section with 'No parameters' and a 'Responses' section. The 'Responses' section contains two entries: a successful 200 response with a JSON example showing user ID and full name, and an unauthorized 401 response with a message about missing authorization.

Code	Description	Links
200	retrieved user successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links

3.1.1.6 API spec สำหรับการจัดการ performer

Player

GET /songs/{songId}/player/assigned/{playerType} Get assigned players of a specific type for a song

GET /songs/{songId}/player/unassigned/{playerType} Get unassigned players of a specific type for a song

POST /songs/{songId}/player/assign Assign a player to a song

POST /songs/{songId}/player/unassign Unassign a player from a song

(1) API spec สำหรับดึงข้อมูล performer ที่เล่นในเพลงนั้น ๆ ตาม ตำแหน่ง

The screenshot shows the API specification for a GET request to `/songs/{songId}/player/assigned/{playerType}`. The request is described as "Get assigned players of a specific type for a song".

Parameters:

Name	Description
<code>songId</code> * required	string (path)
<code>playerType</code> * required	string (path)

Responses:

Code	Description	Links
200	get the assigned player in the song	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links

Example Value:

```
{
  "totalPlayer": 2,
  "currentPlayer": 1,
  "players": [
    {
      "userId": "2b232072-6b3c-4044-98e8-6c6655a2ff91",
      "firstName": "John",
      "lastName": "Doe",
      "nickName": "Johnny",
      "discordId": "123456789012345678",
      "discordUsername": "johndoe#1234",
      "isActive": true
    }
  ]
}
```

Example Value:

```
{
  "message": "Missing authorization header, invalid access token or your discord account does not have access to this application"
}
```

(2) API spec สำหรับดึงข้อมูล performer ที่ยังไม่ถูกเลือกให้เล่นเพลง นั้น ๆ ตามตำแหน่ง

The screenshot shows the API specification for a GET request to `/songs/{songId}/player/unassigned/{playerType}`. The request is described as "Get unassigned players of a specific type for a song".

Parameters:

Name	Description
<code>songId</code> * required	string (path)
<code>playerType</code> * required	string (path)

Responses:

Code	Description	Links
200	get the unassigned player in the song	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links

Example Value:

```
{
  "userId": "2b232072-6b3c-4044-98e8-6c6655a817bc",
  "firstName": "John",
  "lastName": "Doe",
  "nickName": "August",
  "discordId": "123456789012345678",
  "discordUsername": "xenon",
  "isActive": true
}
```

Example Value:

```
{
  "message": "Missing authorization header, invalid access token or your discord account does not have access to this application"
}
```

(3) API spec สำหรับ assign performer ไปยังเพลงนั้น ๆ

The screenshot shows the API specification for the `POST /songs/{songId}/player/assign` endpoint. The endpoint is described as "Assign a player to a song". There are no parameters listed. The request body is required and has a schema of `application/json`. An example value is provided:

```
{
  "playerType": "vocalist",
  "playerId": [
    "2c232072-6b3c-4044-98e8-6c6655a2ff91"
  ],
  "songId": "3c232072-6b3c-4044-98e8-6c6655a2ff92"
}
```

The responses section includes three entries:

- 201**: assign the player to a song. No links.
- 401**: Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions. Media type is `application/json`. Example value: `{"message": "Missing authorization header, invalid access token or your discord account does not have access to this application"}`. No links.
- 403**: Forbidden - Backstage access required. Media type is `application/json`. Example value: `{"message": "Backstage access required"}`. No links.

(4) API spec สำหรับ unassign performer ออกจากเพลงนั้น ๆ

The screenshot shows the API specification for the `POST /songs/{songId}/player/unassign` endpoint. The endpoint is described as "Unassign a player from a song". There are no parameters listed. The request body is required and has a schema of `application/json`. An example value is provided:

```
{
  "playerType": "vocalist",
  "playerId": [
    "2c232072-6b3c-4044-98e8-6c6655a2ff91"
  ],
  "songId": "3c232072-6b3c-4044-98e8-6c6655a2ff92"
}
```

The responses section includes three entries:

- 201**: unassign the player from a song. No links.
- 401**: Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions. Media type is `application/json`. Example value: `{"message": "Missing authorization header, invalid access token or your discord account does not have access to this application"}`. No links.
- 403**: Forbidden - Backstage access required. Media type is `application/json`. Example value: `{"message": "Backstage access required"}`. No links.

3.1.1.7 API spec สำหรับ system admin

The screenshot shows a list of API endpoints under the 'Admin' category:

- POST /admin/activate-user**: Activate new user accounts from Google Sheets data.
- PATCH /admin/transfer**: Transfer system admin privileges to another user.
- PATCH /admin/deactivate**: Deactivate user accounts.

(1) API spec สำหรับเพิ่มบัญชีผู้ใช้ใหม่

The screenshot shows the detailed API specification for the **POST /admin/activate-user** endpoint:

Parameters: No parameters

Responses:

Code	Description	Links
201	add new account by reading google sheets successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
	Media type: application/json	
	Example Value Schema	
	{ "message": "Missing authorization header, invalid access token or your discord account does not have access to this application" }	
403	Forbidden - System admin access required	No links
	Media type: application/json	
	Example Value Schema	
	{ "message": "System admin access required" }	

(2) API spec สำหรับโอนย้ายสิทธิ์ system admin ไปยังผู้ใช้อื่น ๆ

The screenshot shows the detailed API specification for the **PATCH /admin/transfer** endpoint:

Parameters: No parameters

Request body (required): application/json

Example Value | Schema

```
{
  "newAdminId": "e3e3ee51-f1a2-4a2c-9abd-4e40732bf3eb"
}
```

Responses:

Code	Description	Links
200	transfer system admin privilege successfully	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
	Media type: application/json	
	Example Value Schema	
	{ "message": "Missing authorization header, invalid access token or your discord account does not have access to this application" }	
403	Forbidden - System admin access required	No links
	Media type: application/json	
	Example Value Schema	
	{ "message": "System admin access required" }	

(3) API spec สำหรับปิดบัญชีผู้ใช้ที่จบการศึกษาแล้ว

The screenshot shows the API specification for the `PATCH /admin/deactivate` endpoint. The interface includes:

- Parameters:** None
- Responses:**

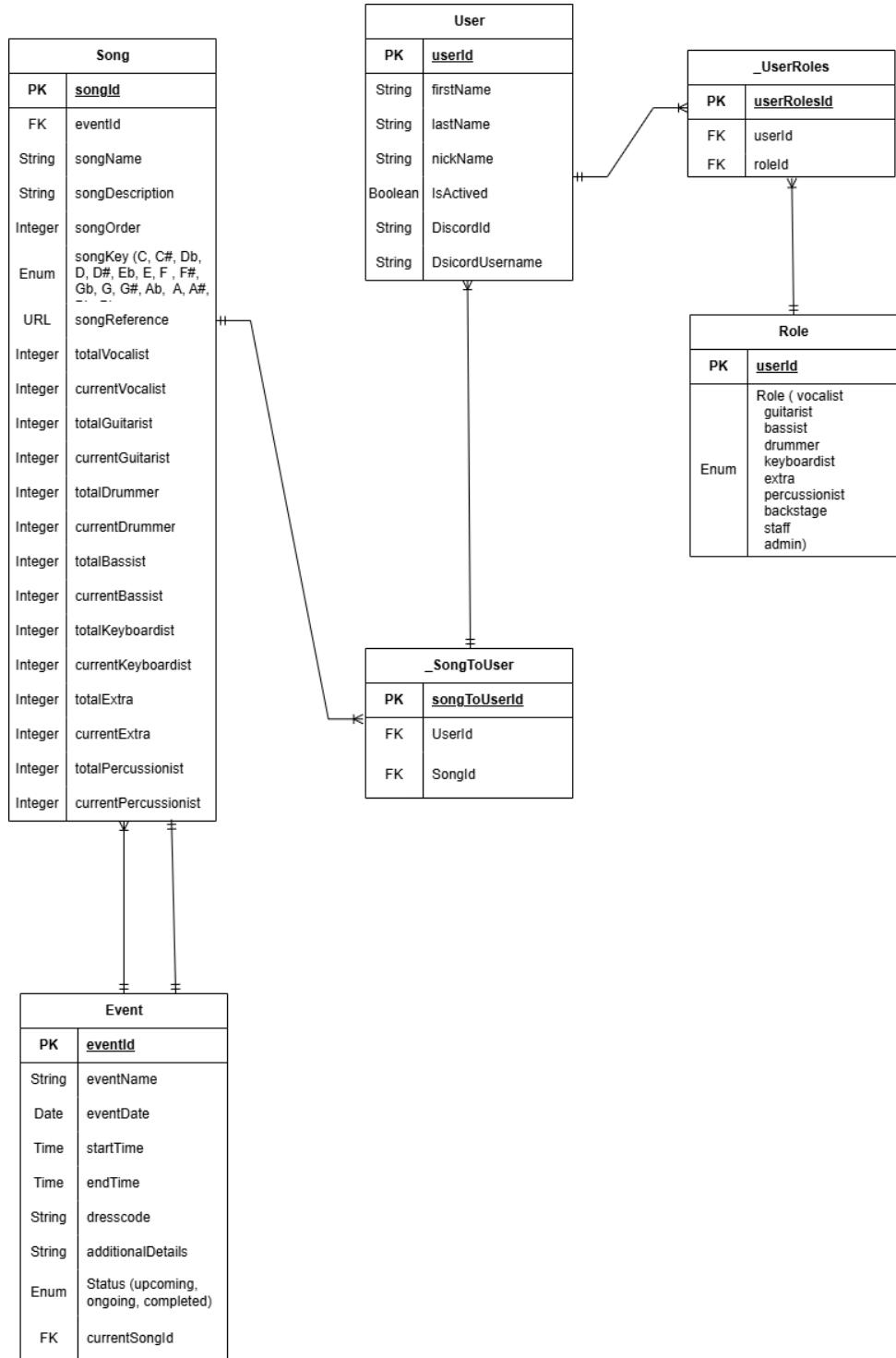
Code	Description	Links
200	successfully deactivate user account	No links
401	Unauthorized - Multiple possible causes: missing authorization header, invalid access token, or insufficient permissions	No links
403	Forbidden - System admin access required	No links
- Example Value:** application/json


```
{
        "message": "Missing authorization header, invalid access token or your discord account does not have access to this application"
      }
```
- Example Value:** application/json


```
{
        "message": "System admin access required"
      }
```

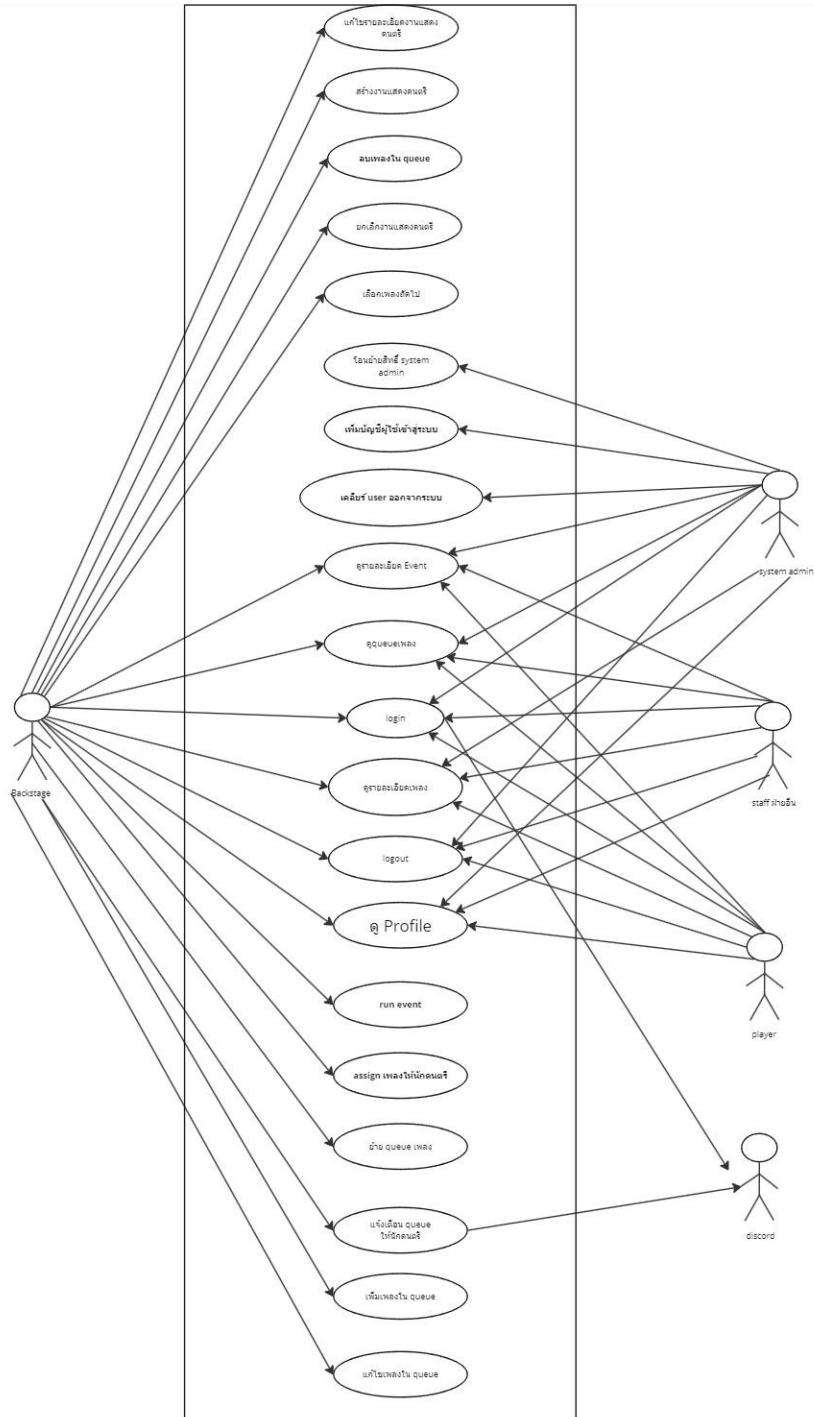
โดยระบบมีการทำ authentication และ authorization เพื่อให้แต่ละ endpoint ได้รับ การป้องกันโดยจำกัดการเข้าถึงเฉพาะผู้ใช้ที่ผ่านการยืนยันตัวตนและมีสิทธิ์ในการใช้งานที่ถูกต้อง เท่านั้น เพื่อป้องกันการเข้าถึงโดยไม่ได้รับอนุญาตและรักษาความปลอดภัยของระบบ

3.1.2 ER diagram สำหรับฐานข้อมูล



3.2 การวิเคราะห์ขอบเขตและความต้องการของระบบ

3.2.1 แผนภาพกรณีศึกษา (Use case diagram)



3.2.1.1 แสดงแผนภาพกรณีศึกษา (Use case diagram)

3.2.2 คำอธิบายแผนภาพกรณีใช้งาน (Use case Description)

3.2.2.1 ตารางรายละเอียดกรณีการใช้งานระบบทั้งหมด

Use case ID	Actor	Use case name	Description
1.	User ทุกประเภท	Login	User สามารถเข้าสู่ระบบเพื่อใช้งานแอปพลิเคชันตามสิทธิ์ของตนเองได้
2.	User ทุกประเภท	ดูรายละเอียดงานแสดงดนตรี	User ทุกคนสามารถดูรายละเอียดงานแสดงดนตรีแต่ละงาน เพื่อทราบถึงข้อมูลงานแสดงนั้นๆได้
3.	Backstage	สร้างงานแสดงดนตรี	Backstage สามารถสร้างงานแสดงดนตรีขึ้นมาเพื่อนำข้อมูลเข้าสู่ระบบและเพื่อให้ผู้ใช้ทุกคนสามารถเข้ามาดูรายละเอียดได้
4.	Backstage	แก้ไขรายละเอียดงานแสดงดนตรี	Backstage สามารถการแก้ไขรายละเอียดงานแสดงดนตรีที่สร้างเอาไว้ เช่น ชื่อ วันเวลา
5.	Backstage	ลบงานแสดงดนตรี	Backstage สามารถลบงานแสดงดนตรีที่ถูกยกเลิกไปแล้วหรือเสร็จสิ้นแล้วได้
6.	User ทุกประเภท	ดู queue เพลง	User สามารถดูคิวเพลงในแต่ละงานแสดงดนตรีได้
7.	Backstage	เพิ่มเพลงลงใน queue	Backstage สามารถเพิ่มเพลงลงใน queue ของงานแสดงดนตรีนั้น ๆ ได้

8.	Backstage	แก้ไขเพลงใน Queue	Backstage สามารถแก้ไขข้อเพลงรายละเอียด และ Performer ที่เล่นในเพลงนั้น ๆ ได้
9.	Backstage	ย้าย Queue เพลง	Backstage สามารถย้าย queue เพลงได้กรณีที่มีการเปลี่ยนแปลงของ queue เพลงก่อนจะเริ่มงานแสดงดนตรี
10.	Backstage	ลบเพลงใน Queue	Backstage สามารถลบเพลงออกจาก Queue ของงานแสดงดนตรีนั้นๆได้
11.	User ทุกประเภท	ดูรายละเอียดเพลง	User ทุกคนสามารถดูรายละเอียดในแต่ละเพลงได้ว่าเพลงนั้นมี Performer คนไหนเล่นบ้าง รวมถึง ชื่อเพลง, คีย์เพลง และ อ้างอิงเพลง (reference)
12.	Backstage	assign เพลงให้ Performer	Backstage สามารถ Assign เพลงให้ Performer เพื่อใช้ในการจัด Queue และแจ้งเตือน Queue ในขณะที่ขึ้นแสดง
13.	Backstage	Run event	Backstage สามารถเริ่มงานแสดงดนตรีที่เลือกได้เมื่อถึงวันแสดงจริง ตามที่ได้กรอกไว้ภายในระบบ เพื่อทำการรับคิวเพลงและเปิดระบบการแสดงแจ้งเตือนให้กับ Performer
14.	Backstage	เลือกเพลงถัดไป	Backstage สามารถเลือกเพลงถัดไปในการแสดงสดเพื่อเป็นการรับคิวเพลง

15.	Backstage	แจ้งเตือน queue ให้ Performer	Backstage สามารถกดแจ้งเตือน ให้นักดนตรีในเพลกัดไปโดยแจ้ง เตือนผ่าน Discord direct message ด้วย Discord bot
16.	System admin	เคลียร์ user ออกจากระบบ	System admin สามารถเคลียร์ User ที่จบการศึกษาไปแล้วออกจากระบบได้และระบบจะทำการลบผู้ใช้ออกจาก discord channels โดยอัตโนมัติ
17.	System admin	เพิ่มบัญชีผู้ใช้เข้าสู่ระบบ	System admin สามารถเพิ่ม User ใหม่ให้สามารถเข้าใช้งานระบบได้และระบบจะทำการเพิ่มผู้ใช้เข้าสู่ discord channels ที่เกี่ยวข้องโดยอัตโนมัติ
18.	User ทุกประเภท	ดู Profile	User ทุกคนสามารถดู Profile ของตัวเองได้ว่าข้อมูลถูกต้อง หรือไม่
19.	User ทุกประเภท	Logout	User สามารถ Logout ออกจากระบบเมื่อไม่ใช้งาน
20.	System admin	โอนย้ายสิทธิ์ system admin	System admin ต้องการโอนย้าย สิทธิ์ผู้ดูแลให้คนในชุมชนในกรณีที่ หมดภาระหรือออกจากชุมชน

3.3 การออกแบบการทำงานของระบบ

3.3.1 คำอธิบาย use case ของระบบ (Use case Specification)

3.3.1.1 รายละเอียดกรณีการใช้งานเข้าสู่ระบบ

Use case ID	1
Use case name	Login
Primary actor	User ทุกประเภท
Description	User สามารถเข้าสู่ระบบเพื่อใช้งานแอปพลิเคชันตามสิทธิ์ของตนเองได้
Trigger	User ต้องการเข้าสู่ระบบเพื่อใช้งานแอปพลิเคชัน
Preconditions	PRE-1 User ต้องมีบัญชี Discord เป็นของตัวเอง PRE-2 System admin ต้องนำข้อมูล Google Forms ที่สมาชิกกรอกตอนสมัครเข้า SCI BAND เข้าสู่ระบบเพื่อให้ผู้ใช้สามารถเข้าสู่ระบบได้
Postconditions	POST-1 User เข้าสู่ระบบและใช้งานแอปพลิเคชันได้ตามสิทธิ์ของตนเอง
Normal flow	<ol style="list-style-type: none"> 1. User กดเข้าสู่ระบบ 2. User กรอก Gmail หรือเบอร์มือถือและรหัสผ่าน 3. User กดยืนยันเพื่อเข้าสู่ระบบ 4. User เข้าสู่ระบบสำเร็จและระบบนำผู้ใช้ไปยังหน้าเมนูหลักพร้อมกับสามารถใช้งานระบบได้ตามสิทธิ์ของผู้ใช้
Alternative flow	User มีแอปพลิเคชัน Discord และทำการเข้าสู่ระบบไว้แล้ว <ol style="list-style-type: none"> 1. User กดเข้าสู่ระบบ 2. User กดยืนยันเพื่อเข้าสู่ระบบ

	<p>3. User เข้าสู่ระบบสำเร็จและระบบนำผู้ใช้ไปยังหน้าเมนูหลักพร้อมกับสามารถใช้งานระบบได้ตามสิทธิ์ของผู้ใช้</p>
Exceptions	<p>System admin ไม่ได้นำเข้าข้อมูลบัญชีผู้ใช้เข้าสู่ระบบ</p> <ol style="list-style-type: none"> 1. User กดเข้าสู่ระบบ 2. User กรอก Gmail หรือเบอร์มือถือและรหัสผ่าน 3. User กดยืนยันเพื่อเข้าสู่ระบบ <ul style="list-style-type: none"> ○ ระบบขึ้นข้อความแจ้งเตือนว่า "คุณไม่มีสิทธิ์ในการเข้าใช้งานระบบนี้" 4. User แจ้ง System admin หรือ Staff ที่เกี่ยวข้อง 5. System admin ทำการเพิ่มบัญชีผู้ใช้เข้าสู่ระบบ 6. User เข้าสู่ระบบอีกครั้ง 7. Use case ดำเนินตามปกติ
Priority	High
Frequency of use	1 ครั้งต่อ 1 คน

3.3.1.2 รายละเอียดกรณีดูรายละเอียดงานแสดงดนตรี

Use case ID	2
Use case name	ดูรายละเอียดงานแสดงดนตรี
Primary actor	User ทุกประเภท

Description	User ทุกคนสามารถดูรายละเอียดงานแสดงดนตรีแต่ละงานเพื่อทราบถึงข้อมูลงานแสดงนั้นๆได้
Trigger	User ต้องการดูข้อมูลงานแสดงดนตรีเพื่อทราบถึงรายละเอียดของงานนั้นๆ
Preconditions	<p>PRE-1 Backstage มีการเพิ่มงานแสดงดนตรีลงในระบบแล้ว</p> <p>PRE-2 User ทำการยืนยันตัวตนสำเร็จ</p>
Postconditions	POST-1 User ดูรายละเอียดงานแสดงดนตรีสำเร็จ
Normal flow	<ol style="list-style-type: none"> 1. User เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมดที่เกี่ยวข้อง 2. User เลือกงานแสดงดนตรีจากรายการงานแสดงดนตรีที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก โดยประกอบไปด้วยชื่อ วันเวลาแสดง Dress code และรายละเอียดเพิ่มเติม 3. หากต้องการดูงานแสดงดนตรีอื่นๆ <ul style="list-style-type: none"> ○ User กดปุ่มกลับไปยังหน้า Event ○ User เลือกดูรายละเอียดงานแสดงดนตรีที่ต้องการ 4. Use case สิ้นสุดเมื่อ user ไม่ต้องการดูงานแสดงดนตรีอีก ๆ แล้ว
Alternative flow	-
Exceptions	Event ยังไม่ได้ถูกเพิ่มลงในระบบ

	<ol style="list-style-type: none"> 1. User เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมดที่เกี่ยวข้อง 2. User ไม่พบงานแสดงดนตรีที่ต้องการในระบบ User ต้องแจ้งปัญหาให้ Backstage 3. User ทำการแจ้งปัญหาไปที่ Backstage ผ่านช่องทางส่วนตัว 4. Backstage ดำเนินการตาม Use case สร้างงานแสดงดนตรี เพื่อเพิ่มEventที่ต้องการ <ul style="list-style-type: none"> ○ ระบบบันทึกงานแสดงดนตรีใหม่และแสดงในรายการ 5. User เข้ามาดูรายละเอียดงานแสดงดนตรีอีกครั้ง 6. Use Case ดำเนินการตามปกติ
Priority	Medium
Frequency of use	1-10 ครั้งต่อหนึ่งงานดนตรี

3.3.1.3 รายละเอียดกรณีสร้างงานแสดงดนตรี

Use case ID	3
Use case name	สร้างงานแสดงดนตรี
Primary actor	Backstage
Description	Backstage สามารถสร้างงานแสดงดนตรีขึ้นมาเพื่อนำข้อมูลเข้าสู่ระบบและเพื่อให้ผู้ใช้ทุกคนสามารถเข้ามาดูรายละเอียดได้
Trigger	Backstage ต้องการสร้างงานแสดงดนตรีใหม่

Preconditions	PRE-1 Backstage ทำการยืนยันตัวตนสำเร็จ
Postconditions	POST-1 Backstage ทำการสร้าง Event ในระบบสำเร็จ
Normal flow	<p>1. Backstage เลือก "Event" จากหน้า Main Menu</p> <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด <p>2. Backstage กดไอคอนรูปเครื่องหมาย (+) จากหน้า Event</p> <ul style="list-style-type: none"> ○ ระบบแสดงฟอร์มสำหรับเพิ่มงานแสดง 丹ตรีใหม่โดยใส่ชื่อ งานแสดงดนตรี, วันเดือนปีที่จัดแสดง, เวลาเริ่มงาน-เวลาสิ้นสุดงาน, การแต่งกายวันงาน (dress code) และรายละเอียดเพิ่มเติม <p>3. Backstage กรอกข้อมูลที่ต้องการและกดยืนยัน</p> <ul style="list-style-type: none"> ○ ระบบบันทึกข้อมูล Event ใหม่ลงในฐานข้อมูล ○ ระบบแสดงข้อความแจ้งเตือน: "สร้าง Event สำเร็จ" <p>4. ระบบกลับไปที่หน้ารายการ Event</p> <p>5. หากต้องการเพิ่มงานแสดงดนตรีอีก</p> <ul style="list-style-type: none"> ○ ระบบนำ Backstage กลับไปยังหน้างานแสดงดนตรีทั้งหมด <p>6. Backstage ทำซ้ำขั้นตอน 2 - 5</p> <p>7. Use case สิ้นสุดเมื่อ Backstage ไม่ต้องการเพิ่มงานแสดงดนตรีอีก ๆ แล้ว</p>
Alternative flow	Backstage ต้องการยกเลิกการเพิ่มงานแสดงดนตรี

	<ol style="list-style-type: none"> 1. ขณะที่อยู่ในฟอร์มเพิ่มงานแสดงดนตรีใหม่ 2. Backstage กดปุ่ม x เพื่อยกเลิกการสร้างงานแสดง ดนตรี <ul style="list-style-type: none"> ○ ระบบขึ้นข้อความให้ backstage ว่า "คุณ ต้องการยกเลิกการสร้าง event หรือไม่" พร้อมปุ่มยืนยันและยกเลิก 3. Backstage กดยืนยัน <ul style="list-style-type: none"> ○ ระบบนำ Backstage กลับไปยังหน้า รายการงานแสดงดนตรีทั้งหมด 4. Alternative Flow สิ้นสุด
Exceptions	<p>ข้อมูลที่กรอกไม่ครบถ้วนหรือไม่ถูกต้อง</p> <ol style="list-style-type: none"> 1. ขณะที่ Backstage กรอกข้อมูลงานแสดงดนตรีใหม่ 2. ระบบตรวจสอบข้อมูลพิเศษ เช่น: <ul style="list-style-type: none"> ○ ชื่อ Event หรือ Dress Code เป็นค่าว่าง 3. ระบบแสดงข้อความแจ้งเตือน: <ul style="list-style-type: none"> ○ “กรุณากรอกชื่อ Event” หรือ “กรุณา กรอก Dress Code” 4. Backstage กรอกข้อมูลให้ถูกต้อง 5. Backstage กดปุ่ม "สร้าง" เพื่อปิดข้อความแจ้ง เตือน: <ul style="list-style-type: none"> ○ ระบบขึ้นข้อความให้ backstage ว่า “สร้าง Event สำเร็จ” 6. Use case ดำเนินตามปกติ
Priority	High

Frequency of use	1 ครั้งต่อหนึ่งงานแสดงดนตรี
------------------	-----------------------------

3.3.1.4 รายละเอียดกรณีแก้ไขงานแสดงดนตรี

Use case ID	4
Use case name	แก้ไขรายละเอียดงานแสดงดนตรี
Primary actor	Backstage
Description	Backstage สามารถแก้ไขรายละเอียดงานแสดงดนตรีที่สร้างเอาไว้ เช่น ชื่อ วันเวลา Dress Code และ รายละเอียดเพิ่มเติม
Trigger	Backstage ต้องการแก้ไขรายละเอียดข้อมูล Event เนื่องจากมีการเปลี่ยนแปลงของข้อมูล
Preconditions	PRE-1 Backstage ทำการยืนยันตัวตนสำเร็จ PRE-2 ในงานแสดงดนตรีนั้น ๆ จะต้องมีการตกลงกันก่อนว่าจะปรับเปลี่ยนข้อมูลอะไรบ้าง
Postconditions	POST-1 งานแสดงดนตรีในระบบถูกแก้ไข
Normal flow	<ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานแสดงดนตรีที่ต้องการจากรายการงานแสดงดนตรีที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก

	<ol style="list-style-type: none"> 3. Backstage ทำการแก้ไขรายละเอียดโดยสามารถแก้ไข ชื่องานแสดงดนตรี, วันเวลา, การแต่งกายวันงาน (dress code) และรายละเอียดเพิ่มเติม 4. Backstage กดยืนยันการแก้ไข <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "อัปเดต Event สำเร็จ" 5. หากต้องการแก้ไขงานแสดงดนตรีอื่นๆเพิ่มเติม <ul style="list-style-type: none"> ○ ระบบนำ Backstage กลับไปยังหน้าเลือกงานแสดงดนตรี 6. Backstage ทำขั้นตอนที่ 2 - 5 7. Use Case สิ้นสุดเมื่อ Backstage ไม่ต้องการแก้ไขงานแสดงดนตรีอื่น ๆ เพิ่มเติม
Alternative flow	<p>Backstage ต้องการยกเลิกการแก้ไข</p> <ol style="list-style-type: none"> 1. ขณะที่ Backstage อยู่ในฟอร์มแก้ไขรายละเอียดงานแสดงดนตรี 2. Backstage กดย้อนกลับไปที่หน้า Event ทั้งหมด 3. Alternative flow สิ้นสุด
Exceptions	<p>Backstage แก้ไขข้อมูลไม่ถูกต้อง</p> <ol style="list-style-type: none"> 1. ขณะที่ Backstage กรอกข้อมูลในฟอร์มแก้ไขรายละเอียดงานแสดงดนตรี 2. ระบบตรวจสอบข้อผิดพลาด เช่น: <ul style="list-style-type: none"> ○ ชื่อ Event หรือ Dress Code เป็นค่าว่าง 3. ระบบแสดงข้อความแจ้งเตือน:

	<ul style="list-style-type: none"> ○ “กรุณากรอกชื่อ Event” หรือ “กรุณากรอก Dress Code” <p>4. Backstage กดปุ่ม “อัปเดต” เพื่อปิดข้อความแจ้งเตือน</p> <ul style="list-style-type: none"> ○ ระบบขึ้นข้อความให้ backstage ว่า “อัปเดต Event สำเร็จ” <p>5. Exception Flow สิ้นสุด หลังจาก Backstage กรอกข้อมูลครบถ้วนและกดยืนยันอีกครั้ง</p>
Priority	High
Frequency of use	0-5 ครั้งต่อหนึ่งงานคนตัวรี

3.3.1.5 รายละเอียดกรณีลับงานแสดงดนตรี

Use case ID	5
Use case name	ลับงานแสดงดนตรี
Primary actor	Backstage
Description	Backstage สามารถลับงานแสดงดนตรีที่ถูกยกเลิกไปแล้ว หรือเสร็จสิ้นแล้วได้
Trigger	งานแสดงดนตรีถูกยกเลิก Backstage จึงต้องการลับงานแสดงดนตรี
Preconditions	PRE-1 Backstage ทำการยืนยันตัวตนสำเร็จ PRE-2 มีการยืนยันแล้วว่างานแสดงดนตรีนั้น ๆ ถูกยกเลิก
Postconditions	POST-1 งานแสดงดนตรีถูกลบออกจากระบบ
Normal flow	<ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานแสดงดนตรีที่ต้องการ

	<p>3. Backstage กดปุ่ม “ลบ Event”</p> <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "ต้องการลบ Event หรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" <p>4. Backstage กดยืนยัน</p> <ul style="list-style-type: none"> ○ ระบบลบงานแสดงดนตรีออกจากรายการ Event ทั้งหมด ○ ระบบแสดงข้อความแจ้งเตือน: "ลบ Event สำเร็จ" <p>5. หากต้องการลบงานแสดงดนตรีอื่น ๆ เพิ่มเติม:</p> <ul style="list-style-type: none"> ○ ระบบนำ Backstage กลับไปที่หน้ารวมงานแสดงดนตรีทั้งหมด ○ Backstage เลือกงานแสดงดนตรีอื่น ๆ และทำซ้ำขั้นตอน 2 - 4 <p>6. Use Case สิ้นสุดเมื่อ Backstage ไม่ต้องการลบ Event เพิ่มเติม</p>
Alternative flow	<p>Backstage ต้องการยกเลิกการลบงานแสดงดนตรี</p> <ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานแสดงดนตรีที่ต้องการ 3. Backstage กดปุ่ม “ลบ Event” ○ ระบบแสดงข้อความแจ้งเตือน: "ต้องการลบ Event หรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก"

	4. Backstage กดปุ่มยกเลิก 5. สิ้นสุด Alternative flow
Exceptions	-
Priority	medium
Frequency of use	0 - 1 ครั้งต่อหนึ่งงานแสดงดนตรี

3.3.1.6 รายละเอียดกรณีดู queue เพลง

Use case ID	6
Use case name	ดู queue เพลง
Primary actor	User ทุกประเภท
Description	User สามารถดูคิวเพลงในแต่ละงานแสดงดนตรีได้
Trigger	User ต้องการดู queue เพลงในงานแสดงดนตรีนั้น ๆ
Preconditions	<p>PRE-1 Backstage มีการเพิ่มงานแสดงดนตรีลงไปในระบบ แล้ว</p> <p>PRE-2 Backstage ได้เพิ่มเพลงลงไปในงานแสดงดนตรีนั้น ๆ แล้ว</p> <p>PRE-3 User ทำการยืนยันตัวตนสำเร็จ</p>
Postconditions	POST-1 User ดู queue เพลงสำเร็จ
Normal flow	<ol style="list-style-type: none"> User เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ระบบแสดงรายการงานแสดงดนตรีทั้งหมด ที่เกี่ยวข้อง User เลือกงานแสดงดนตรีจากรายการที่มี

	<ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก <p>3. User กดปุ่ม "ดูรายชื่อเพลง"</p> <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น ๆ <p>4. หากต้องการดู queue เพลงของงานแสดงดนตรีอื่น ๆ</p> <ul style="list-style-type: none"> ○ User กด tab bar “Event ทั้งหมด” จากหน้ารายละเอียดเพลง ○ ระบบนำ User กลับไปยังหน้าเลือก Event ทั้งหมด <p>5. User ทำขั้นตอน 2 - 4</p> <p>6. Use case สิ้นสุดเมื่อ user ไม่ต้องการดู Event อื่น ๆ 再次</p>
Alternative flow	<p>Performer ต้องการดู queue เพลงเฉพาะเพลงที่ตนเองมีส่วนร่วม</p> <p>1. Performer เลือก "Event" จากหน้า Main Menu</p> <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมดที่เกี่ยวข้อง <p>2. Performer เลือกงานงานการแสดงดนตรีจากรายการที่มี</p> <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก <p>3. Performer กดปุ่ม "ดูรายชื่อเพลง"</p>

	<ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น <p>4. Performer ทำการกดปุ่ม "แสดงเฉพาะเพลงของฉัน"</p> <ul style="list-style-type: none"> ○ ระบบทำการกรองเฉพาะเพลงที่ Performer มีส่วนเกี่ยวข้อง <p>5. หากต้องการดู queue เพลงของ Event อื่นๆ</p> <ul style="list-style-type: none"> ○ User กด tab bar "Event ทั้งหมด" จากหน้ารายละเอียดเพลง ○ ระบบนำ User กลับไปยังหน้าเลือก Event ทั้งหมด <p>6. User ทำขั้นตอน 2 - 5</p> <p>7. Use case สิ่งสุดเมื่อ user ไม่ต้องการดู Event อื่นๆ แล้ว</p> <ul style="list-style-type: none"> ○ User กดไอคอนรูป (Event) จากหน้ารายละเอียดเพลง ○ ระบบนำ User กลับไปยังหน้าเลือกงานแสดงดนตรีทั้งหมด <p>8. User ทำขั้นตอน 2 - 4</p> <p>9. Use case สิ่งสุดเมื่อ User ไม่ต้องการดู queue เพลง อื่น ๆ แล้ว</p>
Exceptions	<p>เพลงยังไม่ได้ถูกเพิ่มลงมาในระบบ</p> <p>1. User เลือก "Event" จากหน้า Main Menu</p> <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมดที่เกี่ยวข้อง

	<p>2. User เลือกงานแสดงดนตรีจากรายการที่มี</p> <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของ Event ที่เลือก <p>3. User กดปุ่ม "ดูรายชื่อเพลง"</p> <p>4. User ไม่พบเพลงที่ต้องการ</p> <ul style="list-style-type: none"> ○ User ทำการแจ้งปัญหาไปที่ Backstage ผ่านช่องทางส่วนตัว <p>5. Backstage ดำเนินการตาม Use case เพิ่มเพลง ใน queue เพื่อเพิ่มเพลงที่ต้องการ</p> <ul style="list-style-type: none"> ○ ระบบบันทึก queue เพลงใหม่และแสดงในรายการ <p>6. Use Case ดำเนินการตามปกติ</p>
Priority	High
Frequency of use	1-10 ครั้งต่อหนึ่งงานดนตรี

3.3.1.7 รายละเอียดกรณีเพิ่มเพลงลงใน queue

Use case ID	7
Use case name	เพิ่มเพลงลงใน queue
Primary actor	Backstage
Description	Backstage สามารถเพิ่มเพลงลงใน queue ของงานแสดงดนตรีนั้น ๆ ได้
Trigger	Backstage ต้องการเพิ่มเพลงลงใน queue เพลงของงานแสดงดนตรีนั้น ๆ
Preconditions	PRE-1 Backstage ทำการยืนยันตัวตนสำเร็จ

	<p>PRE-2 ใน Event นั้น ๆ จะต้องมีการตกลงเพลงกันก่อนจึงจะเพิ่มเพลงได้</p> <p>PRE-3 Backstage จะต้องทำการสร้างงานแสดงดนตรีไว้ในระบบแล้ว</p>
Postconditions	<p>POST-1 เพลงนั้น ๆ ถูกเพิ่มลงไว้ในระบบโดยผูกกับงานแสดงดนตรีนั้น ๆ เท่านั้น</p>
Normal flow	<ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานงานแสดงดนตรีจากรายการที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก 3. Backstage กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น 4. Backstage กดไอคอนรูปเครื่องหมาย (+) จากหน้ารายการเพลง <ul style="list-style-type: none"> ○ ระบบแสดงฟอร์มสำหรับเพิ่มข้อมูลเพลงใหม่โดยใส่ชื่อเพลง, คีย์เพลง (Song Key), อ้างอิงเพลง (Reference) รายละเอียดเพิ่มเติม และกรอกจำนวน Performer ที่ต้องการในแต่ละ part 5. Backstage กรอกข้อมูลเพลงที่ต้องการและกดยืนยัน

	<ul style="list-style-type: none"> ○ ระบบบันทึกข้อมูลเพลงใหม่ลงในฐานข้อมูล ○ ระบบแสดงข้อความแจ้งเตือน: "สร้างเพลงสำเร็จ" โดยเพลงจะเรียงลำดับตามที่เพิ่มเพลง <p>6. หากต้องการเพิ่มเพลงอื่นๆ</p> <ul style="list-style-type: none"> ○ ระบบนำ Backstage กลับไปยังหน้ารายการเพลง <p>7. Backstage ทำขั้นตอน 4 - 5</p> <p>8. Use case สื้นสุดเมื่อ Backstage ไม่ต้องการเพิ่มเพลงอื่น ๆ แล้ว</p>
Alternative flow	<p>Backstage ยกเลิกการเพิ่มเพลง</p> <ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานงานแสดงดนตรีจากรายการที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก 3. Backstage กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น 4. Backstage กดไอคอนรูปเครื่องหมาย (+) จากหน้ารายการเพลง

	<ul style="list-style-type: none"> ○ ระบบแสดงฟอร์มสำหรับเพิ่มข้อมูลเพลงใหม่โดยใส่ชื่อเพลง, คีย์เพลง (Song Key) , อ้างอิงเพลง (Reference) รายละเอียดเพิ่มเติม และกรอกจำนวน Performer ที่ต้องการในแต่ละ part <p>5. Backstage กรอกข้อมูลเพลงที่ต้องการและกดยืนยันลับ</p> <ul style="list-style-type: none"> ○ ระบบนำ Backstage กลับไปที่หน้ารายการเพลงทั้งหมด <p>6. ระบบนำ Backstage กลับไปที่หน้ารายการเพลงทั้งหมด</p> <p>7. Alternative Flow สิ้นสุด</p>
Exceptions	-
Priority	High
Frequency of use	5 - 30 ครั้งต่อหนึ่งงานดนตรี

3.3.1.8 รายละเอียดกรณีแก้ไขเพลงใน Queue

Use case ID	8
Use case name	แก้ไขเพลงใน Queue
Primary actor	Backstage
Description	Backstage สามารถแก้ไขชื่อเพลง รายละเอียด และ Performer ที่เล่นในเพลงนั้น ๆ ได้
Trigger	Backstage ต้องการแก้ไขเพลงใน Queue
Preconditions	PRE-1 Backstage มีการเพิ่มงานแสดงคอนตรีลิ้งไปในระบบแล้ว

	<p>PRE-2 Backstage ได้เพิ่มเพลงลงไปในงานแสดงดนตรีนั้น ๆ แล้ว</p> <p>PRE-3 Backstage ทำการยืนยันตัวตนสำเร็จ</p>
Postconditions	POST-1 เพลงในระบบถูกแก้ไข
Normal flow	<ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานงานแสดงดนตรีจากรายการที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก 3. Backstage กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น 4. Backstage เลือกเพลงที่ต้องการจะแก้ไข <ul style="list-style-type: none"> ○ ระบบนำ Backstage ไปสู่หน้ารายละเอียด 5. Backstage ทำการแก้ไขรายละเอียดโดยสามารถแก้ไข ชื่อเพลง (Song Name), จำนวน Performer ที่ต้องการในแต่ละตำแหน่ง, คีย์เพลง (Song Key), อ้างอิงเพลง (Reference) 6. Backstage กดยืนยันการแก้ไข <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "อัปเดตเพลงสำเร็จ" 7. หากต้องการแก้ไขเพลงอีกเพลงหนึ่งเพิ่มเติม <ul style="list-style-type: none"> ○ ระบบนำ Backstage กลับไปยังหน้าเลือกเพลงที่ต้องการแก้ไข

	<p>8. Backstage ทำข้ามต้อน 4 - 7</p> <p>9. Use Case สิ้นสุดเมื่อ Backstage ไม่ต้องการแก้ไข เพลงอื่นๆเพิ่มเติม</p>
Alternative flow	<p>Backstage ต้องการยกเลิกการแก้ไข</p> <ol style="list-style-type: none"> 1. ขณะที่ Backstage อยู่ในฟอร์มแก้ไขรายละเอียด เพลง 2. Backstage กดย้อนกลับ 3. ระบบนำ Backstage กลับไปยังหน้ารายชื่อเพลง 4. Alternative Flow สิ้นสุด
Exceptions	<p>Backstage กรอกข้อมูลการแก้ไขข้อมูลเพลงผิดรูปแบบ</p> <ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด ที่เกี่ยวข้อง 2. Backstage เลือกงานงานแสดงดนตรีจากรายการ งานแสดงดนตรีที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของ งานแสดงดนตรีที่เลือก 3. Backstage กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดง ของ Event นั้น 4. Backstage เลือกเพลงที่ต้องการจะแก้ไข <ul style="list-style-type: none"> ○ ระบบนำ Backstage ไปสู่หน้ารายละเอียด เพลง 5. Backstage กดปุ่ม "แก้ไขเพลง"

	<p>6. Backstage ทำการแก้ไขรายละเอียดโดยสามารถแก้ไข ชื่อเพลง (Song Name), ตำแหน่งที่เล่น (lineup performer), คีย์เพลง (Song Key), อ้างอิงเพลง (Reference)</p> <p>7. Backstage กดยืนยันการแก้ไข</p> <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "ข้อมูลที่แก้ไขไม่ถูกต้อง (เช่น กรอกข้อมูลไม่ครบถ้วนจากการลบข้อมูลบางแควโดยไม่ตั้งใจ)" <p>8. Backstage ตรวจสอบข้อมูลและลองแก้ไขอีกครั้ง</p> <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "อัปเดตเพลงสำเร็จ" <p>9. Use Case สิ้นสุดเมื่อแก้ไขสำเร็จ</p>
Priority	Medium
Frequency of use	0-5 ครั้งต่อหนึ่งงานดนตรี

3.3.1.9 รายละเอียดกรณีย้าย Queue เพลง

Use case ID	9
Use case name	ย้าย Queue เพลง
Primary actor	Backstage
Description	Backstage สามารถย้าย queue เพลงได้กรณีที่มีการเปลี่ยนแปลงของ queue เพลงก่อนจะเริ่มงานแสดงดนตรี
Trigger	Backstage ต้องการย้าย queue เพลง
Preconditions	PRE-1 Backstage มีการเพิ่มงานแสดงดนตรีลงไว้ในระบบแล้ว

	<p>PRE-2 Backstage ได้เพิ่มเพลงลงไว้ในงานแสดงดนตรีนั้น ๆ แล้ว</p> <p>PRE-3 Backstage ทำการยืนยันตัวตนสำเร็จ</p>
Postconditions	POST-1 Backstage ทำการย้าย queue เพลงได้สำเร็จ
Normal flow	<ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมดที่เกี่ยวข้อง 2. Backstage เลือกงานงานการแสดงดนตรีจากรายการที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก 3. Backstage กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น 4. Backstage เลือกเพลงที่ต้องการย้าย โดยการคลิกที่เพลงค้างเอาไว้และสามารถเลื่อนได้ตามต้องการ 5. ระบบแสดงข้อความแจ้งเตือน: "คุณต้องการย้ายลำดับเพลงหรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" 6. Backstage กดปุ่มยืนยัน <ul style="list-style-type: none"> ○ ระบบจะนำเพลงย้ายจากจุดเดิมไปที่จุดใหม่ 7. หากต้องการย้ายเพลงอื่น ๆ เพิ่มเติม Backstage ทำซ้ำขั้นตอน 4 - 6

	<p>8. Use Case สิ้นสุดเมื่อ Backstage "ไม่ต้องการย้าย เพลงเพิ่มเติม"</p>
Alternative flow	<p>Backstage ยกเลิกการย้าย Queue เพลง</p> <ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงตอนตระทั้งหมด 2. Backstage เลือกงานรายการงานแสดงตอนตระที่มีรายการที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของ รายงานแสดงตอนตระที่เลือก 3. Backstage กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดง ของงานแสดงตอนตระนั้น 4. Backstage เลือกเพลงที่ต้องการย้าย โดยการคลิกที่ เพลงค้างเอาไว้และสามารถเลื่อนได้ตามต้องการ 5. ระบบแสดงข้อความแจ้งเตือน: "คุณต้องการย้าย ลำดับเพลงหรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" 6. Backstage กดปุ่มยกเลิก <ul style="list-style-type: none"> ○ ระบบจะไม่ย้ายเพลงและอยู่ในตำแหน่งเดิม ของรายการเพลงใน Event 7. เพลงยังคงอยู่ในตำแหน่งเดิมของรายการเพลงเดิม โดยไม่มีการเปลี่ยนแปลง 8. ระบบนำ Backstage กลับไปที่หน้ารายชื่อเพลง 9. Alternative Flow สิ้นสุด และกลับเข้าสู่ Normal Flow

Exceptions	-
Priority	Low
Frequency of use	0 – 5 ครั้งต่อหนึ่งงานแสดงดนตรี

3.3.1.10 รายละเอียดกรณีลับเพลงใน Queue

Use case ID	10
Use case name	ลับเพลงใน Queue
Primary actor	Backstage
Description	Backstage สามารถลบเพลงออกจาก Queue ของงานแสดงดนตรีนั้นๆได้
Trigger	Backstage ต้องการลบเพลงออกจาก Queue เพลงของงานแสดงดนตรีนั้น ๆ เนื่องจากไม่มีการเล่นเพลงนั้น ๆ แล้ว
Preconditions	PRE-1 Backstage ทำการยืนยันตัวตนสำเร็จ PRE-2 จะต้องมีเพลงของถูกเพิ่มอยู่ในระบบก่อนแล้ว
Postconditions	POST-1 เพลงนั้น ๆ ถูกลบออกจากระบบ
Normal flow	<ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานงานแสดงดนตรีจากรายการที่มี <ul style="list-style-type: none"> ○ ระบบแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก 3. Backstage กดปุ่ม "ดูรายชื่อเพลง"

	<ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น <ol style="list-style-type: none"> 4. Backstage เลือกเพลงที่ต้องการลบ 5. Backstage ทำการกดปุ่ม “ลบเพลง” 6. ระบบแสดงข้อความแจ้งเตือน: "ต้องการลบเพลงหรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" 7. Backstage กดปุ่มยืนยัน <ul style="list-style-type: none"> ○ ระบบลบเพลงออกจากรายการเพลงใน Event ○ ระบบแสดงข้อความแจ้งเตือน: "ลบเพลงสำเร็จ" 8. หากต้องการลบเพลงอื่น ๆ เพิ่มเติม: <ul style="list-style-type: none"> ○ ระบบนำ Backstage กลับไปที่หน้ารายชื่อเพลง 9. Backstage ทำขั้นตอน 4 - 8 10. Use Case สิ้นสุดเมื่อ Backstage ไม่ต้องการลบเพลงเพิ่มเติม
Alternative flow	<p>Backstage ยกเลิกการลบเพลง</p> <ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายงานการแสดงดนตรีทั้งหมดที่เกี่ยวข้อง 2. Backstage เลือกงานงานแสดงดนตรีจากรายการที่มี

	<ul style="list-style-type: none"> ○ ระบบแสดงหน้ารายละเอียดของงานแสดง คนตระที่เลือก <p>3. Backstage กดปุ่ม "ดูรายชื่อเพลง"</p> <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดง ของงานแสดงคนตระนั้น ๆ <p>4. Backstage เลือกเพลงที่ต้องการลบ โดยเลือกเพลง ลงไปที่ถังขยะ</p> <p>5. ระบบแสดงข้อความแจ้งเตือน: "ต้องการลบเพลง หรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก"</p> <p>6. Backstage กดปุ่มยกเลิก</p> <p>7. เพลงยังคงอยู่ในรายการเพลงเดิมโดยไม่มีการ เปลี่ยนแปลง</p> <p>8. Alternative Flow สิ้นสุด</p>
Exceptions	-
Priority	Medium
Frequency of use	0-3 ครั้งต่อหนึ่งงานคนตระ

3.3.1.11 รายละเอียดกรณีดูรายละเอียดเพลง

Use case ID	11
Use case name	ดูรายละเอียดเพลง
Primary actor	User ทุกประเภท
Description	User ทุกประเภทสามารถดูรายละเอียดในแต่ละเพลงได้ว่า เพลงนั้นมี Performer คนไหนเล่นบ้าง รวมถึง ชื่อเพลง, คีย์ เพลง และอ้างอิงเพลง (reference)
Trigger	User ต้องการดูข้อมูลของแต่ละเพลง

Preconditions	<p>PRE-1 Backstage มีการเพิ่มงานแสดงดนตรีลงไปในระบบแล้ว</p> <p>PRE-2 Backstage ได้เพิ่มเพลงลงไปในงานแสดงดนตรีนั้น ๆ แล้ว</p> <p>PRE-3 Backstage ได้ assign เพลงให้กับ Performer แล้ว</p> <p>PRE-4 User ยืนยันตัวตนสำเร็จ</p>
Postconditions	POST-1 User ดูรายละเอียดเพลงสำเร็จ
Normal flow	<ol style="list-style-type: none"> 1. User เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. User เลือกงานงานแสดงดนตรีจากการรายการที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก 3. User กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น 4. User เลือกเพลงที่ต้องการดูรายละเอียดเพลง 5. ระบบทำการแสดงหน้ารายละเอียดของเพลงโดยจะมี ชื่อเพลง จำนวน performer ในแต่ละตำแหน่ง , คีย์เพลง (Song Key) , อ้างอิงเพลง (reference) 6. หาก User ต้องการดู lineup performer นั้น User สามารถกดปุ่ม “ดูข้อมูล” ตามตำแหน่ง performer ที่ต้องการและระบบจะแสดงชื่อ Performer ที่เกี่ยวข้อง 7. หากต้องการดูรายละเอียดเพลงอื่นๆ <ul style="list-style-type: none"> ○ User กดย้อนกลับ

	<ul style="list-style-type: none"> ○ ระบบนำ User กลับไปยังหน้ารายชื่อเพลงของงานแสดงดนตรีนั้น ๆ 8. User ทำขั้นตอน 4 - 6 9. Use case สิ้นสุดเมื่อ user ไม่ต้องการดูเพลงอื่น ๆ แล้ว
Alternative flow	<p>Performer ต้องการดูรายละเอียดเพลงเฉพาะเพลงที่ตนเองเกี่ยวข้อง</p> <ol style="list-style-type: none"> 1. Performer เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีที่ทั้งหมดที่เกี่ยวข้อง 2. Performer เลือกงานงานแสดงดนตรีจากรายการงานแสดงดนตรีที่มี 3. Performer กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น 4. Performer กด checkbox ตรง "แสดงเฉพาะเพลงของฉัน" <ul style="list-style-type: none"> ○ ระบบทำการกรองเพลงให้ Performer 5. Performer เลือกเพลงที่ต้องการดูรายละเอียดเพลง 6. ระบบทำการแสดงหน้ารายละเอียดของเพลงโดยจะมี ชื่อเพลง จำนวน performer ในแท็ล์ต์ตำแหน่ง , คีย์เพลง (Song Key) , อ้างอิงเพลง (reference) 7. หาก Performer ต้องการดู lineup performer นั้น Performer สามารถกดปุ่มดูข้อมูลที่ต้องการและระบบจะแสดงชื่อ Performer ที่เกี่ยวข้อง 8. หากต้องการดูรายละเอียดเพลงอื่น ๆ

	<ul style="list-style-type: none"> ○ Performer กดย้อนกลับ ○ ระบบนำ Performer กลับไปยังหน้ารายการเพลง <p>9. Performer ทำซ้ำขั้นตอน 5 - 7</p> <p>10. Use case สิ้นสุดเมื่อ user ไม่ต้องการดูเพลงอีกต่อไป</p>
Exceptions	<p>เพลงยังไม่ได้ถูกเพิ่มลงในระบบ</p> <ol style="list-style-type: none"> 1. User เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงตอนต่อไปทั้งหมด 2. User เลือกงานแสดงตอนต่อไปจากรายการที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงตอนต่อไปที่เลือก 3. User กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงตอนต่อไปนั้น 4. User เลือกเพลงที่ต้องการดูรายละเอียดเพลง 5. User ไม่พบเพลงในระบบ <ul style="list-style-type: none"> ○ User ทำการแจ้งปัญหาไปที่ Backstage ผ่านช่องทางส่วนตัว 6. Backstage ดำเนินการตาม Use case: เพิ่มเพลงใน Queue เพื่อเพิ่มเพลงที่ต้องการ <ul style="list-style-type: none"> ○ ระบบบันทึกเพลงใหม่และแสดงในรายการเพลงของงานแสดงตอนต่อไป ๆ 7. User เข้ามาดูรายละเอียดเพลงอีกครั้ง 8. Use Case ดำเนินการตามปกติ

Priority	High
Frequency of use	1-10 ครั้งต่อหนึ่งงานดนตรี

3.3.1.12 รายละเอียดกรณี Assign เพลงให้ Performer

Use case ID	12
Use case name	assign เพลงให้ Performer
Primary actor	Backstage
Description	Backstage สามารถ Assign เพลงให้ Performer เพื่อใช้ในการจัด Queue และแจ้งเตือน Queue ในขณะที่ขึ้นแสดง
Trigger	Backstage ต้องการ Assign เพลงให้ performer
Preconditions	PRE-1 Backstage มีการเพิ่ม Event ลงในระบบแล้ว PRE-2 Backstage ได้เพิ่มเพลงลงไว้ใน Event นั้นๆแล้ว PRE-3 Backstage ยืนยันตัวตนสำเร็จ
Postconditions	POST-1 Performer ถูก Assign ไปยังเพลงที่เกี่ยวข้องอย่างถูกต้อง
Normal flow	<ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานงานแสดงดนตรีจากรายการที่มี 3. Backstage กดปุ่ม "ดูรายชื่อเพลง" 4. ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดงของงานแสดงดนตรีนั้น 5. Backstage เลือกเพลงที่ต้องการจะ assign

	<ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของ เพลงโดยจะมี ชื่อเพลง จำนวน performer ในแต่ละตำแหน่ง , คีย์เพลง (Song Key) , อ้างอิงเพลง (reference) <ol style="list-style-type: none"> 6. Backstage เลือกตำแหน่ง ของ performer ที่ต้องการจะ assign 7. Backstage เลือกชื่อ Performer ทั้งหมดที่ต้องการ และกดยืนยัน <ul style="list-style-type: none"> ○ ระบบบันทึกข้อมูลการ Assign 8. หากต้องการ Assign เพลงเพิ่มเติมในตำแหน่งอื่น ๆ 9. Backstage กดย้อนกลับไปยังหน้ารายชื่อเพลงที่ต้องการ Assign 10. Backstage ทำขั้นตอน 5 - 9 11. Use Case สิ่งสุดเมื่อ Backstage ไม่ต้องการ Assign เพลงเพิ่มเติม
Alternative flow	<p>Backstage ต้องการยกเลิกการ assign</p> <ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานงานแสดงดนตรีจากรายการที่มี <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของ งานแสดงดนตรีที่เลือก 3. Backstage กดปุ่ม "ดูรายชื่อเพลง" <ul style="list-style-type: none"> ○ ระบบแสดงรายชื่อเพลงที่ใช้ในการแสดง ของงานแสดงดนตรีนั้น

	<p>4. Backstage เลือกเพลงที่ต้องการจะ assign</p> <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของเพลงโดยจะมีชื่อเพลง จำนวน performer ในแต่ละตำแหน่ง , คีย์เพลง (Song Key) , อ้างอิงเพลง (reference) <p>5. Backstage เลือกตำแหน่งของ performer ที่ต้องการจะ assign</p> <p>6. Backstage เลือกชื่อ Performer ทั้งหมดที่ต้องการ และกดยืนยัน</p> <p>7. Backstage ตัดสินใจยกเลิกการ assign โดยกดย้อนกลับ</p> <p>8. สิ้นสุด alternative flow</p>
Exceptions	<p>เพลงยังไม่ได้ถูกเพิ่มลงไปในระบบ</p> <p>1. Backstage เลือก "Event" จากหน้า Main Menu</p> <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด <p>2. Backstage เลือกงานงานการแสดงดนตรีที่เกี่ยวข้อง</p> <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก <p>3. Backstage กดปุ่ม "ดูรายชื่อเพลง"</p> <p>4. ระบบแสดงรายการเพลง</p> <p>5. Backstage ไม่พบเพลงที่ต้องการ</p> <ul style="list-style-type: none"> ○ Backstage ดำเนินการตาม Use Case: เพิ่มเพลงใน Queue เพื่อเพิ่มเพลงที่ต้องการ <p>6. ระบบกลับไปที่หน้ารายการเพลง</p>

	<ul style="list-style-type: none"> ○ Backstage เลือกเพลงที่เพิ่งเพิ่มและดำเนินการ Assign เพลงต่อไป <p>7. Use Case ดำเนินการตามปกติ</p>
Priority	High
Frequency of use	4-12 ครั้งต่อเพลง

3.3.1.13 รายละเอียดกรณี Run event

Use case ID	13
Use case name	Run event
Primary actor	Backstage
Description	Backstage สามารถเริ่มงานแสดงดนตรีที่เลือกได้เมื่อถึงวันแสดงจริงตามที่ได้กรอกไว้ภายในระบบ เพื่อทำการรันคิว เพลงและเปิดระบบการแจ้งเตือนให้กับ performer
Trigger	Backstage ต้องการเริ่มงานแสดงดนตรี ในเวลาหนึ่ง
Preconditions	PRE-1 Backstage ยืนยันตัวตนสำเร็จ PRE-2 Backstage มีการเพิ่มงานแสดงดนตรีลงไว้ในระบบ และ PRE-3 Backstage ได้เพิ่มเพลงลงไว้ในงานแสดงดนตรีนั้น ๆ และ PRE-4 วันที่ต้องการ run event ตรงกับวันจัด event ที่ได้กรอกไว้ภายในระบบ
Postconditions	POST-1 Backstage ทำการเริ่มงานแสดงดนตรีสำเร็จ
Normal flow	1. Backstage เลือก "Event" จากหน้า Main Menu

	<ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมดที่เกี่ยวข้อง <p>2. Backstage เลือกงานงานแสดงดนตรีจากรายการงานแสดงดนตรีที่มี</p> <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก <p>3. Backstage กดปุ่ม "เริ่ม Event"</p>
Alternative flow	-
Exceptions	<p>Backstage เลือก Event ไม่ตรงกับวันจริง</p> <p>1. Backstage เลือก "Event" จากหน้า Main Menu</p> <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมดที่เกี่ยวข้อง <p>2. Backstage เลือกงานงานแสดงดนตรีจากรายการที่มี</p> <ul style="list-style-type: none"> ○ ระบบทำการแสดงหน้ารายละเอียดของงานแสดงดนตรีที่เลือก <p>3. Backstage กดปุ่ม "เริ่ม Event"</p> <ul style="list-style-type: none"> ○ ระบบแสดงข้อความการแจ้งเตือน "ไม่สามารถเริ่ม Event ได้เนื่องจากยังไม่ถึงวันกำหนดการ" <p>4. Backstage ตรวจสอบและเลือกงานแสดงดนตรีให้ถูกต้องอีกครั้ง</p> <p>5. Use case ดำเนินตามปกติ</p>
Priority	High

Frequency of use	1 ครั้งต่อ 1 งานดนตรี
------------------	-----------------------

3.3.1.14 รายละเอียดกรณีเลือกเพลงถัดไป

Use case ID	14
Use case name	เลือกเพลงถัดไป
Primary actor	Backstage
Description	Backstage สามารถเลือกเพลงถัดไปในการแสดงสดเพื่อเป็นการรับคิวเพลง
Trigger	เพลงปัจจุบันจบแล้วและ backstage ต้องการเปลี่ยนไปเพลงถัดไป
Preconditions	PRE-1 Backstage ทำการยืนยันตัวตนสำเร็จ PRE-2 Backstage จะต้องทำการกด run event นั้น ๆ แล้ว
Postconditions	POST-1 เพลงที่กำลังเล่นมีการเปลี่ยนแปลง
Normal flow	<ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงาน Event ที่ได้กด run ไปแล้ว <ul style="list-style-type: none"> ○ ระบบจะแสดงว่ากำลังเล่นเพลงอะไรอยู่ ณ ตอนนี้ ด้วยการไฮไลท์สี พร้อมปุ่มแจ้งเตือน และปุ่มสำหรับเลือกเพลงถัดไป 3. Backstage กดปุ่ม “เพลงต่อไป” <ul style="list-style-type: none"> ○ ระบบขึ้นข้อความ “คุณต้องการเลื่อนไปเพลงถัดไปหรือไม่” พร้อมปุ่มยืนยันและยกเลิก 4. Backstage ทำการกดยืนยัน

	<ul style="list-style-type: none"> ○ ระบบจะทำการเปลี่ยนเพลงเป็นเพลงถัดไป จากเพลงที่กำลังแสดงอยู่ตาม Queue ที่ได้สร้างไว้ 5. หากต้องการเปลี่ยนเพลงอีกครั้ง Backstage ทำขั้นตอน 3 - 4 6. Use case สิ้นสุดเมื่อเพลงใน event นั้นถึงเพลงสุดท้ายแล้ว
Alternative flow	<p>Backstage ยกเลิกการเปลี่ยนเพลง</p> <ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมด 2. Backstage เลือกงานงานแสดงดนตรีที่ได้กด run ไปแล้ว <ul style="list-style-type: none"> ○ ระบบจะแสดงว่ากำลังเล่นเพลงอะไรอยู่ ณ ตอนนี้ ด้วยการไฮไลท์สี พร้อมปุ่มแจ้งเตือน และปุ่มสำหรับเลือกเพลงถัดไป 3. Backstage กดปุ่ม “เพลงต่อไป” <ul style="list-style-type: none"> ○ ระบบขึ้นข้อความ “คุณต้องการเลื่อนไป เพลงถัดไปหรือไม่” พร้อมปุ่มยืนยันและยกเลิก 4. Backstage ทำการกดยกเลิก <ul style="list-style-type: none"> ○ ระบบจะไม่ทำการเปลี่ยนเพลงจะเป็นเพลงเดิมที่กำลังแสดงอยู่ 5. สิ้นสุด Alternative flows
Exceptions	-
Priority	High

Frequency of use	5 – 30 ครั้งต่อหนึ่งงานดนตรี
------------------	------------------------------

3.3.1.15 รายละเอียดกรณีแจ้งเตือน Queue ให้ Performer

Use case ID	15
Use case name	แจ้งเตือน queue ให้ Performer
Primary actor	Backstage
Description	Backstage สามารถกดแจ้งเตือนให้นักดนตรีในเพลงถัดไปโดยแจ้งเตือนผ่าน Discord direct message ด้วย Discord bot
Trigger	Backstage ต้องการแจ้งเตือนเพลงให้นักดนตรีระหว่างการจัดแสดงดนตรีในเวลานั้นๆ
Preconditions	PRE-1 Backstage ทำการยืนยันตัวตนสำเร็จ PRE-2 Backstage ได้ทำการเพิ่ม event ลงในระบบแล้ว PRE-3 Backstage ทำการสร้างคิวเพลงและ assign performer ครบเรียบร้อยแล้ว PRE-4 Backstage ทำการกด run event แล้ว
Postconditions	POST-1 performer ที่เกี่ยวข้องได้รับแจ้งเตือนผ่าน discord direct message
Normal flow	<ol style="list-style-type: none"> 1. Backstage เลือก "Event" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงดนตรีทั้งหมดที่เกี่ยวข้อง 2. Backstage เลือกงานงานแสดงดนตรีที่ได้กด run ไปแล้ว

	<ul style="list-style-type: none"> ○ ระบบจะแสดงว่ากำลังเล่นเพลงอะไรอยู่ ณ ตอนนี้ ด้วยการไฮไลท์สี พร้อมปุ่มแจ้งเตือน และปุ่มสำหรับเลือกเพลงถัดไป <p>3. Backstage กดปุ่มแจ้งเตือน</p> <ul style="list-style-type: none"> ○ ระบบแสดงหน้าฟอร์มโดยสามารถเลือกกรอกข้อความเพิ่มเติมถึง performer ได้ <p>4. Backstage ทำการกดปุ่ม "แจ้งเตือน Performer"</p> <ul style="list-style-type: none"> ○ ระบบจะทำการส่งข้อมูลติดต่อไปที่ discord bot ○ discord bot จะส่ง direct message ไปให้ performer ที่เกี่ยวข้องของเพลงนั้นๆ ○ performer ได้รับการแจ้งเตือนเป็นข้อความว่า "เพลงต่อไปเป็นเพลงของคุณแล้วกรุณามาเตรียมตัว" ○ ระบบจะแสดงข้อความ "แจ้งเตือนสำเร็จ" <p>5. หากต้องการส่งแจ้งเตือนเพลงอื่น ๆ</p> <ul style="list-style-type: none"> ○ ระบบจะพาไปที่หน้า queue เพลง <p>6. Backstage ทำ use case เลือกเพลงถัดไป</p> <p>7. Backstage ทำ嘛ขั้นตอน 2 - 4</p> <p>8. Use case จะสิ้นสุดเมื่อ Event นั้นจบลงแล้ว</p>
Alternative flow	<p>Backstage ต้องการเขียน notification message ด้วยตนเอง</p> <p>1. Backstage เลือก "Event" จากหน้า Main Menu</p>

	<ul style="list-style-type: none"> ○ ระบบแสดงรายการงานแสดงตอนต่อไปที่ทั้งหมดที่เกี่ยวข้อง
	<p>2. Backstage เลือกงานงานแสดงตอนต่อไปที่ได้กด run ไปแล้ว</p> <ul style="list-style-type: none"> ○ ระบบจะแสดงว่ากำลังเล่นเพลงอะไรอยู่ ณ ตอนนี้ ด้วยการไฮไลท์สี พร้อมปุ่มแจ้งเตือน และปุ่มสำหรับเลือกเพลงถัดไป
	<p>3. Backstage กดปุ่มแจ้งเตือน</p> <ul style="list-style-type: none"> ○ ระบบแสดงหน้าฟอร์มโดยสามารถเลือกรอขอข้อความเพิ่มเติมถึง performer ได้
	<p>4. Backstage ทำการรอขอข้อความส่งไปที่ performer</p> <ul style="list-style-type: none"> ○ Backstage ทำการกดปุ่ม "แจ้งเตือน Performer" ○ ระบบทำการติดต่อส่งข้อมูลไปที่ discord bot ○ Discord bot ส่งข้อความไปให้ Performer ที่เกี่ยวข้องในเพลงนั้น ๆ ○ performer ได้รับการแจ้งเตือนเป็นข้อความว่า "เพลงต่อไปเป็นเพลงของคุณแล้วกรุณามาเตรียมตัว" และต่อด้วยข้อความเพิ่มเติมที่ Backstage ได้กรอกมา ○ ระบบแสดงข้อความ "แจ้งเตือนสำเร็จ"
	<p>5. หากต้องการส่งแจ้งเตือนเพลงอื่นๆ</p> <ul style="list-style-type: none"> ○ ระบบจะพาไปที่หน้า queue เพลง
	<p>6. Backstage ทำ use case เลือกเพลงถัดไป</p>

	<p>7. Backstage ทำขั้นตอน 3 - 5</p> <p>8. Alternative จะสิ้นสุดแล้วกลับสู่ Normal flow</p>
Exceptions	-
Priority	High
Frequency of use	5 - 30 ครั้งต่อหนึ่งงานดนตรี

3.3.1.16 รายละเอียดกรณีเคลียร์ User ออกจากระบบ

Use case ID	16
Use case name	เคลียร์ user ออกจากระบบ
Primary actor	System admin
Description	System admin สามารถเคลียร์ User ที่จบการศึกษาไปแล้วออกจากระบบได้
Trigger	System admin ต้องการเคลียร์ User ที่จบการศึกษาแล้วออกจากระบบ
Preconditions	<p>PRE-1 System admin รู้ข้อมูลว่า User คนไหนได้จบการศึกษาไปแล้ว</p> <p>PRE-2 System admin ยืนยันตัวตนสำเร็จ</p>
Postconditions	<p>POST-1 User นั้น ๆ ไม่สามารถเข้าสู่ระบบได้อีก</p> <p>POST-2 บัญชีของ User นั้น ๆ ถูกตั้งสถานะเป็น isActive เป็น false ในฐานข้อมูล</p>
Normal flow	<p>1. System admin เลือก "Account Config" จากหน้า Main Menu</p> <ul style="list-style-type: none"> ○ ระบบแสดงชื่อผู้ใช้ที่ยังใช้งานในระบบได้อยู่

	<p>2. System admin เลือกผู้ใช้จากรายการชื่อทั้งหมดที่มี</p> <ul style="list-style-type: none"> ○ System admin ทำการกด checkbox ตามบัญชีที่ต้องการทำการปิดบัญชี <p>3. System admin กดปุ่ม "ปิดบัญชี"</p> <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "ยืนยันการปิดบัญชีหรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" <p>4. System admin กดยืนยัน</p> <ul style="list-style-type: none"> ○ ระบบตั้งสถานะบัญชีนั้น ๆ เป็นปิดการใช้งาน ○ ระบบแสดงข้อความแจ้งเตือน "ปิดบัญชีสำเร็จ" <p>5. หากต้องการปิดบัญชีอื่น ๆ เพิ่มเติม:</p> <ul style="list-style-type: none"> ○ ระบบนำ System admin กลับไปที่หน้ารายชื่อผู้ใช้ <p>6. System admin ทำขั้นตอน 2 - 4</p> <p>7. Use Case สิ้นสุดเมื่อ System admin ไม่ต้องปิดบัญชีแล้ว</p>
Alternative flow	<p>System admin ยกเลิกการปิดบัญชี</p> <p>1. System admin เลือก "Account Config" จากหน้า Main Menu</p> <ul style="list-style-type: none"> ○ ระบบแสดงชื่อผู้ใช้ที่ยังใช้งานในระบบได้อยู่ <p>2. System admin เลือกผู้ใช้จากรายการชื่อทั้งหมดที่มี</p>

	<ul style="list-style-type: none"> ○ System admin ทำการกด checkbox ตามบัญชีที่ต้องการทำการปิดบัญชี <p>3. System admin กดปุ่ม "ปิดบัญชี"</p> <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "ยืนยันการปิดบัญชีหรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" <p>4. System admin กดปุ่มยกเลิก</p> <p>5. บัญชีจะยังสามารถใช้งานได้ปกติ</p> <p>6. Alternative Flow สิ้นสุด</p>
Exceptions	-
Priority	Low
Frequency of use	0 – 10 ครั้งต่อปี

3.3.1.17 รายละเอียดกรณีเพิ่มบัญชีผู้ใช้เข้าสู่ระบบ

Use case ID	17
Use case name	เพิ่มบัญชีผู้ใช้เข้าสู่ระบบ
Primary actor	System admin
Description	System admin สามารถเพิ่ม User ใหม่ให้สามารถเข้าใช้งานระบบได้
Trigger	System admin ต้องการเพิ่ม User เพื่อใช้งานในระบบ
Preconditions	PRE-1 System admin มีข้อมูล Google form ที่สมาชิกได้ทำการกรอกตอนสมัครเข้า Sci Band PRE-2 System admin ยืนยันตัวตนสำเร็จ

	PRE-3 System admin และสมาชิกต้องทำการตรวจสอบความถูกต้องของข้อมูลใน Google form ก่อน
Postconditions	POST-1 ข้อมูลบัญชีผู้ใช้ถูกเพิ่มลงในฐานข้อมูล POST-2 User นั้น ๆ สามารถเข้าสู่ระบบได้
Normal flow	<ol style="list-style-type: none"> 1. System admin เลือก "Account Config" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงหน้าเพิ่มบัญชีผู้ใช้ใหม่ ซึ่งมีปุ่มเพิ่มบัญชี 2. System admin กดปุ่มเพิ่มบัญชี 3. ระบบแสดงข้อความ “เพิ่มบัญชีสำเร็จ” 4. Use Case สิ้นสุด
Alternative flow	-
Exceptions	-
Priority	High
Frequency of use	1 - 5 ครั้งต่อปี

3.3.1.18 รายละเอียดกรานีดู Profile

Use case ID	18
Use case name	ดู Profile
Primary actor	User ทุกประเภท
Description	User ทุกประเภทสามารถดู Profile ของตัวเองได้ว่าข้อมูลถูกต้องหรือไม่
Trigger	User ต้องการดู Profile ของตัวเอง

Preconditions	PRE-1 System admin มีการเพิ่มบัญชีผู้ใช้งานเป็นระบบ แล้ว PRE-2 User ยืนยันตัวตนสำเร็จ
Postconditions	POST-1 User ตรวจสอบดู Profile สำเร็จ
Normal flow	<ol style="list-style-type: none"> 1. User เลือก "Profile" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงหน้า Profile ของผู้ใช้ 2. User ทำการดูรายละเอียด Profile ของตัวเองโดย ประกอบไปด้วยข้อมูล ชื่อ-นามสกุล, ชื่อเล่น, ตำแหน่งในระบบ, discord Id, discord username
Alternative flow	-
Exceptions	-
Priority	Medium
Frequency of use	0 - 5 ครั้งต่อปี

3.3.1.19 รายละเอียดกรณี Logout

Use case ID	19
Use case name	Logout
Primary actor	User ทุกประเภท
Description	User สามารถ logout ออกจากระบบเมื่อไม่ใช้งาน
Trigger	User ต้องการ logout ออกจากระบบ
Preconditions	PRE-1 User ได้ Login ค้างไว้ในแอปพลิเคชัน
Postconditions	POST-1 User ออกจากระบบสำเร็จ
Normal flow	<ol style="list-style-type: none"> 1. User เลือก "Profile" จากหน้า Main Menu

	<ul style="list-style-type: none"> ○ ระบบแสดงหน้า Profile ของผู้ใช้ <p>2. User ทำการกดปุ่ม "Logout"</p> <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "คุณต้องการออกจากระบบหรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" <p>3. User กดยืนยัน</p> <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "ออกจากระบบสำเร็จ" ○ ระบบนำผู้ใช้ไปสู่หน้าเข้าสู่ระบบ
Alternative flow	User ยกเลิกการออกจากระบบ <ul style="list-style-type: none"> 1. User เลือก "Profile" จากหน้า Main Menu ○ ระบบแสดงหน้า Profile ของผู้ใช้ <p>2. User ทำการกดปุ่ม "Logout"</p> <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "คุณต้องการออกจากระบบหรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" <p>3. User กดยกเลิก</p> <p>4. Alternative Flow สิ้นสุด</p>
Exceptions	-
Priority	Medium
Frequency of use	0 – 3 ครั้งต่อปี

3.3.1.20 รายละเอียดกรณีโอนย้ายสิทธิ์ system admin

Use case ID	20
Use case name	โอนย้ายสิทธิ์ system admin

Primary actor	System admin
Description	System admin ต้องการโอนย้ายสิทธิ์ผู้ดูแลให้คนในชุมชน ในกรณีที่หมดวาระหรือออกจากชุมชน
Trigger	System admin ต้องการโอนย้ายสิทธิ์ System admin ให้สมาชิกคนอื่น
Preconditions	PRE-1 ชุมชน Sci Band ได้มีการพูดคุยตกลงกันแล้วว่าจะมอบสิทธิ์ System admin ให้สมาชิกคนใด
Postconditions	POST-1 สิทธิ์ System admin ถูกโอนย้ายไปยังบัญชีใหม่ POST-2 System admin คนเก่าไม่มีสิทธิ์ System admin อีกต่อไป
Normal flow	<ol style="list-style-type: none"> 1. System admin เลือก "Transfer System Admin" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงหน้า Transfer System Admin ซึ่งมีรายชื่อบัญชีผู้ใช้ทั้งหมดที่ยังสามารถใช้งานในระบบได้อยู่ 2. System admin เลือกชื่อสมาชิกจากรายการชื่อทั้งหมดที่มี <ul style="list-style-type: none"> ○ System admin ทำเลือกผู้ใช้ที่ต้องการให้สิทธิ์โดยสามารถโอนย้ายให้ได้เพียงคนเดียวเท่านั้น 3. System admin กดปุ่ม "โอนย้ายสิทธิ์" <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "โปรดยืนยันการโอนสิทธิ์ System admin การดำเนินการนี้จะไม่สามารถย้อนกลับได้" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" 4. System admin กดยืนยัน

	<ul style="list-style-type: none"> ○ ระบบตั้งสถานะบัญชีนั้น ๆ เป็น System admin ในฐานข้อมูล ○ ระบบจะทำการลบสิทธิ์ System admin คนเก่าออกจากระบบ ○ ระบบแสดงข้อความแจ้งเตือน "โอนย้าย สิทธิ์สำเร็จ"
Alternative flow	<p>System admin ยกเลิกการโอนย้ายสิทธิ์</p> <ol style="list-style-type: none"> 1. System admin เลือก "Transfer System Admin" จากหน้า Main Menu <ul style="list-style-type: none"> ○ ระบบแสดงหน้า Transfer System Admin ซึ่งมีปุ่ม 2. System admin เลือกชื่อสมาชิกจากการชื่อห้องหมวดที่มี <ul style="list-style-type: none"> ○ System admin ทำเลือกผู้ใช้ที่ต้องการให้สิทธิ์โดยสามารถโอนย้ายให้ได้เพียงคนเดียวเท่านั้น 3. System admin กดปุ่ม "โอนย้ายสิทธิ์" <ul style="list-style-type: none"> ○ ระบบแสดงข้อความแจ้งเตือน: "โปรดยืนยันการโอนสิทธิ์ System admin การดำเนินการจะไม่สามารถย้อนกลับได้" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" 4. System admin กดยกเลิก 5. Alternative Flow สิ้นสุด
Exceptions	-
Priority	High
Frequency of use	1 ครั้งต่อปี

3.4 ประเด็นที่น่าสนใจและสิ่งที่ท้าทาย

3.4.1 การจัดการบทบาทภายในระบบ

การเป็นสมาชิก SCI BAND นั้น ตัวสมาชิกสามารถเป็นได้มากกว่า 1 บทบาท เช่น เป็นทั้ง Backstage และนักร้อง ซึ่งทั้ง 2 บทบาทนี้นั้นจะมีสิทธิ์ในการใช้งานระบบที่แตกต่างกัน จึงได้กำหนดให้ระบบมีการตรวจสอบบทบาททั้งหมดของสมาชิกนั้น ๆ และให้สิทธิ์ใช้งานระบบตามบทบาททั้งหมดที่ตัวเองเป็น ทำให้บัญชีนั้น ๆ สามารถใช้งานข้ามบทบาทได้โดยไม่จำเป็นต้องเลือกบทบาทตอนเข้าสู่ระบบเพื่อความสะดวกสบายและป้องกันการสับสนในการใช้งาน

3.4.2 การเชื่อมต่อกับ Discord

3.4.2.1 การแจ้งเตือนไปยัง Discord

ระบบแจ้งเตือนคิวเพลงนั้นจะเชื่อมต่อกับ Discord ผ่านทาง Discord bot โดยจะต้องมีการ Implement และกำหนดสิทธิ์ของ Discord bot ให้สามารถส่งข้อความไปยังผู้ใช้โดยตรงได้ รวมไปถึงการเชื่อมต่อระบบกับ Discord นั้นจำเป็นต้องใช้ Discord ID ซึ่งเป็น ID เฉพาะของแต่ละบัญชี โดยสามารถเก็บข้อมูล Discord ID ได้ผ่านการเข้าสู่ระบบผ่าน Discord

3.4.2.2 การจัดการผู้ใช้ใน Discord

ภายในໂປຣຈັນມີການເພີ່ມຜູ້ໃຊ້ເຂົ້າສູ່ Discord channels โดยອັຕໂນມັດໃນຄວັງແຮກທີ່ຜູ້ໃຊ້ໄດ້ເຂົ້າສູ່ระบบ และມີກາລບຜູ້ໃຊ້ອອກຈາກ Discord server โดยອັຕໂນມັດທີ່ລັງຈາກບัญชືຜູ້ໃຊ້ນັ້ນ ຈຸກປິດໃຫ້ຈານ ໂດຍການເພີ່ມແລະລັບຜູ້ໃໝ່ນັ້ນ ຮະບບທີ່ອກການ Channel ID, Discord ID ຂອງຜູ້ໃຊ້ ລວມໄປຖືກການกำหนดสิทธີ່ຂອງ Discord bot ໃຫ້ສາມາດຈັດການເພີ່ມຜູ້ໃຊ້ເຂົ້າສູ່ Discord channels ແລະລັບຜູ້ໃຊ້ອອກຈາກ Discord server ໄດ້ ຈຶ່ງຈະສາມາດຈັດການຜູ້ໃຊ້ກາຍໃນ Discord ໄດ້ຕາມທີ່ຕ້ອງການ

3.5 ผลลัพธ์ที่คาดหวัง

3.5.1 การจัดงานแสดงเป็นไปได้อย่างราบรื่น

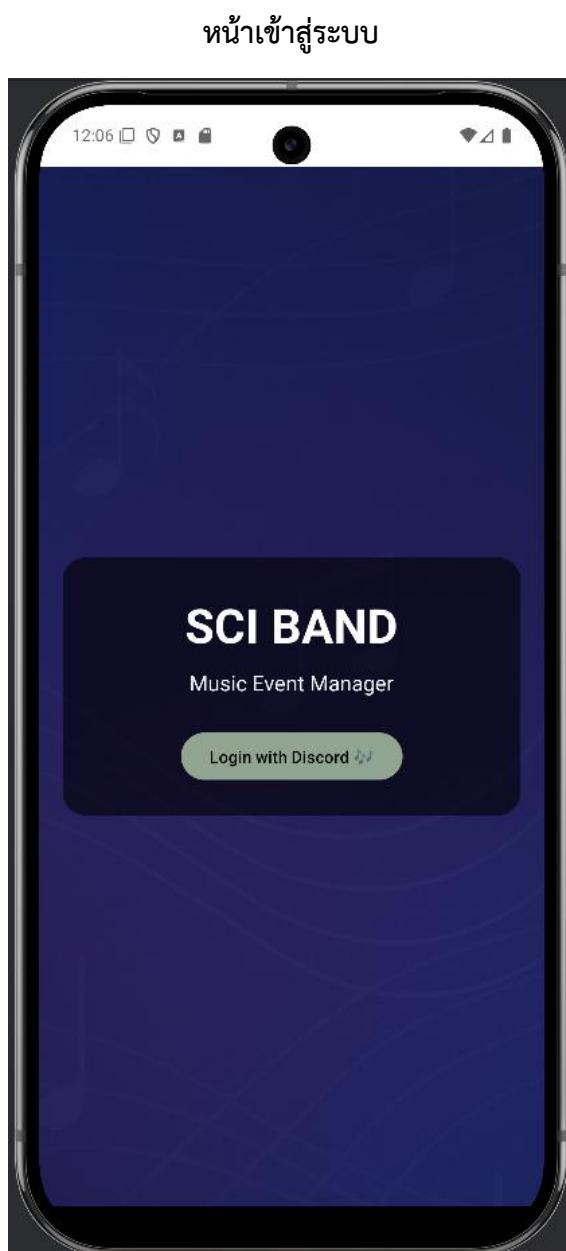
ระบบจะช่วยในการลดความสับสนของคิวการแสดงของ Performer ແຕ່ລະຄນ ທຳໄທການแสดงเป็นไปได้อย่างราบรื่น ຕິດขັດນ້ອຍລົງ ແລະช່ວຍລົດຄວາມສັບສົນໃນການທຳການເຊັ່ນເມື່ອ Staff

ต้องการทราบว่าบุคลากรของคนใดร้องเพลงนั้น ๆ เนื่องจากต้องการให้ Script พูดกับบุคลากร Staff สามารถเข้าสู่ระบบเพื่อดูว่าบุคลากรของเพลงนั้นคือบุคคลใดได้ โดยการแสดงที่ราบรื่นออกจากจะทำให้ SCI BAND ทำงานได้อย่างมีประสิทธิภาพแล้วนั้นยังสามารถเพิ่มชื่อเสียงให้กับ SCI BAND ได้ เนื่องจากการแสดงเป็นไปได้อย่างราบรื่นทำให้ผู้ชมนั้นรู้สึกสนุกไปกับงานดนตรีมากยิ่งขึ้น

3.5.2 การสื่อสารและการเก็บข้อมูลที่ชัดเจนยิ่งขึ้น

จากการใช้งาน Discord ใน การสื่อสารทำให้ข้อมูลเป็นระเบียบและรวมอยู่ในที่เดียว กันได้ภายใน Discord server และการเก็บไฟล์ที่ไม่มีวันหมดอายุทำให้การสื่อสาร มีประสิทธิภาพมากขึ้น รวมถึงการใช้ Discord bot ในการเพิ่มผู้ใช้เข้า Channels ที่ตนเองเกี่ยวข้อง โดยอัตโนมัติช่วยลดปัญหาในการที่ Staff ต้องคอยตรวจสอบว่าสมาชิกใหม่ของ SCI BAND นั้นเข้าร่วม กลุ่มไลน์ที่เกี่ยวข้องครบแล้วหรือไม่ และระบบยังมีการจัดเก็บข้อมูลเกี่ยวกับเพลงที่ Performer จำเป็นต้องรู้ เช่น คีย์เพลง , อ้างอิงเพลง รวมถึง Lineup performer ของเพลงนั้น ๆ ช่วยให้การซ้อม ทั้งส่วนรวมและส่วนตัวนั้นมีประสิทธิภาพยิ่งขึ้น

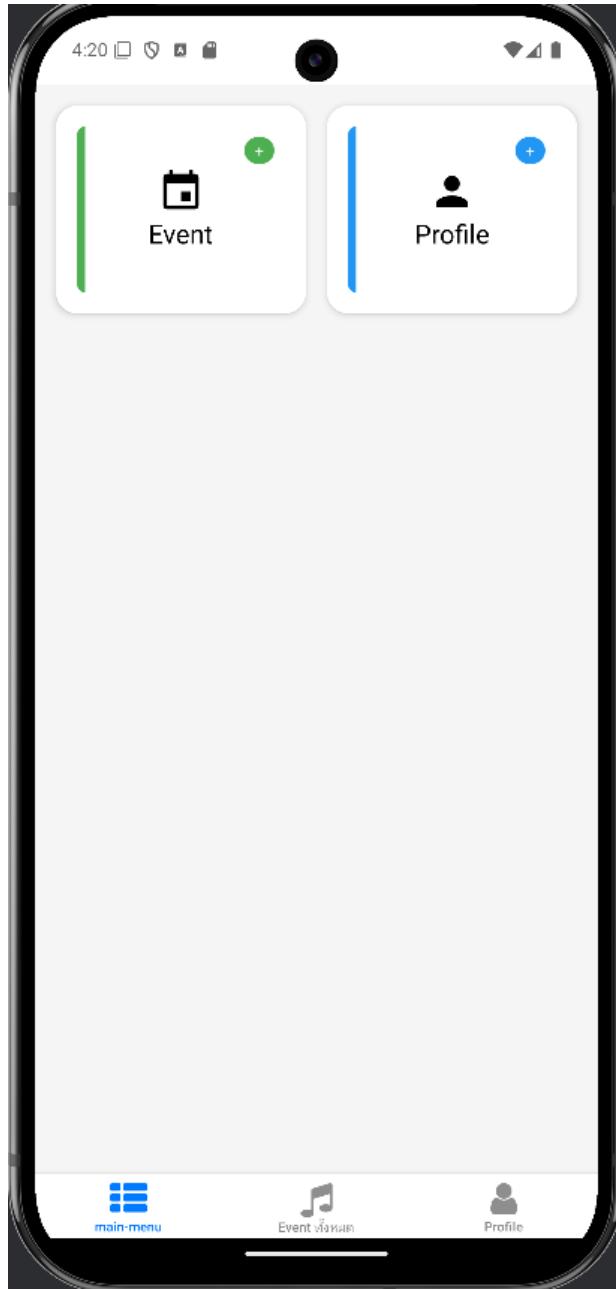
3.6 ระบบตั้นแบบและผลลัพธ์เบื้องต้น



3.6.1 หน้าเข้าสู่ระบบ

หน้า Login เป็นจุดเริ่มต้นที่ผู้ใช้งานต้องยืนยันตัวตนก่อนเข้าสู่ระบบ เพื่อให้สามารถเข้าถึงข้อมูลหรือฟีเจอร์ต่าง ๆ ของแอป Band-Sync Sci ได้อย่างปลอดภัย โดยระบบจะตรวจสอบข้อมูลที่ผู้ใช้กรอกโดยต้องกรอก Username และ Password ของบัญชี Discord โดย Username ต้องตรงตามที่ผู้ใช้ได้กรอกไว้ใน Google Forms สมัครเข้า SCI BAND

หน้า Main menu สำหรับ Performer



3.6.2 หน้า Main menu สำหรับ Performer

หน้าหลักของ Performer เป็นศูนย์กลางที่รวบรวมข้อมูลและฟีเจอร์สำคัญของแอป Band-Sync Sci เพื่อให้ผู้ใช้งานสามารถเข้าถึงส่วนต่าง ๆ ได้สะดวกและรวดเร็ว โดยมีการจัดวางอย่างชัดเจนและตอบสนองต่อความต้องการของผู้ใช้งาน ประกอบไปด้วย Event : โซนหน้า Event ที่

จะกำลังจะถึงทั้งหมด My Event : โชว์หน้า Event ที่เกี่ยวข้องกับบัญชีผู้ใช้โดยเฉพาะ My Profile : หน้า Profile บอกรายละเอียดข้อมูลส่วนตัวของผู้ใช้

หน้า Event สำหรับ Performer & Staff & System admin



3.6.3 หน้า Event สำหรับ Performer & Staff & System admin

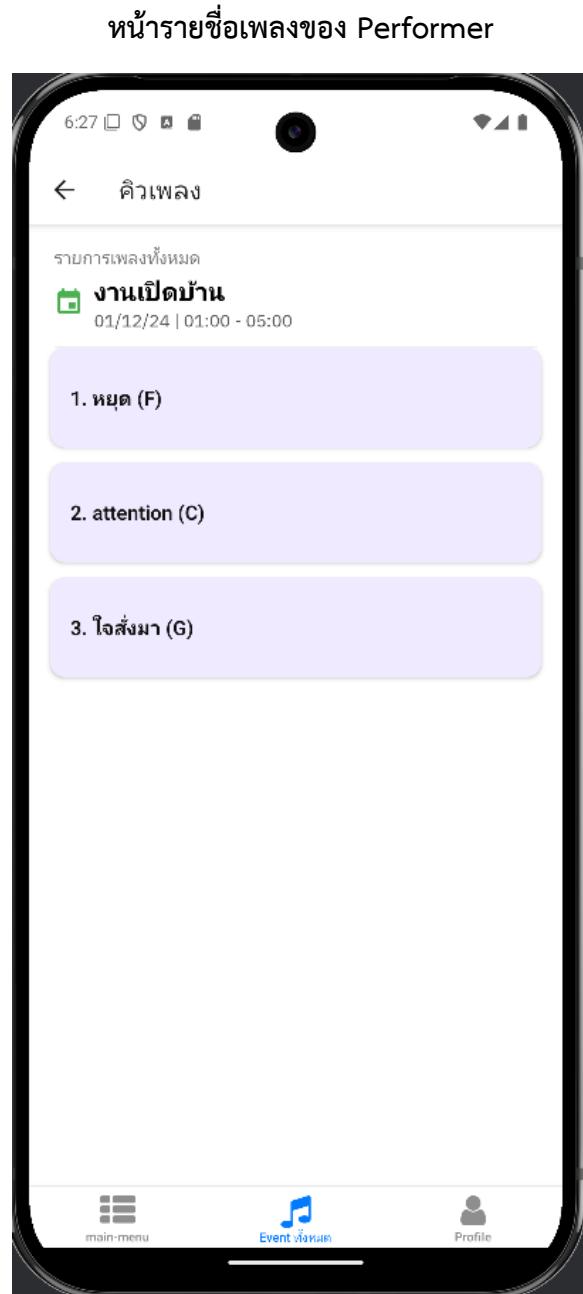
หน้านี้คือหน้า Event สำหรับ Performer Staff และ System admin ซึ่งจะแสดง Event ที่มีทั้งหมด ซึ่งผู้ใช้งานสามารถกดคล่อง Event เพื่อดูรายละเอียด Event นั้น ๆ ได้

หน้ารายละเอียด Event สำหรับ Performer & Staff & System admin



3.6.4 หน้ารายละเอียด Event สำหรับ Performer & Staff & System admin

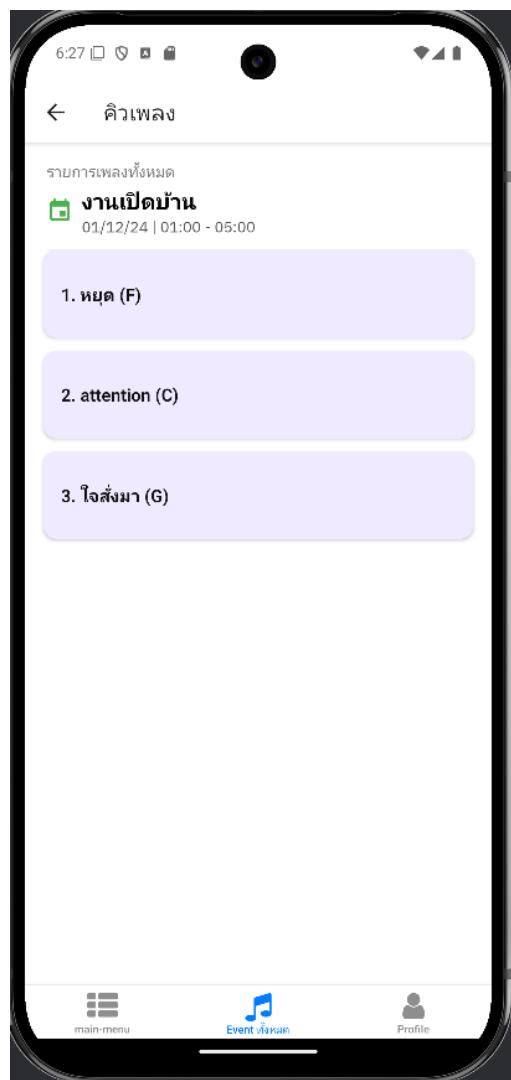
หน้านี้คือหน้ารายละเอียด Event จะประกอบไปด้วยชื่อ Event, วันจัดแสดง, เวลาเริ่มงาน, เวลาสิ้นสุดงาน, การแต่งกายในวันงาน (Dress code) และรายละเอียดเพิ่มเติม และมีปุ่มดูรายชื่อเพลงเพื่อทำรายการถัดไป



3.6.5 หน้ารายชื่อเพลงของ Performer

หน้านี้บอกรายละเอียดชื่อเพลงทั้งหมดของ Event นั้น ๆ โดยมีไฟอร์คันหาชื่อเพลง และเพลงที่เกี่ยวข้องกับฉันเท่านั้น และสามารถกดชื่อเพลงเพื่อเข้าไปดูรายละเอียดเพลงได้

หน้ารายชื่อเพลงของ Staff & System admin



3.6.6 หน้ารายชื่อเพลงของ Performer

หน้านี้แสดงรายชื่อ Performer ในตำแหน่งที่ต้องการในเพลงนั้น ๆ

หน้ารายละเอียดเพลงของ Performer & Staff & System admin

← รายละเอียดเพลง

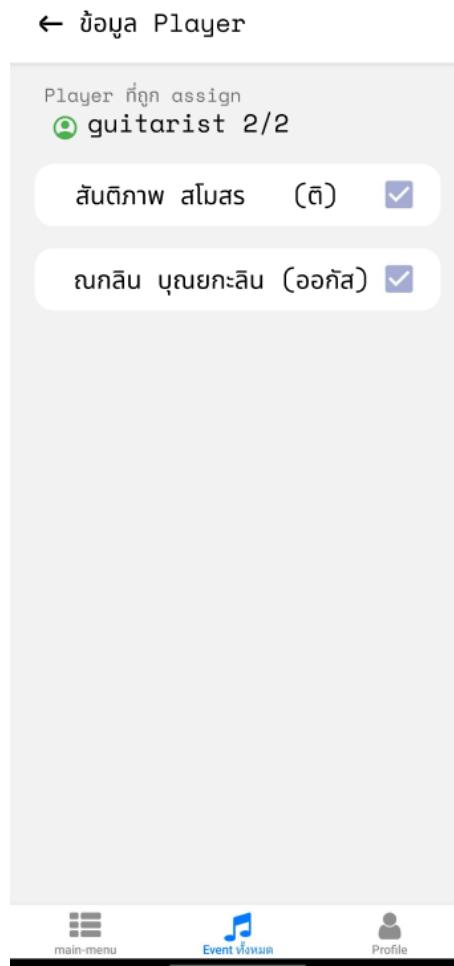
ชื่อเพลง คุกเข่า	key D
ref เพลง i.be/jqtwF4Z2i-0?si=j8GkeLPgdOON34kA	
รายละเอียดเพิ่มเติม (optional) ขั้น intro หลัง data จะมี	
จำนวนนักร้อง 1	ดูข้อมูล
จำนวนกีตาร์ 1	ดูข้อมูล
จำนวนเมส 1	ดูข้อมูล
จำนวนกลอง 1	ดูข้อมูล
จำนวนศิรบอร์ด 0	ดูข้อมูล
จำนวน extra 0	ดูข้อมูล
จำนวน percussion 1	ดูข้อมูล

main-menu
Event ทั้งหมด
Profile

3.6.7 หน้ารายละเอียดเพลงของ Performer & Staff & System admin

หน้านี้คือหน้ารายละเอียดเพลงที่ผู้ใช้เลือกจากภาพก่อนหน้าซึ่งจะบอกรายละเอียดตั้งนี้ ซึ่งเพลง จำนวนนักร้องนักดนตรีของเพลงนั้น, คีย์เพลง (Song Key) และ อ้างอิงเพลง (Reference) ซึ่งสามารถดูรายชื่อของนักร้องนักดนตรีได้ว่ามีใครบ้าง

หน้ารายชื่อนักร้องนักดนตรีของ Performer & Staff & System admin



3.6.8 หน้ารายชื่อนักร้องนักดนตรี Performer & Staff & System admin

หน้านี้แสดงรายชื่อผู้รับผิดชอบของการแสดงแต่ละตำแหน่งในเพลงนั้นๆ

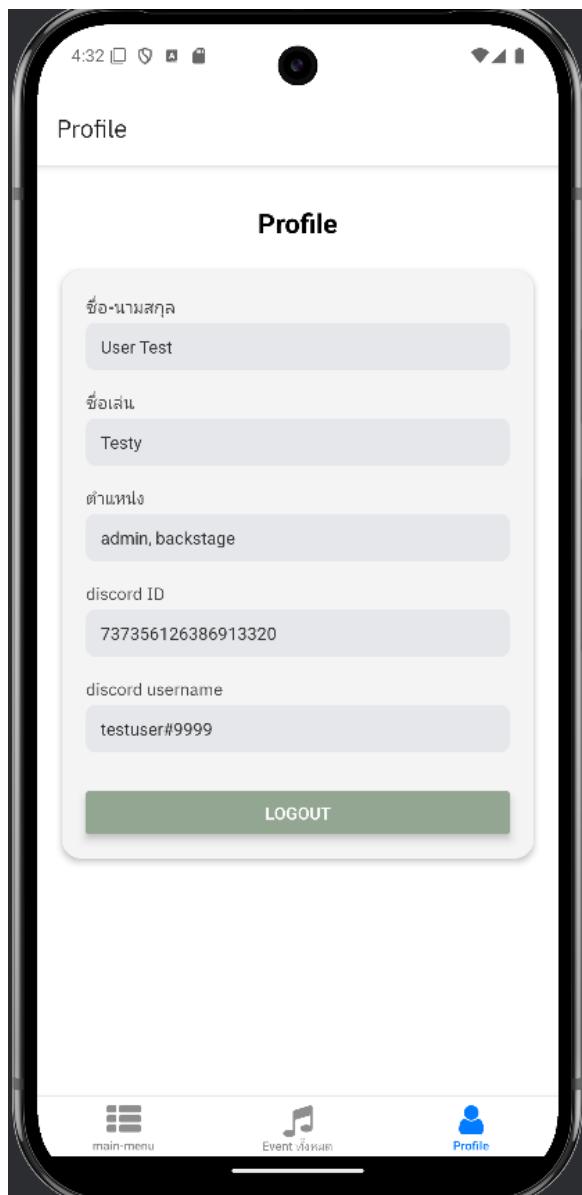
หน้า My Event ของ Performer



3.6.9 หน้า My Event ของ Performer

หน้านี้จะแสดงหน้า Event ที่เกี่ยวข้องกับตัว Performer เท่านั้น

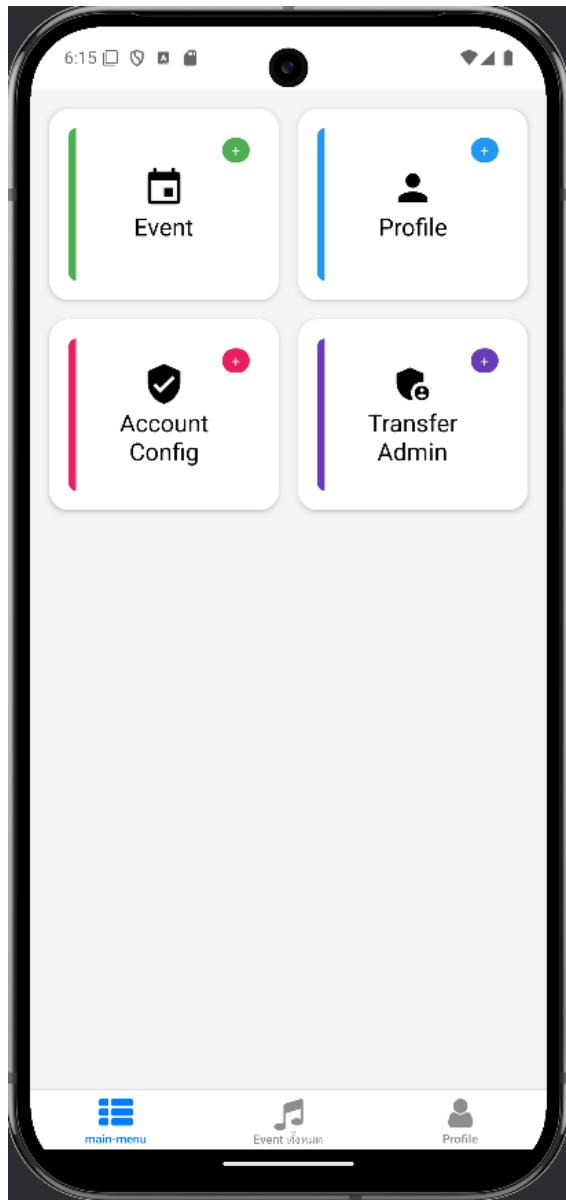
หน้า Profile สำหรับ User ทุกประเภท



3.6.10 หน้า Profile สำหรับ User ทุกประเภท

หน้านี้คือหน้าแสดงรายละเอียดบัญชีของ User โดยประกอบด้วย ชื่อ-นามสกุล, ชื่อเล่น, ตำแหน่ง, Discord ID, Discord username

หน้า Main Menu ของ System admin



3.6.11 หน้า Main Menu ของ System admin

หน้าหลักของ System admin เป็นศูนย์กลางที่รวบรวมข้อมูลและฟีเจอร์สำคัญสำหรับ System admin ประกอบไปด้วย Event : โซนหน้า Event ที่จะกำลังจะถึงทั้งหมด My Profile : หน้า Profile บอกรายละเอียดข้อมูลส่วนตัวของผู้ใช้ Import new members : หน้าสำหรับให้ System admin นำข้อมูลผู้ใช้เข้าสู่ระบบเพื่อให้ผู้ใช้งานเข้าใช้ระบบได้ Transfer system admin :

หน้าที่สามารถทำให้ System admin โอนย้ายสิทธิ์ผู้ดูแลให้คนในชุมชนในกรณีที่หมดภาระหรือออก
จากชุมชนได้ Account config: หน้าที่ทำให้ System admin สามารถเพิ่มและเคลียร์ User ได้

หน้า Transfer system admin privileges ของ System admin



3.6.12 หน้า Transfer system admin privileges ของ System admin

หน้านี้สามารถทำให้ System admin โอนย้ายสิทธิ์ผู้ใช้งานให้สมาชิกชุมชนเป็น System admin ได้โดยสามารถเลือกได้ 1 คนเท่านั้นหลังจากเลือกสมาชิกที่ต้องการแล้ว System admin ทำการกดปุ่มโอนย้ายสิทธิ์

หน้า Deactivate account ของ System admin



3.6.13 หน้า account config ของ System admin

หน้านี้ให้ System admin เพิ่มและปิดบัญชีผู้ใช้ ผู้ใช้ที่ถูกเพิ่มบัญชีเข้ามาในระบบจะสามารถใช้งานระบบได้ กลับกันผู้ใช้ที่โดนปิดบัญชีจะไม่สามารถเข้าใช้งานระบบได้ โดยสามารถเลือกซึ่งผู้ใช้และทำการกดปุ่มปิดบัญชีผู้ใช้

หน้ารวม Event สำหรับ Backstage



3.6.14 หน้ารวม Event สำหรับ Backstage

โดยหน้านี้โดยรวมจะเหมือนหน้ารวม Event ของ User อื่นๆที่ไปแต่ Backstage สามารถสร้าง Event ใหม่ได้โดยกดไอคอนรูป (+) และทางระบบจะขึ้นหน้าฟอร์มที่จะต้องกรอกเพื่อสร้าง Event ใหม่

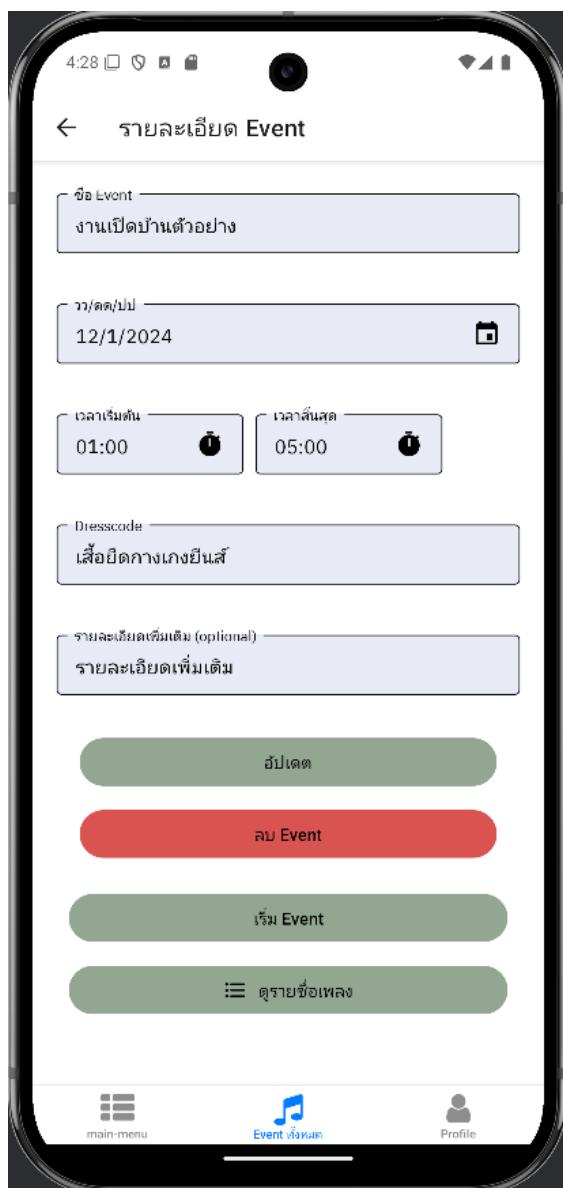
หน้าฟอร์มสร้าง Event สำหรับ Backstage



3.6.15 หน้าฟอร์มสร้าง Event ของ Backstage

หน้านี้เป็นหน้าสำหรับกรอกฟอร์มหลังจาก Backstage ทำการกดไอคอนรูป(+) จากหน้ารวม Event สำหรับ Backstage โดยจะต้องกรอก ชื่อ Event, วันจัดแสดง, เวลาเริ่มงาน, เวลาสิ้นสุดงาน, การแต่งกายในวันงาน (Dress code) และรายละเอียดเพิ่มเติม (ถ้ามี) หลังจากการกรอกเสร็จและทำการบันทึก

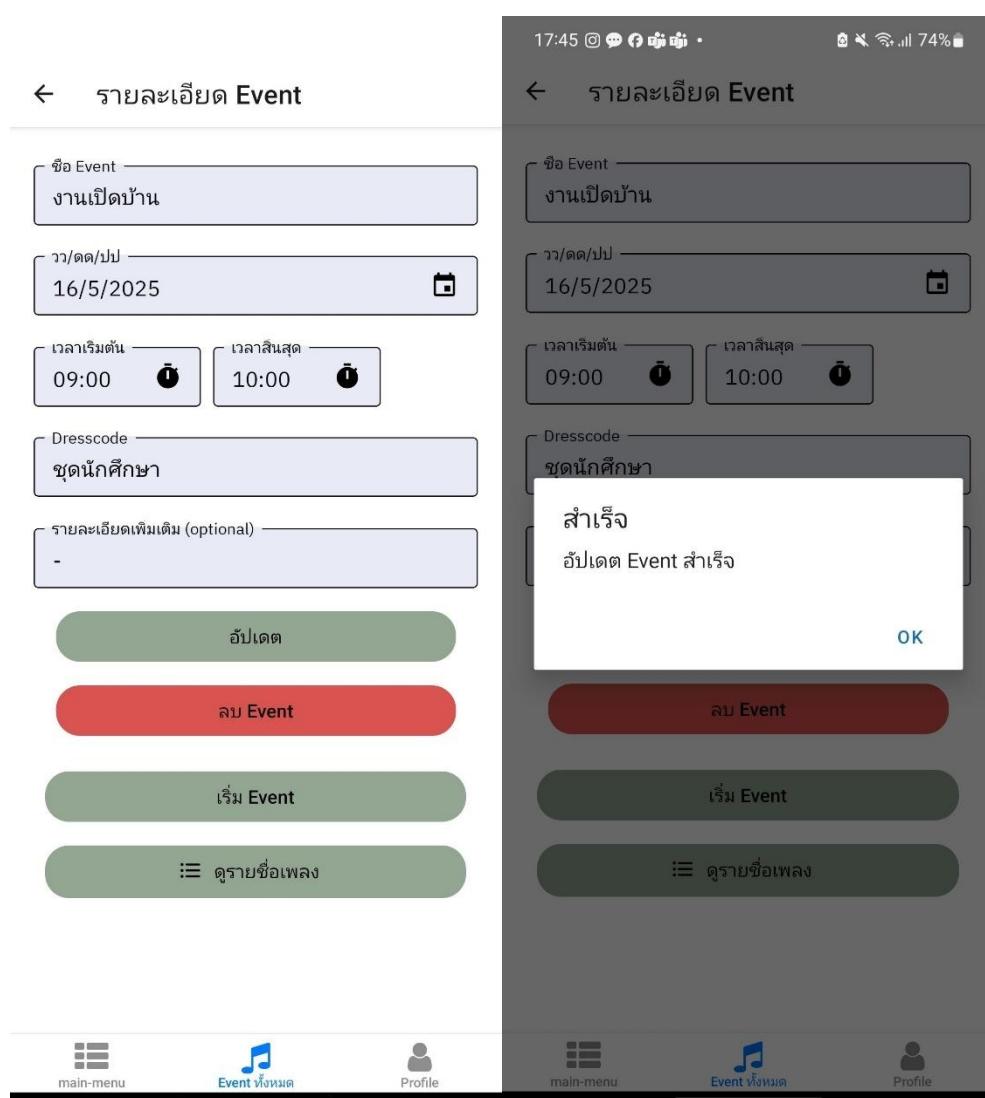
หน้ารายละเอียด Event ของ Backstage



3.6.16 หน้ารายละเอียด Event ของ Backstage

หน้านี้เป็นหน้าสำหรับกรายละเอียด Event เมื่อ User ที่ไม่ได้จัดแท็กต่างจากผู้ใช้ที่ Backstage ทำการแก้ไขได้โดยทำการแก้ไขตามข้อมูลต่างๆ โดยสามารถแก้ไขข้อมูลได้ทั้งหมด หลังจากแก้ไขแล้วกดปุ่มอัปเดตจะทำการอัปเดตฟอร์มตามที่แก้ไข และ Backstage ก็สามารถกดดูรายชื่อเพลงและสามารถแก้ไขรายชื่อเพลงได้อีกด้วย และสามารถเริ่มงาน Event ที่เลือกได้ แต่ถ้า Event ไม่ตรงตามเงื่อนไข เช่น วันที่ไม่ตรงกับเวลาจริง จะไม่สามารถกดเริ่ม Event ได้ หากตรงตามเงื่อนไขแล้ว Backstage กดเริ่มEvent และ Event จะเริ่มทันที และสามารถลบ Event ที่ดูอยู่ได้อีกด้วย

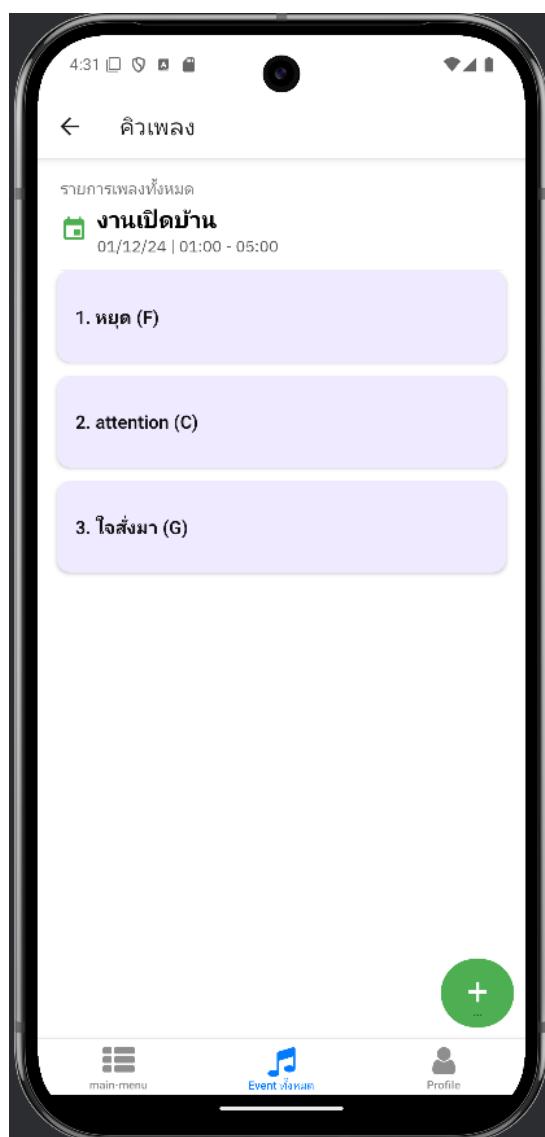
หน้าแก้ไข Event ของ Backstage



3.6.17 หน้าแก้ไข Event ของ Backstage

หน้านี้คือหน้าที่ Backstage ทำการแก้ไขโดยทำการแก้ไขตามข้อมูลต่างๆ สามารถแก้ไขรายละเอียดได้ทั้งหมด ถ้า Backstage แก้ไขเสร็จแล้ว Backstage จะต้องกดปุ่มอัปเดต และการกดยืนยันเป็นอันเสร็จสิ้น

หน้ารายชื่อเพลงของ Backstage



3.6.18 หน้ารายชื่อเพลงของ Backstage

หน้านี้จะแสดงรายชื่อเพลงทั้งหมดหลังจากที่ Backstage กดคูร่ายชื่อเพลงจากหน้าดูรายละเอียด Event โดยจะมีการเพิ่มฟีเจอร์ การย้าย queue เพลง และการเพิ่มเพลงเข้ามา โดยฟีเจอร์แรกการย้าย queue เพลง Backstage สามารถกดเพลงที่ต้องการย้ายค้างไว้และทำการเลื่อนได้ตามต้องการว่าหากให้เพลงอยู่ที่ลำดับไหน ฟีเจอร์ที่ 2 การเพิ่มเพลงโดยให้ Backstage สามารถเพิ่มเพลงได้จากการกดปุ่มไอคอนรูป(+) และทางระบบจะขึ้นฟอร์มการสร้างเพลงให้ Backstage กรอก

หน้าฟอร์มการสร้างเพลงของ Backstage

← สร้างเพลง

← สร้างเพลง

ชื่อเพลง —————
ติดฟัน

ref เพลง —————
<https://youtu.be/jqtwF4Z2i-0?si=j8GkeLP>

รายละเอียดเพิ่มเติม (optional) —————
-

จำนวนนักร้อง —————
2

จำนวนกีตาร์ —————
2

จำนวนเบส —————
1

จำนวนกลอง —————
1

จำนวนศิรบอร์ด —————
1

จำนวน extra —————
0

จำนวน percussion —————
1

ref เพลง —————
<https://youtu.be/jqtwF4Z2i-0?si=j8GkeLP>

รายละเอียดเพิ่มเติม (optional) —————
-

จำนวนนักร้อง —————
2

จำนวนกีตาร์ —————
2

จำนวนเบส —————
1

จำนวนกลอง —————
1

จำนวนศิรบอร์ด —————
1

จำนวน extra —————
0

จำนวน percussion —————
1

สร้างเพลง

main-menu
 Event ทั้งหมด
 Profile
 main-menu
 Event ทั้งหมด
 Profile

3.6.19 หน้าฟอร์มการสร้างเพลงของ Backstage

หน้านี้คือหน้าฟอร์มที่ให้ Backstage กรอกหลังจาก Backstage กดไอคอนรูป (+) หรือการเพิ่มเพลงโดย Backstage จะต้องกรอกชื่อเพลง, จำนวน Performer ที่ต้องการในเพลง, คีย์เพลง (Song key), อ้างอิงเพลง (Reference) และรายละเอียดเพิ่มเติม(ถ้ามี) ถ้ากรอกเสร็จแล้วให้ Backstage กดปุ่มสร้างเพลงเป็นอันสิ้นสุด

หน้ารายละเอียดแต่ละเพลงของ Backstage

รายละเอียดเพลง

ชื่อเพลง	key
คุกเข่า	D
ref เพลง	
<input type="text" value="https://youtu.be/jqtwF4Z2i-0?si=j8GkeLP"/>	
รายละเอียดเพิ่มเติม (optional)	
<input type="text" value="ขึ้น intro หลัง data จบ"/>	
จำนวนนักร้อง	
1	ดูข้อมูล
จำนวนกีตาร์	
1	ดูข้อมูล
จำนวนเบส	
1	ดูข้อมูล
จำนวนกลอง	
1	ดูข้อมูล
จำนวนศิริกอร์ด	
0	ดูข้อมูล
จำนวน extra	
0	ดูข้อมูล
จำนวน percussion	
1	ดูข้อมูล

รายละเอียดเพลง

รายละเอียดเพิ่มเติม (optional)	
<input type="text" value="ขึ้น intro หลัง data จบ"/>	
จำนวนนักร้อง	
1	ดูข้อมูล
จำนวนกีตาร์	
1	ดูข้อมูล
จำนวนเบส	
1	ดูข้อมูล
จำนวนกลอง	
1	ดูข้อมูล
จำนวนศิริกอร์ด	
0	ดูข้อมูล
จำนวน extra	
0	ดูข้อมูล
จำนวน percussion	
1	ดูข้อมูล

← รายละเอียดเพลง →

main-menu
Event ทั้งหมด
Profile
main-menu
Event ทั้งหมด
Profile

แก้ไขเพลง

ลบเพลง

3.6.20 หน้ารายละเอียดแต่ละเพลงของ Backstage

หน้านี้คือหน้ารายละเอียดแต่ละเพลงของ Backstage หลังจากที่ Backstage กดรายชื่อเพลงโดยจะมีฟีเจอร์ ที่ให้ Backstage Assign Performer ได้ และแก้ไขเพลงได้ โดยการ Assign Performer คือการกำหนดว่า Performer คนไหนได้เล่นตำแหน่งอะไร เช่น นายA ได้เป็นนักร้อง นายB เป็นมือกีตาร์จากการกดปุ่มดูข้อมูล อีกฟีเจอร์คือการแก้ไขเพลงสามารถแก้ไขรายละเอียดต่าง ๆ ได้ทั้งหมดตามที่ Backstage ต้องการ

หน้า Assign Performer ของ Backstage



3.6.21 หน้า Assign Performer ของ Backstage

หน้านี้คือหน้าที่ Assign Performer ของ Backstage หลังจากที่ Backstage กดดูข้อมูล จากหน้ารายละเอียดแต่ละเพลงของ โดย Backstage สามารถเลือก Performer ว่าจะให้ใครเล่น ตำแหน่งอะไรโดยไม่สามารถเลือกซ้ำได้ถ้า Backstage เลือกเสร็จแล้วทำการกดยืนยันเป็นอันเสร็จสิ้น

หน้าแก้ไขเพลง ของ Backstage

← รายละเอียดเพลง

ชื่อเพลง คุกเข่า	key D	รายละเอียดเพิ่มเติม (optional) -	รายละเอียดเพิ่มเติม (optional) -
ref เพลง https://youtu.be/jqtWf4Z2i-0?si=j8GkeLP	จำนวนนักร้อง 2	จำนวนกีตาร์ 2	จำนวนเบส 1
จำนวนกลอง 1	จำนวนต่ำบอร์ด 1	จำนวนกลอง 1	จำนวนต่ำบอร์ด 1
จำนวน extra 0	จำนวน extra 0	จำนวน percusion 1	จำนวน percusion 1
จำนวน percusion 1	จำนวน percusion 1	แก้ไขเพลง	
		ลบเพลง	

← รายละเอียดเพลง

จำนวนนักร้อง 2 ดูข้อมูล

จำนวนกีตาร์ 2 ดูข้อมูล

จำนวนเบส 1 ดูข้อมูล

จำนวนกลอง 1 ดูข้อมูล

จำนวนต่ำบอร์ด 1 ดูข้อมูล

จำนวน extra 0 ดูข้อมูล

จำนวน percusion 1 ดูข้อมูล

จำนวน percusion 1 ดูข้อมูล

แก้ไขเพลง

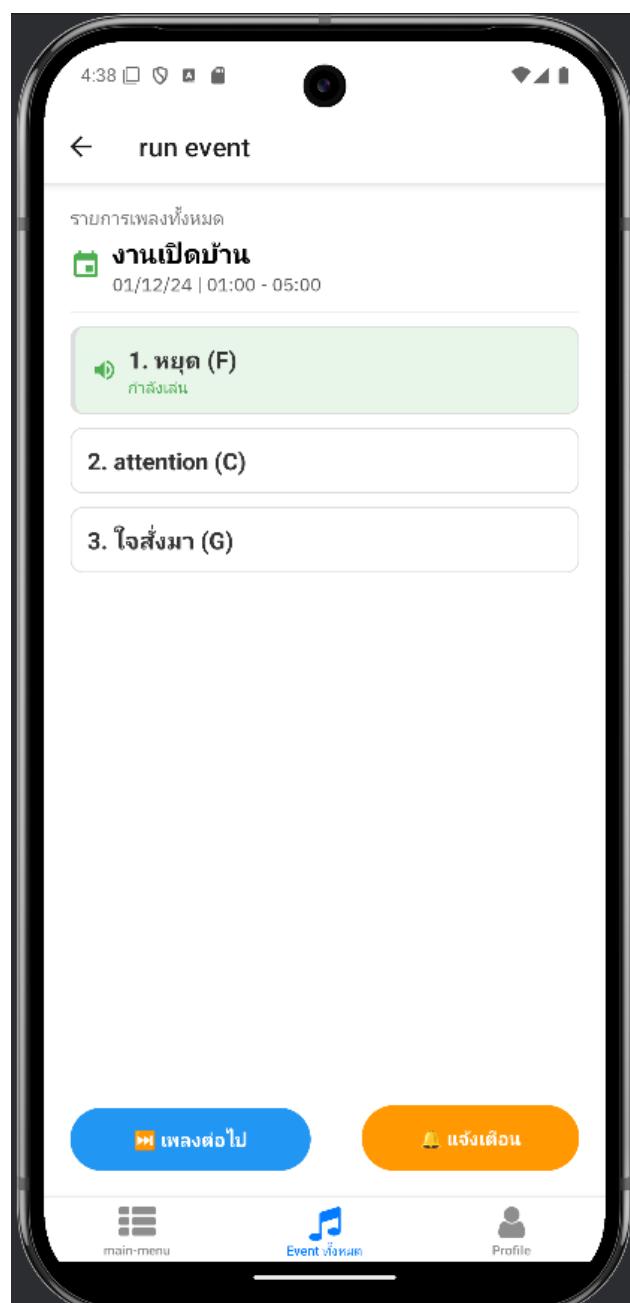
ลบเพลง

main-menu Event ห้างสรรพสินค้า Profile main-menu Event ห้างสรรพสินค้า Profile

3.6.22 หน้าแก้ไขเพลง ของ Backstage

หน้านี้คือหน้าแก้ไขเพลง ของ Backstage หลังจาก Backstage กดดูรายละเอียดเพลง โดย Backstage สามารถแก้ไขข้อมูลได้ทุกส่วน ถ้า Backstage แก้ไขข้อมูลเสร็จแล้วสามารถกดปุ่ม “แก้ไขเพลง” ระบบจะแสดงข้อความแจ้งเตือน “แก้ไขสำเร็จ” และถ้า Backstage ต้องการยกเลิกการแก้ไขให้กดปุ่มย้อนกลับในรูปที่ 1 ระบบแสดงข้อความว่า “ต้องการยกเลิกการแก้ไขหรือไม่” พร้อมปุ่ม ตกลงและยกเลิก

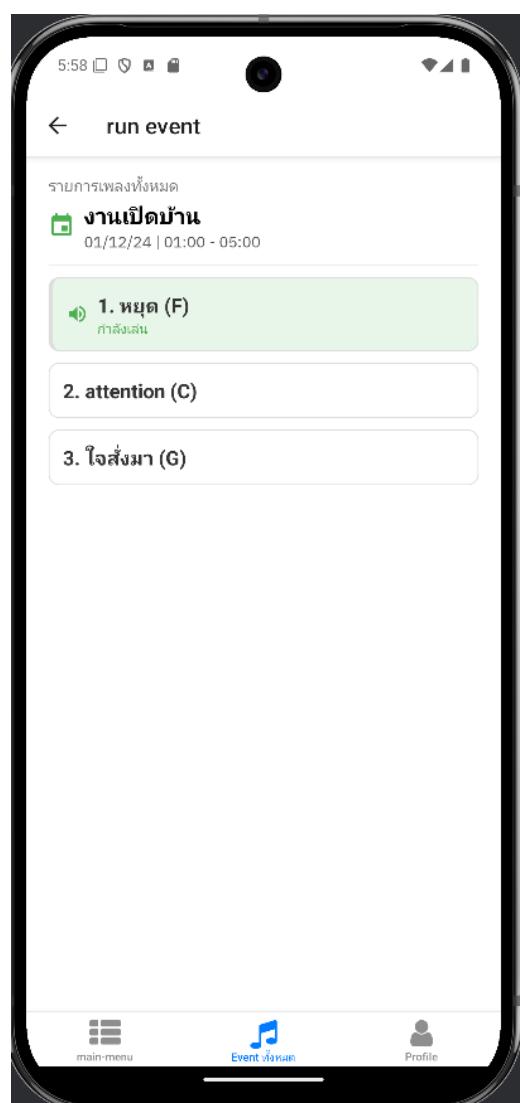
หน้า Run event ของ Backstage



3.6.23 หน้า Run event ของ Backstage

หน้านี้คือหน้า Run event ของ Backstage หลังจาก Backstage กดปุ่ม “เริ่ม Event” จากหน้ารายละเอียด Event โดยจะมีไฟเขียวแจ้งเตือน queue ให้ Performer ในเพลงถัดไปและเปลี่ยนเพลง

หน้า Run event ของ Performer & Staff & System admin



3.6.24 หน้า Run event ของ Performer & Staff & System admin

หน้านี้คือหน้า Run event ของ Performer, Staff, System admin โดยมีลักษณะ
เหมือนกับหน้าดูรายชื่อเพลง แต่จะมีการบ่งบอกถึงเพลงปัจจุบันที่กำลังเล่นอยู่ ณ ตอนนั้น

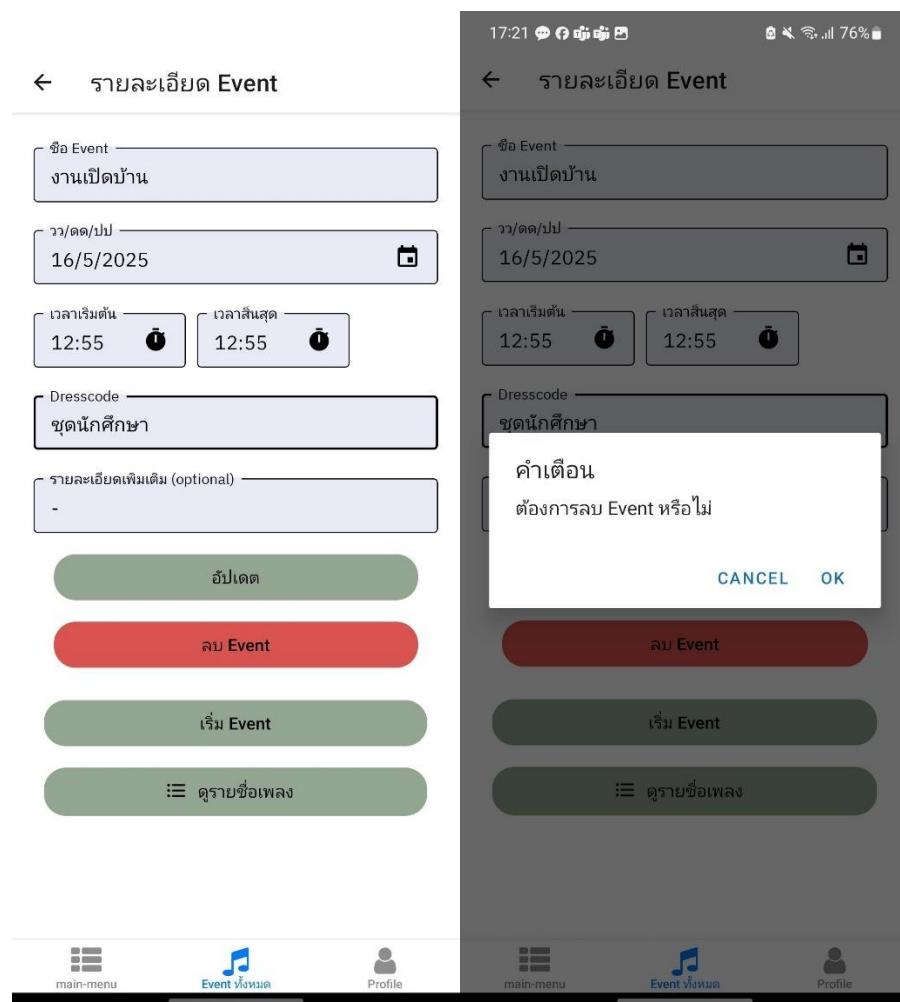
หน้าแจ้งเตือน Performer



3.6.25 หน้าแจ้งเตือน Performer

หน้านี้คือหน้าแจ้งเตือน Performer ของ Backstage หลังจาก Backstage กดปุ่มแจ้งเตือน จากหน้า Run event โดยจะมีฟีเจอร์การใส่ข้อความเพื่อแจ้งให้ Performer โดยตรงโดย Backstage สามารถเลือกใส่หรือไม่ใส่ข้อความได้ จากนั้นถ้า Backstage กดปุ่มแจ้งเตือน Performer ทางระบบ จะแสดงข้อความ "แจ้งเตือนสำเร็จ" ระบบจะส่งข้อความไปยัง Discord ของ Performer ว่า "เพลง ต่อไปเป็นเพลงของคุณแล้วกรุณามาเตรียมตัว"

หน้ายกเลิกการแสดงของ Backstage



3.6.26 หน้ากเลิกการแสดง ของ Backstage

หน้านี้คือหน้าที่ให้ Backstage ลบ Event ที่ยังไม่ได้จัดแสดงได้โดยไม่สามารถลบงาน Event ที่กำลังจัดแสดงอยู่ วิธีการคือกดซ่อน Event ที่ต้องการและกดปุ่มลบ Event จากนั้นระบบจะแสดงข้อความแจ้งเตือน: "ต้องการลบ Event หรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" ถ้ากดยืนยัน ระบบจะลบEventออกจากรายการ Event ทั้งหมด

หน้าลบเพลงใน queue



3.6.27 หน้าลับเพลงใน queue

หน้านี้คือหน้าที่ให้ Backstage ลับเพลงที่ถูกยกเลิกใน Event นั้น ๆ และ วิธีการคือกดชื่อเพลง ที่ต้องการลบ จากนั้นให้ Backstage กดปุ่มลับเพลง จากนั้นระบบจะแสดงข้อความแจ้งเตือน: "ต้องการลบเพลงหรือไม่" พร้อมปุ่ม "ยืนยัน" และ "ยกเลิก" ถ้ากดยืนยัน ระบบจะแสดงข้อความแจ้งเตือน: "ลับเพลงสำเร็จ" ถ้ายกเลิก ระบบจะไม่ลับเพลงออกจากรายการเพลงทั้งหมด

บทที่ 4

ทรัพยากรและแผนการดำเนินงาน

4.1 การจัดเตรียมฮาร์ดแวร์และซอฟต์แวร์

4.1.1 ฮาร์ดแวร์ที่ใช้ในโครงการ

4.1.1.1 คอมพิวเตอร์ส่วนบุคคล

สำหรับการพัฒนาระบบ Frontend และ Backend และรองรับการรัน Emulator รวมถึงเครื่องมือต่าง ๆ ที่จำเป็นในการพัฒนาระบบจะต้องมีความต้องการของระบบขั้นต่ำดังนี้

Processor: Intel Core i3 (8th Gen)

RAM: 8 GB

Storage: 256 GB

Graphics: Integrated graphics

4.1.1.2 เครื่องมือเสริม

อุปกรณ์เสริม เช่น เม้าส์, คีย์บอร์ด และหน้าจอแสดงผลเพิ่มเติมสำหรับการออกแบบ UI

4.1.2 ซอฟต์แวร์ที่ใช้ในโครงการ

4.1.2.1 ระบบปฏิบัติการ (Operating System)

Windows 10 ขึ้นไปซึ่งเป็นระบบปฏิบัติการที่มีความเสถียรและรองรับเครื่องมือพัฒนาระบบต่าง ๆ ที่จำเป็น

4.1.3 ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ

- Visual Studio Code เป็น IDE ที่มีส่วนเสริมหลากหลายหมายสำหรับการเขียนโค้ดหลาย ช่วยเพิ่มและประสิทธิภาพในการพัฒนาระบบ

- Postman ใช้สำหรับทดสอบ REST API เพื่อทดสอบการทำงานของระบบหลังบ้านโดยไม่เข้ากับระบบหน้าบ้าน ช่วยให้พัฒนาระบบทลั้งบ้านและหน้าบ้านพร้อมกันได้

4.2 แผนการดำเนินงาน

4.2.1 ตารางการดำเนินงานที่ผ่านมาจนถึงปัจจุบัน

4.2.2 ตารางแผนการดำเนินงานในอนาคต

ขั้นตอน	มค.68				กพ.68				มีค.68				เมย.68				พค.68			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.ปรับปรุงแก้ไขเอกสาร																				
2.วางแผนพัฒนาระบบ																				
3.พัฒนาและทดสอบ ฟีเจอร์หลัก																				
4.ทดสอบระบบทั้งหมด และปรับปรุง																				
5.เปิดตัวแอปพลิเคชัน																				
6.สรุปผล																				
7.นำเสนอโครงการ																				

4.3 วิธีการประเมินผล

เพื่อให้การพัฒนาระบบเป็นไปอย่างมีประสิทธิภาพและตอบสนองต่อความต้องการของผู้ใช้งานอย่างแท้จริง จึงได้มีการออกแบบกระบวนการประเมินผลที่เน้นการมีส่วนร่วมของสมาชิกชุมชน โดยใช้ทั้งการเก็บข้อเสนอแนะเชิงคุณภาพและข้อมูลเชิงปริมาณควบคู่กันไป โดยหนึ่งในวิธีการสำคัญคือการเปิดช่องทางให้สมาชิกสามารถแสดงความคิดเห็นได้อย่างสม่ำเสมอ ไม่ว่าจะผ่านการสนทนากลุ่ม หรือช่องทางอื่นๆ ที่มีการจัดทำขึ้น เช่น กลุ่มchat บน Facebook หรือ群组 บน LINE ซึ่งจะช่วยให้ผู้ใช้งานสามารถแสดงความคิดเห็นได้อย่างสะดวกและรวดเร็ว ทั้งนี้ ควรกำหนดระยะเวลาและตัวชี้วัดที่ชัดเจน 以便สามารถติดตามและประเมินผลได้ทันท่วงทัน ทั้งนี้ ควรคำนึงถึงความต้องการของผู้ใช้งานเป็นสำคัญ ไม่ใช่แค่การตั้งเป้าหมายที่สูงเกินไป แต่ต้องมีความเหมาะสมและสามารถบรรลุได้จริง

4.3.1 การประเมินผลผ่าน Google Forms

นอกจากการประเมินผลด้วยการเก็บข้อเสนอแนะจากผู้ใช้งานจริงแล้ว ยังมีการสร้างแบบประเมินผ่าน Google Forms เพื่อให้สมาชิกชุมชนสามารถประเมินความพึงพอใจต่อระบบด้วยรูปแบบเชิงปริมาณในด้านต่าง ๆ เช่น ความง่ายในการใช้งาน ความรวดเร็วในการเข้าถึงข้อมูล ความครอบคลุมของฟีเจอร์ ข้อมูลจากแบบประเมินเหล่านี้ช่วยให้ทราบถึงจุดแข็งและจุดที่ควรปรับปรุงของ

ระบบอย่างซัดเจนมากขึ้นและสามารถถูกลายเป็นแนวทางสำคัญในการพัฒนาระบบที่ดียิ่งขึ้น โดยมี เกณฑ์การประเมินผลดังนี้

1. เกณฑ์การประเมินสำหรับ Backstage

- ความสะอาดสวยงามในการสร้าง, แก้ไข, ลบงานแสดงดนตรี (5 คะแนน)
- ความสะอาดสวยงามในการสร้าง, แก้ไข, ลบคิวและรายละเอียดเพลงในแต่ละงาน ดนตรี (5 คะแนน)
- ความสะดวกในการใช้งาน (UX/UI) (5 คะแนน)
- ความรวดเร็วของระบบ (5 คะแนน)
- ความสะอาดสวยงามในการเข้าถึงข้อมูลที่ต้องการ (เช่น ข้อมูลงานแสดงดนตรี ข้อมูลเพลง และ lineup player) (5 คะแนน)

2. เกณฑ์การประเมินสำหรับ Performer, Staff ฝ่ายอื่น

- ความสะดวกในการใช้งาน (UX/UI) (5 คะแนน)
- ความรวดเร็วของระบบ (5 คะแนน)
- ความสะอาดสวยงามในการเข้าถึงข้อมูลที่ต้องการ เช่น ข้อมูลงานแสดงดนตรี ข้อมูลเพลงและ lineup performer (5 คะแนน)

3. เกณฑ์การประเมินสำหรับ System admin

- ความสะดวกในการใช้งาน (UX/UI) (5 คะแนน)
- ความรวดเร็วของระบบ (5 คะแนน)
- ความสะอาดสวยงามในการเข้าถึงข้อมูลที่ต้องการ (เช่น ข้อมูลงานแสดงดนตรี ข้อมูลเพลง และ lineup player) (5 คะแนน)
- ความสะดวกสบายในการจัดการสมาชิกชุมชนภาษาในระบบ (5 คะแนน)
- ความสะดวกสบายในการจัดการสิทธิ์ของสมาชิกภาษาใน Discord server (5 คะแนน)

4.4 เกณฑ์การผ่านการทดสอบ

ระบบต้องได้คะแนนรวมจากแบบประเมินในระดับไม่น้อยกว่า 70% ของคะแนนเต็มในแต่ละกลุ่ม ซึ่งจะช่วยรับประกันว่าระบบมีคุณภาพเพียงพอและตอบสนองต่อความต้องการของผู้ใช้งานส่วนใหญ่ได้อย่างมีประสิทธิภาพ

บทที่ 5

สรุป

5.1 ผลการประเมินผล

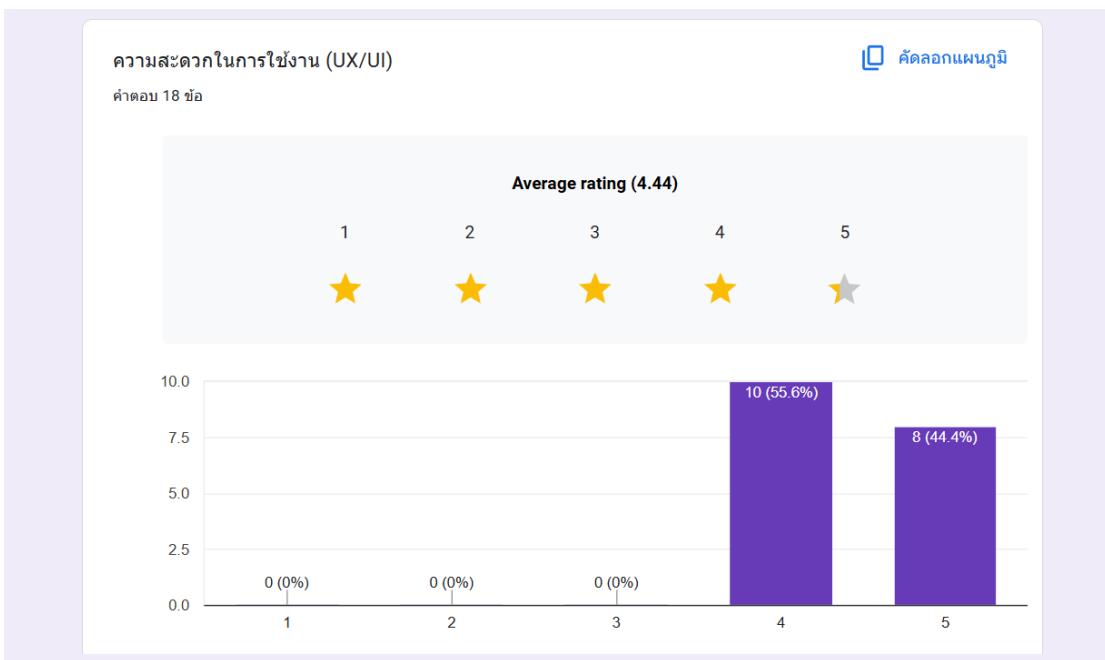
ในการพัฒนาระบบจัดการการแสดงดนตรีสำหรับชุมชน Sci Band นั้นได้มีการดำเนินการปรึกษาและเก็บข้อเสนอแนะจากชุมชน จากหลาย ๆ ฝ่าย ได้แก่ สมาชิกในวงดนตรี ทั้งนักร้อง นักดนตรี และ staff ฝ่ายต่าง ๆ โดยเฉพาะฝ่าย backstage

ผลจากการเก็บข้อเสนอแนะและปรับปรุงระบบอย่างต่อเนื่องนี้ส่งผลให้ระบบสามารถตอบสนองความต้องการของผู้ใช้งานได้ดีขึ้น ในด้านต่าง ๆ เช่น

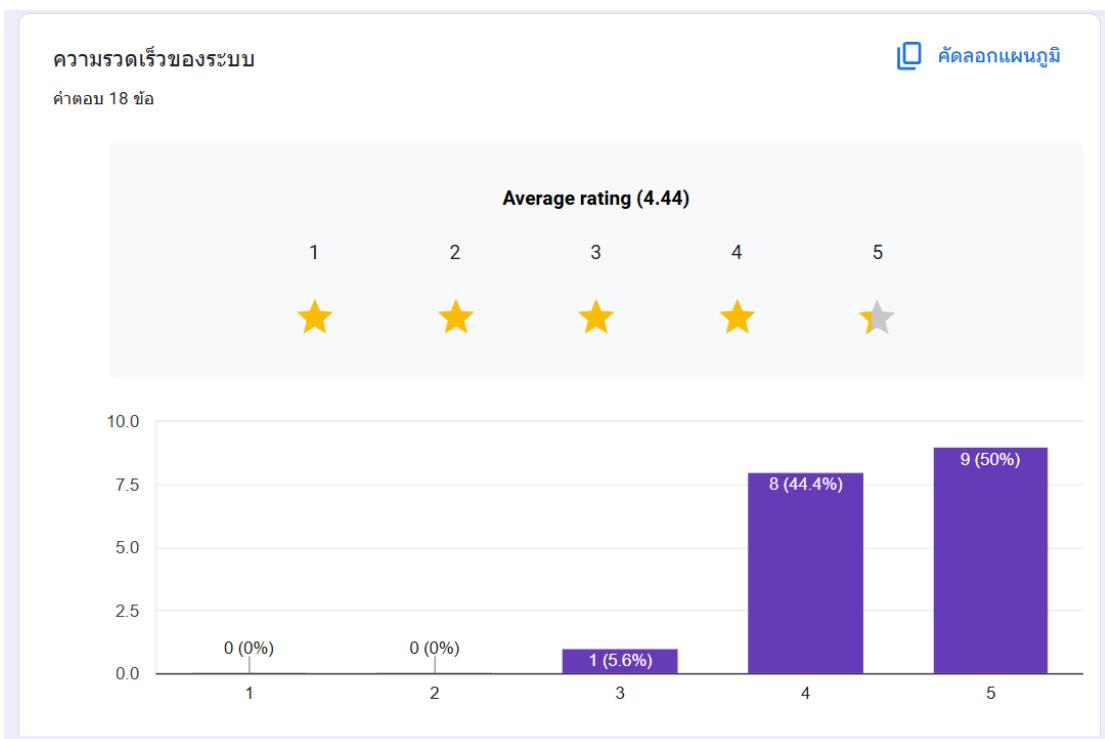
1. การปรับปรุง UI ให้เหมาะสมกับผู้ใช้
2. การปรับปรุง feature เพิ่มบัญชีผู้ใช้ใหม่เข้าสู่ระบบ โดยใช้การเชื่อมต่อ Google Form ที่ใช้สำหรับการเก็บข้อมูลสมาชิกใหม่เข้ากับ Google Sheets เพื่อให้ระบบสามารถอ่านข้อมูล Google Form ผ่านทาง Google Sheets และทำการเพิ่มบัญชีผู้ใช้ใหม่เข้าสู่ระบบโดยอัตโนมัติจากเดิมที่ System admin ต้องเป็นคนนำเข้าไฟล์ Google Form เข้าสู่ระบบด้วยตนเองซึ่งอาจก่อให้เกิดความไม่สะดวกในการใช้งาน เนื่องจากระบบนั้นเป็นโมบายแอปพลิเคชัน
3. การเพิ่มเติม feature ที่ช่วยอำนวยความสะดวก เช่นการเลือกแสดงเฉพาะงานแสดง ดนตรีและเพลงที่ผู้ใช้มีความสนใจมากขึ้น

นอกจากนี้ ยังได้มีการจัดทำแบบประเมินระบบผ่าน Google Form เพื่อวัดความพึงพอใจและประสิทธิภาพในการใช้งานของระบบจากผู้ใช้จริง แบ่งเป็น 3 กลุ่ม ได้แก่ Backstage, Performer และ Staff อื่น ๆ และ System Admin โดยการประเมินครอบคลุมทั่วทั้ง 3 ตามเกณฑ์ เช่น ความสะดวกในการใช้งาน ความเข้าใจง่าย ความเหมาะสมของฟีเจอร์ และประสิทธิภาพโดยรวมของระบบ โดยใช้คะแนนเต็ม 5 คะแนนต่อหัวข้อ

จากการสรุปผลการประเมิน พบร่วมกับ:



5.1.1 ผลการประเมินความสatisfactionในการใช้งาน



5.1.2 ผลการประเมินความรวดเร็วของระบบ

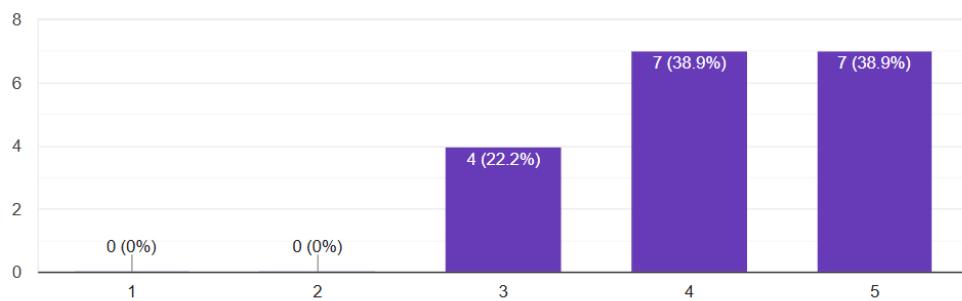
ความสะดวกสบายในการเข้าถึงข้อมูลที่ต้องการ (เช่นข้อมูลงานแสดงดนตรี ข้อมูลเพลง และ lineup player)

ค่าตอบ 18 ข้อ



คัดลอกແຜນກົມ

Average rating (4.17)



5.1.3 ผลการประเมินความสะดวกสบายในการเข้าถึงข้อมูลที่ต้องการ

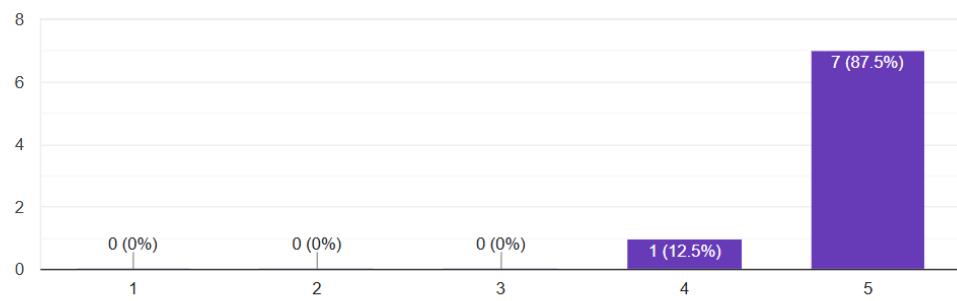
ความสะดวกสบายในการสร้าง, แก้ไข, ลงงานแสดงดนตรี

ค่าตอบ 8 ข้อ

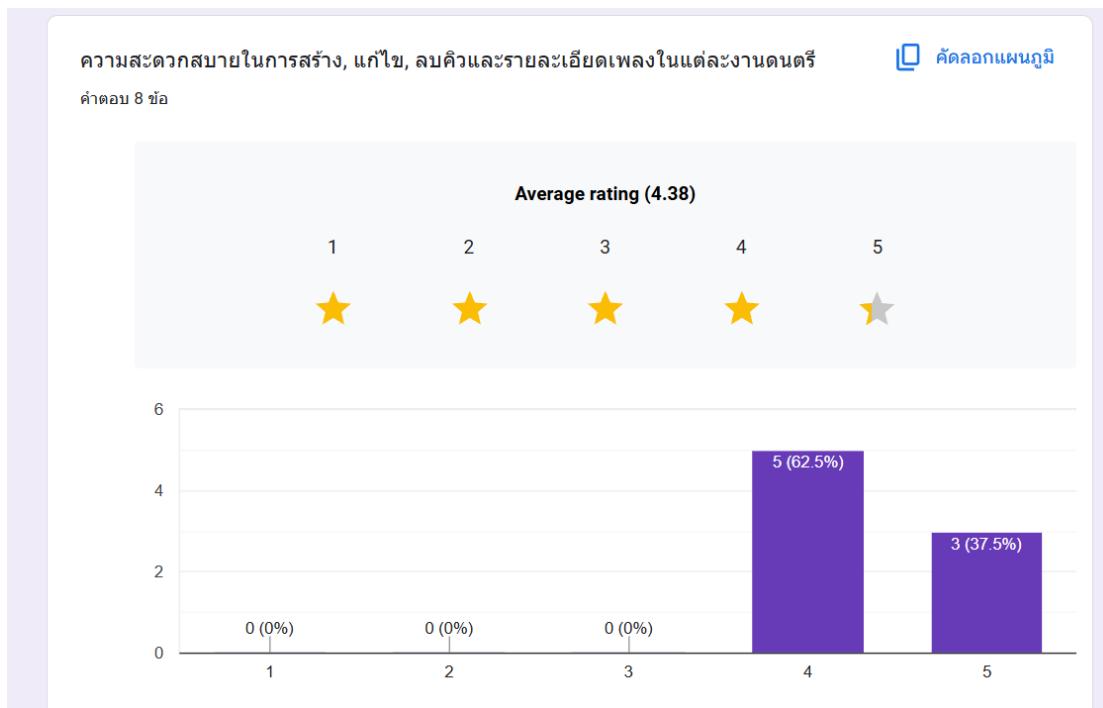


คัดลอกແຜນກົມ

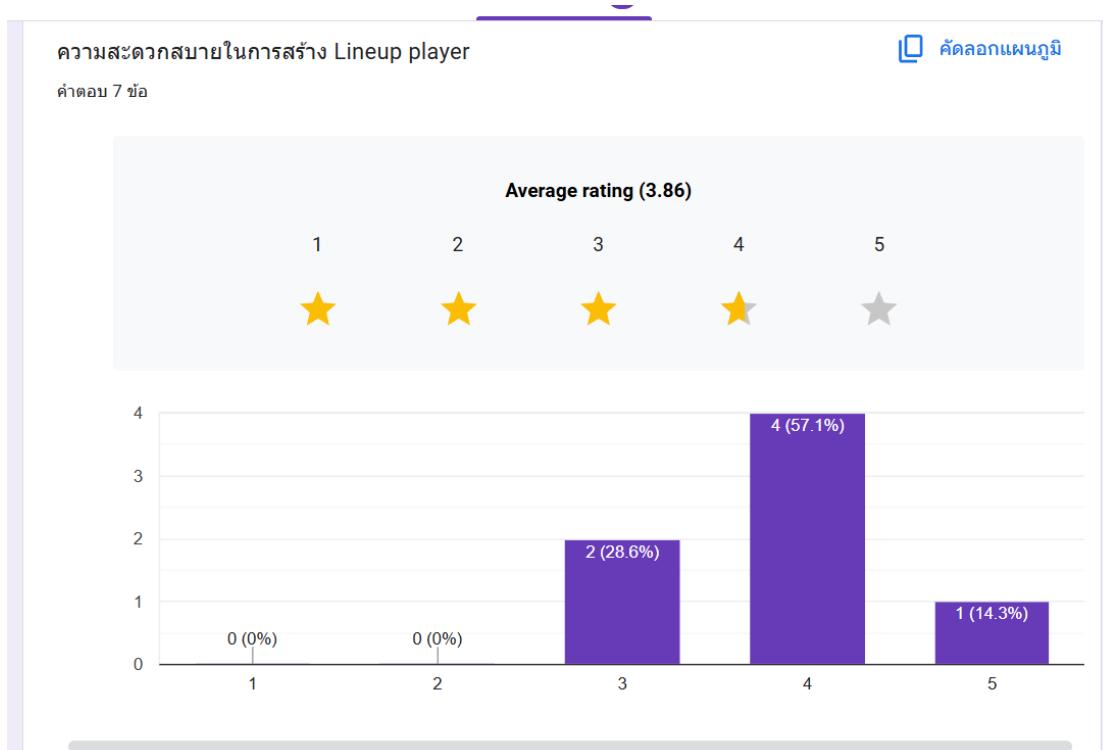
Average rating (4.88)



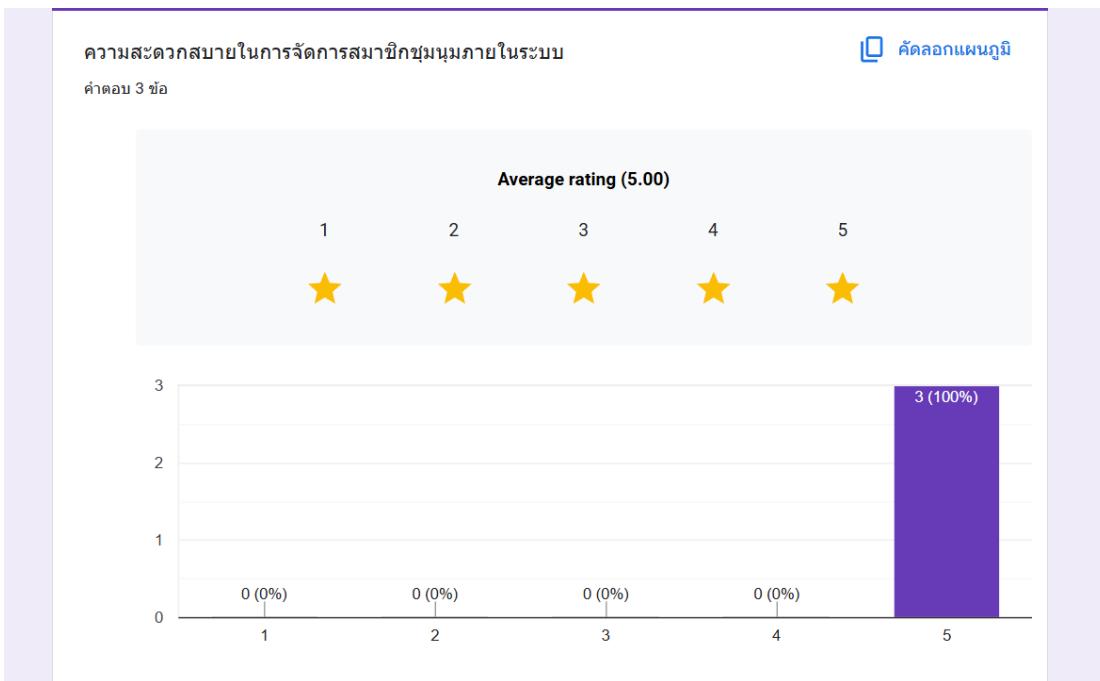
5.1.4 ผลการประเมินความสอดคล้องสบายนในการสร้าง, แก้ไข, ลงงานแสดงดนตรี



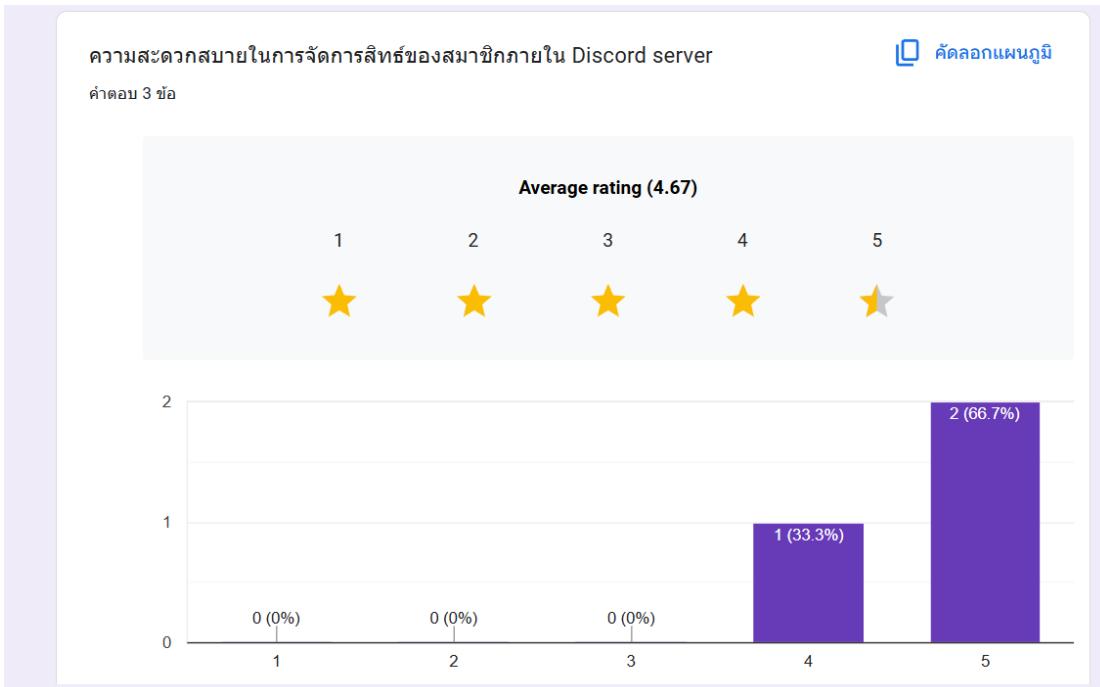
5.1.5 ผลการประเมินการสร้างและแก้ไข รายละเอียดเพลง



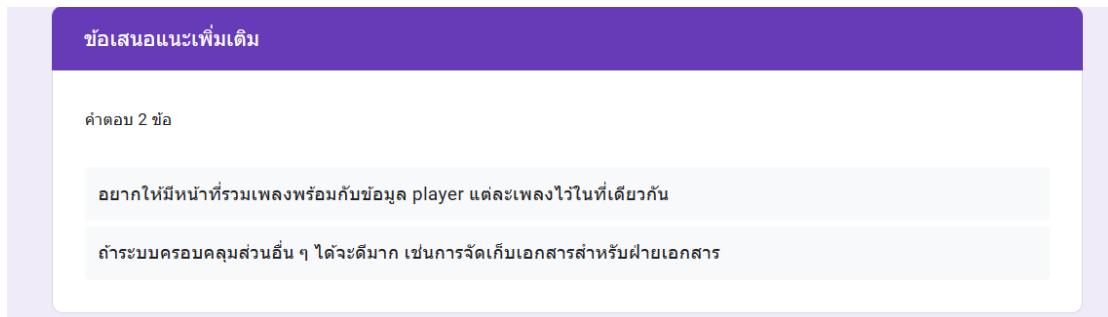
5.1.6 ผลการประเมินความสอดคล้องสบายนในการสร้าง Lineup Performer



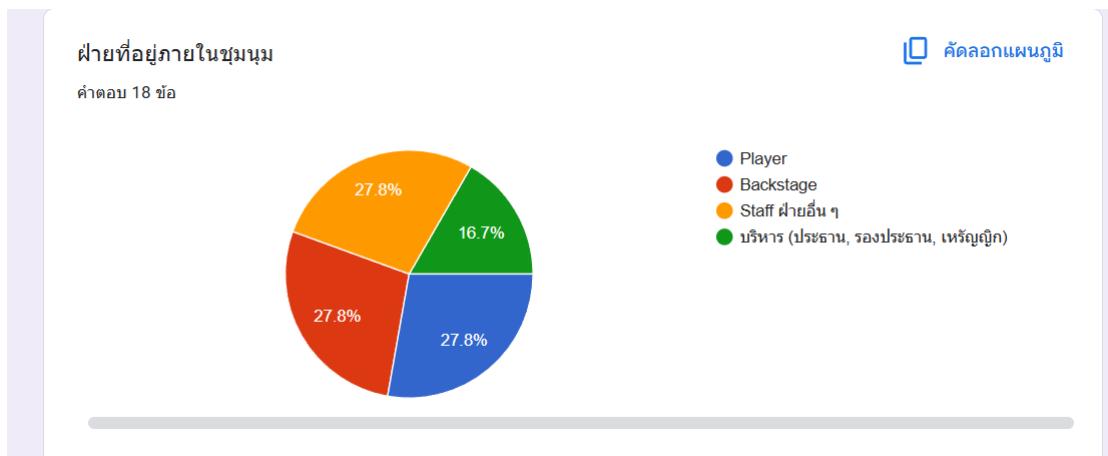
5.1.7 ผลการประเมินความสอดคล้องสบายนในการจัดการสมาชิกชุมชนในระบบ



5.1.8 ผลการประเมินความสอดคล้องสภายการจัดการสิทธิ์ของสมาชิกภายใน Discord server



5.1.9 ผลการประเมินข้อเสนอแนะเพิ่มเติม



5.1.10 จำนวนผู้กรอกแบบฟอร์มและฝ่ายในชุมชน

จากการดำเนินงานและการประเมินความพึงพอใจของผู้ใช้งานพบว่า ระบบสามารถตอบสนองต่อความต้องการของสมาชิกชุมชนได้เป็นอย่างดี โดยมีคะแนนประเมินเฉลี่ยเกินเกณฑ์ร้อยละ 70 ซึ่งถือเป็นระดับที่ “ผ่านเกณฑ์” ตามที่กำหนดไว้

ผลลัพธ์ดังกล่าวสะท้อนให้เห็นว่า การรับฟังข้อเสนอแนะจากผู้ใช้งานเป็นปัจจัยสำคัญที่ส่งผลต่อการพัฒนาและปรับปรุงระบบให้สามารถใช้งานได้จริงในบริบทของชุมชนดูดนตรี อีกทั้งยังเป็นกรณีตัวอย่างที่ดีสำหรับชุมชนหรือองค์กรที่มีลักษณะการทำงานคล้ายคลึงกันในการนำเทคโนโลยีเข้ามาเสริมประสิทธิภาพการจัดการ

จากการประเมินนี้แสดงให้เห็นว่าการรับฟังข้อเสนอแนะจากผู้ใช้งานเป็นปัจจัยสำคัญและนำมาพัฒนาปรับปรุงระบบอย่างสม่ำเสมอช่วยให้การพัฒนาระบบสามารถตอบโจทย์ความต้องการของผู้

ใช้ได้อย่างแท้จริงและส่งผลให้ระบบสามารถนำไปใช้งานได้จริงภายในชั่วโมงและสามารถเป็นตัวอย่างที่ดีได้สำหรับชุมชนดนตรีต่าง ๆ ที่มีลักษณะการทำงานที่คล้ายคลึงกัน

5.2 แนวคิดการต่อยอดและปรับปรุงระบบ

จากการเก็บข้อมูลแบบอย่างสม่ำเสมอจากสมาชิกชุมชนทำให้สามารถมองเห็นแนวทางในการพัฒนาระบบที่ตอบโจทย์การใช้งานจริงได้ดียิ่งขึ้น ทั้งในด้านฟีเจอร์ ความสะดวกของผู้ใช้งาน และความสามารถของระบบในการรองรับสถานการณ์ที่ซับซ้อนมากขึ้น โดยสามารถสรุปแนวคิดการต่อยอดและปรับปรุงได้ดังนี้

การพัฒนาระบบที่รองรับการแสดงผลงาน iPad และอุปกรณ์หลากหลายขนาด
เพื่อให้ผู้ใช้งานสามารถใช้งานระบบได้สะดวกยิ่งขึ้นในสถานการณ์จริง เช่น การใช้งานบนเวทีหรือระหว่างการซ้อม ระบบควรรองรับขนาดหน้าจอที่หลากหลายจะช่วยให้การดูข้อมูลต่าง ๆ ทำได้ง่ายและชัดเจนขึ้น

การจัดการคิวเพลงที่มีความยืดหยุ่นมากยิ่งขึ้น
จากการทดสอบการทำงานจริงพบว่าในช่วงเวลาที่มีงานแสดงหลายงานใกล้เคียงกัน ทาง Performer อาจมีการเลือกกลุ่มเพลงเดิมซ้ำในงานถัดไป และใช้ lineup ชุดเดิม การเพิ่มฟีเจอร์ที่สามารถนำเข้าข้อมูลเพลงพร้อม lineup performer จากงานก่อนหน้าไปยังงานถัดไปได้ จะช่วยลดภาระในการกรอกข้อมูลซ้ำ และทำให้การจัดการคิวเพลงมีประสิทธิภาพมากขึ้น

การขยายขอบเขตของระบบให้ครอบคลุมองค์ประกอบอื่น ๆ ของงานแสดงดนตรี
เนื่องจากงานแสดงดนตรีไม่ได้มีเพียงการเล่นดนตรีเท่านั้น แต่ยังมีองค์ประกอบอื่น ๆ ที่เกี่ยวข้อง เช่น การวางแผนตารางซ้อม การติดตั้งระบบเสียง และการจัดการอุปกรณ์ ระบบสามารถเพิ่มฟีเจอร์ที่รองรับการจัดตารางซ้อม การบันทึกรายการอุปกรณ์เสียงที่ต้องใช้ รวมถึงการจัดเก็บเอกสารประกอบการแสดง เช่น

Stage Plot: แผนผังตำแหน่งของนักดนตรีบนเวทีเพื่อให้ performer สามารถเตรียมตัวลงหน้าได้
Rider List: รายการอุปกรณ์เครื่องดนตรีที่ใช้ในงาน พร้อมรายละเอียดการเชื่อมต่อกับระบบเสียง

การต่อยอดและเพิ่มเติมไฟเซอร์เหล่านี้จะช่วยให้ระบบกล้ายเป็นแพลตฟอร์มจัดการงานแสดงดนตรีแบบครบวงจร ซึ่งจะเป็นประโยชน์ต่อชุมชนมุสลิมในอนาคต ไม่เพียงแค่ตอบโจทย์การใช้งานในปัจจุบันแต่ยังเป็นการวางแผนรากฐานที่มั่นคงสำหรับการเติบโตในระยะยาว การรวมองค์ประกอบต่าง ๆ เข้ามาไว้ในระบบเดียวกันจะช่วยลดความสับสนจากการใช้เครื่องมือหลายตัว กระชับขั้นตอนการทำงานเตรียมงาน และช่วยให้สมาชิกทุกฝ่ายสามารถเข้าถึงข้อมูลที่สำคัญได้อย่างรวดเร็วและตรงจุด

ด้วยแนวทางการพัฒนาที่อิงจากข้อเสนอแนะของผู้ใช้งานจริงและประสบการณ์การทำงานจริง ระบบนี้จึงมีศักยภาพที่จะเติบโตไปพร้อมกับชุมชนมุสลิมและกล้ายเป็นศูนย์กลางของการทำงานร่วมกันอย่างเป็นระบบในหลาย ๆ ขั้นตอนของการจัดงานแสดงดนตรีภายนอกชุมชน มีศักยภาพที่จะสามารถต่อยอดให้มีศักยภาพที่มากยิ่งขึ้น หากสามารถรุ่นต่อ ๆ ไปภายใต้ชุมชนนำระบบนี้ไปพัฒนาต่ออย่างต่อเนื่องตามความต้องการที่เปลี่ยนแปลงไปของชุมชนในอนาคต

รายการอ้างอิง

Rivaldo. (2567). Shannon and Weaver Model of Communication. สีบคัน 10 พฤศจิกายน 2567, จาก <https://www.communicationtheory.org/shannon-and-weaver-model-of-communication/>

Justin Ellingwood. What is an ORM (Object RelationalMapper)? . สีบคัน 11 พฤศจิกายน 2567, จาก <https://www.prisma.io/dataguide/types/relational/what-is-an-orm>

Discord. What is Discord: สีบคัน 11 พฤศจิกายน 2567,<https://discord.com/creators/what-is-discord>

