ДОДАТОК А ТИТУЛЬНИЙ АРКУШ

Міністерство освіти і науки України Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 2

з дисципліни

«Компоненті програмної інженерії 1»

«Специфікування предметної галузі проекту засобами мови UML» Варіант <u>10 «АІС мережі кінотеатрів»</u>

Виконав(ла) Ткаченко Костянтин Олександрович

Перевірив Марченко О.І

Завдання на лабораторну роботу

- 1. Виявити класи, які відносяться до системи що проектується використовуючи діаграму варіантів використання (мінімум 10 класів). Використовувати повний синтаксис атрибутів та операцій в класах.
- 2. Встановити зв'язки між класами. Використати усі чотири типи зв'язків, що наведені в лабораторній роботі. Побудувати цілісну діаграму.
- 3. Коротко текстом (або в таблиці) описати призначення кожного класу та його атрибутів і операцій.
- 4. У звіт обов'язково додати діаграму варіантів використання з першої роботи.

1) Класи, які відносяться до системи

назва: string жанр: string тривалість: double вікове обмеження: int режисер: string опис: string рейтинг: float дата виходу: date id: Long отриматиІнформацію(): змінитиРейтинг(float): void відображенняРозкладу():

Сеанс дата: date час: time зал: Long вільні місця: int ціна: float id: Long забронюватиМісця(int): відмінитиСеанс(): void оновитиІнформацію(): void

 Квиток	
номер: int	
ряд: int місце: int	
ціна: float	
id: Long	
валідувати(): boolean	
відмінити(): void друк(): void	
друк(). Void	

мдміністратор

ім'я: string прізвище: string
електронна пошта: string
рівеньДоступу: AccessLevel

□ Повернення квитків
квиток: Квиток причина: string час повернення: datetime сума: float id операції: Long
обробитиПовернення(): перевіритиУмови():

Користувач		
ім'я: string		
прізвище: string електронна		
пошта: string		
телефон: string		
історія покупок:		

зареєструватися(): void додатиФільм(Film): void оновитиСеанс(Session): void замовитиКвиток(): Ticket оцінитиФільм(float): void переглядЗвітів(): Report[]

відгук
користувач: User фільм:
Film текст: string рейтинг:
float дата: datetime

додатиВідгук(): void
видалитиВідгук(): void
отриматиВсіВідгуки():

Оплата квиток: Ticket сума: float спосібОплати: PaymentMethod часОплати: datetime провестиОплату(): PaymentStatus верифікуватиТранзакцію(): boolean

Розклад

сеанси: Session[] дата:
date кінотеатр: Cinema

отриматиРозклад():
оновитиРозклад(): void
видалитиСеанс(): void

меню: MenuItem[] годиниРоботи: TimeInterval ціни: PriceList
додатиВМеню(): void видалитиЗМеню(): void

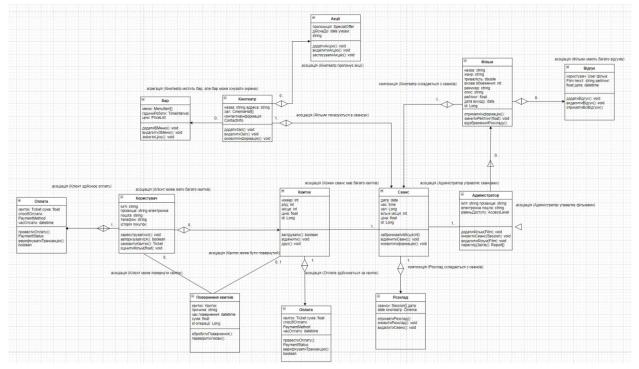
змінитиЦіну(): void

назва: string адреса: string зал: CinemaHall[] контактнаІнформація: ContactInfo
додатиЗал(): void видалитиЗал(): void оновитиІнформацію(): void

Кінотеатр

пропозиція: SpecialOffer дійснаДо: date умови: string
додатиАкцію(): void видалитиАкцію(): void застосуватиАкцію(): void

2) зв'язки між класами.



3) Описання кожного классу

- Фільм: Містить інформацію про фільм, таку як назва, жанр, тривалість, вікове обмеження, рейтинг, дата виходу та методи для отримання цієї інформації.
- Сеанс: Представляє сеанс кінопоказу з датою, часом, залом і ціною. Має методи для забронювання місця, відміни сеансу та оновлення інформації.
- Квиток: Містить дані про квиток, такі як номер, ряд, місце, ціна та ідентифікатор. Має методи для перевірки дійсності та відміни.
- Повернення квитків: Відповідає за обробку повернення квитків, зберігаючи інформацію про квиток, причину повернення, час повернення, суму та ідентифікатор операції.
- Користувач: Зберігає дані про користувача, такі як ім'я, прізвище, електронну пошту та телефон, а також історію покупок.
- Адміністратор: Представляє адміністратора системи з ім'ям, прізвищем, електронною поштою та рівнем доступу. Має методи для додавання фільмів, сеансів, видалення фільмів та генерації звітів.
- Відгук: Представляє відгук користувача про фільм, зберігаючи інформацію про користувача, фільм, текст відгуку та дату.

- Оплата: Містить деталі про оплату квитка, такі як квиток, сума, спосіб оплати, час оплати, а також методи для проведення оплати та перевірки її статусу.
- Розклад: Представляє розклад сеансів для певного кінотеатру, зберігаючи інформацію про сеанси та методи для отримання розкладу та оновлення сеансів.
- Бар: Представляє меню бару в кінотеатрі з переліком пунктів меню, часом роботи та цінами. Має методи для додавання, видалення та змінення меню.
- Кінотеатр: Зберігає інформацію про кінотеатр, таку як назва, адреса, зали та контактні дані. Має методи для додавання та видалення залів і оновлення інформації.
- Акції: Представляє спеціальні пропозиції та акції з інформацією про пропозицію, дійсність та умови. Має методи для додавання, видалення та застосування акцій.

4)

Варіант використання 1: Бронювання квитків

Основний сценарій:

- 1. Клієнт вибирає фільм, дату та час сеансу.
- 2. Система відображає доступні місця у кінозалі.
- 3. Клієнт вибирає місця для бронювання.
- 4. Клієнт вводить свої контактні дані та приступає до оплати.
- 5. Система підтверджує бронювання та відправляє клієнту квитки електронною поштою або через мобільний додаток.

Розширення:

- **Екран вибору місць**: Якщо обране місце вже заброньоване іншим користувачем під час вибору, система повідомляє про це та пропонує вибрати інше місце.
- Оплата: Якщо оплата не пройшла, система пропонує спробувати інший спосіб оплати або повторити спробу.

Включення:

• **Перегляд деталей фільму**: Перед вибором сеансу користувач може переглянути детальну інформацію про фільм, включаючи опис, рейтинг та відгуки.



Варіант використання 2: Управління розкладом сеансів

Основний сценарій:

- 1. Адміністратор вибирає опцію додавання нового сеансу в системі.
- 2. Вводить інформацію про фільм, дату, час та зал показу.
- 3. Система зберігає інформацію про сеанс та автоматично оновлює розклад на сайті кінотеатру.

Розширення:

- **Конфлікт часу**: Якщо обраний час для сеансу перетинається з іншим вже існуючим сеансом у тому ж залі, система попереджає адміністратора та пропонує вибрати інший час.
- **Аналіз попиту**: Система аналізує попит на попередні сеанси та рекомендує адміністратору оптимальні час та дати для нових сеансів.

Включення:

• **Оновлення інформації про фільм**: Перед додаванням нового сеансу, адміністратор може оновити інформац

