

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів

Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Годований Артем Русланович

(прізвище, ім'я, по батькові здобувача освіти)

УДК 004:791.45

КВАЛІФІКАЦІЙНА РОБОТА

“Інформаційна система онлайн продажів квитків у кінотеатрах”

(тема роботи)

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавра кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи

Ковальчук Майя Олегівна

(прізвище, ім'я, по батькові)

к.п.н., доцент кафедри КТіМС

(науковий ступінь, вчене звання)

Житомир – 2023

Висновок кафедри _____

За результатами попереднього захисту: _____

Протокол засідання кафедри _____

№ ___ від «_____» _____ 20___ р.

Завідувач кафедри _____

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

«_____» _____ 20___ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив
(ла)

(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

АНОТАЦІЯ

Годований А.Р. Інформаційна система обліку кадрів. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 122 – комп'ютерні науки. – Поліський національний університет, Житомир, 2023.

Дипломна робота присвячена створенню застосунку для автоматизації процесів онлайн продажу квитків у кінотеатрах. Предметом даної роботи є методи та інформаційні технології автоматизації бізнес-процесів кінотеатру. У першому розділі розглянуто актуальність існуючих систем онлайн продажу квитків у кінотеатрах та аналіз конкурентних переваг запропонованої інформаційної системи. У другому розділі детально розглядаються сценарії використання, алгоритми та рішення програмного забезпечення. У третьому розділі створено прототипи інтерфейсу на платформі Android Studio за допомогою мови програмування JavaScript з використанням HTML та CSS..

У результаті виконаної роботи проаналізовано комплекс застосунків та був розроблений мобільний додаток на платформі Android і розглянута можливість використання інформаційних технологій для автоматизації бізнес-процесів кінотеатру.

Ключові слова: автоматизація, кінотеатр, JavaScript, мобільний додаток

SUMMARY

Hodovanyi A.R. Human Resources Accounting Information System - Qualification work manuscript.

Qualification work for the Bachelor's degree in Computer Science, specialty 122 - Computer Science. - Polissia National University, Zhytomyr, 2023.

The graduate work is dedicated to the development of an application for automating online ticket sales in cinemas. The subject of this work is the methods and information technologies for automating cinema business processes. The first chapter discusses the relevance of existing online ticket sales systems in cinemas and analyzes the competitive advantages of the proposed information system. The second chapter provides a detailed examination of usage scenarios, algorithms, and software solutions. The third chapter involves creating interface prototypes using the Android Studio platform with JavaScript language, HTML, and CSS.

As a result of the completed work, a complex of applications was analyzed, and a mobile application was developed on the Android platform, exploring the possibilities of using technologies for automating cinema business processes.

Keywords: automation, cinema, JavaScript, mobile application

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1: АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ	
1.1. Визначення інформаційних потреб об'єкта інформатизації	9
1.2. Функціональні вимоги до інформаційної технології.....	11
1.3. Огляд існуючих систем онлайн продажу квитків у кінотеатрах.....	15
1.4. Аналіз конкурентних переваг запропонованої інформаційної системи	16
Висновки до першого розділу	19
РОЗДІЛ 2: РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	
ОНЛАЙН ПРОДАЖУ КВИТКІВ У КІНОТЕАТРАХ	
2.1. Узагальнені структурні схеми інформаційної системи	20
2.2. Алгоритми функціонування системи.....	21
2.3. Результати інфологічного моделювання.....	22
2.4. Реалізація функціональних блоків.....	23
Висновки до другого розділу.....	24
РОЗДІЛ 3: ОПИС ІНТЕРФЕЙСУ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	
3.1. Опис інтерфейсу інформаційної системи.....	25
3.2. Порядок встановлення та налаштування системи.....	26
3.3. Керівництво користувача.....	27
3.4. Тестовий приклад.....	29
Висновки до третього розділу.....	30
ЗАГАЛЬНІ ВИСНОВКИ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	33
ДОДАТКИ	36

ВСТУП

З впровадженням електронних інформаційних технологій значна частина сфери життя оптимізувалася. Мережа кінотеатрів є одним з напрямків, де інформаційні системи полегшують життя співробітникам і відвідувачам. Кіноіндустрія розвивається, і в розвиток українського кіно інвестуються мільйони доларів[2], що обов'язково призведе до зростання мереж і незалежних кінотеатрів. Для того, щоб компанія була конкурентоспроможною, необхідно реалізувати стратегічне зростання компанії та вдосконалення основних філій і навіть мереж. У зв'язку зі зміною споживчих уподобань, конкуренції та перерозподілом загальних ринкових тенденцій виникають певні проблеми у виборі методів просування для мережевого кінотеатру [41]. Наприклад, деякі сучасні кінотеатри не мають своїх web-застосунків або мають тільки web-сайт де надана інформація про кінотеатр, фільми, сеанси, але бронювати і купувати квитки онлайн неможливо, це створює незручності. Тому процес розробки додатку, web-сайту в мережі забезпечить приплив нових відвідувачів.

Web – додаток являє собою основний інформаційний ресурс, за допомогою якого можна здійснювати: передачу інформації користувачеві; інформаційну підтримку і контакт з користувачем; здійснювати рекламу.

Тому процес автоматизації роботи кінотеатру включає розробку та впровадження програмного забезпечення для продажу та автоматичного перегляду квитків з урахуванням різних типів місць у кінотеатрі, цінової політики, вибір залів, сеансів. Використання додатку спрощує управління всіма процесами, пов'язаними з прийомом і обробкою замовлень, надаючи швидкість обслуговування, а менеджерам можливість своєчасно отримувати необхідну інформацію.

Тема є **актуальною**, так як, на сьогодні комунікація з аудиторією здійснюється у соціальній мережі, тому мобільний додаток онлайн-бронювання квитків у кінотеатр, це зручний і гнучкий процес, який економить багато часу і робить розваги доступнішими.

Метою роботи є розробка web-додатку із застосуванням мови програмування JavaScript.

Під час виконання даної роботи було поставлено наступні завдання:

-- дослідити інформаційні потреби об'єкта інформатизації та визначення функціональних вимог до інформаційної технології.

-проаналізувати існуючі системи онлайн продажу квитків у кінотеатрах, виділення конкурентних переваг запропонованої інформаційної системи

-розробити узагальнені структурних схем інформаційної системи;

-спроєктувати функціональні блоки інформаційної системи;

-розробити інтерфейс та протестувати інформаційної системи.

Об'єкт дослідження методи і засоби створення додатку для онлайн продажів квитків у кінотеатрах.

Предмет дослідження процес розробки web-додатку для онлайн продажів квитків у кінотеатрах.

Для опрацювання роботи використано такі **методи дослідження**: *метод аналізу літературних джерел* – теоретичне дослідження, вивчення літератури (книг, інтернет-сайтів); *метод аналізу та синтезу* – розглянуто аналоги web-додатків для отримання знань про системи замовлення квитків в кінотеатрі, перегляд і аналіз існуючих сайтів; *метод порівняння* – для аналізу інформаційних потреб об'єкта інформатизації; *системний аналіз* – використовувався для розуміння існуючих систем онлайн продажу квитків у кінотеатрах, виділення конкурентних переваг запропонованої інформаційної системи; аналіз принципів побудови дизайну веб-додатку; *метод моделювання і проектування* – для розробки авторського web-додатку із застосуванням JavaScript та Android Studio.

Отримані результати мають практичне значення у двох аспектах: розробці валідації моделі додатку для кінопоказів і створенні зрозумілої інструкції для користувачів. *Структура та обсяг роботи.* Дипломна робота складається зі вступу, трьох розділів основної частини, загальних висновків, списку використаної літератури та додатків.

РОЗДІЛ 1: АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1. Визначення інформаційних потреб об'єкта інформатизації

Враховуючи специфіку кінотеатральних мереж, їх роль у суспільстві полягає в наданні населенню послуг з показу якісної кінопродукції, організації дозвілля населення, освітньої, культурно-просвітницької та рекламної діяльності (Chandler, A.D., 2007). Відповідно до місії мереж кінотеатрів можна виділити кілька основних завдань (дод.А, рис.1.1), які стоять перед ними: підвищення якості презентації кінопродукції (за рахунок закупівлі іноземно техніки, а також обладнання або необхідного технічного обладнання); розширення мережі кінотеатрів (відкриття нових кінотеатрів у віддалених від міста районах); покращення інвестиційного середовища шляхом участі у світових кінофестивалях; удосконалення системи підготовки кадрів та кваліфікаційної атестації спеціалістів кінопрокату; підвищення інвестиційної привабливості проекту для відкриття нових кінотеатрів; забезпечення новим обладнанням та технологією для презентації кінопродукції [6].

При цьому комплексна система заходів державного впливу на кінотеатри, яка охоплює показ не менше 40-50% вітчизняної кінопродукції виконує наступні функції: здійснює закупівлю кінострічок; складення розкладу кінопоказів; здійснення реклами за допомогою аутсорсингу; касовий продаж квитків; організація кінопоказів; створення звітності за результатами кінопоказів, з продажу, витрат [8, с.4].

Інформаційне забезпечення веб-додатку полягає в наповненні сторінок інформацією, так званим контентом: текстом, графікою, мультимедіа (відео, аудіо та ін.) [5, с. 55-60]. Сучасні інформаційні системи – більшість з яких є комп'ютерними інформаційними системами, які обробляють мільярди даних і виводять необхідну інформацію за дуже короткий час [21][22].

Сучасні інформаційні системи проектуються з урахуванням комплексу вимог до їх роботи. По-перше, це доступ до будь-якої необхідної інформації

в певний момент часу, іншими словами, забезпечення високої продуктивності, надійності та коректності даних.

Найважливіше те, що при розробці інформаційної системи необхідно враховувати стрімкий розвиток інформаційних технологій, тому система повинна мати можливість працювати досить довго, адаптуватися до пристрою та операційної системи, а також мати можливість легко переноситися з однієї платформи на іншу[3, с.348].

Веб-сайт або додаток розроблено з розрахунком на такі групи користувачів: покупці та адміністратори. Інтерфейс буде розроблено для покупців та адміністраторів (адмінпанелі) з урахуванням наступного. Покупці можуть переглядати доступні сеанси, створювати кошики та бронювати квитки. Адміністратори мають доступ до панелі адміністратора, яка має функції для керування сеансами та замовленнями. Технічні вимоги користувача до функцій системи:

Адміністратор. Створення, редагування та видалення фільмів та їх сеанси в панелі адміністратора. 2. Перегляд історії замовлень і даних клієнтів. □

Покупець. 1. Перегляд актуальних фільмів та сеансів у кінотеатрі. 2. Перегляд інформації щодо обраного фільму. 3. Обрання бажаних квитків. 4. Можливість бронювання. 5. Отримання повідомлення на вказаний адрес електронної пошти після успішного бронювання квитків.

Специфікація вимог до даних: □

Фільм. Зберігається його унікальний номер, назва, додаткова інформація (у вигляді жанру та режисера), фотопостер. Покупці можуть переглянути всю цю інформацію на слайдері, на головній сторінці. Адміністратори мають дозвіл редагувати всі властивості, окрім унікального ідентифікатора. Також адміністратори можуть видаляти фільми з сайту. □

Сеанс. Кожен сеанс має унікальний номер, з інформацією про дату і час сеансу, вартість квитка на нього. І призначений лише для одного фільму.

Замовлення. Створюється клієнтом і може містити квиток сеансів. Квитки, з будь-якою кількістю обраних місць. Зарезервовані місця на

часовий проміжок можна забронювати лише один раз. Адміністратор може переглянути номер замовлення, інформацію про нього (фільми, сеанси та вибрані локації, контактні дані), а також видалити це замовлення (у разі форс-мажорних обставин, наприклад, блокування або скасування замовлення).

Покупець. Необхідно зберігати інформацію, надану покупцем при купівлі квитка: ім'я, номер телефону та адресу електронної пошти, на яку йому буде надіслано повідомлення про успішну покупку. Для придбання квитків реєстрація не потрібна. Квитки не можна повернути через веб-інтерфейс[16].

1.2. Функціональні вимоги до інформаційної технології

Залежно від прогнозованої експлуатаційної ресурсомісткості та частоти використання, функціональні вимоги до автоматизованої системи кінотеатру можна розділити на кілька груп:

Ресурсоємні, періодичні операції аналізу, архівування даних, звітності;

Нересурсоємні, часто використовувані функції роботи з сайтом, його адміністрування; Найбільш часто використовуваними операціями є транзакції купівлі квитків, завантаження та потокове передавання інформаційних ресурсів.

Крім того, до таких систем висуваються нефункціональні вимоги, як легкість підтримки серверів, програмного коду, програмного та апаратного забезпечення, швидкість роботи системи, адаптивність до рівнів навантаження, економічність, надійність зберігання даних. Програмний продукт розроблено для кількох груп користувачів – персоналу кінотеатру: адміністраторів, касирів та відвідувачів. При виборі програмної платформи та архітектури майбутнього додатку, в першу чергу варто проаналізувати варіант класичної «монолітної» архітектури (дод. А. рис. 1.2). Це дозволяє створювати програмні додатки, які складаються з багатьох функціональних частин в одній великій структурі. Існує багато програмних платформ,

розроблених спеціально для цієї архітектури: наприклад, Django, AppMaster, Express.js, ASP .NET, MVC, Rails. Усі вони мають перевірені часом інструменти та рішення для подібних завдань, інтеграцію зі сторонніми сервісами, такими як платіжні системи, та автоматизацію складних бізнес-процесів, залишаючись гнучкими та простими в обслуговуванні [18].

Перевагою цього підходу з технічної точки зору є його незалежність від будь-яких зовнішніх ресурсів – його структура, надійність і передбачуваність роботи. Однак цей підхід часто не забезпечує необхідної ефективності для завдань і процесів, навантаження або потоки даних і взаємодії, змінюються. Тому, що монолітна архітектура тісно пов'язана з апаратною частиною. Роботу обслуговує єдиний централізований сервер, який несе витрати на модернізацію та підтримку апаратної інфраструктури у випадку її масштабування та розширення [9]. На противагу, можуть застосовуватися сучасні мікросервіси або “*serverless*” архітектури на основі «*хмарних*» технологій. Цей підхід полягає у використанні багатьох окремих, дуже маленьких програмних компонентів, кожен з яких є окремим веб-додатком [17]. Архітектура мікросервісів — це сервісно-орієнтована архітектурна техніка розробки програмного забезпечення, яка створює додатки як набір незалежних мікросервісів, які спілкуються один з одним на основі «подій», асинхронних повідомлень і ланцюжків виконання, або незалежно і взаємодіють лише з сервером або браузером користувача використовуючи певні механізми, зазвичай HTTP [29] або REST [2]. Serverless архітектура — це спосіб створювати та запускати веб-додатки та служби без необхідності керувати всією інфраструктурою. Підхід передбачає використання сторонньої служби «*backend-as-a-service*» (BaaS) або коду, керованого на платформі «*function-as-a-service*»(FaaS). При використанні цієї архітектури існує підвищена залежність від сторонніх постачальників послуг [4]. Кожен компонент архітектури відповідає певній ролі в роботі програми, а підхід у цілому характеризується автономністю та інтеграцією (дод. А. рис.1.3).

Апаратною інфраструктурою таких систем є віддалені сервери, доступ до яких забезпечує сама технологія на умовах платних послуг. Перевага цього підходу полягає в гнучкості системи, що дозволяє автоматично масштабувати окремі компоненти на льоту. Такі додатки не вимагають підтримки та управління апаратною частиною (сервером), надзвичайно масштабовані та економічно зручні завдяки ще одній особливості: оплата виключно за використані ресурси. Використання такої архітектури накладає певні обмеження на застосовувані технічні рішення, позбавляє цілісності та структурованості, ускладнює процес розробки та тестування. Крім того, для «хмарних» (дод.А, рис.1.4) архітектур характерний негативний вплив «холодного старту». Це включає той факт, що ті частини системи, які не часто використовуються, можуть потребувати більше часу, ніж зазвичай, для початкової ініціалізації. Іноді це може бути кілька секунд, що створює незручності для користувачів інформаційної системи. Узагальнюючи характеристику кожної платформи та архітектури, враховуючи конкретну ситуацію бізнес-процесу в предметній області кінотеатру, недостатньо вибрати лише один метод розгляду, щоб забезпечити найбільш ефективний процес функціонування. Функції розроблених інформаційних систем різноманітні, для отримання оптимального архітектурного рішення пропонується створити нову інформаційну технологію, яка поєднає два розглянутих підходи (дод.А, рис.1.5) і розподіляє інформацію між системами Flow them. для кожного конкретного бізнес-процесу. Тому пропонується автоматизувати функції системи шляхом розробки окремих мікросервісів для забезпечення процесів квиткових кас та онлайн-продажу квитків, надання їх роботі найвищого пріоритету та вільне розширення обчислювальних потужностей. Це забезпечить роботу системи швидкою та стабільною незалежно від навантаження. Цей підхід також використовується для організації доступу користувачів до інформаційних ресурсів і мультимедійного контенту, наприклад, описів фільмів і відеороликів до них, які будуть розміщені в Інтернеті. Саме для задоволення таких потреб

Serverless платформи надають зручний інструментарій – можливість створення CDN [23] (Content Delivery Network) для розповсюдження мультимедійних файлів. Також є готовий інструмент для організації підсистем користувачів і розподілу прав доступу – AWS Cognito [21]. Автоматизувавши процеси, пов'язані з управлінням інформаційних систем та самим кінотеатром, тоді організується єдиний програмний додаток на основі класичної монолітної архітектури. Така підсистема забезпечить надійність, передбачуваність і цілісність процесів і даних у ній, а оскільки вона використовуватиметься в основному персоналом кінотеатрів або мереж кінотеатрів, проблема високих або непередбачуваних навантажень не має значення. Отже, структура інформаційних систем стає розподіленою, а оскільки не всі автоматизовані бізнес-процеси можуть працювати автономно, виникає необхідність забезпечення взаємозв'язку між підсистемами. Взаємодія серверної частини та окремих мікросервісів забезпечується мережним транспортуванням запитів за допомогою стандартного інтерфейсу REST [27] (дод.А, рис.1.6). Representational State Transfer — метод забезпечення взаємодії між інформаційними системами в мережі інтернет.

Сервіси REST [28] дозволяють отримувати доступ до даних з веб-ресурсів за допомогою стандартизованого інтерфейсу запитів і керувати ними за допомогою єдиного стандартного набору операцій[26]. Структура підключення всередині цих підсистем може бути різною, наприклад, мікросервіси є самодостатніми, тому їх можна створювати і запускати незалежно від інших, навіть на зовсім інших мовах програмування [23],(дод.А, рис.1.7). Тим не менш, ми можемо помітити, що в основному існує два типи повідомлень для зв'язку сервісів між собою:

- Синхронні повідомлення – REST зазвичай використовується, якщо клієнт очікує відповіді від служби, оскільки це розподілене середовище, де кожна функція представлена відповідним операційним ресурсом;
- Асинхронні повідомлення – якщо клієнт не очікує відповіді. Зазвичай використовуються протоколи AMQP, STOMP, MQTT, чия бібліотека

повідомлень чітко визначена, і ці повідомлення мають бути сумісними між реалізаціями[7; 33].

Все більше постачальників послуг (Twitter, eBay, Netflix і Amazon) віддають перевагу архітектурі мікросервісів [24].

1.3. Огляд існуючих систем онлайн продажу квитків у кінотеатрах

На сьогоднішній день кількість мобільних додатків для онлайн-бронювання та купівлі квитків у кінотеатр невелика. Більшість сервісів представлено у вигляді веб-сайтів. Кожен з них має переваги й недоліки[1].

Успішний проект повинен не тільки виконувати відповідні вимоги, але й уникати виявлених недоліків[30]. Проаналізуємо популярні веб-сайти. Однією з найбільших кінотеатрів України є “MULTIPLEX” запущений у 2003р. Однією зі стратегій компанії є пошук дистриб'юторів, які володіють правами на найкасовіші фільми. Іншою частиною стратегії компанії є участь у кожному етапі розробки фільму, саме у виробництві, постпродакшні та спецефектах, продажах і дистрибуції[25].

Аналіз (Дод.Б,табл.1.1),(Дод.Б,рис.1.8). Кіноафіші представлені у вигляді каруселі кіноафіш, наведення курсора на них показує останні сеанси та додаткові. Інтерфейс веб-додатку(Дод.Б,рис.1.9) зроблений перелістною стрічкою догори і донизу-що дає гарний вигляд і зручний огляд стрічки. На сторінка вибору зали розташований час сеансу, дата ,ціна. Це надає зручність у виборі-бо не треба переходити до інших сторінок, інформація подана одразу вся і зрозуміла у виборі і використанні.

Далі розглянемо сайт «Планета Кіно». Це друга за величиною мережа кінотеатрів в Україні, тому її було обрано для аналізу та порівняння. Щоб замовити квитки в кіно, користувачі повинні зареєструватися на сайті. Для зареєстрованих користувачів є клубна карта, яка дає право на певні знижки.(Дод. Б,табл. 1.2).

На скріншоті (Дод.Б, рис. 1.10), головна сторінка програми, яка розроблена неефективно, оскільки одночасно видно лише 2- 3 фільми в афіші. Користувач потрапляє на сторінку з розкладом сеансів, вибір за

часовим проміжком: сьогодні, завтра, тиждень або місяць. Є зали різних форматів та технології, користувач може використовувати фільтри, щоб вибрати потрібний. На сторінці вибраного фільму користувач бачить детальну інформацію та короткий опис, а також має можливість придбати квитки з кошика. Придбати квитки можуть як зареєстровані, так і незареєстровані користувачі. Платіжна система прив'язана до сайту «LiqPay», «ApplePay» та «24Pay». Також у «Планети Кіно» [15] є свій робот у популярному месенджері та мобільній версії. Мережа запроваджує систему заохочення для користувачів у вигляді бонусів за покупки.

Інтерфейс веб-додатку «Планета Кіно»(Дод.Б,рис.1.11) зроблений у вигляді каруселі, в огляді тільки один фільм, для бронювання сеансу,зали ,треба робити переходи між сторінками, що не для всіх користувачів зручно.

Наступний аналог – Kinoafisha.ua [13], де можна ознайомитися з розкладом фільмів, що виходять на даний момент практично у всіх кінотеатрах України. Однак не кожен кінотеатр може забронювати квитки, що створює певні незручності для користувачів. Також не можна замовити квитки в міський кінотеатр Житомир. (Дод.Б,табл.1.3)-аналіз застосунку.

Сервіс «КіноАфіша» надає інформацію про фільми в прокаті (Дод.Б,рис.1.12).

Кіноафіша представлена двома каруселями: одна показує вже вийшовші в прокат фільми, а інша – заплановані фільми [19].

Додаток кінотеатру КіноАфіша недійсний.

1.4. Аналіз конкурентних переваг запропонованої інформаційної системи

Проаналізувавши аналоги, можемо виділити мінімальну кількість основних функцій, необхідних замовнику, таких як:

- перегляд кіноафіші;
- перегляд детальної інформації про кінострічку;

-вибір дати, часу сеансу,зали,купівля білету онлайн [14].

Також відзначимо основні недоліки аналогів:

- інформація про придбане місце не передається в режимі реального часу, тому може виникнути ситуація, коли два або більше клієнтів захочуть купити одне і те ж місце і навпаки: якщо користувач зайшов на сторінку, вже є вільне місце буде позначено як придбане, якщо користувач відвідує сторінку замовлення квитку до того, як інший користувач скасує;

-Інтерфейси web-сайтів погано адаптуються до маленьких розмірів екрану, що незручно для мобільних користувачів, яких значна частина. Додатки повинні працювати з кількома користувачами одночасно. Він має працювати 99% часу та сумісний із більшістю поширених браузерів (Google Chrome, Firefox, Opera, Edge) [25] та операційних систем (Windows, Android, iOS) [11] тощо. У результаті функціонального аналізу створена система- наведена в (Дод.Б,табл.1.4). Після виконання зазначених функцій створений застосунок можна вважати кращою порівняно з існуючими аналогічними заявками.

Сьогодні в Україні працює більше десятка мереж кінотеатрів різної форми власності, але “Multiplex cinema” – найбільша з них, займаючи понад 29% ринку. Другою за величиною є мережа «Планета Кіно» (загальна кількість залів – 49) і позиціонується як головний конкурент Multiplex. Це тому, що ці мережі мають велику кількість переваг перед іншими мережами. Наприклад, кінотеатри Multiplex та мережі «Планета Кіно» надають відвідувачам можливість: ознайомитися з афішами фільмів у прокаті, акціями та знижками та іншими новинами на офіційному сайті; купити квитки онлайн в будь-який зручний час; стати учасником “Клубу відвідувачів кінотеатру” та отримати безкоштовні квитки; придбати квитки, попкорн тощо в комфортних умовах; дивитися фільми преміум-класу завдяки унікальним технологіям (IMAX, 4DX, 3D, XpanD тощо), придбаним у світових лідерів, таким як звукові системи від Harman Kardon); придбані квитки, за допомогою мобільного додатку, не потрібно друкувати, просто відсканувати штрих-код, щоб увійти; наявність подарункових сертифікатів номіналом 100, 200 та 500 гривень;

організація цікавих заходів при введенні в прокат нових фільмів, облаштовують фотозони, роблять конкурси та розіграші призів; розміщення відгуків про кінотеатр і можливість отримати feedback від менеджерів і власників компаній.

При створенні програми з використанням інженерного підходу можна визначити ряд стандартних етапів, які необхідно пройти для створення найбільш успішного програмного забезпечення, яке відповідає всім вимогам і придатне для масштабування:

а) Специфікація вимог. На основі аналізу конкуруючих додатків, співпраці із замовниками та власних досліджень створюється перелік функціональних і нефункціональних вимог, яким повинна відповідати система створення, необхідні архітектурні стилі, інструменти розробки, та очікувані обмеження.

б) Створення прототипу інтерфейсу користувача. Сцена для швидкого концептуалізації екранів програми. Структури, зручності та інтуїтивності інтерфейсу, шляхом вдосконалення схеми UI з урахуванням бажаного UX.

в) Розробка детального проекту. Уточнення створеного прототипу, вказані необхідні іконки та зображення, вибір потрібної колірної палітри з урахуванням принципів доступності та теорії кольору. Демонстрація, як програма повинна реагувати на різні ситуації: адаптування до різних розмірів екрана, обробка надто довгого тексту, виведення помилок.

д) Моделювання. Змодельовано концептуальну схему сутностей, баз, які будуть представлені в системі, що розробляється. Для цього зазвичай використовують такі позначення, як UML і ERD, які потім будуть використані розробником для створення відповідної бази даних.

е) Розробка архітектури системи. Визначає концептуальну модель об'єкта та спирається на специфікацію вимог, її підсистем і того, як вони взаємодіють одна з одною. Це особливо важливо для створення програм.

ж) Впровадження програмного забезпечення. На основі інформації, визначеної вище, кодування системи.

з) Конфігурація системи розгортання. Процес адаптації системи до роботи з урахуванням взаємодії створених сервісів і потенційних відмінностей програмного середовища.

і) Перевірити та виправити виявлені недоліки, тестування. Створену систему перевіряють на недоліки за допомогою методів тестування та виправляються виявлені недоліки. Процес є повторюваним, і кожного разу, коли дефект виправляється, систему потрібно тестувати знову, щоб підтвердити її працездатність.

к) Документація. Під час цього етапу створюється документація, примітки та коментарі, щоб допомогти розробники для усунення несправностей і розширення системи, а також чіткі, зручні інструкції з використання[31].

Висновки до першого розділу

Під час виконання першого розділу було проведено аналіз предметної області, описано функції веб-додатку і їх вимоги при створенні. Здійснено порівняльну характеристику існуючих web-сайтів та діючих додатків, які дозволяють замовляти квитки в кінотеатр: Multiflex, Planetakino.ua, Kinoafisha.ua.

На основі проведеного аналізу визначені основні задачі розробки, створено порівняльні таблиці, сформовано висновки щодо універсальності: реєстрація, аутентифікації користувача в системі; створення нового профілю користувача; вхід в систему користувача за раніше створеним профілем; відображення списку фільмів для відтворення; відтворення демонстрації з інформацією про фільм; відображення дати та часу вибраного сеансу; можливість вибору місць, покупки та перегляду придбаних квитків.

РОЗДІЛ 2: РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН ПРОДАЖУ КВИТКІВ У КІНОТЕАТРАХ

2.1. Узагальнені структурні схеми інформаційної системи

Основою функціонування будь-якої системи є процес, а основою інформаційної системи є процес виробництва інформації. Отже, призначення інформаційних систем — виробляти інформацію для потреб організації для забезпечення ефективного управління її діяльністю.

Структура програмного модуля на основі аналізу функціональних вимог програмних модулів мобільний додаток бронювання має структурну схему (дод. В рис. 2.13), [32]. Ця структурна схема розділена на два блоки: клієнтський блок і серверний блок. Загальні складові структурної схеми:

- база даних, елементами якої є дані користувача; дані про фільми, конференції, вільні місця та час, номери купонів;
- блок, який взаємодіє з базою даних, що відповідає за підключення клієнтської частини. програми з базою даних.

Компоненти клієнтського блоку діаграми структури:

- інтерфейс користувача, виконує передачу інформації між підрозділом;
- Робочі блоки з профілями користувачів, авторизацію у системі.
- Блокує вхід із зареєстрованими профілями, на основі даних.
- Робочий блок фільмів, відповідальний за можливість перегляду всіх фільмів, наявних у базі даних, вибір і перегляд опису фільму.
- Блок роботи з кінотеатрами, надання даних про вільні місця на сеансі.
- Блок купонів, який надає користувачам код ваучера.

Компоненти серверного блоку на структурній схемі:

- Блок оновлення сеансу, надання оновлених даних про сеанс системі;
- Блоки для оновлення вільних локацій, надає системі дані про локації;
- Блок оновлення доступного часу, дані про доступний час;
- Блок генерації облікових даних, користувач може повторно ввести персональні дані та скористатися функціями сервісу; – Блок оновлення списку фільмів, додавання нових фільмів до бази даних в сервісі;

Діаграми послідовності показують життєвий цикл об'єктів, порядок, у якому надсилаються та отримуються повідомлення між об'єктами, показують різні процеси або об'єкти, які існують, у вигляді вертикальних ліній різні процеси. Надіслані повідомлення відображаються горизонтальними лініями в порядку їх надсилання(дод.В рис. 2.14) . Опис процесу авторизації профілю користувача (дод.В рис. 2.15) включає:

- клієнт надсилає запит на авторизацію в системі; – сервер аналізує надані дані; – якщо аналіз успішний, якщо сервер знаходить адресу електронної пошти та пароль, доступні в базі даних; – сервер генерує маркер авторизації користувача. і записати його в базу даних; – сервер надсилає повідомлення про авторизацію профілю користувачу.

Опис процесу отримання списку перегляду користувача (дод.В рис. 2.16) передбачає: – користувач надсилає запит на отримання списку доступних фільмів; – сервер завантажує доступні фільми з даних ;

Опис процесу отримання списку доступних сеансів для фільму(дод.В рис. 2.17) включає:

- користувач надсилає запит на отримання списку доступних сеансів;
- сервер вибирає доступний сеанс із бази даних;
- якщо сервер не знаходить фільм, користувач отримує повідомлення про помилку;
- якщо сервер знаходить фільм, сервер надсилає користувачеві список доступних сеансів користувачу.

2.2. Алгоритми функціонування системи

*Клієнт.*Клієнтам має бути надано доступ до інформації про програму кінотеатру, яка має бути актуальною та достовірною. Система повинна допомогти користувачеві вибрати необхідні послуги, дозволяти користувачеві розміщувати замовлення на квитки, обробляти це замовлення та отримувати квитки. Система повинна надавати можливість користувачеві вибрати, на який сеанс і яке вільне місце він може зробити замовлення.

Система має дозволити користувачам повертати квитки в кінотеатр із поверненням коштів; дозволяти користувачеві забронювати квиток для подальшої покупки квитка. Клієнтська система не повинна дозволяти користувачам купувати квитки на неіснуючі сеанси. Система не повинна дозволяти користувачам відшкодовувати квитки за 10 хвилин до початку сеансу. Система не повинна дозволяти зарезервовані місця, які не були викуплені, бронювання потрібно скинути За 20 хвилин до початку сеансу.

Касири. Система має допомогти касирам відстежувати наявні для продажу місця в аудиторіях. Система має звести до мінімуму роботу касира, використовуючи шаблони та допомагаючи клієнтам правильно оформляти замовлення. Надіслати звіт про продажі до фінансового відділу та відділу статистики. Повинна дозволяти касирам контролювати ліміти бронювання та скасування квитків. Система не повинна надавати неправдиві дані у звітах або надавати інформацію про сеанс (Дод.Д.рис.2.18).

Алгоритм створення замовлення для покупця (дод Д, рис.2.19):

Веб-сайт повинен функціонувати належним чином незалежно від операційної системи або браузера користувача. Однак зауважимо, що для успішного використання та розміщення замовлення необхідно мати підключення до Інтернету (дод Д.рис.2.20).

2.3. Результати інфологічного моделювання

Інфологічна модель БД. Інфологічний рівень – це інформаційно-логічна модель (ІЛМ) предметної області, в якій виключається надмірність даних і відображаються інформаційні особливості об'єкту управління незалежно від характеристик і деталей конкретної СУБД.(дод К.рис.2.21)..

Мета інфологічного проектування є створення структурованої інформаційної моделі програмного забезпечення, для якої буде розроблена база даних. При проектуванні інформаційного рівня створюється інформаційна логічна модель, яка повинна відповідати таким вимогам: Коректність схеми бази даних, тобто повне відображення програмного

забезпечення моделювання; Простота використання на наступному етапі проектування, тобто ІЛМ має легко відображатися в моделях БД.(дод К.рис.2.22), які підтримуються добре відомими СУБД (сіткові, ієрархічні, реляційні); ІЛМ має бути описано мовою, яку можуть зрозуміти розробники баз даних, програмісти, адміністратори та майбутні користувачі бази даних.

Основними компонентами інфографічної моделі є атрибути, які необхідно проаналізувати та певним чином згрупувати для подальшого зберігання в базі даних[35].

2.4. Реалізація функціональних блоків

Сучасне суспільство характеризується безперервним розвитком і активним впровадженням процесів інформатизації в більшості сфер людської діяльності [13]. Автоматизація бізнес-процесів – це впровадження програмних систем, які виконують типові процедури за сучасними алгоритмами, напр., таким як складання і виписка документів, контроль виконання облікових, складських операцій. Завдяки подібним нововведенням рівень роботи підприємства якісно підвищується. Щоб побудувати модель інформаційної системи, спочатку опишемо функцію всієї системи у вигляді контекстної діаграми(дод Л.рис.2.23).

Кожен кінотеатр має власну систему залів і місця розташування.Зали мають такі особливості: · Кількість рядів;· Кількість місць в кожному з рядів.

Продаж квитків у кінотеатр можливий через службу черги на місці та за попереднім записом (по телефону або на сайті кінотеатру). Місця на конкретну подію можуть мати кілька статусів: –· вільно;–· заброньовано; –· придбано ;–· не обслуговується

Щоб забронювати квиток, клієнт повинен надати в касу або на веб-сайт таку інформацію: Назва фільму; Дата сеансу; Час сеансу; Кількість квитків; Номер ряду (рядів); · Номер місць (а); · Прізвище (або інше кодове слово).

Під час бронювання місця в кінотеатрі на цей сеанс ніхто інший не зможе купити квиток на це місце.[14].

Замовники – це ті, хто формує попит на послуги кінотеатру.

Репертуар – група фільмів або інших товарів, які демонструються в розкладі показів. Входить: назва фільму; інформація про фільм; виконавці-актори; афіша.

Список усіх сеансів, що проводяться в кінотеатрі входить: назва фільму; дата сеансу і час сеансу; тривалість фільму; ціна квитків за класом А, В, С; схеми кінозалу та вільних місць

Квитки – право клієнта відвідати певний сеанс.

Відшкодування – ситуація, коли клієнт повертає гроші за кінотеатр і отримує гроші, витрачені на квиток. Заброньовано – коли клієнт повертає квиток у кінотеатр і отримує гроші, витрачені на квиток бронювання або коли термін бронювання закінчиться.

Зняття броні – звільнення місця в залі. Внесення його в продаж.

В ході розробки другого розділу було розглянуто інформаційне забезпечення сайту, який визначає основні вимоги, що пред'являються до нього. Визначено, що інформаційне забезпечення системи бронювання квитків у кінотеатр включає інформацію про фільм, інформацію про кінотеатр та інформацію про кіносеанси. Проаналізовано основні структури, функції сайту.

Висновки до другого розділу

У процесі роботи над другим розділом розглянуто інформаційні потреби об'єкта інформатизації та визначенні функціональні вимоги до інформаційної технології, зроблений огляд систем і структури автоматизації процесу вибору та купівлі квитків на кіносеанси.

Сьогодні за допомогою сучасних технологій і передового програмного забезпечення можна ефективно організувати практично будь-який бізнес.

Було проаналізовано основні види структур побудови інформаційної системи сайтів, додатків, розглянуті блоки і діаграми послідовності, визначено зв'язки між ними, які в ній зберігаються.

РОЗДІЛ 3: ОПИС ІНТЕРФЕЙСУ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Опис інтерфейсу інформаційної системи

Інтерфейс розробляється для створення зовнішнього образу, надання йому конструктивної привабливості. Спрямовання на те, щоб зробити програму максимально зрозумілою для користувача. Інтерфейс має бути простим у реалізації, повинен використовувати терміни, зрозумілі користувачам, а об'єкти, якими керує система, мають бути безпосередньо пов'язані з робочим середовищем користувача.

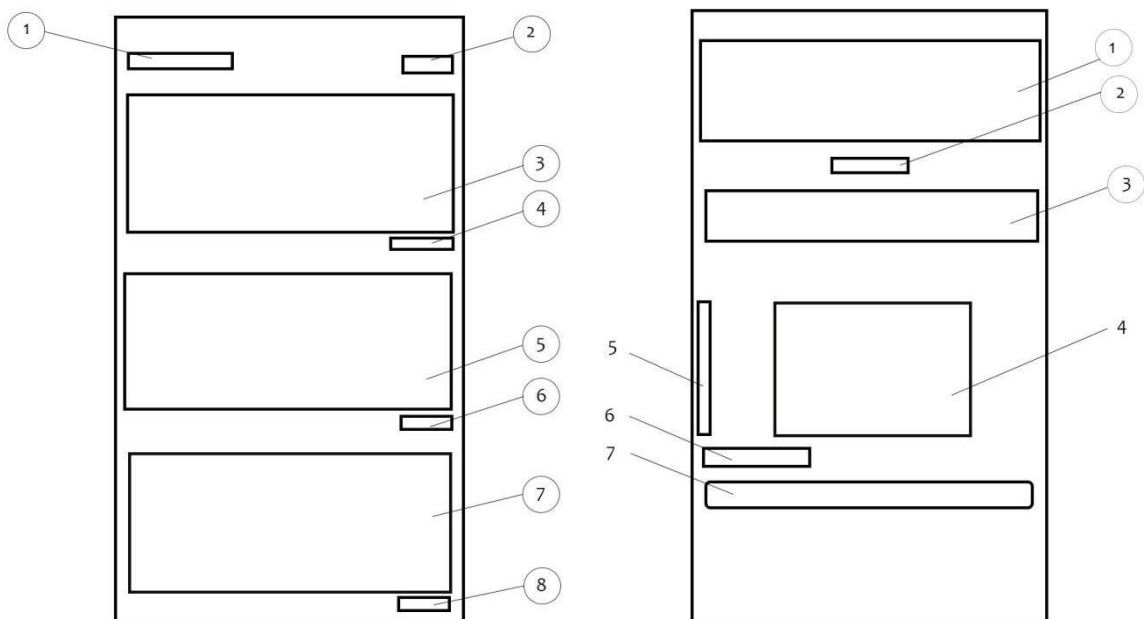


Рисунок 3.24– Схема.Зовнішній вигляд домашньої сторінки -зліва

Згідно зі схемою рисунка 3.21 головна сторінка виглядає наступним чином, з таких блоків: 1. текстова інформація сайту; 2.тестовий блок3. навігація/ список фільмів; 4..тестовий блок; 5. навігація/ список фільмів; 6.тестовий блок; 7. навігація/ список сеансів; 8.тестовий блок;

Рисунок 3.24 – Зовнішній вигляд сторінки особистого кабінету- з права.

Згідно з рисунком 3.24 сторінка особистого кабінету виглядає так, на ній розміщені наступні блоки: 1. кіноафіша; 2. тестовий блок, 3. рухома навігація : час сеансів; 4. схема залу; 5.схема рядів; 6. інфографіка щодо місць у залі; 7. підтвердження броні;

Згідно (дод.М,рис. 3.25) інтерфейс має такий вигляд і використовує блоки: описані вгорі.

3.2. Порядок встановлення та налаштування системи

Зростання продажів пристроїв на базі Android- стрімке, і цим відкриває чудову можливість для розробників програм Android.

Будь-хто може взяти участь, отримання вихідного коду Android, дізнатися принципи операційної системи з відкритим кодом і отримати інформацію про ліцензування. Відкритість платформи сприяє швидкому оновленню. Щоб розробити програми для операційної системи Android, потрібно встановити Android Studio. При розробці додатків для Android використовується Java – одну з найпоширеніших мов програмування. Програмування графічного інтерфейсу керується подіями, які реагують на події, ініційовані користувачем, такі як торкання екрана. Окрім безпосереднього написання коду програми, також можна використовувати середовища розробки Eclipse та Android Studio. Ці середовища розробки дозволяють швидко та легко створювати, тестувати та налагоджувати програми Android. Компоненти графічного інтерфейсу в Android називаються уявленнями (views).

Графічний інтерфейс користувача, створений за допомогою стандартних процедур, містить компонент TextView.

XML (Extensible Markup Language)-розширювана мова розмітки — це природний спосіб опису графічних інтерфейсів. Розмітку XML читають як люди, так і комп'ютери; у контексті Android вона використовується для опису макета використовуваних компонентів і їхніх властивостей: розмір, положення, колір, розмір тексту, поля та відступи. Android Studio аналізує розмітку XML, щоб відобразити макет у редакторі макета, і під час виконання генерує код Java, який складає графічний інтерфейс користувача. Створіть віртуальний пристрій Android. Інструменти Android включають емулятор віртуального пристрою Android (AVD). Емулятор AVD дозволяє тестувати програми на віртуальному мобільному пристрої з операційною

системою Android. Емулятор дозволяє створювати кілька віртуальних пристроїв з різними конфігураціями. Проект — це група пов'язаних файлів (таких як файли коду, ресурси та графічні файли), які складають додаток. Робота програми починається зі створення проекту. (дод.М,рис. 3.26),(дод.М,рис. 3.27)(дод.М,рис. 3.28)(дод.М,рис. 3.29) (дод.М,рис. 3.30).

```

153
154     if [ $CHECK -ne 0 ] && [ $CHECK2 -eq 0 ] ; then          ### Added a condition
155         eval `echo args$i`="`cygpath --path --ignore --mixed "$arg"`
156     else
157         eval `echo args$i`="\"$arg\""
158     fi
159     i=`expr $i + 1`
160 done
161 case $i in
162 0) set -- ;;
163 1) set -- "$args0" ;;
164 2) set -- "$args0" "$args1" ;;
165 3) set -- "$args0" "$args1" "$args2" ;;
166 4) set -- "$args0" "$args1" "$args2" "$args3" ;;
167 5) set -- "$args0" "$args1" "$args2" "$args3" "$args4" ;;
168 6) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" ;;
169 7) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" ;;
170 8) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" "$args7" ;;
171 9) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" "$args7" "$args8" ;;
172 esac
173 fi
174
175 # Escape application args
176 save () {
177     for i do printf %s\n "$i" | sed "s/'/\\"/g;1s/'/'/;\\$s/\\$/ '\\\\' ; done
178     echo " "

```

Рисунок 3.31. коди створення місць в залі.

3.3. Керівництво користувача

Дослідження юзабіліті дозволяють отримувати великі обсяги різномірних і якісних даних, які необхідно обробляти і зберігати для отримання корисних рекомендацій при вирішенні завдань проектування інформаційної системи. З цією метою у світі створено багато експериментальних веб-додатків для оцінки інтерфейсів, відстеження та аналізу експериментальних результатів. Аналіз поведінки користувачів (UBA) — це метод збору, об'єднання та аналізу кількісних і якісних даних користувачів, щоб зрозуміти, як користувачі взаємодіють із продуктом або веб-сайтом і чому під час запуску мобільного додатку головне вікно відкривається за допомогою відображення списку фільмів, доступних у системі (рис. 3.32). Після натискання вибраного фільму відкриється вікно фільму з постером фільму, доступними датами, доступний час, та кнопка Бронювання (рис. 3.33).

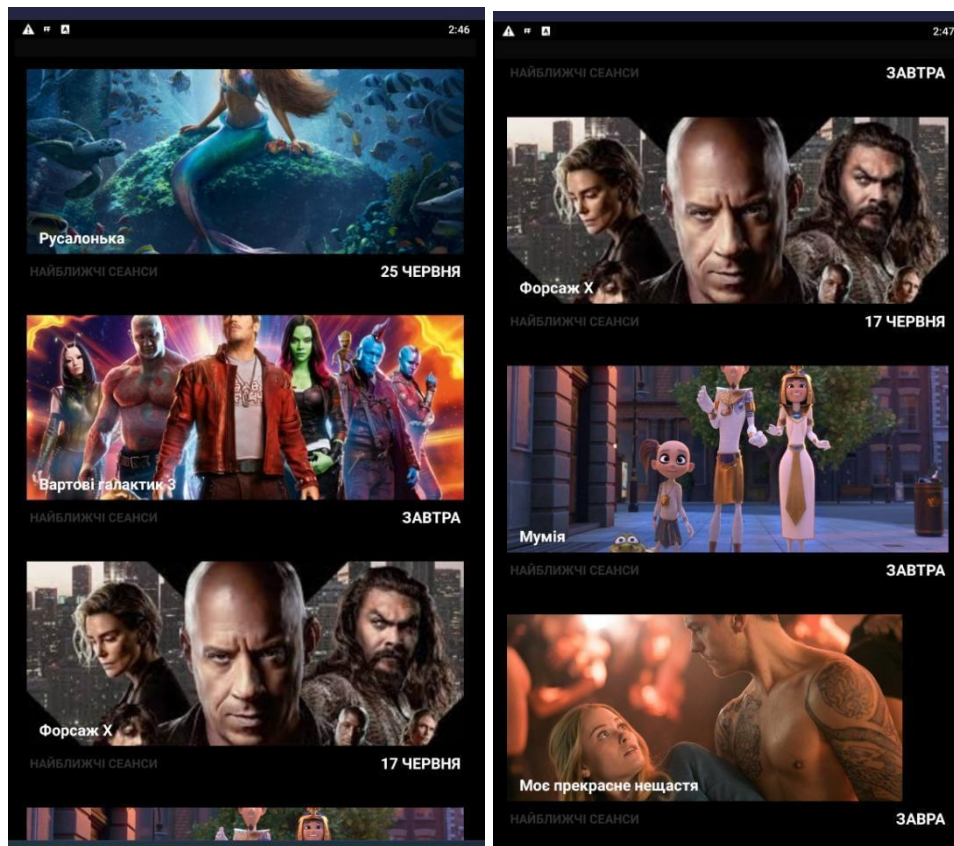


Рис. 3.32. Навігація

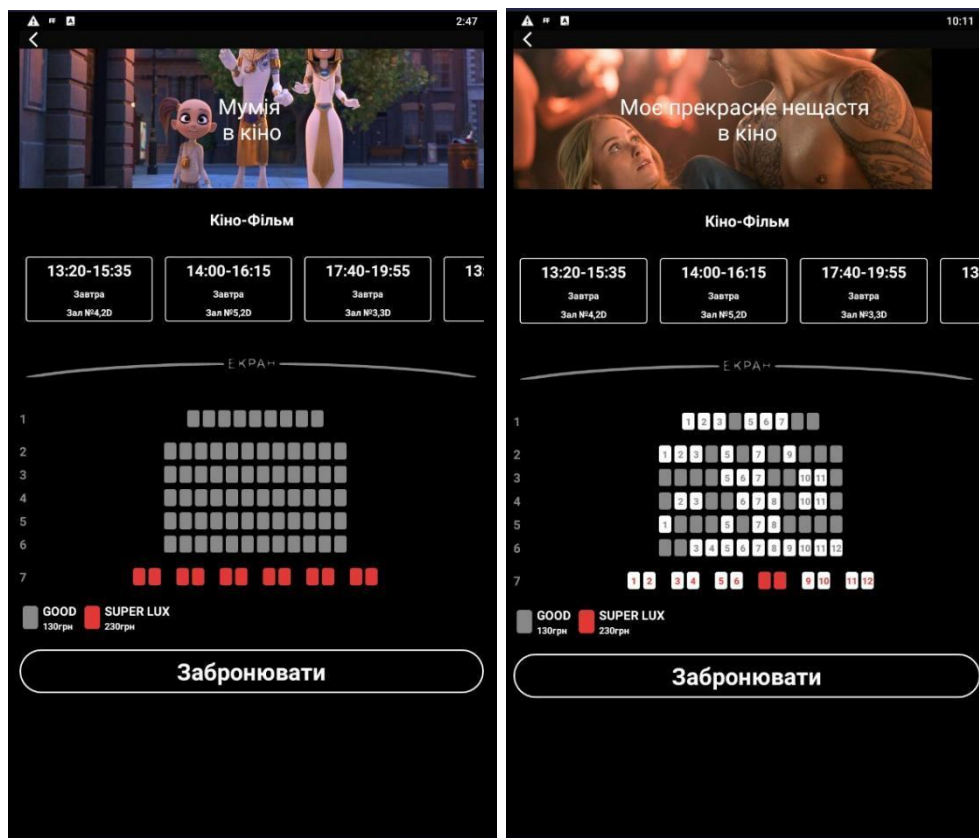


Рис. 3.33. Форма бронювання

Після вибору вільної дати та часу користувач натискає кнопку «Забронювати», після чого користувач отримує код електронного квитка.

3.4. Тестовий приклад

Тестування є невід'ємною фазою будь-якого процесу розробки Програмне забезпечення, як засіб підтвердження працездатності продукту. Чи потрібне тестування в рамках цієї роботи також залежить від Логіки програми, тобто взаємодія з онлайн-сервісами. Тестування програмного забезпечення – це процес, який дозволяє оцінити функціональність програми, щоб дізнатися, чи відповідає розроблене програмне забезпечення встановленим вимогам, і виявляти дефекти таким чином переконатися, що продукт не містить їх після запуску. Відповідно до стандарту ANSI/IEEE 1059, тестування – це «процес аналізу програмні елементи для виявлення відмінностей між існуючими і необхідні умови (тобто недоліки) та характеристики елементів оцінювання програмне забезпечення».

На ринку існує кілька методів тестування програмного забезпечення, але по суті, це дослідницький процес, і необхідно враховувати дійсне тестування, детальну інформацію про продукти в розробці.

План тестування програмного забезпечення, включаючи метод, список компоненти тесту, критерії проходження тесту, список тестів і створення звітів про завершення тестування системи, а також звіт де сталася помилка в системі.

Тестовий компонент – авторизація користувача;

- реєстрація користувача; – перегляд доступних фільмів; – переглянути доступні сеанси; – переглянути схему залу; – підбір вільних місць; – оплата квитка; – огляд придбаних квитків;

Компоненти, що не тестуються :- Система управління базами даних(СУБД);

– API платіжної системи.

Якщо не знайдено жодних помилок перевірка вважається успішною. Всі рівні перевірені на наявність помилок і виконуються всі вимоги, зазначені в методиці завдання.

Для бронювання мобільного додатку для тестування використовується ручне тестування. Метою тесту, наведеного в контрольному прикладі, є перевірка коректності операції сцени купівлі квитків користувачем. У (дод.табл. 3.1). Демонструє покроковий опис тесту, включаючи дії користувача, очікувані та фактичні результати.

Висновки до третього розділу

У третьому розділі було створено додаток онлайн купівлі квитків у кінотеатр на основі досліджених аналогів, який дозволяє: здійснювати пошук фільму, сеансів, часу, демонстрація вільних місць, бронювання квитків, обробка даних користувача, замовлення їжі з бару кінотеатру. Розроблено інтерфейс Web-додатку із використанням JavaScript та інтегрованого середовища Android Studio. Підготовлено інструкцію для користувача з детальним описом функціонування сторінок додатку.

ЗАГАЛЬНІ ВИСНОВКИ

У процесі кваліфікаційного дослідження було розроблено інтерфейс та функціональну частину web-додатку для онлайн продажу квитків у кінотеатр на основі проаналізованих аналогів, із врахуванням їх переваг та недоліків.

У першому розділі було проведено аналіз існуючих сайтів кінотеатрів, який показав, що незважаючи на високий рівень технологічного прогресу, багато кінотеатрів досі не мають власних додатків або розвинутих веб-сайтів. Тому було поставлено завдання створити продукт, який задовольняв би чітко описані умови.

У другому розділі здійснено комплекс робіт спрямованих на проектування інформаційної системи для онлайн продажу квитків, структурних схем, баз і схем даних, діаграм, розроблено узагальнену структурну схему інформаційної системи.

Третій практичний розділ присвячено розробці інтерфейсу мовою програмування JavaScript на платформі Android. Створено інструкцію користувача web-додатку для онлайн продажу квитків у кінотеатр, охарактеризовано розробку. Розроблено структурну схему, інтерфейс користувача для передачі інформації яка складається з компонентів мобільного додатку бронювання квитків. Визначено мінімальні системні вимоги для смартфонів з операційними системами Android та iOS не нижче 12. Алгоритм, реалізований програмно: складова частина кіноафіші; спливаючий компонент для фільмів; компонент відображення дати й часу.

Розроблений додаток може використовуватися для комерційних цілей, оскільки вдосконалено та розширено функціональні можливості: додавання фільтрів (за датою, за фільмом тощо); створення та впровадження облікових записів користувачів (для зручнішої комунікації з клієнтами, системи пропозицій – акцій, знижок тощо); підключення платіжних систем для забезпечення можливості онлайн-платежів; введення статистичної звітності адміністраторам (перегляд статистики сеансів, інформацію про продаж

квитків, можливість збереження звіту у вигляді Excel-листа); оформлення замовлення їжі з бару кінотеатру.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жлуктарьову А. А. Комплекс застосунків автоматизації бізнеспроцесів кінотеатру. Миколаїв. 2022. с.91
2. Інформаційна система забезпечення ефективності функціонування бізнес-процесів організації на прикладі кінотеатру. Технічні науки. Вісник Хмельницького національного університету, Том 2, №6, 2018. с.146-150.
3. Іванов В. Г., Сучасні інформаційні системи і технології. Конспект лекцій. Харків. 2014. с. 348
4. Мацюк В.О. Комп'ютерні науки. Реалізація інформаційної системи на основі реляційної та нереляційної бази даних. нац.ун-т Києво-Могилянська академія. Київ. 2021. с.46
5. Романюк О.Н. / Веб-дизайн і комп'ютерна графіка : навч. посіб./ О.Н. Романюк, Д.І. Кательніков, О.П. Косовець. Вінниця : ВНТУ. 2007. 147с.
6. Стангурська Б.О. Стратегічний інструментарій розвитку кінотеатральних мереж. Львів. 2022. С. 65-80.
7. Прикладні системи та технології в інформаційному суспільстві: зб. тез доповідей і наук. повідомл. учасників IV Міжнародної науково-практичної конференції (Київ, 30 вересня 2020 р.) [за заг. ред. В.Л. Плєскач, В.Л. Міронова]. К.: Київський нац. ун-т імені Тараса Шевченка, 2020. 253 с.
8. Ушакова І. О. Плєханова Г. О. Конспект лекцій. Інформаційна система та технології підприємств. Харків : Вид. ХНЕУ. 2009. с.4
9. Якимець Р. В., Яременко К. М. Масштабування навантаження Web-додатків. Міжнародний науковий журнал. 2016. 6(2). С. 45-47. URL: http://nbuv.gov.ua/UJRN/mnj_2016_6%282%29__11
10. Автоматизація кінотеатрів, театрів, концерт-холів, музеїв, фестивалей. URL: <https://expertsolution.com.ua/uk/avtomatizacija-kinoteatrov--teatrov--operi>
11. Зараз у кіно, Афіша – Кінотеатр Multiplex | Київ, Харків, Дніпро, Херсон і вся Україна. URL: <https://multiplex.ua>
13. Кіноафіша. URL: <http://kinoafisha.ua>

14. Краліна Г., Баков Н. Операційні системи: популярність, характеристики, вибір InterConf. 2020. URL: <https://marketer.ua/ua/stats-of-browsers-2017/>
15. Планета Кіно в Києві (Blockbuster) — мережа кінотеатрів. URL: <https://planetakino.ua>
16. Реалізація інформаційної системи на основі реляційної та нереляційної базаданих. URL: <https://ekmair.ukma.edu.ua/server/api/core/bitstreams/d577f09f-9181-4c05-aa2a-b6e366453b1a/content>
17. Сидоренко Г.Ю. Використання Serverless підходу для створення веб-додатку моніторингу товарів. URL: http://repository.kpi.kharkov.ua/bitstream/KhPIPress/37361/1/vestnik_KhPI_2018_21_Sydorenko_Vykorystannia_Serverless.pdf
18. Упіров І.С. Система управління режимами функціонування електричної мікромережі. URL: https://ela.kpi.ua/bitstream/123456789/32308/1/Upirov_magistr.pdf
19. Фільми в кінотеатрах України на KINOafisha.ua. URL: <https://kinoafisha.ua/>
20. Шеремета Б. О. Аналіз кінопрокатного ринку України та напрями його розвитку. URL: <https://science.lpnu.ua/sites/default/files/journal-paper/2019/apr/16490/semi12019-55-61.pdf>
21. Information Systems and Future. URL: <https://witanworld.com/article/2019/10/03/information-systems-and-future/>
22. Components Of Information System. URL: <https://www.geeksforgeeks.org/components-of-information-system/>
23. Microservices design patterns. URL: <https://www.javacodegeeks.com/2015/04/microservice-design-patterns.html>
24. Optimal Inter-Release Time between Sequentially Released Products. K. Sudhir (Yale School of Management).

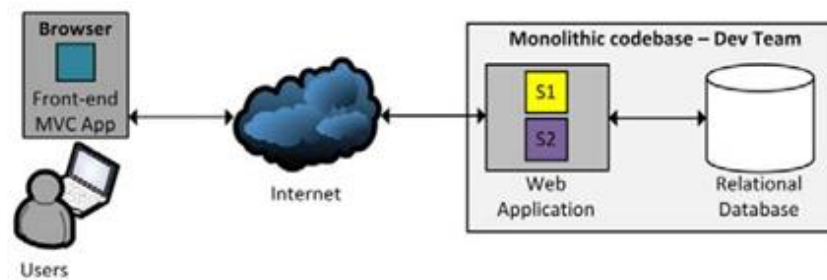
26. Representational State Transfer (REST) URL:
<http://hinews.pp.ua/kompyuteri/4865-rest-api-scho-ce-rest-pereklad-representational-state-transfer.html>
27. REST. URL: <https://uk.wikipedia.org/wiki/REST>
28. Використання RESTful. URL: <http://edu.asu.in.ua/mod/book/tool/print/index.php?id=117>
29. HTTP. URL: <https://ru.wikipedia.org/wiki/HTTP>
30. eZdorovya. Адміністратор центральної бази даних eHealth України.
URL: <https://ehealth.gov.ua>
31. Introduction to Apache Kafka. URL: <https://kafka.apache.org/intro>
32. Levy E. Kafka vs. RabbitMQ: Architecture, Performance & Use Cases. Blog Upsolver, 2019. URL: <https://www.upsolver.com/blog/kafka-versus-rabbitmq-architecture-performance-use-case>
33. Блог компанії ITSumma. 2018. URL:
[:https://habr.com/ru/company/itsumma/blog/416629/](https://habr.com/ru/company/itsumma/blog/416629/)
34. Порівняльний аналіз баз даних SQL ТА NOSQL. URL:
<https://conf.ztu.edu.ua/wp-content/uploads/2021/05/3-2.pdf>
35. Основні поняття теорії інфологічного моделювання URL:
<https://studfile.net/preview/10045748/page:3/>

ДОДАТКИ

Додаток А



Рис. 1.1. Складові інформаційних систем



(дод. А. Рис. 1.2. «Монолітна» архітектура веб-додатку

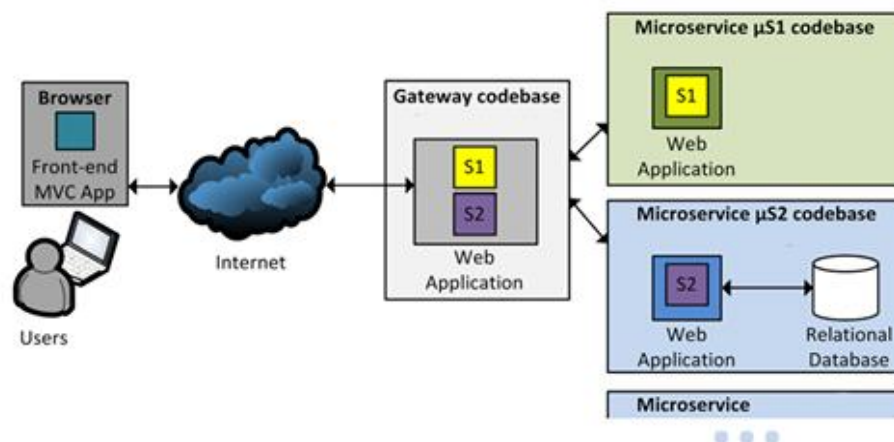


Рис.1.3. Мікросервісна архітектура веб-додатку

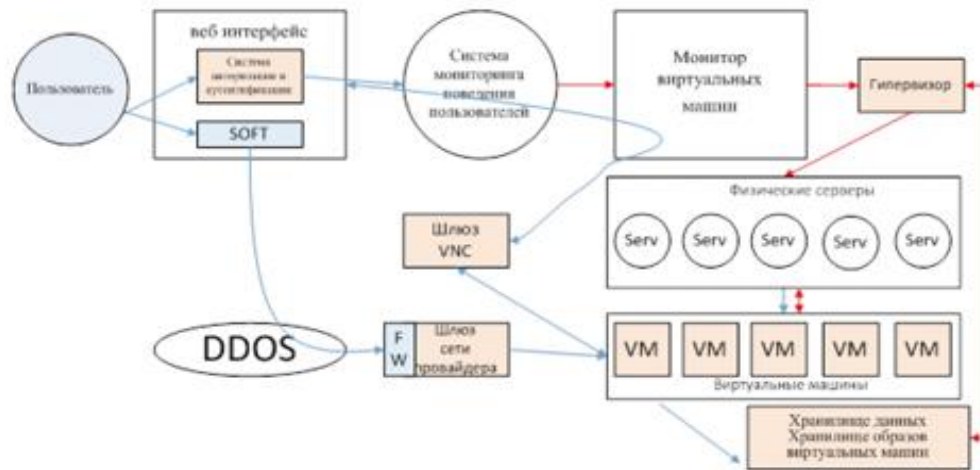


рис.1.4. Приклад моделі системи «хмарних» обчислень

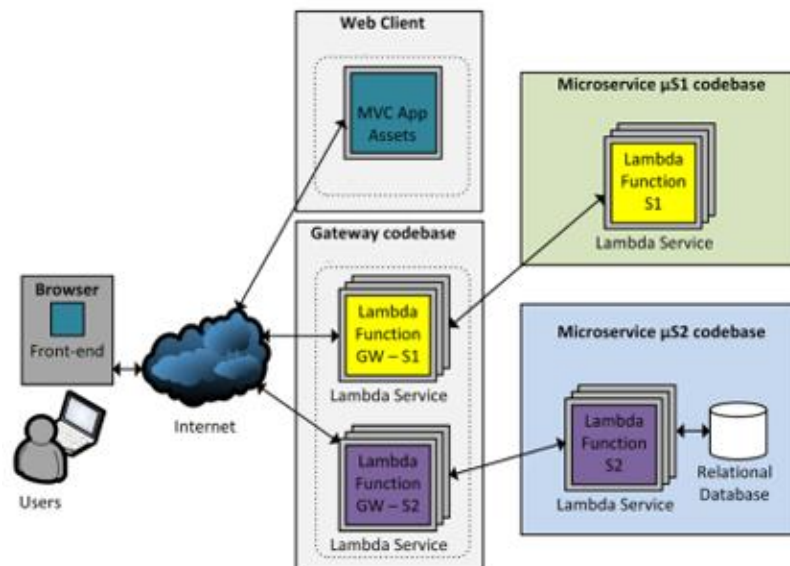


Рис.1.5. Комбінована архітектура, що поєднує мікросервіси та «монолітний» MVC додаток в якості серверної частини.

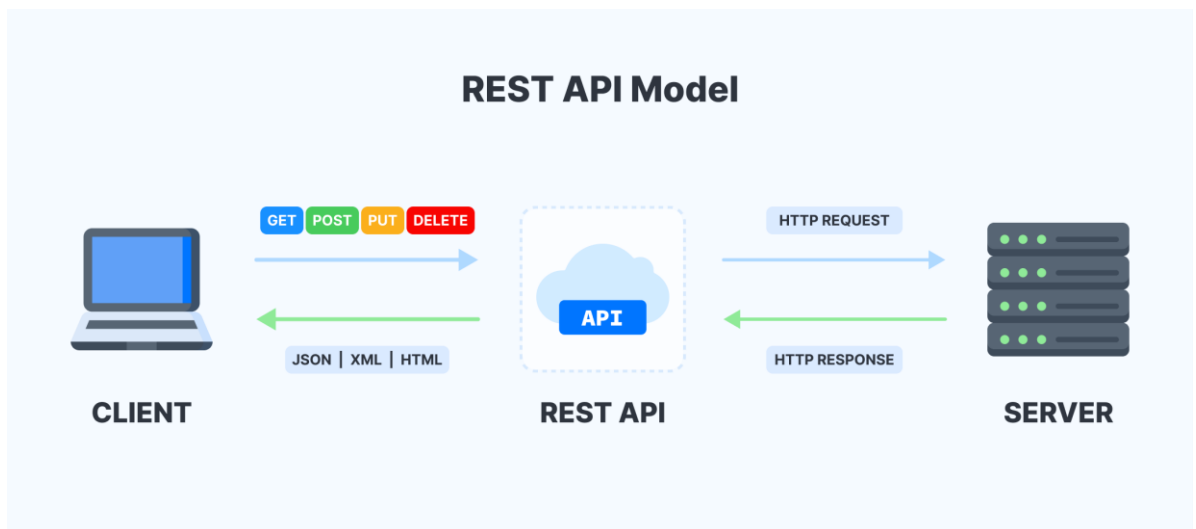


Рис.1.6. REST

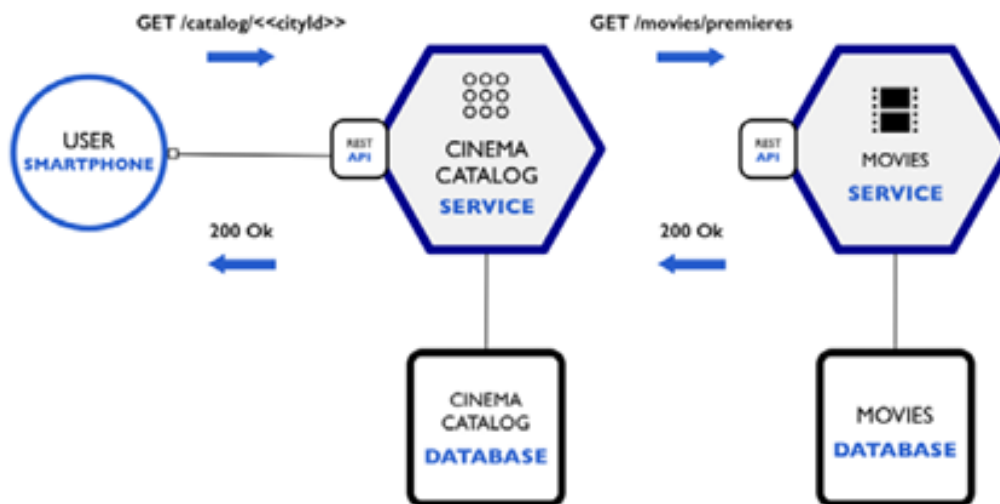


Рис.1.7. Схема взаємодії частин розроблюваної інформаційної системи

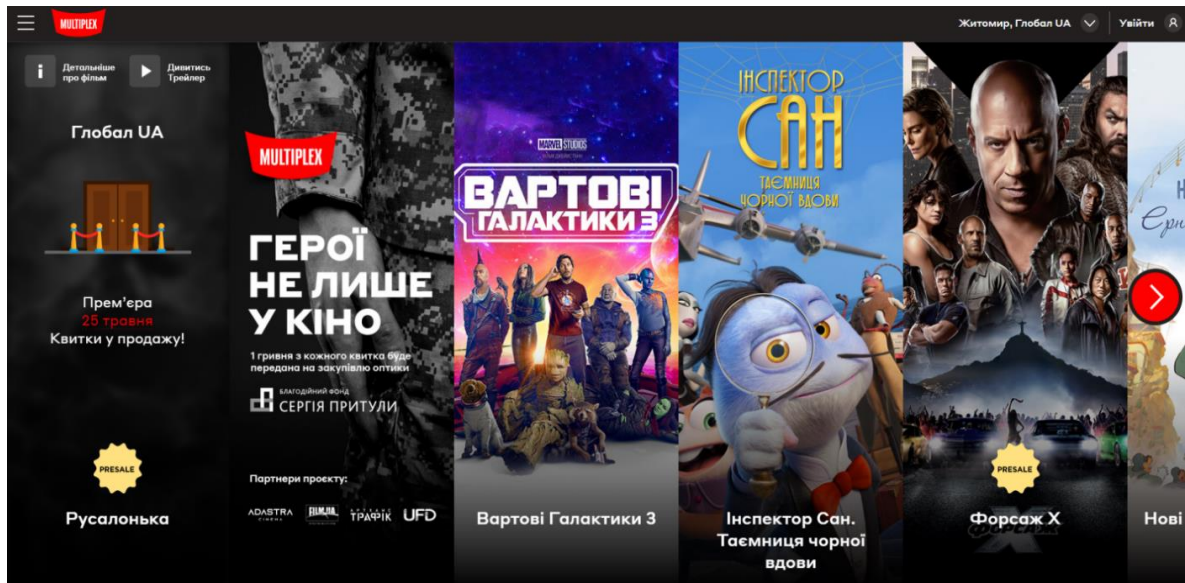


Рис. 1. 8 – Інтерфейс веб-сайту «Multiplex»

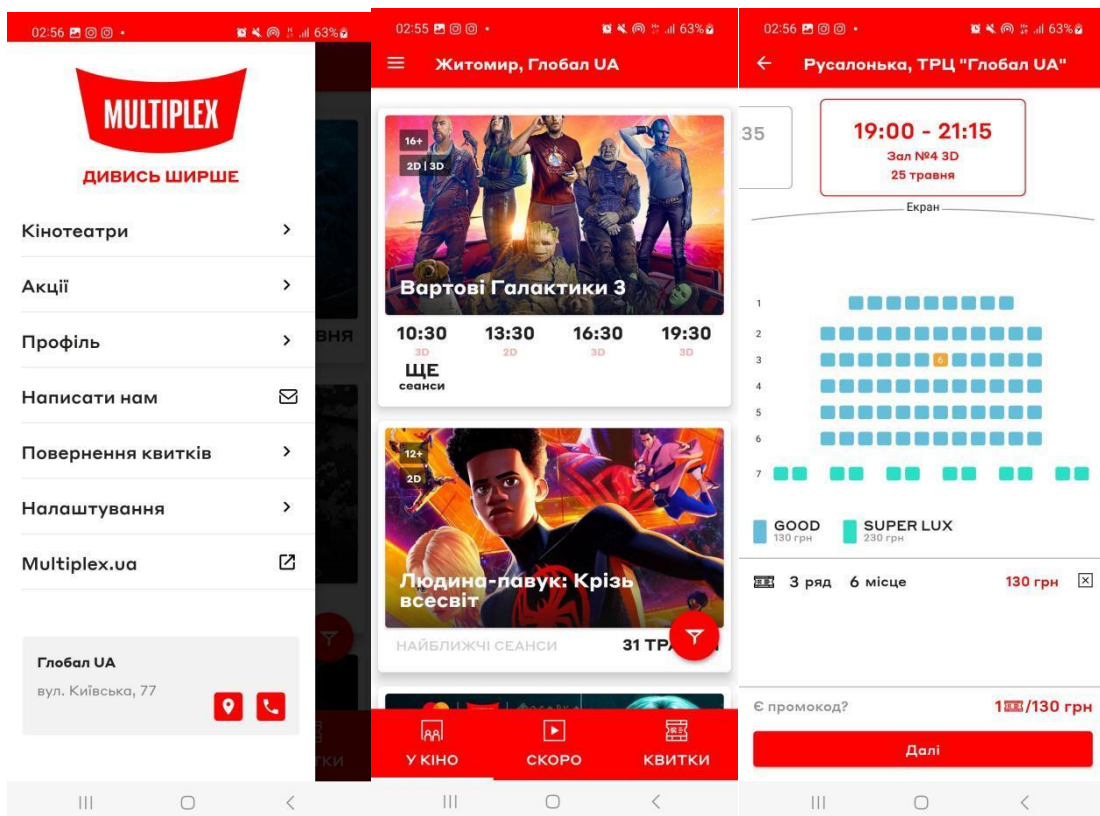


Рис. 1. 9 – Інтерфейс веб-додатку «Multiplex»

Таблиця 1.1 – Аналіз web-застосунку «Multiplex»

Назва характеристики	Значення характеристики
Назва	Multiplex
Відомі особливості реалізації	Frontend використовує мову

	програмування JavaScript разом з бібліотекою jQuery. Backend генерує сторінки та відсилає їх браузеру. Використовується AJAX[30].
Функції	-Дивіться афіші фільмів; <input type="checkbox"/> -Купуйте квитки онлайн; <input type="checkbox"/> -Дивіться деталі фільму; -Дивіться фільми високої якості, завдяки унікальним технологіям (IMAX, 4DX, 3D, Xpand та ін.)
Обов'язковість реєстрації	ні
Макети залів	має
Переваги	-автоматично визначає місто користувача; <input type="checkbox"/> -надає можливість купувати квитки онлайн; <input type="checkbox"/> -має лаконічний та сучасний дизайн.
Недоліки	-інформація про куплені місця не оновлюється в режимі реального часу; <input type="checkbox"/> – сайт погано адаптується до малого розміру екрану.

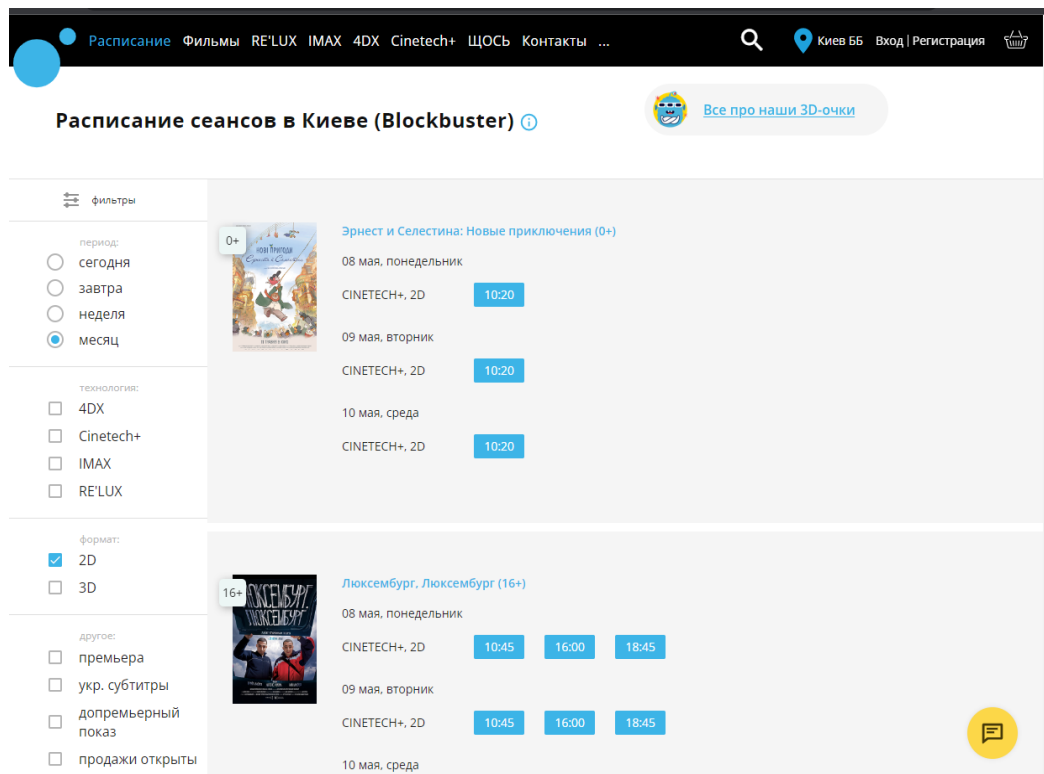


Рис. 1.10 – Інтерфейс веб-сайту мережевого кінотеатру «Планета Кіно»

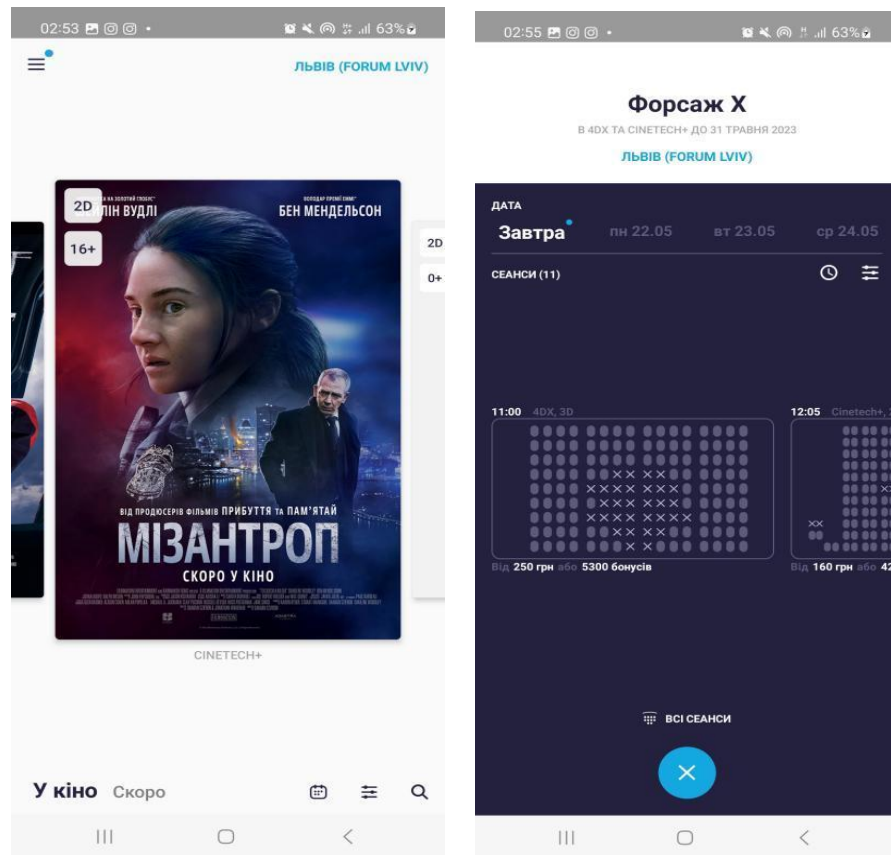


Рис.1.11 – Інтерфейс веб-додатку «Планета Кіно»

Таблиця 1.2 – Аналіз вебзастосунку «Планета Кіно»

Назва характеристики	Значення характеристики
Назва	Планета Кіно
Відомі особливості реалізації	Frontend використовує мову програмування JavaScript разом з бібліотекою jQuery. Backend переважно генерує сторінки та відсилає її браузеру, але інколи дані відправляються за методологією REST. Для комунікації використовується технологія AJAX[29].
Функції	-дивитися кіноафіші; <input type="checkbox"/> -купувати квитки онлайн; <input type="checkbox"/> -переглядати детальну інформацію про фільми; -перегляду кіно (IMAX, 4DX, Cinetech+);
Обов'язковість реєстрації	так
Макети залів	має
Переваги	- Автоматично визначати місто

	користувача; <input type="checkbox"/> н-Надати можливість купувати місця онлайн; дати вам більше можливостей для відображення планування залу та різних типів місць.
Недоліки	-Інформація про купівлю місць не оновлюватиметься в режимі реального часу; <input type="checkbox"/> - Інтерфейс незручний, не відповідає критеріям хорошої взаємодії з користувачем UX.

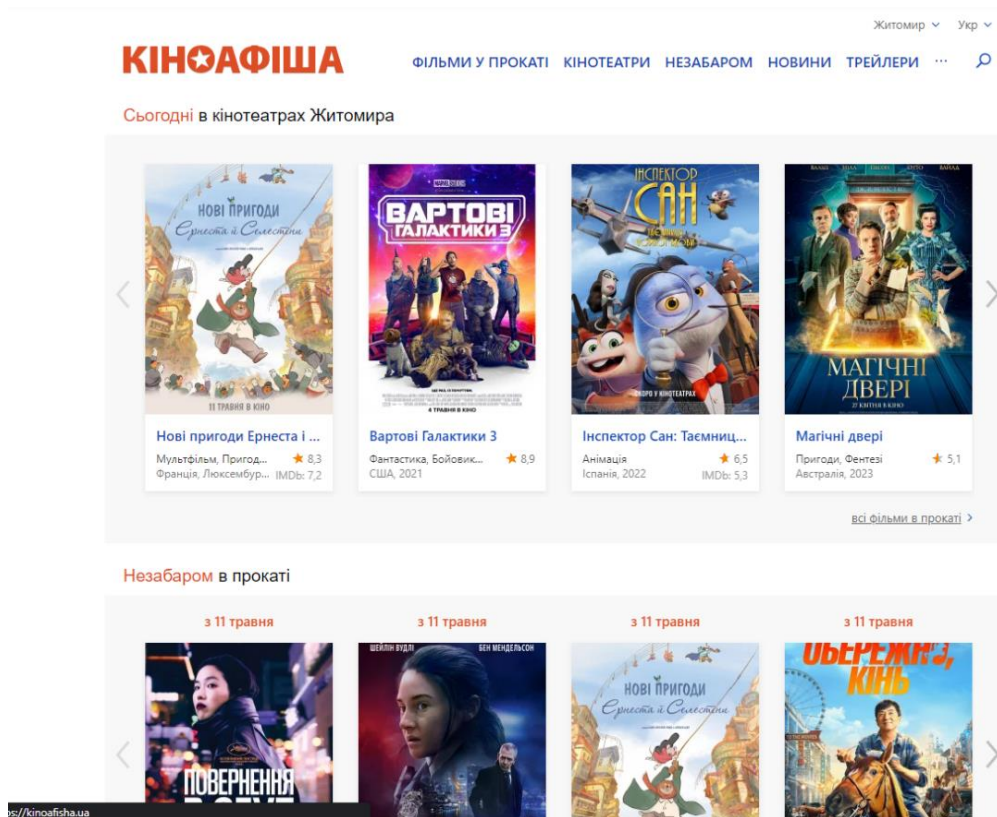


Рис.1.12– Інтерфейс застосунку «КіноАфіша»

Таблиця 1.3 – Аналіз web-застосунку «КіноАфіша»

Назва характеристики	Значення характеристики
Назва	КіноАфіша
Відомі особливості реалізації	Frontend використовує мову програмування JavaScript разом з бібліотекою jQuery. Backend генерує сторінки та відсилає її браузеру. Для комунікації використовується технологія AJAX

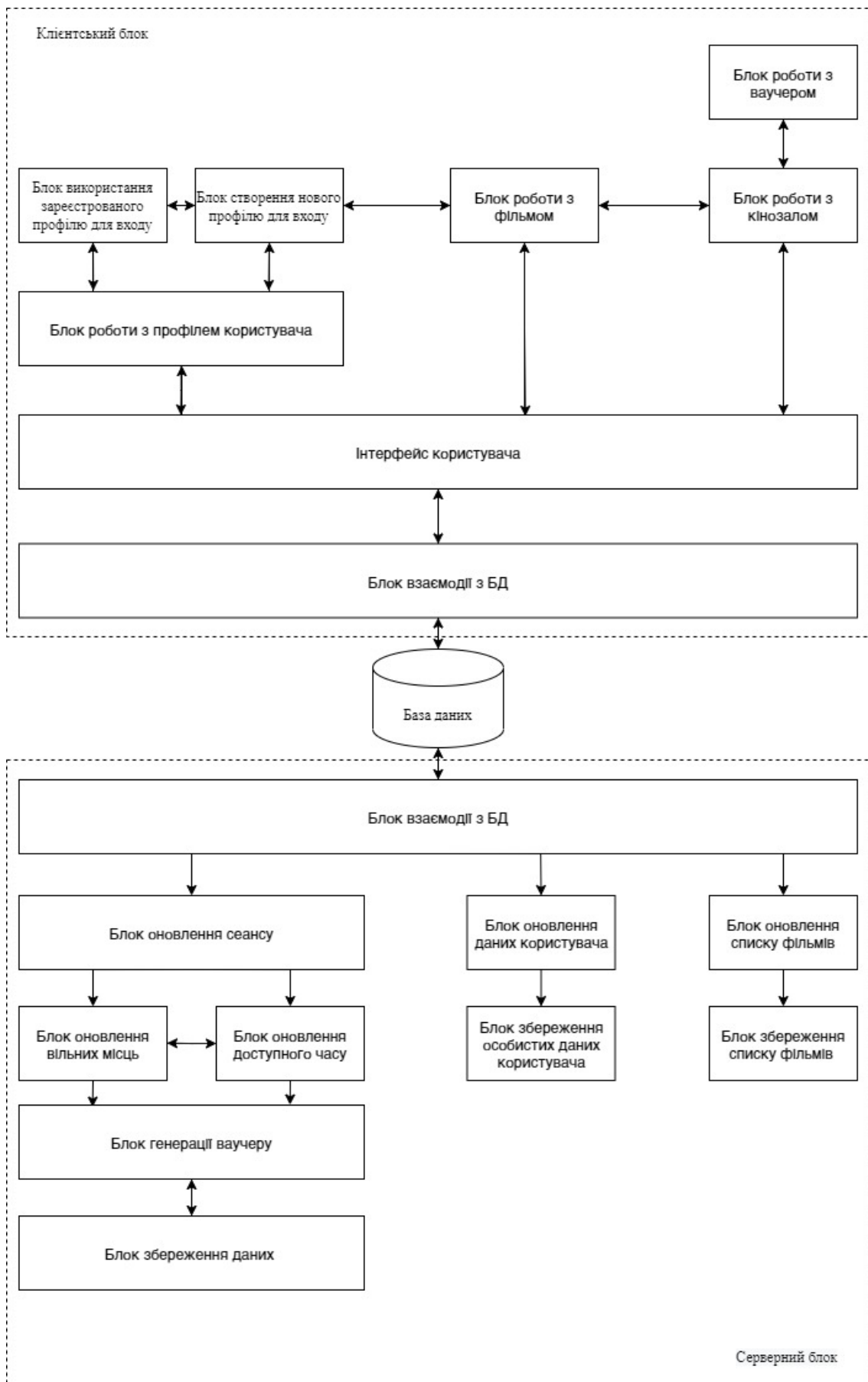
Функції	-перегляд кіноафіші; <input type="checkbox"/> -купівля квитків онлайн; <input type="checkbox"/> -коментування фільмів; <input type="checkbox"/> -відображення новин кіноіндустрії.
Обов'язковість реєстрації	ні
Макети залів	має
Переваги	-автоматично визначає місто користувача; -надає можливість купувати квитки онлайн; <input type="checkbox"/> -має потужну базу даних з світу кіно.
Недоліки	- інформація про куплені місця не оновлюється в режимі реального часу; -для покупки квитків сайт перенаправляє на зовнішній ресурс, таким чином, КіноАфіша залежить від його стану для виконання однієї з своїх основних функцій. -інформація про куплені місця не оновлюється в режимі реального часу;

Таблиця 1.4 – Опис створюваної системи

Додаток Б

Користувачі	<input type="checkbox"/> клієнт; <input type="checkbox"/> контент-менеджер; <input type="checkbox"/> адміністратор зали.
Функції	-Перегляд афіш фільмів; -Перегляд деталей фільмів; -Купівля квитків онлайн у реальному часі; -Додавання та редагування афіш фільмів; <input type="checkbox"/> -Додавання та редагування існуючих залів у кінотеатрах; -Призначення зустрічей; -Авторизація адміністраторів лобі та адміністраторів контенту у відповідних частинах аплікаційний комплекс.

Додаток В



Додаток В Рис. 2.13. Структурна схема додатку бронювання квитків

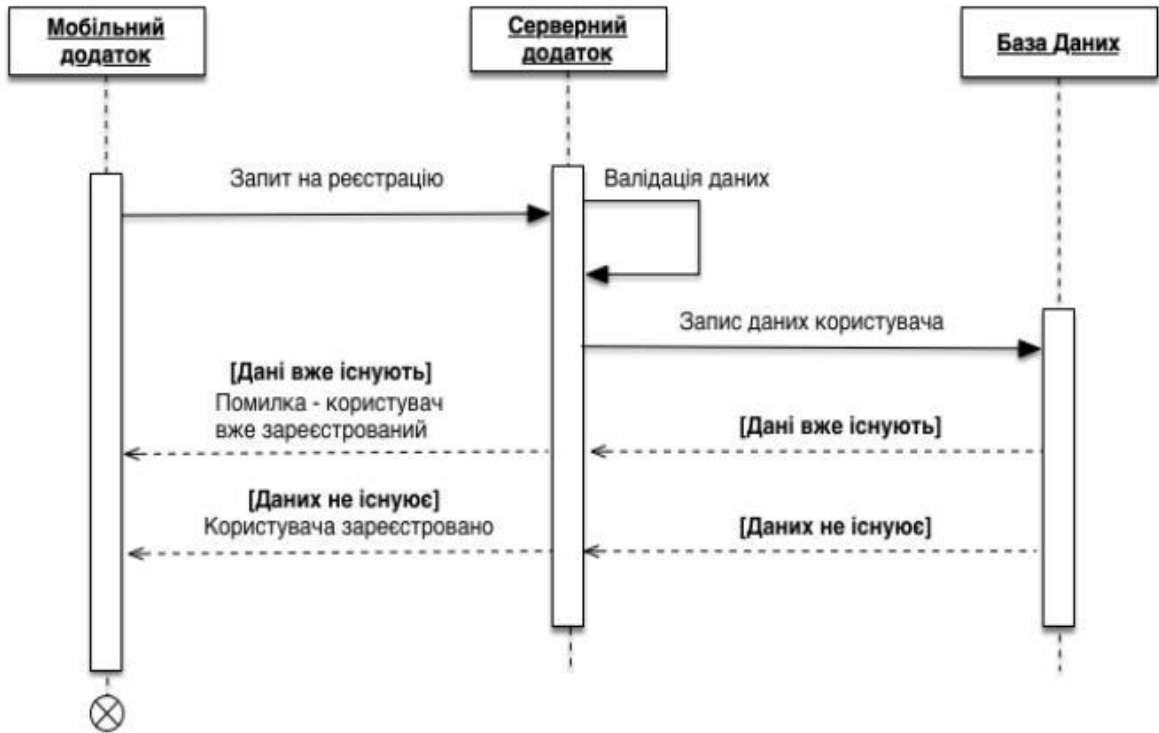


Рис. 2.14. Архітектура сервера та бізнес-логіка демонструються завдяки діаграмі послідовності. Схема реєстрації профілю користувача



Рис. 2.15. Схема авторизації профілю користувача.



Рис. 2.16. Схема отримання списку фільмів

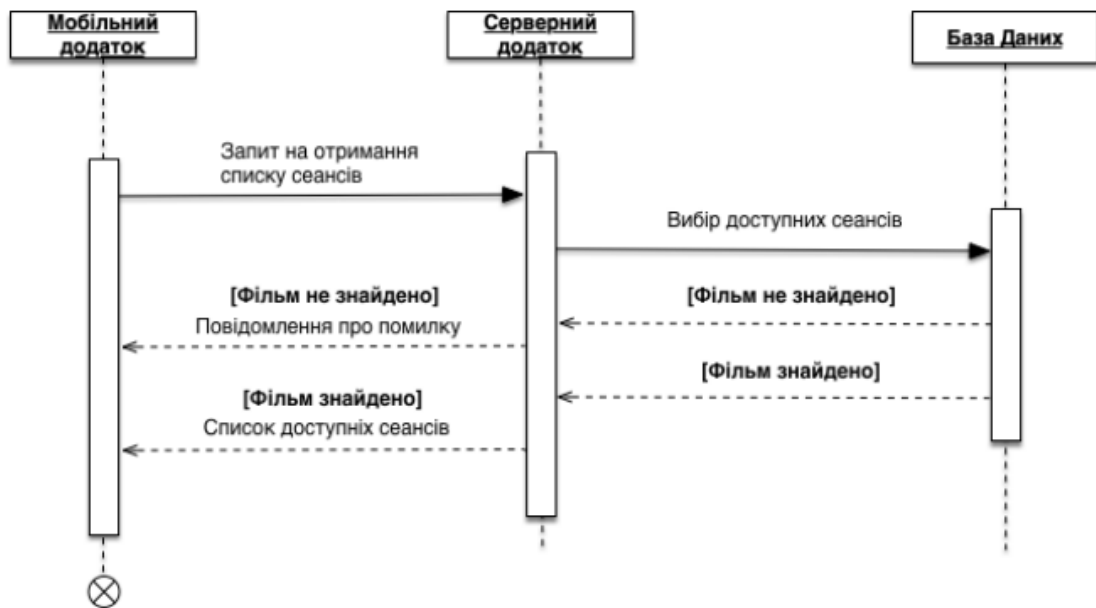


Рис. 2.17. Схема отримання списку доступних сеансів

Додаток Д

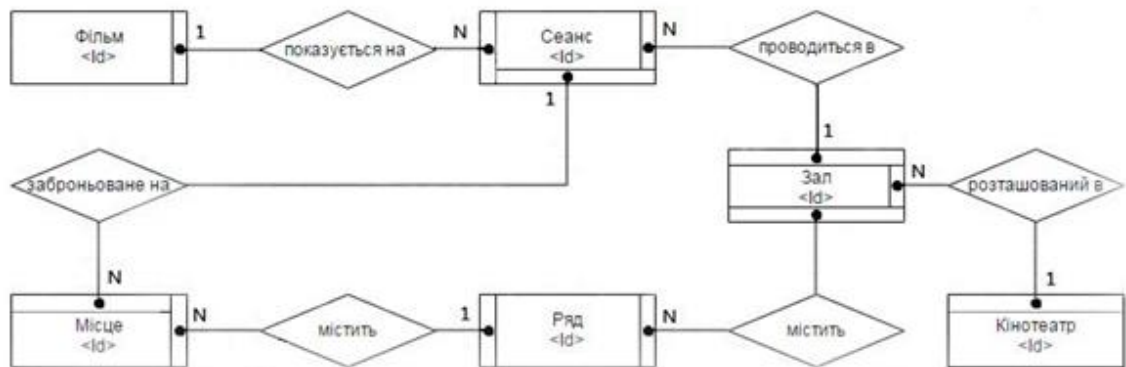


Рис.2.18.Результуюча ER-модель предметної області



Рис 2.19 –Алгоритм формування замовлень для покупуців

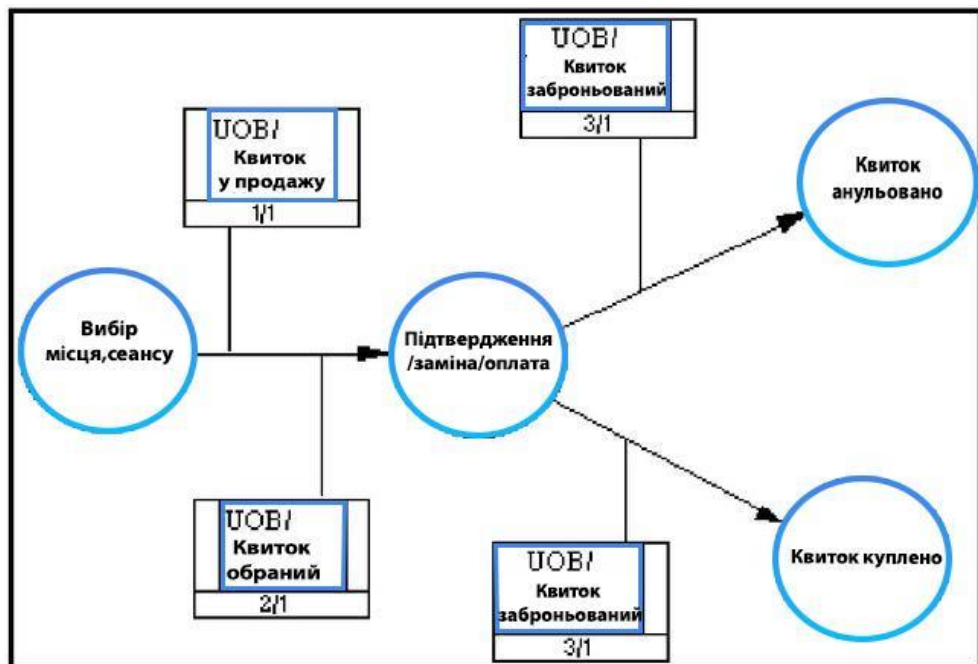


Рис.2.20. Фрагмент IDEF3-діаграма (OSTN)

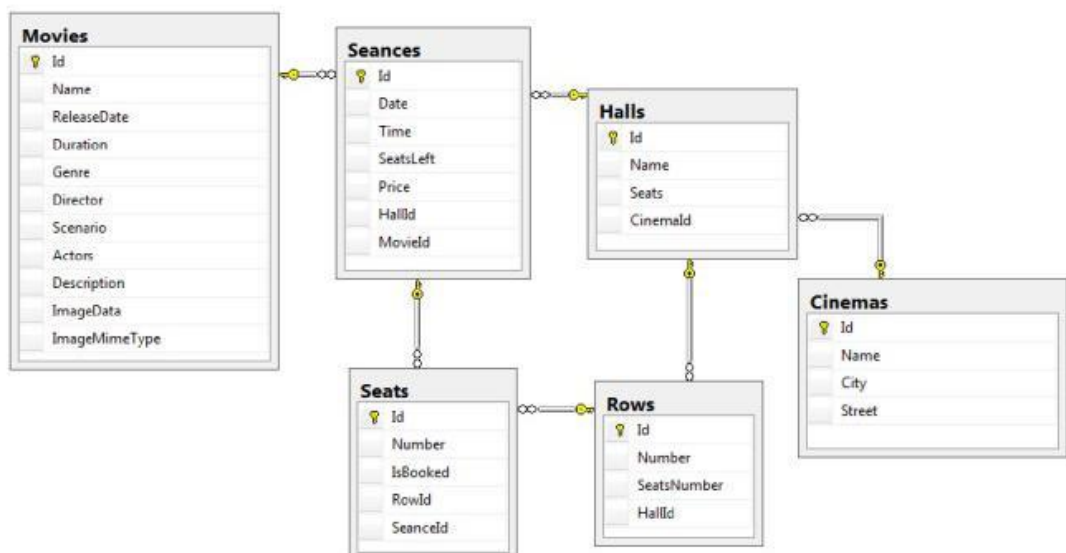


Рис.2.21. Схема даних та зв'язки між ними

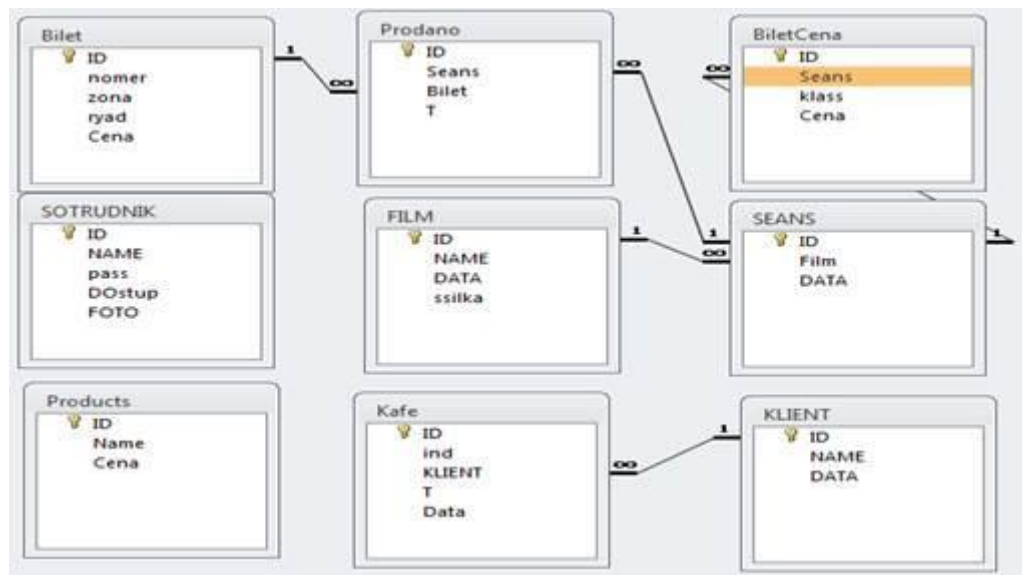


Рисунок 2.22 – Схема БД

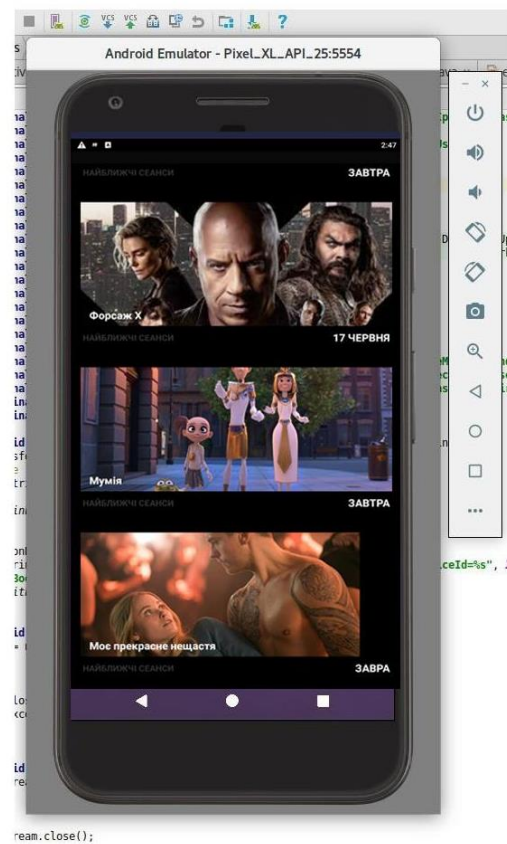


Рисунок 3.25 – Зовнішній вигляд додатку

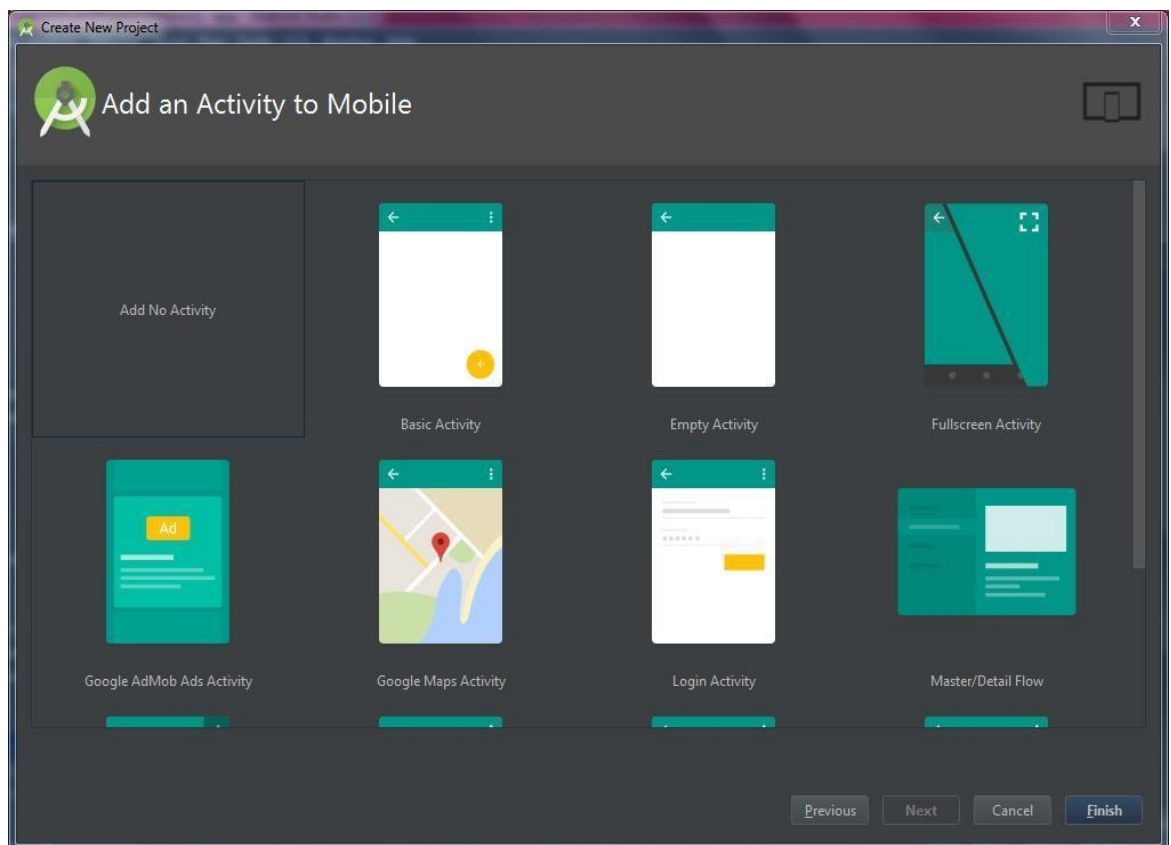


Рис 3.26 Вибір головної Activity.

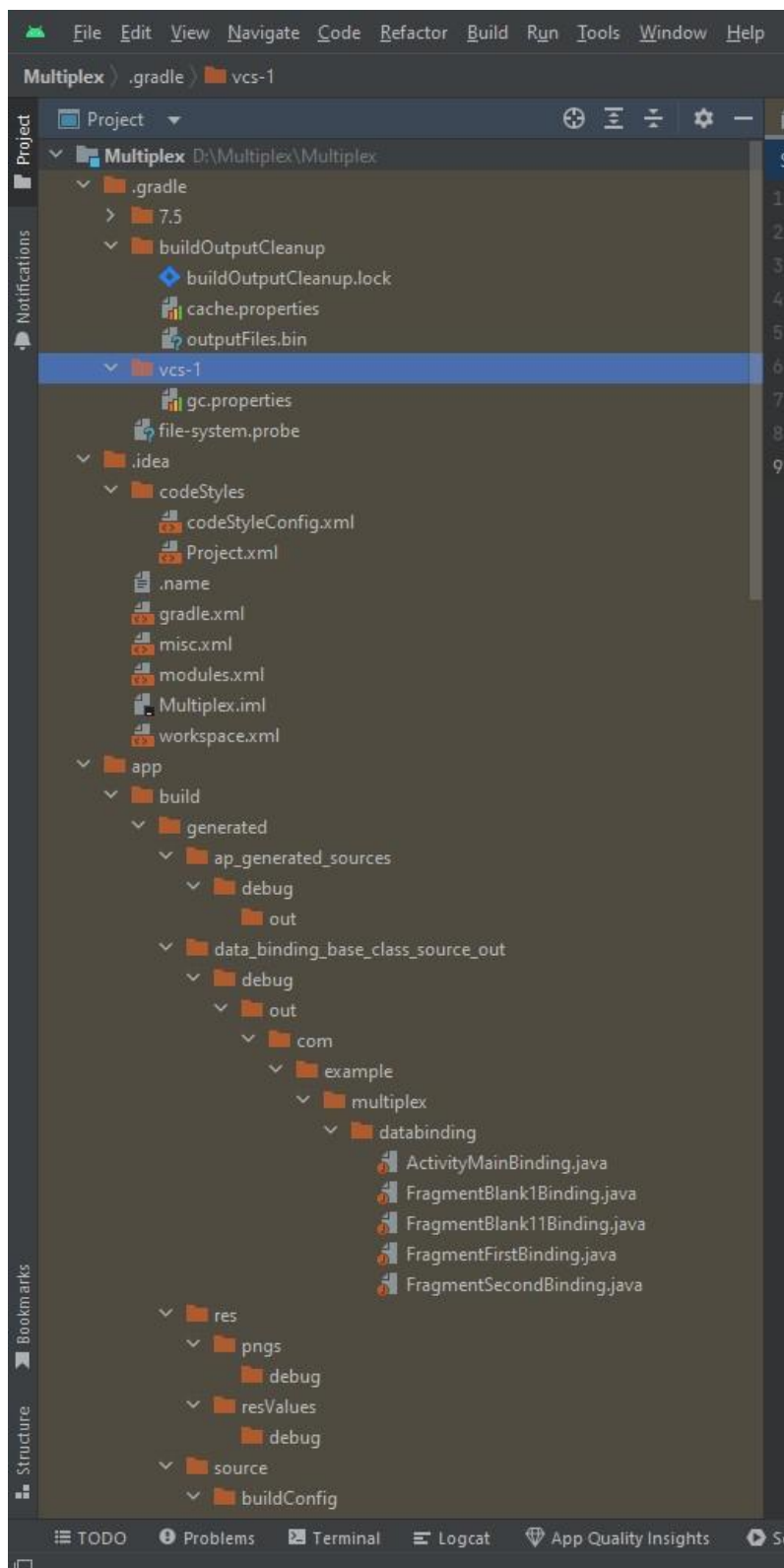


Рис 3.27 Структура Android проекта.

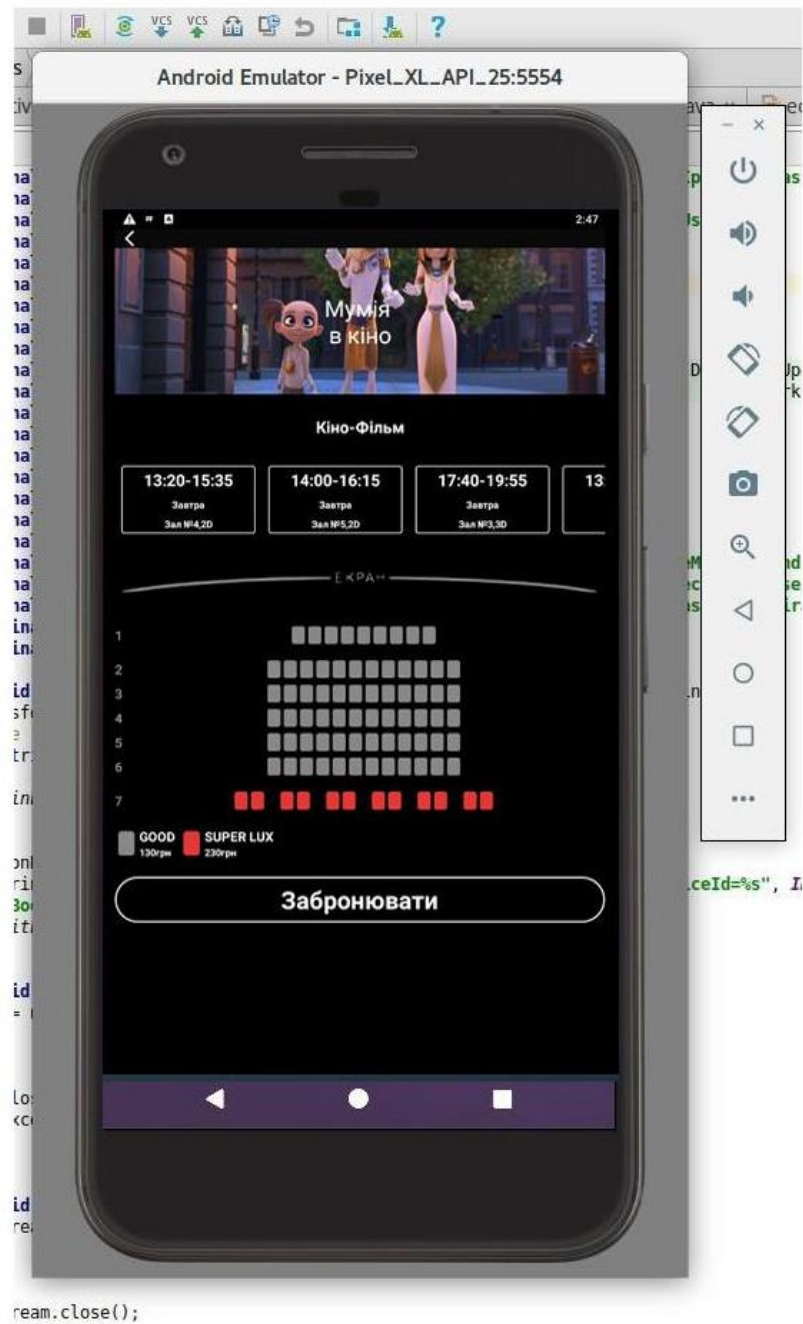
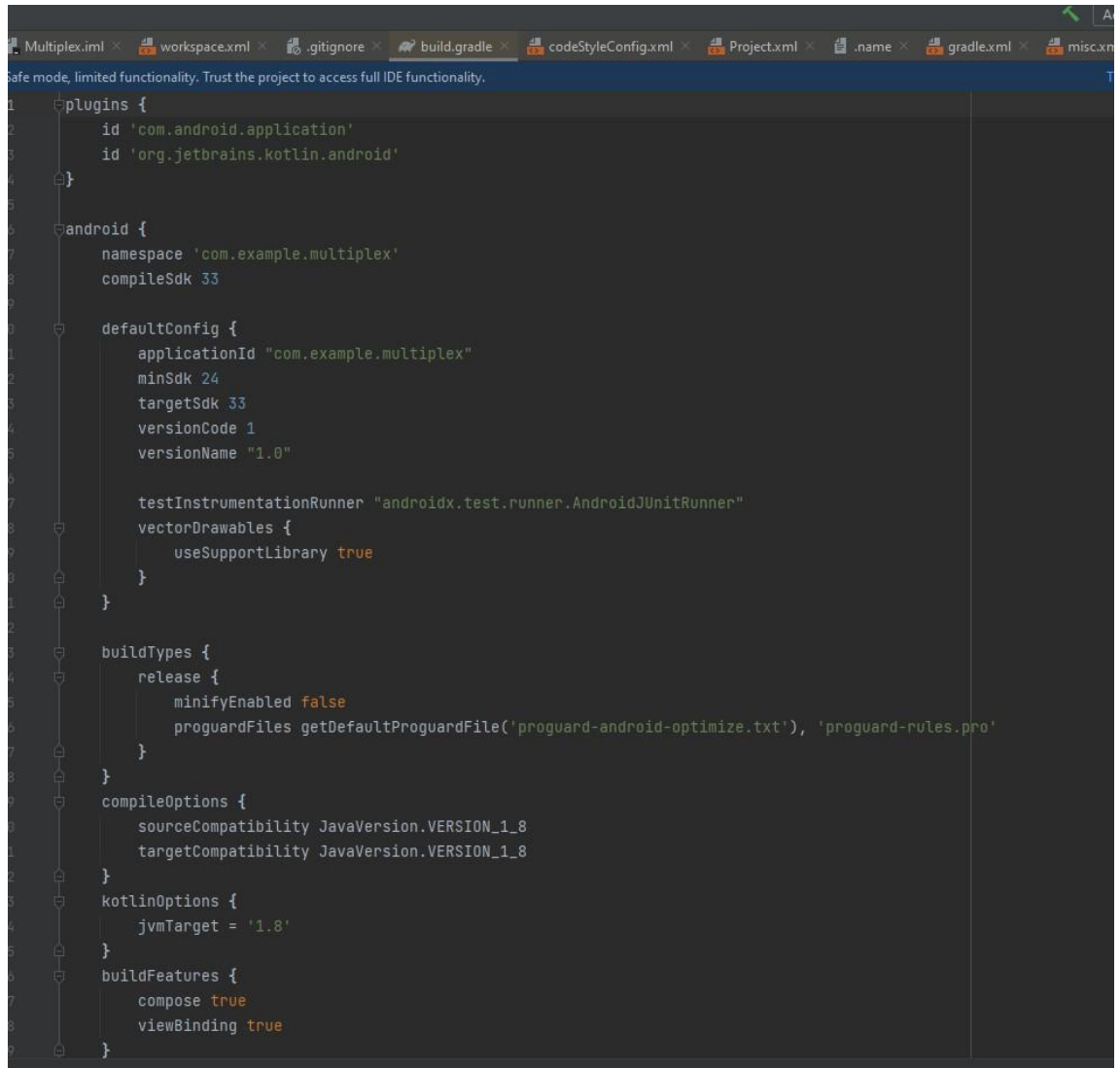


Рисунок 3.28. Віртуальний девайс в Android Studio.



```
1 plugins {  
2     id 'com.android.application'  
3     id 'org.jetbrains.kotlin.android'  
4 }  
5  
6 android {  
7     namespace 'com.example.multiplex'  
8     compileSdk 33  
9  
10    defaultConfig {  
11        applicationId "com.example.multiplex"  
12        minSdk 24  
13        targetSdk 33  
14        versionCode 1  
15        versionName "1.0"  
16  
17        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
18        vectorDrawables {  
19            useSupportLibrary true  
20        }  
21    }  
22  
23    buildTypes {  
24        release {  
25            minifyEnabled false  
26            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
27        }  
28    }  
29    compileOptions {  
30        sourceCompatibility JavaVersion.VERSION_1_8  
31        targetCompatibility JavaVersion.VERSION_1_8  
32    }  
33    kotlinOptions {  
34        jvmTarget = '1.8'  
35    }  
36    buildFeatures {  
37        compose true  
38        viewBinding true  
39    }  
40 }
```

Рисунок 3.29. коди побудови додатку в Android Studio.

```

85
86 # Determine the Java command to use to start the JVM.
87 if [ -n "$JAVA_HOME" ]; then
88     if [ -x "$JAVA_HOME/jre/sh/java" ]; then
89         # IBM's JDK on AIX uses strange locations for the executables
90         JAVACMD="$JAVA_HOME/jre/sh/java"
91     else
92         JAVACMD="$JAVA_HOME/bin/java"
93     fi
94     if [ ! -x "$JAVACMD" ]; then
95         die "ERROR: JAVA_HOME is set to an invalid directory: $JAVA_HOME"
96     fi
97     Please set the JAVA_HOME variable in your environment to match the
98     location of your Java installation."
99     fi
100 else
101     JAVACMD="java"
102     which java >/dev/null 2>&1 || die "ERROR: JAVA_HOME is not set and no 'java' command could be found in your PATH."
103 fi
104 Please set the JAVA_HOME variable in your environment to match the
105 location of your Java installation."
106 fi
107
108 # Increase the maximum file descriptors if we can.
109 if [ "$cygwin" = "false" -a "$darwin" = "false" -a "$nonstop" = "false" ]; then
110     MAX_FD_LIMIT=`ulimit -H -n`
111     if [ $? -eq 0 ]; then
112         if [ "$MAX_FD" = "maximum" -o "$MAX_FD" = "max" ]; then
113             MAX_FD="$MAX_FD_LIMIT"
114         fi
115         ulimit -n $MAX_FD
116         if [ $? -ne 0 ]; then
117             warn "Could not set maximum file descriptor limit: $MAX_FD"
118         fi
119     else
120         warn "Could not query maximum file descriptor limit: $MAX_FD_LIMIT"
121     fi
122 fi
123

```

Рисунок 3.30.продовження :коди побудови додатку в Android Studio.

```

148 # Now convert the arguments - kLudge to limit ourselves to /bin/sh
149 i=0
150 for arg in "$@" ; do
151     CHECK=`echo "$arg"|egrep -c "$SOURCE_PATTERN" -`
152     CHECK2=`echo "$arg"|egrep -c "^\-"`          ### Determine if an option
153
154     if [ $CHECK -ne 0 ] && [ $CHECK2 -eq 0 ]; then    ### Added a condition
155         eval `echo args$i`=`cygpath --path --ignore --mixed "$arg"`
156     else
157         eval `echo args$i`=`"$arg"`
158     fi
159     i=`expr $i + 1`
160 done
161 case $i in
162     0) set -- ;;
163     1) set -- "$args0" ;;
164     2) set -- "$args0" "$args1" ;;
165     3) set -- "$args0" "$args1" "$args2" ;;
166     4) set -- "$args0" "$args1" "$args2" "$args3" ;;
167     5) set -- "$args0" "$args1" "$args2" "$args3" "$args4" ;;
168     6) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" ;;
169     7) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" ;;
170     8) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" "$args7" ;;
171     9) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" "$args7" "$args8" ;;
172 esac
173 fi
174
175 # Escape application args
176 save () {
177     for i do printf %s\n "$i" | sed 's/'"/'"/g;s/'/'/;s/\/$/' \\\'/' ; done
178     echo " "
179 }
180 APP_ARGS=`save "$@"`
181
182 # Collect all arguments for the java command, following the shell quoting and substitution rules
183 eval set -- $DEFAULT_JVM_OPTS $JAVA_OPTS $GRADLE_OPTS "\-Dorg.gradle.appname=$APP_BASE_NAME\" -classpath "\$CLASSPATH\" org.gradle.
184
185 exec "$JAVACMD" "$@"
186

```

Рисунок 3.31. коди створення місць в залі.

Таблиця 3.1 .Тестування

Дія	Дія Очікуваний результат	Фактичний результат
Відкрити додаток та Виповнити вхід в профіль	Відбувається перехід до головного вікна програми	Пройдено
Кроки тесту		
Користувач обирає фільм зі списку	відображається на екрані опис фільму та кнопка "бронювання"	Пройдено
Користувач обирає сеанс	На екрані демонструється схема залу. На схемі позначені вільні та зайняті місця	Пройдено
Користувач обирає вільні місця	Місце, вибране користувачем Виділяються кольором.	Пройдено
Користувач натискає кнопку «Забронювати»	Бронювання виконане успішно, на екрані демонструється електронний код квитка.	Пройдено
Постумова		
Завершити роботу програмного модуля	Робота програмного модуля завершена	Пройдено

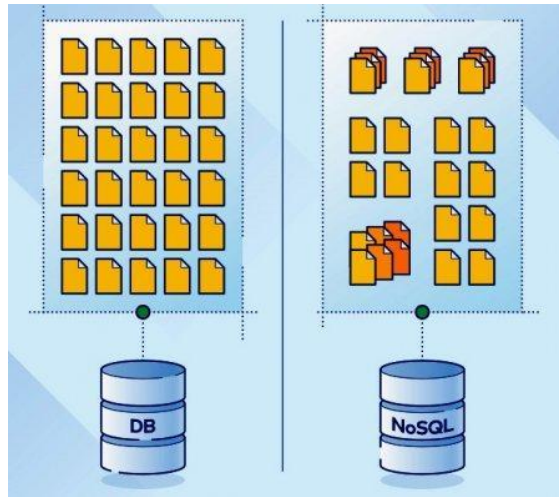


Рис 3.34. Реляційні і нереляційні бази даних. Одне ключова відмінність між базами даних NoSQL і традиційними реляційними БД полягає в тому, що перша є формою неструктурованого сховища.