

ระบบจัดการภาพยนตร์
Cinema Management System

นายณิณภัฏ สัจจพรรณ์	รหัสนักศึกษา 6706022610268 Sec 1
นายธราธิป ศิลปานุรักษ์	รหัสนักศึกษา 6706022610411 Sec 1
นางสาวธนพร ไทยาชีวะ	รหัสนักศึกษา 6706022610381 Sec 1

โครงการนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้า

พระนครเหนือ

ภาคเรียนที่ 1 ปีการศึกษา 2568

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

คำนำ

การจัดทำโครงงาน “ระบบจัดการภาพยนตร์ (Cinema Management System)” เป็นส่วนหนึ่งของการเรียนรายวิชา Computer Programming มีวัตถุประสงค์เพื่อพัฒนาโปรแกรมสำหรับงานด้านหลังบ้าน (Back-Office) ของโรงภาพยนตร์ โดยมุ่งเน้นให้ผู้ดูแลระบบสามารถจัดการข้อมูลภาพยนตร์ได้อย่างเป็นระบบ เช่น การเพิ่ม แก้ไข ลบ และตรวจสอบข้อมูล รวมถึงการสรุปรายงานในรูปแบบที่ชัดเจนและเข้าใจง่าย

ระบบนี้พัฒนาด้วยภาษา Python โดยเลือกใช้การจัดเก็บข้อมูลแบบไฟล์ (CSV/DAT) แทนฐานข้อมูลขนาดใหญ่ เพื่อความเรียบง่ายและเหมาะสมกับโรงภาพยนตร์ขนาดเล็กหรือระบบต้นแบบสำหรับการศึกษา นอกจากนี้ยังมีการออกแบบฟังก์ชันตรวจสอบความถูกต้องของข้อมูล และการสร้างรายงานสรุปเพื่อให้ผู้จัดการสามารถวิเคราะห์และตัดสินใจได้สะดวกยิ่งขึ้น

คณะผู้จัดทำหวังว่าโครงงานนี้จะเป็นประโยชน์ต่อผู้ที่สนใจการพัฒนาโปรแกรมด้านการจัดการข้อมูล ตลอดจนเป็นแนวทางเบื้องต้นสำหรับการนำไปปรับใช้หรือพัฒนาต่อยอดในระบบจริง หากมีข้อผิดพลาดประการใด คณะผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้ด้วย

สารบัญ

สารบัญภาพ	ก
สารบัญตาราง	ข
บทที่ 1	1
1.1 วัตถุประสงค์	1
1.2 ขอบเขตของโครงการ	1
1.3 ประโยชน์ที่ได้รับ	2
1.4 เครื่องมือที่คาดว่าจะต้องใช้	2
บทที่ 2	3
2.1 แฟ้มเก็บข้อมูลภาพยนตร์ movies.dat	3
2.2 แฟ้มเก็บประวัติการแก้ไข moviesLog.dat	6
2.3 แฟ้มเก็บข้อมูลรอบฉาย screening.dat	8
2.4 ไฟล์ report.txt	10
บทที่ 3	13
3.1 การใช้งานโปรแกรมระบบจัดการภาพยนตร์	13
3.2 การใช้งานโปรแกรมการเพิ่มข้อมูล	14
3.3 การใช้งานโปรแกรมแก้ไขข้อมูล	17
3.4 การใช้งานโปรแกรมการลบข้อมูล	19
3.5 การใช้งานโปรแกรมดูข้อมูล	23
3.6 การสร้างรายงานผล (.txt)	27

บทที่ 4	30
4.1 main.py	30
4.2 project.py	35
บทที่ 5	56
5.1 สรุปผลการดำเนินงาน	56
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	56
5.3 ข้อเสนอแนะ	57
5.4 สิ่งที่ได้จัดทำได้รับจากการพัฒนาโครงการ	57

สารบัญภาพ

ภาพที่ 1	ไฟล์ report	10
ภาพที่ 2	ตัวอย่างการใช้งาน	14
ภาพที่ 3	ตารางข้อมูล	14
ภาพที่ 4	การแสดงผลภาพยนตร์	15
ภาพที่ 5	อัปเดตตาราง	15
ภาพที่ 6	การแสดงตารางภาพยนตร์ทั้งหมด	16
ภาพที่ 7	ตารางที่อัปเดตข้อมูล	16
ภาพที่ 8	การใช้งานโปรแกรม update	17
ภาพที่ 9	ตารางข้อมูล	17
ภาพที่ 10	ผลบรรทัดยืนยันว่าแก้ไขเรียบร้อย	18
ภาพที่ 11	ตารางแสดงรอบฉาย	19
ภาพที่ 12	ตารางแสดงอัปเดต	19
ภาพที่ 13	การใช้งานโปรแกรม delete	20
ภาพที่ 14	สถานะของภาพยนตร์	20
ภาพที่ 15	การแสดงผลกรลบ	21
ภาพที่ 16	เลือกภาพยนตร์ที่ต้องการลบ	21
ภาพที่ 17	แสดงข้อความยืนยัน	22
ภาพที่ 18	การแสดงผลหลังจากลบ	22
ภาพที่ 19	การใช้งาน view	23
ภาพที่ 20	การดูข้อมูลเฉพาะเรื่อง	23
ภาพที่ 21	การดูข้อมูลทั้งหมด	24
ภาพที่ 22	กรองข้อมูลที่ตรงตามเงื่อนไข	25
ภาพที่ 23	การดูสถิติข้อมูล	26
ภาพที่ 24	การสร้างรายงานแสดงผล	27
ภาพที่ 25	ตัวอย่างไฟล์	28
ภาพที่ 26	การออกจากไฟล์	29
ภาพที่ 27	การ import	30
ภาพที่ 28	ฟังก์ชันรับค่าตัวเลข	31

ภาพที่ 29	ฟังก์ชันแสดงเมนู	32
ภาพที่ 30	ฟังก์ชันแสดงเมนูย่อย4	32
ภาพที่ 31	ฟังก์ชันแสดงเมนูย่อย"แก้ไขข้อมูล"	33
ภาพที่ 32	ฟังก์ชันแสดงเมนู"การดูข้อมูล"	33
ภาพที่ 33	ส่วนควบคุมเริ่มต้นโปรแกรม	34
ภาพที่ 34	นำเข้าโมดูล	35
ภาพที่ 35	นำเข้าข้อมูล datetime	36
ภาพที่ 36	เก็บค่าที่ใช้ซ้ำบ่อย	36
ภาพที่ 37	ฟังก์ชันคืนค่า วันและเวลา	37
ภาพที่ 38	ฟังก์ชันทำหน้าที่คืนค่าเวลา	37
ภาพที่ 39	ฟังก์ชันหยุดการทำงานโปรแกรม	37
ภาพที่ 40	ฟังก์ชันอ่านไฟล์	38
ภาพที่ 41	ฟังก์ชันอ่านข้อมูลทั้งหมด	39
ภาพที่ 42	ฟังก์ชันชื่อภาพยนตร์	39
ภาพที่ 43	ฟังก์ชันเขียนข้อมูลภาพยนตร์	40
ภาพที่ 44	ฟังก์ชันอ่านข้อมูลไฟล์	40
ภาพที่ 45	ฟังก์ชันเขียนข้อมูลรอบฉาย	41
ภาพที่ 46	ฟังก์ชันค้นหาภาพยนตร์	41
ภาพที่ 47	ฟังก์ชันบันทึกเหตุการณ์	42
ภาพที่ 48	ฟังก์ชันรับค่าจำนวนเต็ม	42
ภาพที่ 49	ฟังก์ชันรับค่าทศนิยม	43
ภาพที่ 50	ฟังก์ชันรับค่าจากผู้ใช้	43
ภาพที่ 51	ฟังก์ชันคืนค่าความจุสูงสุด	44
ภาพที่ 52	ฟังก์ชันเพิ่มภาพยนตร์	44
ภาพที่ 53	ต่อจากภาพที่ 52	45
ภาพที่ 54	ฟังก์เพิ่มรอบฉาย	45
ภาพที่ 55	ฟังก์ชันคืนค่าภาพยนตร์	46

ภาพที่ 56 ฟังก์ชันแสดงข้อมูลภาพยนตร์	46
ภาพที่ 57 ฟังก์ชันแก้ไขข้อมูลภาพยนตร์	47
ภาพที่ 58 ฟังก์ชันแก้ไขรอบฉาย	48
ภาพที่ 59 ฟังก์ชันลบข้อมูลภาพยนตร์	49
ภาพที่ 60 ฟังก์ชันค้นหาหัวตาราง	50
ภาพที่ 61 ฟังก์ชันจัดรูปแบบจำนวนเงิน	50
ภาพที่ 62 ฟังก์ชันรวมข้อมูลภาพยนตร์	51
ภาพที่ 63 ฟังก์ชันจัดรูปแบบ	51
ภาพที่ 64 ฟังก์ชันดูรายการเฉพาะเรื่อง	52
ภาพที่ 65 ฟังก์ชันดูรายการทั้งหมด	52
ภาพที่ 66 ฟังก์ชันกรองการแสดงผล	53
ภาพที่ 67 ฟังก์ชันสรุปสถิติรวม	54
ภาพที่ 68 ฟังก์ชันสร้างไฟล์รายงานข้อความ	55

สารบัญตาราง

ตารางที่ 1 แฟ้มเก็บข้อมูลภาพยนต์	3
ตารางที่ 2 แฟ้มเก็บประวัติแก้ไข	6
ตารางที่ 3 แฟ้มเก็บข้อมูลรอบฉาย	8

บทที่ 1

บทนำ

1.1 วัตถุประสงค์

- 1.1.1 เพื่อสร้างโปรแกรมที่ช่วยเก็บและจัดการข้อมูลภาพยนตร์ได้อย่างเป็นระบบ
- 1.1.2 เพื่อให้ผู้ดูแลสามารถเพิ่ม แก้ไข และลบข้อมูลได้สะดวก
- 1.1.3 เพื่อให้ค้นหาและดูข้อมูลภาพยนตร์ได้รวดเร็วและง่ายขึ้น
- 1.1.4 เพื่อสรุปรายงานภาพรวม เช่น จำนวนหนังที่ยังฉายอยู่และที่ออกจากโปรแกรมแล้ว
- 1.1.5 เพื่อฝึกทักษะการเขียนโปรแกรมและการจัดการไฟล์ข้อมูลให้พร้อมต่อยอดใช้งานจริง

1.2 ขอบเขตของโครงการ

- 1.2.1 ระบบจัดการภาพยนตร์มีฟังก์ชันพื้นฐานทั้งหมด 17 ฟังก์ชัน โดยมี 7 ฟังก์ชันหลัก ดังนี้

1. เพิ่ม (Add)

1.1 เพิ่มภาพยนตร์

1.2 เพิ่มรอบฉาย

2. แก้ไข (Update)

1.3 แก้ไขภาพยนตร์

1.4 แก้ไขรอบฉาย

3. ลบ (Deleted)

3.1 ลบภาพยนตร์ (เปลี่ยนสถานะเป็น Deleted)

3.2 ลบภาพยนตร์ออกจากข้อมูลจริง (Hard Deleted)

4. ดู (View)

4.1 เรียกดูรายการเดียว View One

4.2 เรียกดูรายการทั้งหมด View All

4.3 เรียกดูแบบกรอง Filter View

4.4 เรียกดูเป็นสถิติ Summary Statistics

5. สร้างรีพอร์ต (Generate Report)

6. กลับไปที่เมนู

7. เมฆออกจากหน้าปัจจุบัน

1.2.2 ระบบจัดการภาพยนตร์ประกอบด้วย 4 ไฟล์ ได้แก่

1. แฟ้มเก็บข้อมูลภาพยนตร์หลัก movies.dat

2. แฟ้มเก็บประวัติการแก้ไข moviesLog.dat

3. แฟ้มเก็บข้อมูลรอบฉาย screening.dat

4. ไฟล์ report.txt

1.2.3 ระบบจัดการภาพยนตร์มีการจัดเก็บข้อมูลหนังสือไว้ในไฟล์ text file ชื่อ report.txt ซึ่งมีจำนวนรวมของรอบฉาย จำนวนรวมของที่นั่งทั้งหมด จำนวนรวมของที่นั่งที่ถูกจอง จำนวนรายได้รวมทั้งหมดของวัน อัตราที่นั่งที่ถูกจอง สถิติราคาตั๋ว ประเภทของหนังที่ยังฉายอยู่ ไอทีรอบฉาย ไอทีภาพยนตร์ ชื่อภาพยนตร์ ประเภทภาพยนตร์ การจำกัดอายุเข้าชม โรงภาพยนตร์ ระยะเวลาของภาพยนตร์(หน่วยนาทีก) แก้วี่ทั้งหมดใน1โรง ที่นั่งที่ถูกจอง รายได้ทั้งหมดต่อวัน สถานะ

1.3 ประโยชน์ที่ได้รับ

1.3.1 ช่วยจัดการข้อมูลภาพยนตร์ได้อย่างเป็นระเบียบ

1.3.2 ลดความซ้ำซ้อนและความผิดพลาดของข้อมูล

1.3.3 ค้นหาและดูข้อมูลได้รวดเร็วขึ้น

1.3.4 ผู้จัดการสามารถดูรายงานสรุปเพื่อใช้ตัดสินใจได้ง่าย

1.4 เครื่องมือที่คาดว่าจะต้องใช้

1.4.1 โปรแกรม Visual Studio Code

1.4.2 Microsoft Office

บทที่ 2

ระบบจัดการภาพยนตร์

2.1 แฟ้มเก็บข้อมูลภาพยนตร์ movies.dat

แฟ้มเก็บข้อมูลภาพยนตร์ประกอบด้วย 9 फिल्ตหลัก ซึ่งแต่ละ फिल्ตมีรายละเอียดและความสำคัญดังนี้

फिल्ต	ชนิด	ขนาด(bytes)	คำอธิบาย
movie_id	I	4	รหัสของภาพยนตร์
status	I	4	สถานะ Active = 1, Deleted = 0
duration_min	I	4	ระยะเวลาของภาพยนตร์ (นาทิจ)
ticket_Price	F	4	ราคาตั๋ว เช่น
title	32s	32	ชื่อภาพยนตร์ Avengers
genre	16s	16	ประเภทของภาพยนตร์ เช่น
rating	8s	8	เรตติ้ง
created_at	I	4	timestamp
Updated_at	I	4	timestamp

ตารางที่ 1 แฟ้มเก็บข้อมูลภาพยนตร์

2.1.1 movie_id รหัสภาพยนตร์

movie_id เป็นรหัสที่ใช้ระบุภาพยนตร์แต่ละเรื่องในระบบ โดยไม่ซ้ำกัน กำหนดในรูปแบบจำนวนเต็ม (integer) เช่น 7001, 7002, 7003 เป็นต้น รหัสนี้เป็น primary key ที่ช่วยให้ค้นหาและอ้างอิงได้อย่างแม่นยำและรวดเร็ว

2.1.2 status สถานะ

status ใช้เก็บสถานะการใช้งานของภาพยนตร์ โดยกำหนดเป็นจำนวนเต็ม (integer)

1 = Active (ยังฉายอยู่ในระบบ) 0 = Deleted (ถูกลบออกจากระบบแล้ว)

การมีสถานะนี้ช่วยให้ระบบสามารถควบคุมและคัดกรองข้อมูลได้โดยไม่จำเป็นต้องลบระเบียนทิ้งจริงซึ่งเป็นประโยชน์ต่อการอ้างอิงย้อนหลังหรือการจัดทำรายงานสรุป นอกจากนี้ยังช่วยป้องกันปัญหาการสูญหายของข้อมูลเดิมเมื่อมีการอัปเดตระบบ

2.1.3 duration_min ความยาวภาพยนตร์

duration_min ใช้สำหรับจัดเก็บข้อมูลความยาวของภาพยนตร์ หน่วยเป็นนาที เช่น 90, 120 หรือ 150 นาที ความยาวของภาพยนตร์ถือเป็นปัจจัยสำคัญที่มีผลต่อการจัดรอบฉายในโรงภาพยนตร์ เนื่องจากจะต้องคำนวณระยะเวลาฉายจริง รวมถึงช่วงเวลาพักระหว่างรอบ

2.1.4 ticket_price ราคาตั๋ว

ticket_price เป็นฟิลด์ที่ใช้เก็บข้อมูลราคาตั๋วของภาพยนตร์หนึ่งเรื่อง โดยเก็บเป็นตัวเลขทศนิยม (float) เช่น 200.00, 220.00 หรือ 250.00 บาท การบันทึกราคาตั๋วมีความสำคัญเพราะใช้เป็นพื้นฐานในการคำนวณรายได้และสร้างสถิติที่เกี่ยวข้องกับผลประกอบการของโรงภาพยนตร์ อีกทั้งยังช่วยให้สามารถเปรียบเทียบราคาเฉลี่ยของภาพยนตร์ประเภทต่าง ๆ และใช้เป็นข้อมูลประกอบการตัดสินใจทางธุรกิจ

2.1.5 title ชื่อภาพยนตร์

title คือชื่อของภาพยนตร์แต่ละเรื่อง ใช้แสดงเพื่อให้ผู้ใช้งานจดจำและอ้างอิงได้ง่าย ข้อมูลนี้ถูกจัดเก็บในรูปแบบข้อความ (string) โดยมีความยาวสูงสุด 32 ตัวอักษร เช่น Avengers, Avatar หรือ Doraemon การมีชื่อเรื่องทำให้ผู้ใช้งานสามารถค้นหาภาพยนตร์ได้สะดวก และยังเป็นข้อมูลที่มีความสำคัญในการสร้างรายงานหรือตารางสรุป เนื่องจากชื่อเรื่องเป็นข้อมูลที่สามารถสื่อสารกับผู้ใช้งานได้โดยตรง

2.1.6 genre ประเภทภาพยนตร์

genre ใช้สำหรับจัดเก็บประเภทหรือหมวดหมู่ของภาพยนตร์ เช่น Action, Comedy, Horror, Romance หรือ Animation เป็นต้น การกำหนดประเภทมีความสำคัญเนื่องจากช่วยให้ระบบสามารถจัดหมวดหมู่และกรองข้อมูลได้สะดวกยิ่งขึ้น

2.1.7 rating เรตติ้ง

เป็นข้อมูลที่ใช้บ่งบอกกลุ่มผู้ชมที่เหมาะสมกับภาพยนตร์ โดยอ้างอิงตามเกณฑ์มาตรฐาน เช่น G (ทั่วไป), PG-13 (เด็กอายุต่ำกว่า 13 ควรเป็นผู้ปกครองแนะนำ), 18+ (จำกัดผู้ชมที่มีอายุเกิน 18 ปี) เป็นต้น ข้อมูลนี้ถูกเก็บในรูปแบบข้อความสั้น (string) ยาวไม่เกิน 8 ตัวอักษร การมีเรตติ้งช่วยให้โรงภาพยนตร์สามารถกำหนดข้อจำกัดในการเข้าชมได้ชัดเจน และยังเป็นการปฏิบัติตามข้อกำหนดด้านกฎหมายหรือข้อบังคับที่เกี่ยวข้องกับการเผยแพร่สื่อภาพยนตร์

2.1.8 created_at วันเวลาที่สร้าง

created_at เป็นฟิลด์ที่เก็บค่าเวลาหรือ timestamp ของการสร้างข้อมูลภาพยนตร์ครั้งแรก การบันทึกเวลาที่สร้างข้อมูลมีความสำคัญต่อการติดตามประวัติและลำดับการเพิ่มภาพยนตร์ในระบบ อีกทั้งยังช่วยให้ผู้ใช้งานสามารถตรวจสอบได้ว่าภาพยนตร์เรื่องใดถูกเพิ่มเข้ามาใหม่ในช่วงเวลาใด

2.1.9 updated_at วันเวลาที่แก้ไขล่าสุด

updated_at เป็นฟิลด์ที่ใช้บันทึกค่าเวลาหรือ timestamp ของการแก้ไขข้อมูลภาพยนตร์ครั้งล่าสุด ฟิลด์นี้ช่วยให้สามารถตรวจสอบและติดตามการเปลี่ยนแปลงของข้อมูลได้ เช่น การแก้ไขชื่อเรื่อง การปรับเปลี่ยนราคาตั๋ว หรือการปรับสถานะของภาพยนตร์ การเก็บข้อมูลเวลาล่าสุดนี้ทำให้ระบบสามารถควบคุมคุณภาพและความถูกต้องของข้อมูลได้ดียิ่งขึ้น และช่วยลดความสับสนในการใช้งานจริง

2.2 แฟ้มเก็บประวัติการแก้ไข moviesLog.dat

แฟ้มเก็บประวัติการแก้ไขประกอบด้วย 9 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด(bytes)	คำอธิบาย
ts	I	4	timestamp เหตุการณ์
op_code	I	4	1=ADD, 2=UPDATE, 3=DELETE, 4=VIEW
movie_id	I	4	รหัสภาพยนตร์
status_after	I	4	สถานะหลังการแก้ไข
duration_after	I	4	ระยะเวลาของภาพยนตร์หลังการแก้ไข
price_after	F	4	ราคาตัวหลังการแก้ไข
revenue_after	F	4	รายได้ทั้งหมดของภาพยนตร์ในวันนั้น

ตารางที่ 2 แฟ้มเก็บประวัติแก้ไข

2.2.1 ts เวลาที่เกิดเหตุการณ์

ts เป็นฟิลด์ที่เก็บค่าเวลา (timestamp) ของเหตุการณ์แต่ละครั้ง เพื่อบอกว่าการเปลี่ยนแปลงเกิดขึ้นเมื่อใด เช่น วันที่และเวลาที่ผู้ใช้ทำการเพิ่ม ลบ หรือแก้ไขภาพยนตร์ ข้อมูลนี้ช่วยให้สามารถตรวจสอบประวัติย้อนหลังได้อย่างถูกต้องและแม่นยำ

2.2.2 op_code รหัสการทำงาน

op_code ใช้เก็บรหัสตัวเลขที่บ่งบอกประเภทของการทำงานที่เกิดขึ้น โดยกำหนดเป็นจำนวนเต็ม (integer) ได้แก่ 1 = Add (เพิ่มภาพยนตร์ใหม่), 2 = Update (แก้ไขข้อมูลภาพยนตร์), 3 = Delete (ลบหรือเปลี่ยนสถานะภาพยนตร์), 4 = View (เข้าดูข้อมูลภาพยนตร์) การกำหนดรหัสการทำงานในลักษณะนี้ช่วยให้ระบบสามารถจำแนกและประมวลผลเหตุการณ์ได้สะดวก รวมทั้งสามารถสรุปจำนวนครั้งของการใช้งานแต่ละประเภทในภายหลังได้

2.2.3 movie_id รหัสภาพยนตร์

movie_id เป็นรหัสของภาพยนตร์ที่เกี่ยวข้องกับเหตุการณ์นั้น ๆ เพื่อให้สามารถเชื่อมโยงกลับไปยังข้อมูลหลักในแฟ้ม **movies.dat** ได้อย่างถูกต้อง การจัดเก็บฟิลด์นี้ช่วยให้สามารถวิเคราะห์ประวัติการเปลี่ยนแปลงของหนังแต่ละเรื่องได้อย่างเป็นระบบ

2.2.4 status_after สถานะหลังการแก้ไข

status_after เป็นฟิลด์ที่เก็บค่าสถานะของภาพยนตร์หลังจากมีการดำเนินการ เช่น

1 = Active (ยังอยู่ในระบบ)

0 = Deleted (ถูกนำออกจากระบบแล้ว)

ข้อมูลนี้มีความสำคัญเพราะช่วยบอกได้ทันทีว่าผลลัพธ์หลังเหตุการณ์คืออะไร เช่น ภาพยนตร์ยังคงฉายในระบบ หรือถูกยกเลิกไปแล้ว

2.2.5 duration_after ความยาวหลังการแก้ไข

duration_after ใช้เก็บระยะเวลาของภาพยนตร์ (นาทิจ) หลังจากที่มีการแก้ไข ตัวอย่างเช่น ภาพยนตร์บางเรื่องอาจมีการปรับลดหรือเพิ่มความยาวเพื่อจัดรอบฉายใหม่ ข้อมูลนี้ช่วยให้สามารถติดตามการเปลี่ยนแปลงได้ว่าภาพยนตร์แต่ละเรื่องเคยถูกปรับแก้ความยาวมาแล้วกี่ครั้ง

2.2.6 price_after ราคาตัวหลังการแก้ไข

price_after ใช้เก็บราคาตัวใหม่ของภาพยนตร์ภายหลังการแก้ไข เช่น การปรับราคาในช่วงโปรโมชั่นหรือปรับขึ้นตามนโยบายโรงภาพยนตร์ ข้อมูลนี้ช่วยให้สามารถวิเคราะห์ประวัติการเปลี่ยนแปลงราคาตัว และนำไปคำนวณรายได้ที่สอดคล้องกับช่วงเวลานั้นได้

2.2.7 revenue_after รายได้หลังการแก้ไข

revenue_after เป็นฟิลด์ที่ใช้เก็บข้อมูลรายได้รวมของภาพยนตร์ในวันที่มีการแก้ไข ซึ่งคำนวณจากจำนวนที่นั่งที่ถูกจองรวมกับราคาตั๋วในวันนั้น การเก็บข้อมูลรายได้ ณ เวลาที่มีการเปลี่ยนแปลงช่วยให้สามารถตรวจสอบผลกระทบของการแก้ไข (เช่น การปรับราคา) ต่อรายได้ของภาพยนตร์ได้อย่างเป็นรูปธรรม

2.3 แฟ้มเก็บข้อมูลรอบฉาย screening.dat

แฟ้มเก็บข้อมูลภาพยนตร์ประกอบด้วย 9 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียดและความสำคัญดังนี้

ฟิลด์	ชนิด	ขนาด(bytes)	คำอธิบาย
screening_id	I	4	รหัสรอบฉาย
movie_id	I	4	อ้างอิง movies.dat
hall	I	4	เลขโรง
capacity	I	4	จำนวนที่นั่งในโรงนั้น
booked	I	4	จำนวนที่ถูกจองแล้ว

ตารางที่ 3 แฟ้มเก็บข้อมูลรอบฉาย

2.3.1 screening_id รหัสรอบฉาย

screening_id เป็นรหัสประจำรอบฉายภาพยนตร์แต่ละรอบ ใช้ในรูปแบบจำนวนเต็ม (integer) เช่น 101, 102, 103 เป็นต้น การมีรหัสนี้ทำให้สามารถแยกการฉายที่ต่างกันของภาพยนตร์เรื่องเดียวกันได้ เช่น *Avengers* อาจมีหลายรอบ แต่ละรอบจะถูกกำหนดด้วย screening_id ไม่ซ้ำกัน

2.3.2 movie_id รหัสภาพยนตร์ (อ้างอิง)

movie_id เป็นรหัสที่เชื่อมโยงไปยังแฟ้ม *movies.dat* เพื่อระบุว่ารอบฉายนั้นเป็นของภาพยนตร์เรื่องใด ตัวอย่างเช่น หาก movie_id = 7001 หมายถึงรอบฉายนั้นเป็นของภาพยนตร์

Tha Rae: The Exorcist การเก็บข้อมูลนี้ช่วยให้ระบบสามารถจัดตารางรอบฉายแยกตามภาพยนตร์ได้อย่างถูกต้อง

2.3.3 hall เลขโรง

hall ใช้บันทึกหมายเลขโรงภาพยนตร์ที่ใช้ฉายในรอบนั้น เช่น 1, 2, 3 หรือ 4 แต่ละโรงอาจมีจำนวนที่นั่งและขนาดต่างกัน การบันทึกเลขโรงมีความสำคัญต่อการจัดการตารางการฉาย เนื่องจากต้องวางแผนการใช้งานโรงฉายให้สอดคล้องกับความต้องการของผู้ชม

2.3.4 capacity จำนวนที่นั่ง

capacity ใช้บันทึกจำนวนที่นั่งทั้งหมดของโรงภาพยนตร์ในรอบฉายนั้น เช่น 100, 150 หรือ 200 ที่นั่งฟิลด์นี้เป็นข้อมูลพื้นฐานที่ใช้ในการคำนวณรายได้สูงสุดของรอบฉาย และใช้เปรียบเทียบกับจำนวนที่ถูกจองจริงเพื่อวิเคราะห์อัตราการใช้

2.3.5 booked จำนวนที่ถูกจองแล้ว

booked ใช้เก็บจำนวนที่นั่งที่มีการจองแล้วในรอบนั้น เช่น 56, 120, 180 เป็นต้น ข้อมูลนี้ช่วยให้ระบบสามารถแสดงผลจำนวนที่นั่งว่าง ($\text{capacity} - \text{booked}$) และใช้ในการรายงานสถิติการเข้าชมภาพยนตร์ของแต่ละรอบ

2.4 ไฟล์ report.txt

ไฟล์ **report.txt** เป็นไฟล์รายงานสรุปข้อมูลจากระบบจัดการโรงภาพยนตร์ ซึ่งใช้สำหรับเก็บข้อมูลภาพรวมของภาพยนตร์ทั้งหมดในระบบ ทั้งส่วนของรายละเอียดภาพยนตร์ รอบฉาย รายได้ และข้อมูลสถิติที่ได้จากการประมวลผลของระบบ โดยมีวัตถุประสงค์เพื่อให้ผู้ใช้หรือผู้ดูแลระบบสามารถตรวจสอบข้อมูลสรุปได้อย่างรวดเร็วในรูปแบบข้อความ (Text File) ที่อ่านง่ายและเข้าใจได้ทันที

Cinema Management System – Summary Report (Sample)											
Generated At : 2025-10-10 12:00:00 (+07:00)											
App Version : 1.0											
Endianness : Little-Endian											
Encoding : UTF-8 (fixed-length)											
ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	Tha Rae: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Active
106	7006	School Life	Drama	G	4	105	100	70	180.00	12,600.00	Deleted
Summary											
- Total Screenings : 6											
- Total Capacity : 1010											
- Total Booked : 850											
- Total Revenue : 183,600.00 THB											
- Avg Fill Rate : 84.16 %											
Ticket Price Statistics (Active only)											
- Min : 200.00											
- Max : 230.00											
- Avg : 218.00											
Movies by Genre (Active only)											
- Horror : 3 รอบ											
- Romance : 1 รอบ											
- Action : 1 รอบ											

ภาพที่ 1 ไฟล์ report

2.4.1 header_text ส่วนหัวรายงาน

เป็นส่วนที่ใช้ระบุชื่อรายงาน เช่น “Cinema Management System – Summary Report” เพื่อแสดงให้เห็นทราบว่าไฟล์นี้เป็นรายงานสรุปผลจากระบบจัดการโรงภาพยนตร์ ซึ่งช่วยให้ผู้ใช้งานสามารถแยกแยะไฟล์รายงานจากไฟล์ข้อมูลอื่น ๆ ได้อย่างชัดเจน

2.4.2 generated_at วันที่และเวลาที่สร้างรายงาน

ระบุวันและเวลาที่ระบบสร้างรายงานในขณะนั้น เช่น “Generated At : 2025-10-10 12:00:00 (+07:00)” ข้อมูลส่วนนี้ช่วยให้ทราบถึงช่วงเวลาที่ยังงานถูกจัดทำ ซึ่งเป็นประโยชน์ต่อการติดตามประวัติการอัปเดตข้อมูลหรือการวิเคราะห์ข้อมูลย้อนหลัง

2.4.3 app_version เวอร์ชันของโปรแกรม

ใช้แสดงหมายเลขเวอร์ชันของระบบที่ใช้สร้างรายงาน เช่น “App Version : 1.0” ข้อมูลนี้ช่วยในการตรวจสอบความสอดคล้องของไฟล์รายงานกับเวอร์ชันของโปรแกรม หากมีการปรับปรุงระบบหรือเพิ่มฟังก์ชันใหม่ในอนาคต

2.4.4 encoding การเข้ารหัสไฟล์

เป็นการระบุรูปแบบการเข้ารหัสข้อความของไฟล์ เช่น “Encoding : UTF-8 (fixed-length)” เพื่อให้แน่ใจว่าเมื่อเปิดไฟล์บนเครื่องอื่นหรือในระบบปฏิบัติการต่างกัน ข้อมูลจะยังคงแสดงผลได้ถูกต้องครบถ้วน

2.4.5 table_header ส่วนหัวของตารางข้อมูล

เป็นส่วนที่แสดงชื่อคอลัมน์ของข้อมูลภาพยนตร์ เช่น “ScreeningID | MovieID | Title | Genre | Rating | Hall | Duration(min) | Capacity | Booked | Ticket Price | Revenue(THB/day) | Status” หัวตารางนี้ช่วยให้การอ่านข้อมูลเป็นระบบระเบียบ เข้าใจง่าย และสามารถนำไปใช้งานวิเคราะห์หรือจัดทำรายงานอื่นต่อได้

2.4.6 movie_records รายการข้อมูลภาพยนตร์

ส่วนนี้เป็นข้อมูลรายละเอียดของภาพยนตร์แต่ละเรื่องในระบบ ซึ่งแสดงในรูปแบบตาราง ได้แก่ 1. รหัสรอบฉาย (ScreeningID) 2. รหัสภาพยนตร์ (MovieID) 3. ชื่อภาพยนตร์ (Title) 4. ประเภท (Genre) 5. เรตติ้ง (Rating) 6. โรงฉาย (Hall) 7. ระยะเวลา (Duration) 8. จำนวนที่นั่งทั้งหมด (Capacity) 9. จำนวนที่นั่งที่ถูกจอง (Booked) 10. ราคาตั๋ว (Ticket Price) 11. รายได้ต่อวัน (Revenue per day) 12. สถานะ (Status – Active/Deleted)

ข้อมูลในส่วนนี้ถือเป็นส่วนสำคัญของรายงาน เนื่องจากใช้แสดงรายละเอียดภาพยนตร์ทั้งหมดในระบบ

2.4.7 summary_section สรุปข้อมูลภาพรวมของระบบ

เป็นส่วนที่แสดงข้อมูลสรุปผลจากระบบในลักษณะสถิติ เช่น

จำนวนรอบฉายทั้งหมด (Total Screenings) จำนวนที่นั่งรวม (Total Capacity)

จำนวนที่ถูกจอง (Total Booked) รายได้รวมทั้งหมด (Total Revenue)

อัตราการจองเฉลี่ย (Average Fill Rate)

ข้อมูลเหล่านี้ช่วยให้ผู้ดูแลระบบสามารถประเมินภาพรวมของการดำเนินงานโรงภาพยนตร์ได้อย่างรวดเร็วและแม่นยำ

2.4.8 statistics_section ข้อมูลทางสถิติของภาพยนตร์

ส่วนนี้จะแสดงข้อมูลเพิ่มเติมเกี่ยวกับสถิติในระบบ เช่น

ราคาตั๋วต่ำสุด สูงสุด และเฉลี่ย (Min / Max / Avg Ticket Price)

จำนวนภาพยนตร์ในแต่ละประเภท เช่น Horror, Romance, Action เป็นต้น

จำนวนภาพยนตร์ที่ยัง Active อยู่ในระบบ

ส่วนนี้มีประโยชน์ในการวิเคราะห์แนวโน้มของประเภทภาพยนตร์ที่ได้รับความนิยม และช่วยให้ผู้ดูแลสามารถนำข้อมูลไปใช้วางแผนปรับปรุงระบบในอนาคตได้

บทที่ 3

การใช้งานระบบจัดการภาพยนตร์

โปรแกรมการจัดการภาพยนตร์ (Cinema Management System) พัฒนาขึ้นเพื่อช่วยให้ผู้ดูแลสามารถจัดการข้อมูลภาพยนตร์ได้อย่างสะดวก โดยการทำงานทั้งหมดจะอยู่ในรูปแบบเมนู Text-Based ผู้ใช้สามารถเลือกเมนูเพื่อทำงานต่าง ๆ เช่น เพิ่มข้อมูลภาพยนตร์, แก้ไขข้อมูลภาพยนตร์, ลบข้อมูลภาพยนตร์, และดูข้อมูลในระบบได้

ข้อมูลภาพยนตร์ที่จัดเก็บในระบบประกอบด้วย MovieID, Title, Genre, Rating, Duration, Ticket Price, และ Status โดยระบบสามารถค้นหาและแสดงผลตาม MovieID หรือแสดงทั้งหมดได้ตามที่ผู้ใช้เลือก การทำงานจะถูกออกแบบให้ใช้งานง่าย เพียงพิมพ์ตัวเลขเมนูที่ต้องการแล้วกด Enter ระบบจะประมวลผลและแสดงผลตามคำสั่งที่เลือก

สำหรับผู้ใช้งานโปรแกรม

3.1 การใช้งานโปรแกรมระบบจัดการภาพยนตร์

3.1.1 เมื่อผู้ใช้งานเปิดโปรแกรม ระบบจะปรากฏเมนูหลักบนหน้าจอ ซึ่งประกอบด้วยตัวเลือกการทำงานต่าง ๆ โดยผู้ใช้งานสามารถเลือกการทำงานได้โดยการพิมพ์หมายเลขของเมนูที่ต้องการแล้วกด Enter เมนูหลักมีรายละเอียดดังนี้

1. **Add (เพิ่ม)** : ใช้สำหรับเพิ่มข้อมูลภาพยนตร์หรือรอบฉายใหม่
2. **Update (แก้ไข)** : ใช้สำหรับปรับปรุงหรือแก้ไขข้อมูลภาพยนตร์ที่มีอยู่
3. **Delete (ลบ)** : ใช้สำหรับลบข้อมูลภาพยนตร์ โดยเปลี่ยนสถานะเป็น Deleted
4. **View (ดู)** : ใช้สำหรับแสดงข้อมูลภาพยนตร์และรอบฉายทั้งหมดในระบบ
5. **Generate Report (.txt)** : ใช้สำหรับสร้างรายงานสรุปข้อมูลในรูปแบบไฟล์ข้อความ
6. **Exit (ออก)** : ใช้สำหรับออกจากโปรแกรม

ตัวอย่างการเลือกใช้งาน หากผู้ใช้งานกรอกหมายเลข 1 แล้วกด Enter ระบบจะเข้าสู่ฟังก์ชัน

Add (เพิ่ม) เพื่อเพิ่มข้อมูลภาพยนตร์หรือรอบฉาย

- 1) Add (เพิ่ม)
- 2) Update (แก้ไข)
- 3) Delete (ลบ)
- 4) View (ดู)
- 5) Generate Report (.txt)
- 0) Exit (ออก)

เลือกเมนู: 1

ภาพที่ 2 ตัวอย่างการใช้งาน

3.2 การใช้งานโปรแกรมการเพิ่มข้อมูล

3.2.1 เมื่อผู้ใช้งานเลือกหมายเลข 1 ระบบจะแสดงเมนูย่อยเพื่อเลือกการเพิ่มข้อมูล โดยมีตัวเลือกดังนี้

- 1) เพิ่มหนัง
- 2) เพิ่มรอบฉาย
- 0) Back – กลับสู่เมนูก่อนหน้า

เมื่อเลือกหมายเลข 1 (เพิ่มหนัง) ระบบจะแสดงตารางข้อมูลทั้งหมดมาให้ดูว่าตอนนี้มี

ภาพยนตร์ทั้งหมดกี่เรื่อง

- 1) เพิ่มหนัง
- 2) เพิ่มรอบฉาย
- 0) Back

เลือกเมนู: 1

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	Tha Rae: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Active
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Active
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 3 ตารางข้อมูล

3.2.2 ระบบจะให้ผู้ใช้งานกรอกข้อมูลภาพยนตร์ใหม่ตามลำดับ 1. Movie ID (รหัสภาพยนตร์ใหม่) 2. Title (ชื่อของภาพยนตร์) 3. Genre (ประเภทของภาพยนตร์) 4. Rating (เรตติ้ง) 5. Duration (ระยะเวลา(นาที)ของภาพยนตร์) 6. Ticket Price (ราคาของตั๋ว)

เมื่อข้อมูลถูกบันทึกเรียบร้อยแล้ว ระบบจะแสดงผลภาพยนตร์เรื่องใหม่พร้อมตารางภาพยนตร์ทั้งหมด เพื่อให้ผู้ใช้งานตรวจสอบความถูกต้อง

Movie ID (ใหม่): 7008
Title: Sekiro
Genre: Action
Rating: 18+
Duration (mins): 62
Ticket Price: 180

เพิ่ม Movie 7008 : Sekiro (Active)

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	Tha Rae: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Active
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Active
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Active
-	7008	Sekiro	Action	18+	-	62	0	0	180.00	0.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 4 การแสดงผลภาพยนตร์

3.2.3 หลังจากนั้นระบบจะแสดงให้เพิ่มรอบฉาย ที่ประกอบไปด้วย ScreeningID โรงที่ฉาย จำนวนคนที่จองที่นั่ง เมื่อกรอกครบถ้วน ระบบจะแสดงข้อความยืนยันการเพิ่มรอบฉาย พร้อมทั้งอัปเดตตารางรอบฉายของภาพยนตร์เรื่องนั้น ๆ ให้เห็นทันที

กรุณาเพิ่มรอบฉายสำหรับตัวอย่างน้อย 1 รอบ:
Screening ID: 110
Hall (1-4): 3
Booked (0-150): 33
เพิ่ม Screening 110 ของหนัง Sekiro เรียบร้อย

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
110	7008	Sekiro	Action	18+	3	62	150	33	180.00	5,940.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 5 อัปเดตตาราง

3.2.4 หากผู้ใช้งานเลือกเพิ่มรอบฉาย ระบบจะแสดงตารางภาพยนตร์ทั้งหมดที่มีอยู่ เพื่อให้เลือกกว่าต้องการเพิ่มรอบฉายของภาพยนตร์เรื่องใด

1) เพิ่มหนัง
 2) เพิ่มรอบฉาย
 0) Back
 เลือกเมนู: 2

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	Tha Rae: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Active
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Active
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Active
110	7008	Sekiro	Action	18+	3	62	150	33	180.00	5,940.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 6 การแสดงตารางภาพยนตร์ทั้งหมด

3.2.5 หลังจากนั้นจึงทำการกรอกข้อมูลรอบฉายใหม่ เช่น Screening ID เลขโรง และจำนวนที่ถูกจอง ระบบจะแสดงตารางที่อัปเดตพร้อมข้อมูลรอบฉายใหม่ทันที

กรอก Movie ID ที่ต้องการเพิ่มรอบฉาย: 7008
 Screening ID: 111
 Hall(1-4): 2
 Booked (0-180): 180
 เพิ่มรอบฉายแล้ว

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	Tha Rae: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Active
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Active
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Active
110	7008	Sekiro	Action	18+	3	62	150	33	180.00	5,940.00	Active
111	7008	Sekiro	Action	18+	2	62	180	180	180.00	32,400.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 7 ตารางที่อัปเดตข้อมูล

3.3 การใช้งานโปรแกรมแก้ไขข้อมูล

3.3.1 เมื่อเข้าสู่เมนูหลัก ให้กรอกหมายเลข 2 แล้วกด Enter เพื่อเข้าสู่โหมด Update (แก้ไข)

- 1) Add (เพิ่ม)
- 2) Update (แก้ไข)
- 3) Delete (ลบ)
- 4) View (ดู)
- 5) Generate Report (.txt)
- 0) Exit (ออก)

เลือกเมนู: 2

ภาพที่ 8 การใช้งานโปรแกรม update

3.3.2 ระบบจะแสดงเมนูย่อย 3 ตัวเลือก พร้อม ตารางข้อมูลล่าสุด เพื่อช่วยตัดสินใจ

- 1) แก้ไขข้อมูล Movie – ปรับรายละเอียดของภาพยนตร์
- 2) แก้ไขข้อมูล Screening – ปรับรายละเอียดรอบฉาย
- 0) Back – กลับสู่เมนูก่อนหน้า

- 1) แก้ไขข้อมูล Movie
- 2) แก้ไขข้อมูล Screening
- 0) Back

เลือกเมนู: 1

ข้อมูลปัจจุบัน:

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	Tha Rae: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Active
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Active
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Active
110	7008	Sekiro	Action	18+	3	62	150	33	180.00	5,940.00	Active
111	7008	Sekiro	Action	18+	2	62	180	180	180.00	32,400.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 9 ตารางข้อมูล

3.3.3 แก้ไขข้อมูลภาพยนตร์ (Movie)

ระบบจะให้ระบุ MovieID ที่ต้องการแก้ไข จากนั้นจะแสดงแบบฟอร์มที่ละบรรทัด โดยมีค่าเดิมอยู่ในวงเล็บเหลี่ยม

Title ชื่อเรื่องใหม่ (กด Enter เพื่อคงค่าเดิม)

Genre ประเภท (รับค่าเดียว เช่น Action, Drama, Horror ฯลฯ)

Rating เรต เช่น G, 13+, 15+, 18+, 20+

Duration (mins) ความยาวเป็น “นาที” (ตัวเลข ≥ 0)

Ticket price ราคาตั๋ว (ตัวเลข ≥ 0)

Status (Active/Deleted) สถานะการฉาย

เมื่อกรอกจบ ระบบจะแสดงบรรทัดยืนยันผลว่าแก้ไขเรียบร้อยแล้ว พร้อมตารางที่สะท้อนค่าล่าสุดของเรื่องนี้

Movie ID ที่ต้องการแก้ไข: 7008
กำลังแก้ไข Movie 7008 : Sekiro กด 'Enter' ทกไม่ต้องการแก้ไข
Title [Sekiro]: One punch man
Genre [Action]: Action
Rating [18+]: 15+
Duration(mins) [62]: 90
Ticket price [180.00]: 160
Status (Active/Deleted) [Active]: Active

ฉันได้เรียบร้อยแล้ว: Movie 7008 : One punch man

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
110	7008	One punch man	Action	15+	3	90	150	33	160.00	5,280.00	Active
111	7008	One punch man	Action	15+	2	90	180	180	160.00	28,800.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 10 ผลบรรทัดยืนยันว่าแก้ไขเรียบร้อยแล้ว

3.3.4 เลือกแก้ไขข้อมูลรอบฉาย (Screening)

ระบบจะแสดงตารางรอบฉายทั้งหมด (มี ScreeningID, MovieID, Hall, Duration, Capacity, Booked, Ticket Price, Status) เพื่อใช้เป็นข้อมูลอ้างอิงก่อนแก้ไข

1) แก้ไข Movie
2) แก้ไขรอบฉาย Screening
3) Back

แก้ไขรอบฉาย: 2

ข้อมูลรอบฉาย:

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	The Rite: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Active
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Active
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Active
110	7008	One punch man	Action	15+	3	90	150	33	160.00	5,280.00	Active
111	7008	One punch man	Action	15+	2	90	180	180	160.00	28,800.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 11 ตารางแสดงรอบฉาย

3.3.4 แก้ไขรายละเอียดรอบฉาย

ระบุ ScreeningID ที่ต้องการแก้ไข ระบบจะแสดงแบบฟอร์มที่ละบรรทัด พร้อมค่าเดิมใน ให้พิมพ์ค่าใหม่ หรือกด Enter เพื่อคงเดิม เช่น Hall หมายเลขโรง Booked จำนวนที่ถูจอง (ต้องเป็นตัวเลข และไม่เกิน Capacity) Ticket Price ราคาตัวของรอบนั้น (ถ้ารอบยึดราคาตามหนัง ให้คงเดิม) Status Active/Deleted เมื่อยืนยัน ระบบจะแสดงตารางที่อัปเดตแล้วทันที เพื่อให้ตรวจสอบความถูกต้องของรอบฉายที่แก้ไข

Movie ID ที่ต้องการแก้ไข: 7008
กำลังแก้ไข Movie 7008 : Sekiro กด Enter' หากไม่ต้องการแก้ไข
Title [Sekiro]: One punch man
Genre [Action]: Action
Rating [18+]: 15+
Duration(mins) [62]: 90
Ticket price [180.00]: 160
Status (Active/Deleted) [Active]: Active

อัปเดตเรียบร้อยแล้ว: Movie 7008 : One punch man

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
110	7008	One punch man	Action	15+	3	90	150	33	160.00	5,280.00	Active
111	7008	One punch man	Action	15+	2	90	180	180	160.00	28,800.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 12 ตารางแสดงอัปเดต

3.4 การใช้งานโปรแกรมการลบข้อมูล

3.4.1 เมื่อผู้ใช้เข้าสู่เมนูการลบ (Delete Menu)

เมื่อผู้ใช้เปิดโปรแกรม ระบบจะแสดงเมนูหลัก (Main Menu) บนหน้าจอ โดยมีตัวเลือกการทำงานต่าง ๆ เช่น เพิ่ม แก้ไข ลบ และดูข้อมูล เมื่อผู้ใช้ต้องการลบข้อมูล ให้กรอกหมายเลข 3 แล้วกด Enter เพื่อเข้าสู่เมนูย่อยของ ลบ (Delete)

- ```

1) Add (เพิ่ม)
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Generate Report (.txt)
0) Exit (ออก)

```

เลือกเมนู: 3

ภาพที่ 13 การใช้งานโปรแกรม delete

### 3.4.2 การลบแบบชั่วคราว (Soft Delete)

เมื่อผู้ใช้เลือกหมายเลข 1 ระบบจะแสดงตารางข้อมูลภาพยนตร์ทั้งหมดในระบบ เพื่อให้ผู้ใช้สามารถตรวจสอบข้อมูลก่อนทำการลบ จากนั้นผู้ใช้สามารถระบุ Movie ID ที่ต้องการลบได้ ระบบจะทำการเปลี่ยนสถานะของภาพยนตร์จาก Active เป็น Deleted โดยไม่ได้ลบข้อมูลออกจากฐานข้อมูลจริง ซึ่งทำให้สามารถเรียกคืนข้อมูลได้ในภายหลัง หากเกิดข้อผิดพลาดหรือมีการเปลี่ยนแปลงในภายหลัง

- ```

-----
1) ลบ Movie (ตั้งสถานะเป็น Deleted)
2) ลบ Movie ออกจากข้อมูลจริง (Hard delete)
0) Back
-----

```

เลือกเมนู: 1

ข้อมูลปัจจุบัน:

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	Tha Rae: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Active
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Active
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Active
110	7008	One punch man	Action	15+	3	90	150	33	160.00	5,280.00	Active

กด Enter เพื่อดำเนินการต่อ...

Movie ID ที่ต้องการลบ (soft delete): 7007
 จะตั้งสถานะ Deleted ให้ Movie 7007 : Super_man
 ตั้งสถานะ Deleted แล้ว: Movie 7007 : Super_man

ภาพที่ 14 สถานะของภาพยนตร์

3.3.5 แสดงผลหลังการลบแบบชั่วคราว

เมื่อการลบเสร็จสิ้น ระบบจะแสดงผลตารางข้อมูลภาพยนตร์ใหม่ โดยภาพยนตร์ที่ถูกลบแบบ Soft Delete จะยังคงปรากฏอยู่ในตาราง แต่สถานะจะเปลี่ยนเป็น Deleted เพื่อแสดงให้เห็นว่าข้อมูลนั้นถูกระงับการใช้งานชั่วคราว และไม่สามารถทำรายการอื่น ๆ ได้จนกว่าจะถูกแก้ไขกลับเป็น Active

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	Tha Rae: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Deleted
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Deleted
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Deleted
110	7008	One punch man	Action	15+	3	90	150	33	160.00	5,280.00	Active

ภาพที่ 15 การแสดงผลการลบ

3.4.4 การลบแบบถาวร (Hard Delete)

หากผู้ใช้เลือกหมายเลข 2 จากเมนูการลบ ระบบจะทำการเปิดตารางข้อมูลภาพยนตร์ทั้งหมดอีกครั้ง เพื่อให้ผู้ใช้สามารถเลือกภาพยนตร์ที่ต้องการลบออกจากฐานข้อมูลอย่างถาวร โดยผู้ใช้ต้องระบุ

Movie ID ของรายการที่ต้องการลบ

- 1) ลบ Movie (ตั้งสถานะเป็น Deleted)
 2) ลบ Movie ออกจากข้อมูลจริง (Hard delete)
 0) Back

เลือกเมนู: 2

ข้อมูลปัจจุบัน:

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
101	7001	Tha Rae: The Exorcist	Horror	18+	1	120	200	150	220.00	33,000.00	Active
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Deleted
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Deleted
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Deleted
110	7008	One punch man	Action	15+	3	90	150	33	160.00	5,280.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 16 เลือกภาพยนตร์ที่ต้องการลบ

3.4.5 เมื่อผู้ใช้ยืนยันการลบ ระบบจะแสดงข้อความยืนยัน เช่น

ยืนยันลบ Movie 7001 : Tha_Rae_The_Exorcist ออกจากทุกไฟล์หรือไม่ (Y/N)

หากผู้ใช้พิมพ์ Y ระบบจะดำเนินการลบข้อมูลภาพยนตร์เรื่องนั้นออกจากฐานข้อมูลทั้งหมดอย่างถาวร

Movie ID ที่ต้องการลบออกจากข้อมูลจริง: 7001

ยืนยันลบ Movie 7001 : Tha_Rae_The_Exorcist ออกจากทุกไฟล์? (Y/N): y

ลบเรียบร้อยแล้ว: Movie 7001 : Tha_Rae_The_Exorcist ออกจากทุกไฟล์แล้ว

ภาพที่ 17 แสดงข้อความยืนยัน

3.4.6 แสดงผลหลังการลบแบบถาวร

หลังจากการลบเสร็จสิ้น ระบบจะแสดงตารางภาพยนตร์อีกครั้ง โดยข้อมูลภาพยนตร์ที่ถูกลบแบบถาวรจะไม่ปรากฏอยู่ในตารางอีกต่อไป ซึ่งหมายความว่าภาพยนตร์ดังกล่าวได้ถูกลบออกจากฐานข้อมูลทั้งหมดเรียบร้อยแล้ว

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Deleted
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Deleted
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Deleted
110	7008	One punch man	Action	15+	3	90	150	33	160.00	5,280.00	Active

ภาพที่ 18 การแสดงผลหลังจากลบ

3.5 การใช้งานโปรแกรมดูข้อมูล

3.5.1 ผู้ใช้เลือกหมายเลข **4) View (ดู)** จากรายการหลัก เพื่อเข้าสู่เมนูสำหรับการดูข้อมูลภาพยนตร์ ดังแสดงในภาพ เมื่อพิมพ์หมายเลข 4 แล้วกด Enter ระบบจะเข้าสู่เมนูย่อยของการดูข้อมูลเมื่อเข้าสู่เมนู View ระบบจะแสดงเมนูย่อยให้ผู้ใช้เลือกได้

```

-----
1) Add (เพิ่ม)
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Generate Report (.txt)
0) Exit (ออก)
-----

```

เลือกเมนู: 4

ภาพที่ 19 การใช้งาน view

3.5.2 การดูข้อมูลเฉพาะเรื่อง (View One)

ในกรณีที่ผู้ใช้เลือกหมายเลข 1) View One ระบบจะให้กรอก Movie ID ของภาพยนตร์ที่ต้องการดู เมื่อกรอกหมายเลขภาพยนตร์ ระบบจะแสดงข้อมูลของภาพยนตร์เรื่องนั้นโดยละเอียด เช่น ชื่อเรื่อง ประเภท เรตอายุ ระยะเวลาฉาย ราคาตั๋ว และสถานะของภาพยนตร์ (Active / Deleted) ตัวอย่างเช่น เมื่อเลือกดูภาพยนตร์ที่มีรหัส 7002 ระบบจะแสดงข้อมูลของภาพยนตร์เรื่อง *Death Whisperer 3* ตามที่ปรากฏในภาพ

```

-----
1) Add (เพิ่ม)
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Generate Report (.txt)
0) Exit (ออก)
-----

```

เลือกเมนู: 4

```

-----
1) View One
2) View All
3) Filter View
4) Summary Statistics
0) Back
-----

```

เลือกเมนู: 1

กรอก Movie ID: 7002

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 20 การดูข้อมูลเฉพาะเรื่อง

3.5.2 การดูข้อมูลทั้งหมด (View All)

หากผู้ใช้เลือกหมายเลข 2) View All ระบบจะแสดงรายการภาพยนตร์ทั้งหมดในระบบโดยไม่ต้องการกรอกข้อมูลเพิ่มเติม โดยแต่ละแถวจะแสดงรายละเอียดครบถ้วนของแต่ละเรื่อง เช่น รหัสรอบฉาย (Screening ID), รหัสภาพยนตร์ (Movie ID), ชื่อเรื่อง, ประเภท, เรตติ้ง, ระยะเวลา, จำนวนที่นั่ง, จำนวนที่จอง, ราคาตั๋ว และรายได้ต่อวัน ผลลัพธ์จะแสดงในรูปแบบตารางเพื่อความชัดเจนในการตรวจสอบข้อมูล

1) View One
2) View All
3) Filter View
4) Summary Statistics
0) Back

เลือกเมนู: 2

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
105	7005	Warrior of Siam	Action	13+	2	130	180	180	230.00	41,400.00	Deleted
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
107	7007	Super man	Action	G	3	160	150	140	220.00	30,800.00	Deleted
108	7007	Super man	Action	G	1	160	200	46	220.00	10,120.00	Deleted
109	7007	Super man	Action	G	4	160	100	56	220.00	12,320.00	Deleted
110	7008	One punch man	Action	15+	3	90	150	33	160.00	5,280.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 21 การดูข้อมูลทั้งหมด

3.5.3 การกรองข้อมูล (Filter View)

ในกรณีที่ผู้ใช้เลือกหมายเลข 3) Filter View ระบบจะให้เลือกเงื่อนไขการกรองข้อมูล ได้แก่

1. สถานะ (Status)
2. ประเภทภาพยนตร์ (Genre)
3. เรตติ้ง (Rating)
4. ราคาตั๋ว (Ticket Price)

เช่น หากผู้ใช้เลือกกรองตามสถานะ (Active / Deleted) ระบบจะแสดงเฉพาะข้อมูลของภาพยนตร์ที่มีสถานะตามที่ระบุ เช่น “Active”

การเลือกตัวเลือกอื่น ๆ เช่น การกรองตามประเภท เรตติ้ง หรือราคาตั๋ว จะมีลักษณะการทำงานเช่นเดียวกัน กล่าวคือ ระบบจะกรองข้อมูลและแสดงเฉพาะรายการที่ตรงตามเงื่อนไขที่ผู้ใช้เลือกไว้

1) View One
2) View All
3) Filter View
4) Summary Statistics
0) Back

เลือกเมนู: 3
คำกรอง: 1) Status 2) Genre 3) Rating 4) Ticket price (<=)
เลือก: 1
กรอง Active/deleted: Active

ScreeningID	MovieID	Title	Genre	Rating	Hall	duration(min)	Capacity	Booked	Ticket Price	Revenue(THB/day)	Status
102	7002	Death Whisperer 3	Horror	18+	2	118	180	160	220.00	35,200.00	Active
103	7003	The Red Envelope	Horror	18+	1	125	200	170	220.00	37,400.00	Active
104	7004	Bangkok Love Story	Romance	15+	3	110	150	120	200.00	24,000.00	Active
106	7006	Avenger	Action	G	3	183	150	140	220.00	30,800.00	Active
110	7008	One punch man	Action	15+	3	90	150	33	160.00	5,280.00	Active

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 22 กรองข้อมูลที่ตรงตามเงื่อนไข

3.5.4 การดูสถิติข้อมูล (Summary Statistics)

เมื่อผู้ใช้เลือกเมนู 4) Summary Statistics ระบบจะประมวลผลและแสดงผลสถิติภาพรวมของโรงภาพยนตร์ เช่น จำนวนรอบฉายทั้งหมด จำนวนที่นั่งและที่ถูกจอง รายได้รวม ค่าเฉลี่ยอัตราการจอง รวมถึงราคาตั๋วขั้นต่ำ สูงสุด และค่าเฉลี่ยของโรงภาพยนตร์ที่ยัง Active พร้อมทั้งแสดงจำนวนโรงภาพยนตร์แยกตามประเภท เช่น Horror, Romance และ Action เพื่อให้ผู้ใช้สามารถประเมินภาพรวมของการดำเนินงานได้อย่างรวดเร็วและชัดเจน

-
- 1) View One
 - 2) View All
 - 3) Filter View
 - 4) Summary Statistics
 - 0) Back
-

เลือกเมนู: 4

Summary

- Total Screenings : 9
- Total Capacity : 1460
- Total Booked : 1045
- Total Revenue : 227,320.00 THB
- Avg Fill Rate : 71.58 %

Ticket Price Statistics (Active only)

- Min : 160.00
- Max : 220.00
- Avg : 204.00

Movies by Genre (Active only)

- Horror : 2 เรื่อง
- Romance : 1 เรื่อง
- Action : 2 เรื่อง

กด Enter เพื่อดำเนินการต่อ...

ภาพที่ 23 การดูสถิติข้อมูล

3.6 การสร้างรายงานผล (.txt)

เมื่อผู้ใช้เลือกเมนู 5) Generate Report (.txt) ระบบจะสร้างไฟล์รายงานอัตโนมัติในรูปแบบ .txt ซึ่งสรุปข้อมูลจากเมนู Summary Statistics พร้อมรายละเอียดของภาพยนตร์ที่อยู่ในสถานะ Active เช่น รายได้รวม รายการฉาย และสถิติราคาตั๋ว โดยไฟล์จะถูกบันทึกไว้ในโฟลเดอร์ report/ ชื่อ report.txt เพื่อให้สามารถเปิดดูหรือพิมพ์ภายหลังได้

```
-----
1) Add (เพิ่ม)
2) Update (แก้ไข)
3) Delete (ลบ)
4) View (ดู)
5) Generate Report (.txt)
0) Exit (ออก)
-----
เลือกเมนู: 5
สร้างไฟล์ report/report.txt แล้ว

กด Enter เพื่อดำเนินการต่อ...
-----
```

ภาพที่ 24 การสร้างรายงานแสดงผล

3.6.1 ตัวอย่างไฟล์ report

```

1 Cinema Backend Report
2 Generated at: 2025-10-02 22:59:29
3
4 Summary
5 - Total Movies      : 7
6 - Active Movies     : 5
7 - Deleted Movies    : 2
8 - Total Screenings  : 9
9 - Total Capacity    : 1460
10 - Total Booked      : 1045
11 - Total Revenue     : 227,320.00 THB
12 - Avg Fill Rate     : 71.58 %
13
14 Ticket Price Statistics (Active movies)
15 - Min : 160.00
16 - Max : 220.00
17 - Avg : 204.00
18
19 Movies by Genre (Active movies)
20 - Horror : 2 เรื่อง
21 - Romance : 1 เรื่อง
22 - Action : 2 เรื่อง
23
24 Table A - Active screenings
25 -----
26 | ScreeningID | MovieID | Title                | Genre | Rating | Hall | duration(min) | Capacity | Booked | Ticket Price | Revenue(THB/day)|Status |
27 -----
28 | 102         | 7002    | Death Whisperer 3    | Horror | 18+    | 2    | 118            | 180      | 160    | 220.00       | 35,200.00       | Active |
29 | 103         | 7003    | The Red Envelope     | Horror | 18+    | 1    | 125            | 200      | 170    | 220.00       | 37,400.00       | Active |
30 | 104         | 7004    | Bangkok Love Story   | Romance | 15+    | 3    | 110            | 150      | 120    | 200.00       | 24,000.00       | Active |
31 | 105         | 7005    | Warrior of Slam      | Action | 13+    | 2    | 130            | 180      | 180    | 230.00       | 41,400.00       | Deleted |
32 | 106         | 7006    | Avenger              | Action | G       | 3    | 183            | 150      | 140    | 220.00       | 30,800.00       | Active |
33 | 107         | 7007    | Super man            | Action | G       | 3    | 160            | 150      | 140    | 220.00       | 30,800.00       | Deleted |
34 | 108         | 7007    | Super man            | Action | G       | 1    | 160            | 200      | 46    | 220.00       | 10,120.00       | Deleted |
35 | 109         | 7007    | Super man            | Action | G       | 4    | 160            | 100      | 56    | 220.00       | 12,320.00       | Deleted |
36 | 110         | 7008    | One punch man        | Action | 15+    | 3    | 90             | 150      | 33    | 160.00       | 5,280.00        | Active |
37 -----

```

ภาพที่ 25 ตัวอย่างไฟล์

3.7 การออกจากโปรแกรม (Exit)

เมื่อผู้ใช้เลือกหมายเลข 0) Exit (ออก) ระบบจะสิ้นสุดการทำงาน และแสดงข้อความ “ขอบคุณที่ใช้บริการ” เพื่อยืนยันว่าผู้ใช้ได้ออกจากโปรแกรมสำเร็จแล้ว

-
- 1) Add (เพิ่ม)
 - 2) Update (แก้ไข)
 - 3) Delete (ลบ)
 - 4) View (ดู)
 - 5) Generate Report (.txt)
 - 0) Exit (ออก)
-

เลือกเมนู: 0

ขอบคุณที่ใช้บริการ

ภาพที่ 26 การออกจากไฟล์

บทที่ 4


อธิบายการทำงานของ Code

4.1 main.py

ไฟล์ main.py ทำหน้าที่เป็นจุดเริ่มต้นหลักของระบบจัดการข้อมูลโรงภาพยนตร์ ซึ่งเป็นไฟล์ที่ใช้ในการเรียกโปรแกรมทั้งหมดให้ทำงาน โดยจะเชื่อมโยงกับไฟล์ project.py เพื่อเข้าถึงฟังก์ชันการทำงานต่าง ๆ ของระบบ เช่น การเพิ่มข้อมูลภาพยนตร์ การเพิ่มรอบฉาย การแก้ไขข้อมูล การลบข้อมูล การดูข้อมูล รวมถึงการสรุปและสร้างรายงานผลในรูปแบบไฟล์ข้อความ (.txt)

4.1.1 from project import (...)

เป็นคำสั่งนำเข้าฟังก์ชันจากไฟล์ project.py เพื่อให้สามารถเรียกใช้งานได้ภายใน main.py ประกอบด้วยฟังก์ชันที่ใช้ในการจัดการข้อมูลหลัก เช่น การเพิ่ม แก้ไข ลบ ดู และสรุปรายงานของภาพยนตร์และรอบฉาย



```


1  from project import (
2      add_movie, add_screening,
3      update_movie, update_screening,
4      delete_menu,
5      view_one, view_all, filter_view, summary_stats,
6      generate_report,
7  )

```

ภาพที่ 27 การ import

4.1.2 def input_int(prompt, minv=None, maxv=None):

ฟังก์ชันนี้ทำหน้าที่รับค่าตัวเลขจากผู้ใช้โดยตรวจสอบความถูกต้องของข้อมูลที่ป้อนเข้ามา
ป้องกันการกรอกค่าที่ไม่ใช่ตัวเลข หรืออยู่นอกช่วงที่กำหนด และส่งค่าที่ผ่านการตรวจสอบกลับไปใช้งาน
ต่อในเมนูต่าง ๆ



```


1 def input_int(prompt, minv=None, maxv=None):
2     while True:
3         x = input(prompt).strip()
4         try:
5             v = int(x)
6             if minv is not None and v < minv:
7                 print("ค่าน้อยเกินไป"); continue
8             if maxv is not None and v > maxv:
9                 print("ค่ามากเกินไป"); continue
10            return v
11        except:
12            print("กรุณาใส่ตัวเลข")

```

ภาพที่ 28 ฟังก์ชันรับค่าตัวเลข

4.1.3 def print_menu():

ฟังก์ชันนี้ทำหน้าที่แสดงเมนูหลัก (Main Menu) ของโปรแกรม ประกอบด้วยตัวเลือกหลัก ได้แก่ 1) เพิ่มข้อมูล 2) แก้ไข 3) ลบ 4) ดูข้อมูล 5) สร้างรายงาน (.txt) 0) ออกจากโปรแกรม โดยจะเรียกใช้ฟังก์ชัน input_int() เพื่อรับคำตอบจากผู้ใช้ และส่งค่ากลับให้ฟังก์ชัน run_menu() ใช้งาน



```


1 def print_menu():
2     print("-----")
3     print("1) Add (เพิ่ม)")
4     print("2) Update (แก้ไข)")
5     print("3) Delete (ลบ)")
6     print("4) View (ดู)")
7     print("5) Generate Report (.txt)")
8     print("0) Exit (ออก)")
9     print("-----")
10    return input_int("เลือกเมนู: ", 0, 5)

```

ภาพที่ 29 ฟังก์ชันแสดงเมนู

4.1.4 def print_add_menu():

ฟังก์ชันนี้ทำหน้าที่แสดงเมนูย่อยสำหรับการ “เพิ่มข้อมูล” ภายในระบบ โดยให้ผู้ใช้เลือกว่า ต้องการเพิ่มข้อมูลภาพยนตร์ (Movie) หรือรอบฉาย (Screening) พร้อมส่งค่าที่ผู้ใช้เลือกกลับไปยังเมนูหลักเพื่อประมวลผลต่อ



```

1 def print_add_menu():
2     print("-----")
3     print("1) เพิ่มหนัง")
4     print("2) เพิ่มรอบฉาย")
5     print("0) Back")
6     print("-----")
7     return input_int("เลือกเมนู: ", 0, 2)

```

ภาพที่ 30 ฟังก์ชันแสดงเมนูย่อย

4.1.5 def print_update_menu():

ฟังก์ชันนี้ทำหน้าที่แสดงเมนูย่อยสำหรับ “แก้ไขข้อมูล” โดยให้ผู้ใช้เลือกว่า ต้องการแก้ไขข้อมูลภาพยนตร์ หรือแก้ไขข้อมูลรอบฉาย และส่งค่าการเลือกกลับไปยังเมนูหลักเพื่อดำเนินการแก้ไข



```
1 def print_update_menu():
2     print("-----")
3     print("1) แก้ไขข้อมูล Movie")
4     print("2) แก้ไขข้อมูล Screening")
5     print("0) Back")
6     print("-----")
7     return input_int("เลือกเมนู: ", 0, 2)
```

ภาพที่ 31 ฟังก์ชันแสดงเมนูย่อย"แก้ไขข้อมูล"

4.1.6 def print_view_menu():

ฟังก์ชันนี้ทำหน้าที่แสดงเมนูย่อยสำหรับ “การดูข้อมูล (View)” ให้ผู้ใช้เลือกได้หลายรูปแบบ เช่น ดูข้อมูลเฉพาะเรื่องเดียว ดูข้อมูลทั้งหมด ดูแบบกรองข้อมูล (Filter) ดูสถิติภาพรวม (Summary Statistics) จากนั้นส่งค่าที่ผู้ใช้เลือกกลับไปยังเมนูหลักเพื่อเรียกใช้ฟังก์ชันที่เกี่ยวข้อง



```
1 def print_view_menu():
2     print("-----")
3     print("1) View One")
4     print("2) View All")
5     print("3) Filter View")
6     print("4) Summary Statistics")
7     print("0) Back")
8     print("-----")
9     return input_int("เลือกเมนู: ", 0, 4)
10
```


ภาพที่ 32 ฟังก์ชันแสดงเมนู"การดูข้อมูล"

4.1.7 def run_menu():

ฟังก์ชันหลักของโปรแกรม ทำหน้าที่ควบคุมลำดับการทำงานทั้งหมดของระบบผ่านเมนูหลัก (Main Menu) เป็นศูนย์กลางในการเรียกใช้เมนูย่อย เช่น เพิ่มข้อมูล แก้ไขข้อมูล ลบข้อมูล ดูข้อมูล สร้างรายงานและเป็นจุดวนลูปหลักของโปรแกรมที่ทำงานต่อเนื่องจนกว่าผู้ใช้จะเลือกออกจากระบบ

if __name__ == "__main__":

ส่วนควบคุมการเริ่มต้นโปรแกรม ทำหน้าที่ตรวจสอบว่าถ้าไฟล์นี้ถูกเรียกใช้งานโดยตรงให้เริ่มทำงานโดยเรียกฟังก์ชัน run_menu() และมีระบบป้องกันการหยุดโปรแกรมผิดปกติ (KeyboardInterrupt)



```

1  def run_menu():
2      while True:
3          ch = print_menu()
4          if ch == 0:
5              print("ขอบคุณที่ใช้บริการ")
6              break
7          elif ch == 1:
8              sub = print_add_menu()
9              if sub == 1: add_movie()
10             elif sub == 2: add_screening()
11          elif ch == 2:
12              sub = print_update_menu()
13              if sub == 1: update_movie()
14              elif sub == 2: update_screening()
15          elif ch == 3:
16              delete_menu()
17          elif ch == 4:
18              while True:
19                  vv = print_view_menu()
20                  if vv == 0: break
21                  elif vv == 1: view_one()
22                  elif vv == 2: view_all()
23                  elif vv == 3: filter_view()
24                  elif vv == 4: summary_stats()
25          elif ch == 5:
26              generate_report()
27
28  if __name__ == "__main__":
29      try:
30          run_menu()
31      except KeyboardInterrupt:
32          print("\nออกจากโปรแกรม")
33

```

ภาพที่ 33 ส่วนควบคุมเริ่มต้นโปรแกรม

4.2 project.py

ไฟล์ `project.py` เป็นไฟล์หลักที่ใช้ในการประมวลผลและจัดการข้อมูลของระบบทั้งหมด โดยมีหน้าที่เกี่ยวข้องกับการจัดเก็บ แก้ไข และเรียกดูข้อมูลของภาพยนตร์ รวมถึงการบันทึกประวัติการดำเนินงาน และการสร้างรายงานสรุปผลการดำเนินการในระบบจัดการตัวภาพยนตร์

กล่าวได้ว่าไฟล์นี้เป็น “แกนกลางของระบบ (Core Processing Unit)” ซึ่งทำงานร่วมกับไฟล์ `main.py` ที่เป็นส่วนควบคุมการทำงานหลัก โดยฟังก์ชันภายใน `project.py` จะรับคำสั่งจากผู้ใช้ผ่าน `main.py` แล้วดำเนินการจัดการข้อมูลจริงในไฟล์ฐานข้อมูล `.dat` ที่เก็บไว้ในระบบ

4.2.1 import os

ฟังก์ชันนี้ทำหน้าที่นำเข้าโมดูล `os` เพื่อใช้คำสั่งที่เกี่ยวข้องกับระบบปฏิบัติการ เช่น ตรวจสอบว่ามีไฟล์หรือโฟลเดอร์อยู่หรือไม่ (`os.path.exists`), สร้างโฟลเดอร์ใหม่สำหรับจัดเก็บรายงาน (`os.makedirs`) และใช้ขั้นตอนการจัดการไฟล์ของระบบโดยรวม เช่น ในส่วนของการสร้างรายงานสรุป (`generate_report`)



```
1 import os
```

ภาพที่ 34 นำเข้าโมดูล

4.2.2 import datetime

ฟังก์ชันนี้ทำหน้าที่นำเข้าโมดูล `datetime` เพื่อจัดการข้อมูลด้านวันและเวลา เช่น การบันทึกวันที่สร้างหรืออัปเดต (`created_at`, `updated_at`) ของข้อมูลภาพยนตร์, รอบฉาย, และการบันทึกเวลาในไฟล์ Log (`MovieLog.dat`)



1 import datetime

ภาพที่ 35 นำเข้าข้อมูล datetime

4.2.3 ค่าคงที่ (Constants)

ส่วนนี้ทำหน้าที่เก็บค่าที่ใช้ซ้ำบ่อยทั่วทั้งระบบ เช่น

MOVIE_TXT : ชื่อไฟล์หลักที่เก็บข้อมูลภาพยนตร์

SCREEN_TXT : ชื่อไฟล์ที่เก็บข้อมูลรอบฉาย

LOG_TXT : ไฟล์บันทึกประวัติการทำรายการ

REPORT_DIR / REPORT_TXT : โฟลเดอร์และชื่อไฟล์สำหรับจัดเก็บรายงานสรุป

HALL_CAPACITY : ความจุสูงสุดของแต่ละโรงฉาย

GENRES : รายชื่อแนวภาพยนตร์ที่ระบบรองรับ

ค่าคงที่เหล่านี้ช่วยให้การอ้างอิงในทุกฟังก์ชันตรงกันและลดการเขียนซ้ำ



```

1  MOVIE_TXT = "MovieInfo.dat"
2  SCREEN_TXT = "ScreeningInfo.dat"
3  LOG_TXT = "MovieLog.dat"
4  REPORT_DIR = "report"
5  REPORT_TXT = "report/report.txt"
6
7  HALL_CAPACITY = {1: 200, 2: 180, 3: 150, 4: 100}
8
9  GENRES = {
10     "Action", "Comedy", "Horror", "Romance", "Drama", "Sci-Fi", "Thriller",
11     "Animation", "Adventure", "Crime", "Fantasy", "Mystery", "Family"
12 }
```

ภาพที่ 36 เก็บค่าที่ใช้ซ้ำบ่อย

4.2.4 def now()

ฟังก์ชันนี้ทำหน้าที่คืนค่า “วันและเวลาในปัจจุบัน” โดยจัดรูปแบบเป็น ISO มาตรฐาน (YYYY-MM-DD HH:MM:SS) เพื่อให้ระบบสามารถบันทึกเวลาที่เกิดการเปลี่ยนแปลงข้อมูลได้อย่างแม่นยำ



```
1 def now():
2     return datetime.datetime.now().isoformat(sep=" ", timespec="seconds")
```

ภาพที่ 37 ฟังก์ชันคืนค่า วันและเวลา

4.2.5 def now_ts()

ฟังก์ชันนี้ทำหน้าที่คืนค่าเวลาปัจจุบันในรูปแบบ Timestamp (จำนวนวินาทีตั้งแต่ปี 1970) ใช้สำหรับบันทึกใน Log (MovieLog.dat) เพื่อเรียงลำดับเหตุการณ์ตามเวลาได้อย่างถูกต้อง



```
1 def now_ts():
2     return int(datetime.datetime.now().timestamp())
```

ภาพที่ 38 ฟังก์ชันทำหน้าที่คืนค่าเวลา

4.2.6 def pause()

ฟังก์ชันนี้ทำหน้าที่หยุดการทำงานของโปรแกรมชั่วคราว โดยให้ผู้ใช้กด Enter เพื่อดำเนินการต่อ เหมาะสำหรับจุดที่มีการแสดงผลข้อมูลสำคัญ เช่น หลังจากเพิ่ม แก้ไข หรือลบข้อมูล เพื่อให้ผู้ใช้มีเวลาตรวจสอบข้อมูลก่อนเข้าสู่ขั้นตอนถัดไป




```
1 def pause():
2     input("\nกด Enter เพื่อดำเนินการต่อ...")
```

ภาพที่ 39 ฟังก์ชันหยุดการทำงานของโปรแกรม

4.2.7 def _parse_movie_line(s: str)

ฟังก์ชันนี้ทำหน้าที่อ่าน 1 บรรทัดจากไฟล์ข้อมูลภาพยนตร์ (MovieInfo.dat) และแปลงเป็นรูปแบบ dictionary เพื่อให้ระบบสามารถเข้าถึงข้อมูลแต่ละฟิลด์ได้ เช่น movie_id, status, duration, price, title, genre, rating, created_at, updated_at โดยมีการตรวจสอบกรณีข้อมูลไม่ครบหรือชื่อภาพยนตร์มีการใช้ชื่อกลาง _ แทนช่องว่าง



```


1 def _parse_movie_line(s: str):
2     parts = s.split()
3     if len(parts) < 8:
4         return None
5     try:
6         movie_id = int(parts[0])
7         duration = int(parts[2])
8         price = float(parts[3])
9     except:
10        return None
11    status = parts[1]
12    if len(parts) >= 9:
13        title = parts[4]
14        genre = parts[5]
15        rating = parts[6]
16        created_at = parts[7]
17        updated_at = parts[8]
18    else:
19        title_token = parts[4]
20        if "_" in title_token:
21            tokens = title_token.split("_")
22            if len(tokens) >= 2:
23                genre = tokens[-1]
24                title = "_".join(tokens[:-1])
25            else:
26                genre = "Unknown"
27                title = title_token
28        else:
29            genre = "Unknown"
30            title = title_token
31        rating = parts[5]
32        created_at = parts[6]
33        updated_at = parts[7]
34    return {
35        "movie_id": movie_id,
36        "status": status,
37        "duration": duration,
38        "price": price,
39        "title": title,
40        "genre": genre,
41        "rating": rating,
42        "created_at": created_at,
43        "updated_at": updated_at,
44    }

```

ภาพที่ 40 ฟังก์ชันอ่านไฟล์

4.2.8 def read_movies()

ฟังก์ชันนี้ทำหน้าที่อ่านข้อมูลทั้งหมดจากไฟล์ MovieInfo.dat โดยจะข้ามบรรทัดว่างและหัวตาราง จากนั้นใช้ _parse_movie_line() แปลงข้อมูลแต่ละบรรทัดเป็น dictionary และรวมเก็บไว้ในลิสต์ก่อนส่งกลับให้ส่วนอื่นของโปรแกรมใช้งาน เช่น ในการเพิ่ม อัปเดต หรือลบข้อมูลภาพยนตร์



```


1 def read_movies():
2     rows = []
3     with open(MOVIE_TXT, "r", encoding="utf-8") as f:
4         header = True
5         for line in f:
6             s = line.strip()
7             if not s: continue
8             if header:
9                 header = False
10                if s and s[0].isdigit():
11                    rec = _parse_movie_line(s)
12                    if rec: rows.append(rec)
13                    continue
14                rec = _parse_movie_line(s)
15                if rec: rows.append(rec)
16    return rows

```

ภาพที่ 41 ฟังก์ชันอ่านข้อมูลทั้งหมด

4.2.9 def _format_title_for_write(title: str)

ฟังก์ชันนี้ทำหน้าที่ปรับรูปแบบชื่อภาพยนตร์ก่อนเขียนกลับลงไฟล์ โดยแทนที่ช่องว่าง () ด้วยขีดล่าง (_) เพื่อป้องกันความผิดพลาดของการจัดคอลัมน์ในไฟล์



```

1 def _format_title_for_write(title: str):
2     return title.replace(" ", "_")

```

ภาพที่ 42 ฟังก์์รับชื่อภาพยนตร์

4.2.7 write_movies(rows)

ฟังก์ชันนี้ทำหน้าที่เขียนข้อมูลภาพยนตร์ทั้งหมดกลับลงไฟล์ MovieInfo.dat โดยจัดรูปแบบหัวตารางให้ครบ และใช้ `_format_title_for_write()` เพื่อรักษารูปแบบข้อมูลให้สม่ำเสมอในทุกครั้งที่มีการบันทึก

```

1 def write_movies(rows):
2     with open(MOVIE_TXT, "w", encoding="utf-8") as f:
3         f.write("movie_id status duration_min ticket_price title genre rating created_at updated_at\n")
4         for m in rows:
5             title_token = _format_title_for_write(m['title'])
6             f.write(f"{m['movie_id']} {m['status']} {m['duration']} {m['price']:.2f} {title_token} {m['genre']} {m['rating']} {m['created_at']} {m['updated_at']}\n")

```

ภาพที่ 43 ฟังก์ชันเขียนข้อมูลภาพยนตร์

4.2.10 def read_screens()

ฟังก์ชันนี้ทำหน้าที่อ่านข้อมูลจากไฟล์ ScreeningInfo.dat และแปลงเป็น dictionary สำหรับแต่ละรอบฉาย เช่น `screening_id`, `movie_id`, `hall`, `capacity`, `booked`, `status` โดยมีระบบตรวจสอบค่าที่ขาดหายและกำหนดสถานะเริ่มต้นเป็น “Active”

```

1 def read_screens():
2     rows = []
3     with open(SCREEN_TXT, "r", encoding="utf-8") as f:
4         header = True
5         for line in f:
6             s = line.strip()
7             if not s: continue
8             if header:
9                 header = False
10            if s and s[0].isdigit():
11                parts = s.split()
12                if len(parts) >= 5:
13                    rows.append({
14                        "screening_id": int(parts[0]),
15                        "movie_id": int(parts[1]),
16                        "hall": int(parts[2]),
17                        "capacity": int(parts[3]),
18                        "booked": int(parts[4]),
19                        "status": parts[5] if len(parts) >= 6 else "Active",
20                    })
21                continue
22            parts = s.split()
23            if len(parts) < 5: continue
24            rows.append({
25                "screening_id": int(parts[0]),
26                "movie_id": int(parts[1]),
27                "hall": int(parts[2]),
28                "capacity": int(parts[3]),
29                "booked": int(parts[4]),
30                "status": parts[5] if len(parts) >= 6 else "Active",
31            })
32    return rows

```

ภาพที่ 44 ฟังก์ชันอ่านข้อมูลไฟล์

4.2.11 def write_screens(rows)

ฟังก์ชันนี้ทำหน้าที่เขียนข้อมูลรอบฉายทั้งหมดกลับลงไฟล์ ScreeningInfo.dat พร้อมหัวตาราง และจัดรูปแบบข้อมูลให้ถูกต้อง เพื่อให้ข้อมูลล่าสุดถูกบันทึกอย่างสมบูรณ์ทุกครั้งที่มีการเพิ่มหรือแก้ไข



```

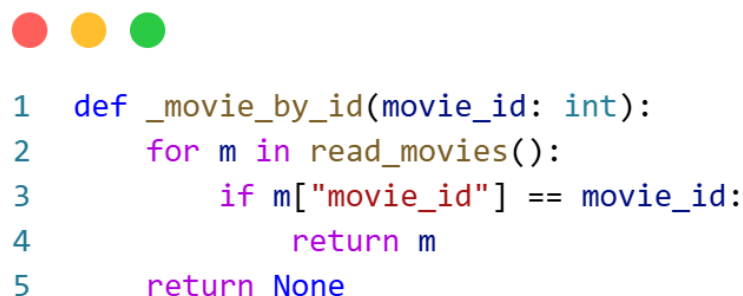
1 def write_screens(rows):
2     with open(SCREEN_TXT, "w", encoding="utf-8") as f:
3         f.write("screening_id movie_id hall capacity booked status\n")
4         for s in rows:
5             f.write(f"{s['screening_id']} {s['movie_id']} {s['hall']} {s['capacity']} {s['booked']} {s['status']}\n")
6
7 def _movie_by_id(movie_id: int):
8     for m in read_movies():
9         if m["movie_id"] == movie_id:
10             return m
11     return None
12
13 def _sum_booked_active(movie_id: int):
14     total = 0
15     for s in read_screens():
16         if s["movie_id"] == movie_id and s["status"] == "Active":
17             total += s["booked"]
18     return total

```

ภาพที่ 45 ฟังก์ชันเขียนข้อมูลรอบฉาย

4.2.12 def _movie_by_id(movie_id: int)

ฟังก์ชันนี้ทำหน้าที่ค้นหาภาพยนตร์จาก movie_id ที่ระบุ หากพบจะคืนค่าข้อมูลของภาพยนตร์เรื่องนั้นในรูปแบบ dictionary หากไม่พบจะคืนค่า None ใช้ในหลายส่วน เช่น ตรวจสอบก่อนเพิ่มรอบหรือก่อนอัปเดตข้อมูล



```

1 def _movie_by_id(movie_id: int):
2     for m in read_movies():
3         if m["movie_id"] == movie_id:
4             return m
5     return None

```

ภาพที่ 46 ฟังก์ชันค้นหาภาพยนตร์

4.2.13 def append_movie_log(op_code: int, movie_id: int)

ฟังก์ชันนี้ทำหน้าที่บันทึกเหตุการณ์ต่าง ๆ ลงในไฟล์ MovieLog.dat เช่น การเพิ่มภาพยนตร์ (op_code=1), การแก้ไข (op_code=2), หรือการลบ (op_code=3) โดยบันทึกข้อมูลประกอบ เช่น timestamp, สถานะ, ระยะเวลา, ราคาตัว และรายได้โดยประมาณหลังการดำเนินการ



```

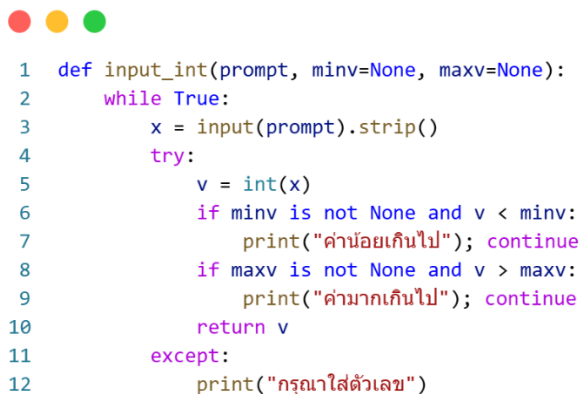
1 def append_movie_log(op_code: int, movie_id: int):
2     m = _movie_by_id(movie_id)
3     if not m:
4         return
5     status_bit = "1" if str(m.get("status", "")).lower().startswith("a") else "0"
6
7     booked_sum = 0
8     for s in read_screens():
9         if s["movie_id"] == movie_id and str(s.get("status", "")).lower().startswith("a"):
10             booked_sum += s.get("booked", 0)
11
12     revenue_after = float(m["price"]) * float(booked_sum)
13
14     with open(LOG_TXT, "a", encoding="utf-8") as f:
15         f.write(
16             f"{now_ts()} {op_code} {movie_id} {status_bit} "
17             f"{int(m['duration'])} {float(m['price']):.2f} {revenue_after:.2f}\n"
18         )

```

ภาพที่ 47 ฟังก์ชันบันทึกเหตุการณ์

4.2.14 def input_int(prompt, minv=None, maxv=None)

ฟังก์ชันนี้ทำหน้าที่รับค่าจำนวนเต็มจากผู้ใช้ โดยตรวจสอบให้แน่ใจว่าเป็นตัวเลขและอยู่ในช่วงค่าที่กำหนด ถ้าไม่ถูกต้องจะให้กรอกใหม่จนกว่าจะถูกต้อง เช่น ใช้ในการรับ Movie ID, Screening ID หรือ Hall



```


1 def input_int(prompt, minv=None, maxv=None):
2     while True:
3         x = input(prompt).strip()
4         try:
5             v = int(x)
6             if minv is not None and v < minv:
7                 print("ค่าน้อยเกินไป"); continue
8             if maxv is not None and v > maxv:
9                 print("ค่ามากเกินไป"); continue
10            return v
11        except:
12            print("กรุณาใส่ตัวเลข")

```

ภาพที่ 48 ฟังก์ชันรับค่าจำนวนเต็ม

4.2.15 def input_float(prompt, minv=None, maxv=None)

ฟังก์ชันนี้ทำหน้าที่รับค่าทศนิยมจากผู้ใช้ เช่น ราคาตั๋ว (Ticket Price) พร้อมตรวจสอบช่วงค่าขั้นต่ำและขั้นสูงเพื่อป้องกันข้อผิดพลาดของข้อมูล



```


1  def input_float(prompt, minv=None, maxv=None):
2      while True:
3          x = input(prompt).strip()
4          try:
5              v = float(x)
6              if minv is not None and v < minv:
7                  print("ค่าน้อยเกินไป"); continue
8              if maxv is not None and v > maxv:
9                  print("ค่ามากเกินไป"); continue
10             return v
11         except:
12             print("กรุณาใส่ตัวเลข")

```

ภาพที่ 49 ฟังก์ชันรับค่าทศนิยม

4.2.16 def input_str(prompt, allow_empty=False, maxlen=None)

ฟังก์ชันนี้ทำหน้าที่รับข้อความจากผู้ใช้ โดยสามารถกำหนดได้ว่าจะอนุญาตให้เว้นว่างหรือจำกัดความยาวสูงสุด เช่น ใช้ในการรับชื่อเรื่อง ประเภท หรือเรตติ้งของภาพยนตร์



```

1  def input_str(prompt, allow_empty=False, maxlen=None):
2      while True:
3          s = input(prompt).strip()
4          if not s and not allow_empty:
5              print("ห้ามเว้นว่าง"); continue
6          if maxlen and len(s) > maxlen:
7              print("ยาวเกินไป"); continue
8          return s

```

ภาพที่ 50 ฟังก์ชันรับค่าจากผู้ใช้

4.2.17 def get_capacity_for_hall(hall: int)

ฟังก์ชันนี้ทำหน้าที่คืนค่าความจุสูงสุดของโรงฉายจาก HALL_CAPACITY ตามหมายเลขโรงที่ระบุ เพื่อใช้ตรวจสอบจำนวนจอง (booked) ว่าไม่เกินขีดจำกัด



```
1 def get_capacity_for_hall(hall: int):
2     return HALL_CAPACITY.get(hall)
```

ภาพที่ 51 ฟังก์ชันคืนค่าความจุสูงสุด

4.2.18 def add_movie()

ฟังก์ชันนี้ทำหน้าที่เพิ่มภาพยนตร์ใหม่เข้าสู่ระบบ โดยรับข้อมูลจากผู้ใช้ เช่น Movie ID, Title, Genre, Rating, Duration และ Price จากนั้นตั้งค่าสถานะเป็น “Active” และบันทึกวันที่สร้าง ก่อนเขียนข้อมูลลงไฟล์ MovieInfo.dat พร้อมบันทึก Log จากนั้นระบบจะบังคับให้เพิ่มอย่างน้อย 1 รอบฉายของภาพยนตร์นั้น โดยระบุ Screening ID, Hall, และจำนวนจอง (booked) ก่อนบันทึกลง ScreeningInfo.dat และบันทึก Log ซ้ำอีกครั้ง สุดท้าย ระบบจะแสดงผลสรุปของภาพยนตร์เรื่องที่เพิ่มเสร็จสิ้น เพื่อยืนยันความถูกต้องของข้อมูล

```
1 def add_movie():
2     view_all()
3     movies = read_movies()
4     screens = read_screens()
5
6     movie_id = input_int("Movie ID (ใหม่): ", 1)
7     if any(m["movie_id"] == movie_id for m in movies):
8         print("Movie ID ซ้ำ")
9         return
10
11     title = input_str("Title: ", 100)
12     genre = input_str("Genre: ", 30)
13     rating = input_str("Rating: ", 10)
14     dur = input_int("Duration (mins): ", 1)
15     price = input_float("Ticket Price: ", 0.0)
16
17     today = datetime.date.today().isoformat()
18     movies.append({
19         "movie_id": movie_id,
20         "title": title,
21         "genre": genre,
22         "rating": rating,
23         "duration": dur,
24         "price": price,
25         "status": "Active",
26         "created_at": today,
27         "updated_at": today
28     })
29     write_movies(movies)
30     append_movie_log(1, movie_id)
```

ภาพที่ 52 ฟังก์ชันเพิ่มภาพยนตร์

```

31
32 print(f"\nเพิ่ม Movie {movie_id} : {title} (Active)\n")
33 view_all()
34 while True:
35     print("กรุณาเพิ่มรอบฉายสำหรับหนังอย่างน้อย 1 รอบ:")
36     screening_id = input_int("Screening ID: ", 1)
37     if any(s["screening_id"] == screening_id for s in screens):
38         print("Screening ID ซ้ำ กรุณาใส่ใหม่")
39         continue
40     hall = input_int("Hall (1-4): ", 1, 4)
41     cap = HALL_CAPACITY.get(hall, 0)
42     booked = input_int(f"Booked (0-{cap}): ", 0, cap)
43
44     screens.append({
45         "screening_id": screening_id,
46         "movie_id": movie_id,
47         "hall": hall,
48         "capacity": cap,
49         "booked": booked,
50         "status": "Active"
51     })
52     write_screens(screens)
53     append_movie_log(1, movie_id)
54
55     print(f"เพิ่ม Screening {screening_id} ของหนัง {title} เสร็จเรียบร้อย\n")
56     break
57
58 _print_view_by_movie_id(movie_id)
59 pause()

```

ภาพที่ 53 ต่อจากภาพที่ 52

4.2.19 def add_screening()

ฟังก์ชันนี้ทำหน้าที่เพิ่มรอบฉายใหม่ให้กับภาพยนตร์ที่มีอยู่ในระบบ โดยตรวจสอบว่า Movie ID ที่กรอกมีอยู่จริง และ Screening ID ไม่ซ้ำ จากนั้นระบุโรง (hall) เพื่อดึงค่าความจุอัตโนมัติ และรับค่าจอง (booked) ก่อนบันทึกข้อมูลลงไฟล์ ScreeningInfo.dat พร้อมบันทึก Log เพื่อเก็บประวัติการเพิ่มข้อมูล

```

1 def add_screening():
2     view_all()
3     movies = read_movies()
4     if not movies:
5         return
6     screens = read_screens()
7     movie_id = input_int("กรอก Movie ID ที่ต้องการหุ้กับรอบฉาย: ", 1)
8     if not any(m["movie_id"] == movie_id for m in movies):
9         print("ไม่พบ Movie ID"); return
10    screening_id = input_int("Screening ID: ", 1)
11    if any(s["screening_id"] == screening_id for s in screens):
12        print("มี Screening ID นี้แล้ว"); return
13    while True:
14        hall = input_int("Hall (1-4): ", 1)
15        cap = get_capacity_for_hall(hall)
16        if cap is None:
17            print("Hall นี้ยังไม่ถูกกำหนดความจุ"); continue
18        break
19    booked = input_int(f"Booked (0-{cap}): ", 0, cap)
20    screens.append({
21        "screening_id": screening_id,
22        "movie_id": movie_id,
23        "hall": hall,
24        "capacity": cap,
25        "booked": booked,
26        "status": "Active",
27    })
28    write_screens(screens)
29    append_movie_log(1, movie_id)
30    print("เพิ่มรอบฉายแล้ว\n")
31    view_all()
32    pause()

```

ภาพที่ 54 ฟังก์ชันเพิ่มรอบฉาย

4.2.20 def _movie_title(movie_id: int)

ฟังก์ชันนี้ทำหน้าที่คืนค่าชื่อภาพยนตร์จาก movie_id ที่ระบุ หากไม่พบจะคืนข้อความว่า “ไม่พบชื่อ” ใช้เพื่อแสดงชื่อในส่วนที่เกี่ยวกับรอบฉายหรือการแก้ไขข้อมูล



```
1 def _movie_title(movie_id: int):
2     m = _movie_by_id(movie_id)
3     return m["title"] if m else "(ไม่พบชื่อ)"
```

ภาพที่ 55 ฟังก์ชันคืนค่าภาพยนตร์

4.2.21 def _print_view_by_movie_id(mid: int)

ฟังก์ชันนี้ทำหน้าที่แสดงข้อมูลของภาพยนตร์แต่ละเรื่องพร้อมรอบฉายทั้งหมด โดยใช้ _join_for_view_left() เพื่อรวมข้อมูลจากตารางภาพยนตร์และรอบฉาย จากนั้นกรองเฉพาะรอบที่มีสถานะ “Active” และแสดงผลในรูปแบบตารางที่มีหัวข้อชัดเจน



```
1 def _print_view_by_movie_id(mid: int):
2     pairs = [(s, m) for (s, m) in _join_for_view_left() if m["movie_id"] == mid]
3     pairs = [p for p in pairs if (p[0] is None) or (p[0]["status"] == "Active")]
4     if not pairs:
5         print("ไม่พบข้อมูล")
6         return
7     print(_header())
8     for s, m in pairs:
9         print(_render_row(s, m))
10    print("-----")
```

ภาพที่ 56 ฟังก์ชันแสดงข้อมูลภาพยนตร์

4.2.22 def update_movie()

ฟังก์ชันนี้ทำหน้าที่แก้ไขข้อมูลภาพยนตร์ โดยแสดงข้อมูลเดิมก่อนให้ผู้ใช้กรอกข้อมูลใหม่เฉพาะฟิลด์ที่ต้องการเปลี่ยน (หากไม่กรอกจะใช้ข้อมูลเดิม) จากนั้นตรวจสอบความถูกต้องของข้อมูล เช่น ประเภทข้อมูล ตัวเลข และช่วงค่า ก่อนบันทึกกลับลงไฟล์ MovieInfo.dat อัปเดตวันแก้ไข (updated_at) และบันทึก Log การแก้ไข

```

1 def update_movie():
2     print("\nข้อมูลปัจจุบัน:")
3     view_all()
4     print("")
5
6     movies = read_movies()
7     movie_id = input_int("Movie ID ที่ต้องการแก้ไข: ", 1)
8     idx = next((i for i, mm in enumerate(movies) if mm["movie_id"] == movie_id), None)
9     if idx is None:
10         print("ไม่พบ Movie นี้")
11         return
12
13     m = movies[idx]
14     print(f"กำลังแก้ไข Movie {m['movie_id']} : {m['title']} กด 'Enter' หากไม่ต้องการแก้ไข")
15
16     t = input_str(f"Title [{m['title']}]: ", allow_empty=True)
17     g = input_str(f"Genre [{m['genre']}]: ", allow_empty=True)
18     r = input_str(f"Rating [{m['rating']}]: ", allow_empty=True)
19     d = input_str(f"Duration(mins) [{m['duration']}]: ", allow_empty=True)
20     p = input_str(f"Ticket price [{m['price']:.2f}]: ", allow_empty=True)
21     s = input_str(f"Status (Active/Deleted) [{m['status']}]: ", allow_empty=True)
22
23     if t: m["title"] = t
24     if g: m["genre"] = g
25     if r: m["rating"] = r
26     if d:
27         try:
28             dv = int(d)
29             if dv > 0:
30                 m["duration"] = dv
31         except:
32             print("ห้ามการแก้ไข Duration (รูปแบบไม่ถูกต้อง)")
33     if p:
34         try:
35             pv = float(p)
36             if pv >= 0:
37                 m["price"] = pv
38         except:
39             print("ห้ามการแก้ไข Price (รูปแบบไม่ถูกต้อง)")
40     if s:
41         m["status"] = "Active" if s.lower().startswith("a") else "Deleted"
42
43     m["updated_at"] = datetime.date.today().isoformat()
44     movies[idx] = m
45     write_movies(movies)
46     append_movie_log(2, movie_id)
47
48     print(f"\nอัปเดตเรียบร้อยแล้ว: Movie {m['movie_id']} : {m['title']}\n")
49     _print_view_by_movie_id(movie_id)
50     pause()

```

ภาพที่ 57 ฟังก์ชันแก้ไขข้อมูลภาพยนตร์

4.2.23 def update_screening()

ฟังก์ชันนี้ทำหน้าที่แก้ไขข้อมูลรอบฉาย เช่น เปลี่ยนโรง (hall), ปรับจำนวนที่นั่ง (capacity), จำนวนจอง (booked) หรือสถานะ (Active/Deleted) โดยระบบจะตรวจสอบค่าที่ผู้ใช้ป้อนว่าถูกต้องตามรูปแบบ หากถูกต้องจะเขียนกลับลงไฟล์และบันทึก Log เพื่อเก็บประวัติการแก้ไข

```

1 def update_screening():
2     print("\nข้อมูลปัจจุบัน:")
3     view_all()
4     print("")
5
6     screens = read_screens()
7     scr_id = input_int("Screening ID ที่ต้องการแก้ไข: ", 1)
8     idx = next((i for i, ss in enumerate(screens) if ss["screening_id"] == scr_id), None)
9     if idx is None:
10         print("ไม่พบ Screening นี้")
11         return
12
13     s = screens[idx]
14     title = _movie_title(s["movie_id"])
15     print(f"กำลังแก้ไข Screening {s['screening_id']} ของหนัง {title} (Movie {s['movie_id']})")
16
17     h = input_str(f"Hall [{s['hall']}] : ", allow_empty=True)
18     b = input_str(f"Booked [{s['booked']}] : ", allow_empty=True)
19     st = input_str(f"Status (Active/Deleted) [{s['status']}] : ", allow_empty=True)
20
21     if h:
22         try:
23             hv = int(h)
24             if hv > 0:
25                 cap = HALL_CAPACITY.get(hv)
26                 if cap is None:
27                     print("Hall นี้ยังไม่ถูกกำหนดความจุ - ข้ามการแก้ Hall")
28                 else:
29                     s["hall"] = hv
30                     s["capacity"] = cap
31                     if s["booked"] > cap:
32                         s["booked"] = cap
33             except:
34                 print("ข้ามการแก้ Hall (รูปแบบไม่ถูกต้อง)")
35
36     if b:
37         try:
38             bv = int(b)
39             if 0 <= bv <= s["capacity"]:
40                 s["booked"] = bv
41             else:
42                 print(f"ค่าจะต้องอยู่ระหว่าง 0..{s['capacity']} - ข้ามการแก้ Booked")
43             except:
44                 print("ข้ามการแก้ Booked (รูปแบบไม่ถูกต้อง)")
45
46     if st:
47         s["status"] = "Active" if st.lower().startswith("a") else "Deleted"
48
49     screens[idx] = s
50     write_screens(screens)
51     append_movie_log(2, s["movie_id"])
52
53     print(f"\nอัปเดตเรียบร้อยแล้ว: Screening {s['screening_id']} ของหนัง {title}\n")
54     _print_view_by_movie_id(s["movie_id"])
55     pause()

```

ภาพที่ 58 ฟังก์ชันแก้ไขรอบฉาย

4.2.24 def delete_menu()

ฟังก์ชันนี้ทำหน้าที่ลบข้อมูลภาพยนตร์ 2 รูปแบบ

1. Soft Delete – เปลี่ยนสถานะของภาพยนตร์เป็น “Deleted” โดยไม่ลบข้อมูลออกจากไฟล์
2. Hard Delete – ลบข้อมูลภาพยนตร์ออกจากไฟล์ถาวร รวมถึงลบรอบฉายและข้อมูล Log ที่เกี่ยวข้องทั้งหมด

การลบทั้งสองแบบจะอัปเดตวัน (updated_at) และบันทึก Log การดำเนินการไว้ทุกครั้ง

```


1 def delete_menu():
2     print("-----")
3     print("1) ลบ Movie (ตั้งสถานะเป็น Deleted)")
4     print("2) ลบ Movie ออกจากข้อมูลจริง (Hard delete)")
5     print("0) Back")
6     print("-----")
7     ch = input_int("เลือกเมนู: ", 0, 3)
8     if ch == 0:
9         return
10
11     print("\nข้อมูลปัจจุบัน:")
12     view_all()
13     print("")
14
15     if ch == 1:
16         movies = read_movies()
17         movie_id = input_int("Movie ID ที่ต้องการลบ (soft delete): ", 1)
18         idx = next((i for i,m in enumerate(movies) if m["movie_id"]==movie_id), None)
19         if idx is None:
20             print("ไม่พบ Movie ID"); return
21         title = movies[idx]["title"]
22         print(f"จะตั้งสถานะ Deleted ให้ Movie {movie_id} : {title}")
23         movies[idx]["status"]="Deleted"
24         movies[idx]["updated_at"]=datetime.date.today().isoformat()
25         write_movies(movies)
26         append_movie_log(3, movie_id)
27         print(f"ตั้งสถานะ Deleted แล้ว: Movie {movie_id} : {title}")
28
29
30     elif ch == 2:
31         movies = read_movies()
32         screens_all = read_screens()
33         movie_id = input_int("Movie ID ที่ต้องการลบออกจากข้อมูลจริง: ", 1)
34         m = next((mm for mm in movies if mm["movie_id"] == movie_id), None)
35         if not m:
36             print("ไม่พบ Movie ID"); return
37         title = m["title"]
38         confirm = input_str(f"ยืนยันลบ Movie {movie_id} : {title} ออกจากทุกไฟล์? (Y/N): ").lower()
39         if confirm not in ("y","yes"):
40             print("ยกเลิกการลบ")
41             return
42
43         movies = [mm for mm in movies if mm["movie_id"] != movie_id]
44         write_movies(movies)
45
46         screens = [s for s in screens_all if s["movie_id"] != movie_id]
47         write_screens(screens)
48
49         if os.path.exists(LOG_TXT):
50             with open(LOG_TXT, "r", encoding="utf-8") as f:
51                 lines = f.readlines()
52             with open(LOG_TXT, "w", encoding="utf-8") as f:
53                 for line in lines:
54                     parts = line.strip().split()
55                     if not parts: continue
56                     if len(parts) >= 3 and parts[2].isdigit() and int(parts[2]) == movie_id:
57                         continue
58                     f.write(line)
59
60     print(f"ลบเรียบร้อยแล้ว: Movie {movie_id} : {title} ออกจากทุกไฟล์แล้ว")
61     pause()
62

```

ภาพที่ 59 ฟังก์ชันลบข้อมูลภาพยนตร์

4.2.25 def _header()

ฟังก์ชันนี้ทำหน้าที่คืนข้อความหัวตารางมาตรฐานที่ใช้แสดงผลในหน้าข้อมูลหรือรายงาน โดยมีคอลัมน์ เช่น ScreeningID, MovieID, Title, Genre, Rating, Hall, Duration, Capacity, Booked, Ticket Price, Revenue, และ Status เพื่อให้การแสดงผลเป็นรูปแบบเดียวกันทั่วระบบ



```


1 def _header():
2     return (
3         "-----\n"
4         "| ScreeningID | MovieID | Title                | Genre | Rating | Hall | duration(min) | Capacity | Booked | Ticket Price |Revenue(THB/day)|Status | \n"
5         "-----\n"
6     )

```

ภาพที่ 60 ฟังก์ชันคืนค่าหัวตาราง

4.2.26 def _fmt_money(v)

ฟังก์ชันนี้ทำหน้าที่จัดรูปแบบจำนวนเงินให้เป็นเลขทศนิยม 2 ตำแหน่ง และคืนหลักพัน เพื่อให้มองเห็นมูลค่าชัดเจนดูรายงานหรือสถิติ



```


1 def _fmt_money(v):
2     return f"{v:,.2f}"

```

ภาพที่ 61 ฟังก์ชันจัดรูปแบบจำนวนเงิน

4.2.27 def _join_for_view_left()

ฟังก์ชันนี้ทำหน้าที่รวมข้อมูลภาพยนตร์ (จาก read_movies()) เข้ากับข้อมูลรอบฉาย (จาก read_screens()) แบบ Left Join เพื่อให้หนังทุกเรื่องปรากฏในผลลัพธ์แม้จะยังไม่มีรอบฉาย โดยคืนค่าเป็นคู่ (screen, movie) ใช้เป็นพื้นฐานสำหรับฟังก์ชันการดูข้อมูล สรุป และสร้างรายงานสถิติในส่วนถัดไป



```

1 def _join_for_view_left():
2     movies = read_movies()
3     screens = read_screens()
4     by_movie = {}
5     for s in screens:
6         by_movie.setdefault(s["movie_id"], []).append(s)
7     out = []
8     for m in movies:
9         lst = by_movie.get(m["movie_id"])
10        if not lst:
11            out.append((None, m))
12        else:
13            for s in lst:
14                out.append((s, m))
15    return out

```

ภาพที่ 62 ฟังก์ชันรวมข้อมูลภาพยนตร์

4.2.28 def _render_row(s, m)

ฟังก์ชันนี้ทำหน้าที่ “จัดรูปแบบหนึ่งแถว” ของตารางแสดงผลจากคู่ข้อมูล (s, m)

กรณีไม่มีรอบ (s is None) จะแสดง - และค่า 0 ในคอลัมน์ของฝั่งรอบ

กรณีมีรอบ จะเติม screening_id, hall, capacity, booked และคำนวณ revenue =

price * booked

ผลลัพธ์เป็นสตริงหนึ่งบรรทัดพร้อมจัดระยะตัวอักษรให้สวยงามโดยคง Title ไม่เกิน 22 ตัวอักษร



```

1 def _render_row(s, m):
2     title = m["title"].replace("_", " ")
3     if len(title) > 22: title = title[:22]
4     if s is None:
5         screening_id = "-"
6         hall = "-"
7         capacity = 0
8         booked = 0
9     else:
10        screening_id = s["screening_id"]
11        hall = s["hall"]
12        capacity = s["capacity"]
13        booked = s["booked"]
14    revenue = (m["price"] * booked) if s is not None else 0.0
15    return (
16        f"| {str(screening_id).ljust(11)} | "
17        f"| {str(m['movie_id']).ljust(7)} | "
18        f"| {title.ljust(22)} | "
19        f"| {m['genre'][:8].ljust(8)} | "
20        f"| {m['rating'][:4].ljust(6)} | "
21        f"| {str(hall).ljust(4)} | "
22        f"| {str(m['duration']).ljust(16)} | "
23        f"| {str(capacity).ljust(8)} | "
24        f"| {str(booked).ljust(6)} | "
25        f"| {_fmt_money(m['price']).rjust(11)} | "
26        f"| {_fmt_money(revenue).rjust(14)} | "
27        f"| {m['status'][:8].ljust(8)} | "
28    )

```

ภาพที่ 63 ฟังก์ชันจัดรูปแบบ

4.2.29 def view_one()

ฟังก์ชันนี้ทำหน้าที่ “ดูรายการเฉพาะเรื่อง” โดยให้ผู้ระบุ movie_id แล้วดึงข้อมูล (s, m) เฉพาะเรื่องนั้นจาก _join_for_view_left() จากนั้นพิมพ์หัวตารางและแต่ละแถวด้วย _render_row() โดยข้ามรอบที่ status != "Active" เพื่อให้เห็นภาพรวมของเรื่องนั้นในสถานะใช้งานจริง

```

1 def view_one():
2     mid = input_int("man Movie ID: ", 1)
3     pairs = [(s, m) for (s, m) in _join_for_view_left() if m["movie_id"] == mid]
4     if not pairs:
5         print("ไม่พบข้อมูล"); return
6     print(_header())
7     for s, m in pairs:
8         if s is not None and s["status"] != "Active":
9             continue
10        print(_render_row(s, m))
11    print("-----")
12    pause()

```

ภาพที่ 64 ฟังก์ชันดูรายการเฉพาะเรื่อง

4.2.30 def view_all()

ฟังก์ชันนี้ทำหน้าที่ “ดูรายการทั้งหมด” โดยดึงทุกคู่ข้อมูลจาก _join_for_view_left() แล้วกรองให้แสดงเฉพาะ

ภาพยนตร์ที่ไม่มีรอบ (คู่ (None, m)) เพื่อให้รู้ว่าไม่มีเรื่องใดยังไม่จัดรอบ

หรือคู่ที่ฝั่งรอบ status == "Active"

จากนั้นพิมพ์หัวตารางและทุกรายการด้วย _render_row() เป็นมุมมองสรุปทั้งระบบ

```

1 def view_all():
2     pairs = _join_for_view_left()
3     pairs = [p for p in pairs if (p[0] is None) or (p[0]["status"] == "Active")]
4     if not pairs:
5         print("ไม่พบข้อมูล"); return
6     print(_header())
7     for s, m in pairs:
8         print(_render_row(s, m))
9     print("-----")
10    pause()

```

ภาพที่ 65 ฟังก์ชันดูรายการทั้งหมด

4.2.31 def filter_view()

ฟังก์ชันนี้ทำหน้าที่ “กรองการแสดงผล” ตามเงื่อนไขที่ผู้ใช้เลือก ได้แก่

1. สถานะ (Active/Deleted ของฝั่งภาพยนตร์)
2. ประเภท (Genre)

3. เรตติ้ง (Rating)
4. ราคา (Ticket Price \leq ค่าเพดานที่กำหนด)

โดยทุกกรณีเริ่มจากชุดข้อมูลพื้นฐานที่ได้จาก `_join_for_view_left()` ซึ่งถูกกรองให้เหลือเฉพาะคู่ที่รอบฉาย Active หรือไม่มีรอบ แล้วจึงกรองขั้นที่ 2 ตามเงื่อนไขที่ผู้ใช้เลือก ก่อนพิมพ์ผลลัพธ์ด้วย `_render_row()`

```

1 def filter_view():
2     print("ตัวเลือก: 1) Status 2) Genre 3) Rating 4) Ticket price (<=)")
3     from_choice = input_int("เลือก: ", 1, 4)
4     pairs = _join_for_view_left()
5     pairs = [p for p in pairs if (p[0] is None) or (p[0]["status"] == "Active")]
6     if from_choice == 1:
7         want = "Active" if input_str("รอบฉาย Active/Deleted: ").lower().startswith("a") else "Deleted"
8         pairs = [(s, m) for (s, m) in pairs if m["status"] == want]
9     elif from_choice == 2:
10        g = input_str("รอบฉาย Genre: ").lower()
11        pairs = [(s, m) for (s, m) in pairs if m["genre"].lower() == g]
12    elif from_choice == 3:
13        r = input_str("รอบฉาย Rating: ").lower()
14        pairs = [(s, m) for (s, m) in pairs if m["rating"].lower() == r]
15    else:
16        pmax = input_float("ราคาสูงสุด: ", 0)
17        pairs = [(s, m) for (s, m) in pairs if m["price"] <= pmax]
18    if not pairs:
19        print("ไม่พบข้อมูล"); return
20    print(_header())
21    for s, m in pairs:
22        print(_render_row(s, m))
23    print("-----")
24    pause()

```

ภาพที่ 66 ฟังก์ชันการกรองการแสดงผล

4.2.32 def summary_stats()

ฟังก์ชันนี้ทำหน้าที่ “สรุปสถิติภาพรวม” ของระบบจากชุดข้อมูล left join โดยโฟกัสเฉพาะรอบฉายที่ Active ได้แก่

จำนวนรอบฉายทั้งหมดที่ Active

ผลรวมความจุ (Capacity) และยอดจอง (Booked)

รายได้รวมโดยประมาณ (Σ price * booked)

ค่า Avg Fill Rate = (Booked / Capacity)*100 เมื่อ Capacity > 0

สถิติราคาตัวของภาพยนตร์ที่ Active (Min/Max/Avg)

การนับจำนวนเรื่องตามประเภท (Movies by Genre)

ผลลัพธ์พิมพ์เป็นบล็อกสรุปอ่านง่าย ใช้ประกอบการตัดสินใจจัดรอบ/ตั้งราคา



```

1 def summary_stats():
2     pairs = _join_for_view_left()
3     active_screens = [(s, m) for (s, m) in pairs if (s is not None and s["status"] == "Active")]
4     total_scr = len(active_screens)
5     total_cap = sum(s["capacity"] for s, _ in active_screens)
6     total_booked = sum(s["booked"] for s, _ in active_screens)
7     total_rev = sum(m["price"] * s["booked"] for s, m in active_screens)
8     fill_rate = (total_booked / total_cap * 100.0) if total_cap > 0 else 0.0
9     movies = read_movies()
10    active_movies = [m for m in movies if m["status"] == "Active"]
11    prices = [m["price"] for m in active_movies]
12    pmin = min(prices) if prices else 0.0
13    pmax = max(prices) if prices else 0.0
14    pavg = (sum(prices) / len(prices)) if prices else 0.0
15    gcount = {}
16    for m in active_movies:
17        gcount[m["genre"]] = gcount.get(m["genre"], 0) + 1
18    print("Summary")
19    print(f"- Total Screenings : {total_scr}")
20    print(f"- Total Capacity : {total_cap}")
21    print(f"- Total Booked : {total_booked}")
22    print(f"- Total Revenue : {_fmt_money(total_rev)} THB")
23    print(f"- Avg Fill Rate : {fill_rate:.2f} %")
24    print("")
25    print("Ticket Price Statistics (Active only)")
26    print(f"- Min : {pmin:.2f}")
27    print(f"- Max : {pmax:.2f}")
28    print(f"- Avg : {pavg:.2f}")
29    print("")
30    print("Movies by Genre (Active only)")
31    for g, c in gcount.items():
32        print(f"- {g} : {c} เรื่อง")
33    pause()

```

ภาพที่ 67 ฟังก์ชันสรุปสถิติรวม

4.2.33 def generate_report()

ฟังก์ชันนี้ทำหน้าที่ “สร้างไฟล์รายงานแบบข้อความ” (TXT) ที่โฟลเดอร์ report/report.txt โดยสรุปสาระสำคัญของระบบ ณ ขณะเรียกใช้ ประกอบด้วย

1. ส่วน Summary: จำนวนเรื่อง Active/Deleted, จำนวนรอบ Active, Total Capacity, Total Booked, Total Revenue, Avg Fill Rate
2. ส่วน Ticket Price Statistics (เฉพาะเรื่อง Active): Min/Max/Avg
3. ส่วน Movies by Genre (เฉพาะเรื่อง Active): นับจำนวนเรื่องต่อประเภท
4. ส่วน Table A – Active screenings: ถ้ามีรอบ Active จะพิมพ์หัวตาราง (_header()) และทุกรายการด้วย _render_row() ลงไฟล์ (หากไม่มีจะระบุ <empty>\n)

ก่อนบันทึกงานจะ `os.makedirs(REPORT_DIR, exist_ok=True)` เพื่อให้แน่ใจว่าโฟลเดอร์มีอยู่แล้ว
จึงปลอดภัยต่อการเขียนไฟล์รายงานทุกครั้งที่สั่ง “Generate Report” จากเมนูหลัก

```

1 def generate_report():
2     os.makedirs(REPORT_DIR, exist_ok=True)
3     pairs_left = _join_for_view_left()
4     movies = read_movies()
5     active_movies = [m for m in movies if m["status"] == "Active"]
6     active_screens = [(s, m) for (s, m) in pairs_left if (s is not None and s["status"] == "Active")]
7     movies_no_screen = [m for (s, m) in pairs_left if s is None]
8     total_scr = len(active_screens)
9     total_cap = sum(s["capacity"] for s, _ in active_screens)
10    total_booked = sum(s["booked"] for s, _ in active_screens)
11    total_rev = sum(m["price"] * s["booked"] for s, m in active_screens)
12    fill_rate = (total_booked / total_cap * 100.0) if total_cap > 0 else 0.0
13    prices = [m["price"] for m in active_movies]
14    pmin = min(prices) if prices else 0.0
15    pmax = max(prices) if prices else 0.0
16    pavg = (sum(prices) / len(prices)) if prices else 0.0
17    gcount = {}
18    for m in active_movies:
19        gcount[m["genre"]] = gcount.get(m["genre"], 0) + 1
20    with open(REPORT_TXT, "w", encoding="utf-8") as r:
21        r.write("Cinema Backend Report\n")
22        r.write(f"Generated at: {now()}\n\n")
23        r.write("Summary\n")
24        r.write(f"- Total Movies      : {len(movies)}\n")
25        r.write(f"- Active Movies      : {sum(1 for m in movies if m['status']=='Active')}\n")
26        r.write(f"- Deleted Movies     : {sum(1 for m in movies if m['status']=='Deleted')}\n")
27        r.write(f"- Total Screenings   : {total_scr}\n")
28        r.write(f"- Total Capacity     : {total_cap}\n")
29        r.write(f"- Total Booked       : {total_booked}\n")
30        r.write(f"- Total Revenue      : {fmt_money(total_rev)} THB\n")
31        r.write(f"- Avg Fill Rate      : {fill_rate:.2f} %\n")
32        r.write("Ticket Price Statistics (Active movies)\n")
33        r.write(f"- Min : {pmin:.2f}\n")
34        r.write(f"- Max : {pmax:.2f}\n")
35        r.write(f"- Avg : {pavg:.2f}\n")
36        r.write("Movies by Genre (Active movies)\n")
37        if not gcount:
38            r.write("- (none)\n\n")
39        else:
40            for g, c in gcount.items():
41                r.write(f"- {g} : {c} เรื่อง\n")
42            r.write("\n")
43            r.write("Table A - Active screenings\n")
44            r.write(_header() + "\n")
45            if not active_screens:
46                r.write("(empty)\n")
47            else:
48                for s, m in active_screens:
49                    r.write(_render_row(s, m) + "\n")
50            r.write("-----\n")
51    print(f"สร้างไฟล์ {REPORT_TXT} แล้ว")
52    pause()

```

ภาพที่ 68 ฟังก์ชันสร้างไฟล์รายงานข้อความ

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงการ “ระบบจัดการโรงภาพยนตร์” ถูกพัฒนาขึ้นเพื่ออำนวยความสะดวกแก่ผู้ดูแลระบบในการจัดการข้อมูลภาพยนตร์ การเพิ่ม ลบ แก้ไข และดูข้อมูลได้อย่างเป็นระบบ รวมถึงการสร้างรายงานสรุปผลในรูปแบบไฟล์ .txt เพื่อใช้ในการวิเคราะห์ข้อมูลภายหลัง

ระบบนี้ถูกออกแบบให้ทำงานผ่าน Command Line Interface (CLI) โดยใช้ภาษา Python ในการพัฒนา และจัดเก็บข้อมูลด้วยไฟล์แบบ Binary Data (.dat) เพื่อให้สามารถเข้าถึงและจัดการข้อมูลได้อย่างรวดเร็วและแม่นยำ

ผลการดำเนินงานโดยรวม ระบบสามารถทำงานได้ถูกต้องตามวัตถุประสงค์ เช่น การจัดเก็บข้อมูลภาพยนตร์ การบันทึกประวัติการแก้ไข การคำนวณรายได้ รวมถึงการแสดงผลสถิติภาพรวมของโรงภาพยนตร์ได้ครบถ้วน

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

ในการพัฒนาระบบจัดการโรงภาพยนตร์ พบปัญหาหลักคือข้อจำกัดของการจัดการข้อมูลผ่านไฟล์ไบนารี ซึ่งทำให้การปรับแก้โครงสร้างข้อมูลหรือเพิ่มฟังก์ชันใหม่ทำได้ยากกว่าเมื่อใช้ฐานข้อมูลแบบสัมพันธ์ (Relational Database)

อีกทั้งในบางกรณีอาจเกิดความคลาดเคลื่อนในการคำนวณรายได้หรือจำนวนรอบฉาย หากข้อมูลที่บันทึกไว้ไม่ครบถ้วน นอกจากนี้การแสดงผลผ่าน CLI อาจไม่สะดวกต่อผู้ใช้ที่ไม่คุ้นเคยกับระบบบรรทัดคำสั่ง อย่างไรก็ตาม ปัญหาเหล่านี้สามารถแก้ไขได้ด้วยการวางแผนปรับโครงสร้างระบบให้ยืดหยุ่นมากขึ้นในอนาคต

5.3 ข้อเสนอแนะ

เพื่อพัฒนาระบบให้สมบูรณ์และเหมาะสมต่อการใช้งานจริงในอนาคต มีข้อเสนอแนะดังนี้

5.3.1 ควรปรับระบบจัดเก็บข้อมูลให้รองรับฐานข้อมูลแบบสัมพันธ์ (Relational Database)

เช่น MySQL หรือ SQLite เพื่อความปลอดภัยและความยืดหยุ่นในการจัดเก็บข้อมูล

5.3.2 เพิ่มระบบการยืนยันตัวตน (Authentication) เพื่อจำกัดสิทธิ์การเข้าถึงข้อมูลของผู้ดูแลแต่ละระดับ

5.3.3 ปรับปรุงการแสดงผลให้อยู่ในรูปแบบกราฟิก (GUI) เพื่อความสะดวกในการใช้งานและเข้าใจข้อมูลได้ง่ายขึ้น

5.3.4 พัฒนาฟังก์ชันการวิเคราะห์ข้อมูลเชิงลึก เช่น การเปรียบเทียบรายได้รายเดือน หรือสถิติการเข้าชมของภาพยนตร์แต่ละเรื่อง เพื่อสนับสนุนการตัดสินใจทางธุรกิจ

5.4 สิ่งที่ได้ผู้จัดทำได้รับจากการพัฒนาโครงงาน

จากการพัฒนาโครงงาน ผู้จัดทำได้รับความรู้และประสบการณ์จริงในการออกแบบและพัฒนา ระบบด้วยภาษา Python โดยเฉพาะการจัดการไฟล์แบบ Binary, การใช้โมดูล struct, การแสดงผลแบบ CLI รวมถึงการจัดการโครงสร้างข้อมูลอย่างเป็นระบบ

นอกจากนี้ยังได้ฝึกการทำงานเป็นทีม การแบ่งหน้าที่ความรับผิดชอบ และการวิเคราะห์ปัญหาเชิงระบบ ซึ่งสามารถนำไปประยุกต์ใช้ในงานพัฒนาซอฟต์แวร์จริงในอนาคตได้อย่างมีประสิทธิภาพ