

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему МОБІЛЬНИЙ ДОДАТОК КУПІВЛІ КВИТКІВ ДО
КІНОТЕАТРУ «POPCORNPALMS»
MOBILE APPLICATION TO BUY TICKETS TO THE
«POPCORNPALMS» CINEMA

Виконав(ла): студент(ка) 4 курсу, групи КНТ-151
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)
забезпечення

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

ЯКОВЛЕВА В.М.
(ПРИЗВИЩЕ та ініціали)

Керівник ШИТІКОВА О.В.
(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.
(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЯКОВЛЕВОЇ Вероніки Миколаївни
 (ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Мобільний додаток купівлі квитків до кінотеатру «PopcornPalms». Mobile Application to Buy Tickets to the «PopcornPalms» Cinema

керівник проєкту (роботи) к.т.н., ШИТИКОВА Олена Вікторівна,
 (науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 02 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Основні рішення щодо реалізації компонентів системи. 4. Керівництво програміста. 5. Керівництво оператора. 6. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-6 Основна частина	ШИТІКОВА О.В., доцент		
Нормоконтроль	КАЛІНІНА М.В., асистент		

7. Дата видачі завдання « 14 » квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

Підпис Вероніка ЯКОВЛЕВА
 (підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

Олена ШИТІКОВА
 (підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 135 с., 17 табл., 55 рис., 3 дод., 22 джерел.

КУПІВЛЯ КІНОКВИТКІВ, POPCORNPALMS, КІНОТЕАТР,
МОБІЛЬНИЙ ДОДАТОК, MONGODB COMPASS, WPF, C#.

Об'єкт дослідження – процес розроблення програмного забезпечення автоматизації функціональних можливостей для купівлі квитків до кінотеатру.

Предмет дослідження – методи та програмні засоби для купівлі квитків до кінотеатру з урахуванням сучасних вимог користувачів.

Мета роботи – підвищення швидкості та зручності персоналізованого вибору та придбання квитків до кінотеатру, а також додаткових послуг за допомогою мобільного додатка.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, мова програмування C#, технології WPF, база даних MongoDB, інтегроване середовище розробки Microsoft Visual Studio 2022, персональний IBM-сумісний комп'ютер з процесором AMD RYZEN 5, оперативною пам'яттю 8 ГБ, монітором 15.6'', клавіатурою та маніпулятором типу «миша», що функціонує під управлінням операційної системи Windows 11.

Результати. Створено мобільний додаток для кінотеатру «PopcornPalms», що дозволяє користувачам авторизуватись, переглядати афішу, шукати фільми за жанром чи назвою, переглядати трейлери, вибирати місця, замовляти попкорн та напої, здійснювати оплату та переглядати квитки. Реалізовано профіль користувача з можливістю перегляду історії замовлень, накопичених бонусів, доданих карток, розташування кінотеатру, службою підтримки та контактною інформацією про кінотеатр. Для адміністратора доступні функції додавання, видалення фільмів, обробки запитів від

користувачів.

Висновки. Розроблено програмний продукт, який автоматизує процес купівлі квитків, вибору місць та снєків, а також покращує взаємодію з клієнтами кінотеатру, знижуючи витрати часу та підвищуючи зручність користувачів.

Галузь використання – онлайн-сервіси з купівлі квитків та взаємодії з клієнтами кінотеатра.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 135 pages, 17 tables, 55 figures, 3 appendixes, 22 sources.

MOVIE TICKETING, POPCORNPALMS, CINEMA, MOBILE APPLICATION, MONGODB COMPASS, WPF, C#.

The object of research is the process of developing software for automating the functionality of purchasing cinema tickets.

The subject of the study is methods and software tools for purchasing cinema tickets, taking into account modern user requirements.

The purpose of the study is to increase the speed and convenience of personalized selection and purchase of cinema tickets, as well as additional services using a mobile application.

Materials, methods and techniques: structural and object-oriented programming, C# programming language, WPF technologies, MongoDB database, Microsoft Visual Studio 2022 integrated development environment, personal IBM-compatible computer with AMD RYZEN 5 processor, 8 GB RAM, 15.6" monitor, keyboard and mouse manipulator running Windows 11.

Results. A mobile application for the «PopcornPalms» cinema was created that allows users to log in, view the poster, search for movies by genre or title, watch trailers, select seats, order popcorn and drinks, make payments, and view tickets. A user profile has been implemented with the ability to view the history of orders, accumulated bonuses, added cards, cinema location, support service, and contact information about the cinema. The administrator can add, delete movies, and process requests from users.

Conclusions. A software product has been developed that automates the process of buying tickets, choosing seats and snacks, and improves interaction with cinema customers, reducing time costs and increasing user convenience.

The field of use is online ticket purchasing services and interaction with cinema customers.

3MICT

	С.
Перелік скорочень та умовних познач 11	11
Вступ..... 12	12
1 Аналіз предметної області..... 14	14
1.1 Особливості онлайн – замовлення квитків до кінотеатру 14	14
1.2 Опис предмета та мети дослідження 15	15
1.3 Огляд існуючих додатків для онлайн – купівлі квитків 15	15
1.3.2 Multiplex..... 17	17
1.3.3 SmartCinema..... 18	18
1.4 Формальна постановка задачі 20	20
1.5 Висновки за розділом 1 22	22
2 Матеріали і методи..... 24	24
2.1 Вибір середовища розробки програмного забезпечення 24	24
2.1.1 Обране середовище: Visual Studio..... 24	24
2.1.2 Альтернативи середовища розробки 24	24
2.2 Вибір шаблону архітектури проєктування 27	27
2.2.1 Обрана архітектура проєктування: Model-View-ViewModel (MVVM) 27	27
2.2.2 Альтернативи архітектури проєктування..... 27	27
2.3 Вибір бази даних 30	30
2.3.1 Обрана база даних: MongoDB 30	30
2.3.2 Альтернативи бази даних..... 30	30
2.4 Вибір мови програмування 33	33
2.4.1 Обрана мова програмування: C# 33	33
2.4.2 Альтернативи мови програмування 34	34
2.5 Висновки за розділом 2 37	37
3 Основні рішення щодо реалізації компонентів системи..... 38	38
3.1 Функціональні можливості програми..... 38	38
3.2 Структурна схема програми 39	39

3.3 Алгоритм функціонування розробленої програми.....	40
3.4 Вибір шаблону проектування	44
3.4.1 Діаграма класів.....	44
3.4.2 Діаграма компонентів.....	45
3.4.3 Діаграма кооперації основного процесу системи.....	47
3.5 Концептуальна модель бази даних.....	48
3.6 Взаємодія компонентів системи	53
3.6.1 Діаграма прецедентів.....	54
3.6.2 Діаграма послідовності	57
3.7 Висновки за розділом 3	59
4 Керівництво програміста.....	60
4.1 Призначення й умови застосування програми.....	60
4.2 Характеристики програми.....	61
4.3 Звертання до програми	61
4.4 Початкові та вихідні дані	62
4.5 Повідомлення	62
4.6 Висновки за розділом 4	64
5 Керівництво оператора	65
5.1 Призначення програми	65
5.2 Умови виконання програми	65
5.2.1 Апаратні вимоги програми	65
5.3 Виконання програми.....	66
5.4 Повідомлення оператору	89
5.5 Висновки за розділом 5	91
6 Експлуатація, тестування та експериментальне дослідження програми	92
6.1 Загальні положення.....	92
6.2 Підготовка до експлуатації	92
6.3 Створення чек-листа для тестування програми.....	92
6.4 Методика тестування.....	94
6.5 Результати тестування	96

	10
6.6 Висновки за розділом 6	97
Висновки	99
Перелік джерел посилання	101
Додаток А Технічне завдання	103
Додаток Б Текст програми	114
Додаток В Слайди презентації.....	125

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

APP – Application;

MDB – MongoDB;

MOB – Mobile;

PP – PopcornPalms;

VS – Visual Studio;

WPF – Windows Presentation Foundation;

ПЗ – програмне забезпечення.

ВСТУП

У сучасному світі інформаційні технології відіграють ключову роль у забезпеченні ефективного функціонування різноманітних сфер діяльності людини. Однією з таких сфер є індустрія розваг, де автоматизація процесів, зручність для користувача та доступність є основними вимогами до програмного забезпечення. Сучасні мобільні додатки змінюють підхід до взаємодії між кіноглядачами та кінотеатрами, надаючи можливість здійснювати купівлю квитків, обирати місця та додаткові послуги безпосередньо зі смартфона.

Актуальність розробки мобільного додатку для кінотеатру обумовлена зростаючою популярністю цифрових рішень для повсякденних завдань. За останні роки спостерігається значний попит на інтегровані сервіси, які об'єднують перегляд інформації про фільми, бронювання місць, придбання квитків та додаткових послуг в одному інтерфейсі. Крім того, персоналізований підхід до обслуговування, зокрема можливість збереження даних користувачів, використання бонусних програм та швидкий доступ до історії придбаних квитків, є важливими конкурентними перевагами.

Об'єктом дослідження є процес розроблення програмного забезпечення автоматизації функціональних можливостей для купівлі квитків до кінотеатру.

Предмет дослідження – методи та програмні засоби для купівлі квитків до кінотеатру з урахуванням сучасних вимог користувачів.

Метою роботи є підвищення швидкості та зручності персоналізованого вибору та придбання квитків до кінотеатру, а також додаткових послуг за допомогою мобільного додатка.

Для досягнення поставленої мети у роботі будуть вирішені наступні завдання:

- проведення аналізу існуючих мобільних додатків для кінотеатрів та визначення їх сильних і слабких сторін;

- вибір середовища та інструментів розробки;
- розробка архітектури мобільного додатка та бази даних для збереження інформації про фільми, квитки та користувачів;
- реалізація функціоналу авторизації, перегляду афіші, вибору місць, придбання квитків та додаткових послуг;
- створення інтерфейсу для адміністратора для управління фільмами та відповідей на запитання користувачів;
- тестування розробленого додатка на коректність роботи та зручність для користувачів.

Результати розробки можуть бути застосовані для впровадження у кінотеатрах, які прагнуть підвищити якість обслуговування клієнтів, скоротити час обслуговування та оптимізувати роботу персоналу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Особливості онлайн – замовлення квитків до кінотеатру

Кіноіндустрія залишається однією з провідних сфер розваг, що забезпечує значні доходи у всьому світі. Водночас стрімкий розвиток технологій висуває нові вимоги до обслуговування клієнтів, змушуючи кінотеатри модернізувати підходи до взаємодії з глядачами. Сьогодення суттєво вплинуло на ключові аспекти роботи кінотеатрів – від управління розкладом сеансів до організації купівлі квитків і надання додаткових послуг.

Процес купівлі квитків є одним із найважливіших напрямів для оптимізації. Традиційна система продажу через касу часто створює черги, незручності та витрати часу для клієнтів. Крім того, оперативна зміна інформації про кіносеанси, особливо у випадках технічних збоїв або змін у розкладі, є ще одним викликом для ефективного управління.

Впровадження мобільних додатків для автоматизації процесів обслуговування клієнтів стане оптимальним рішенням цих проблем. Вони будуть забезпечувати швидкість і зручність купівлі квитків, дозволяючи користувачам самостійно переглядати розклад, вибирати місця, бронювати та оплачувати послуги онлайн.

Предметом дослідження стануть методи та програмні засоби для купівлі квитків до кінотеатру з урахуванням сучасних вимог користувачів. Такий продукт буде спрямований на підвищення зручності для клієнтів, зменшення часу обслуговування та оптимізацію внутрішніх бізнес-процесів.

Актуальність розробки програмного забезпечення для автоматизації процесів у кінотеатрах зумовлена зростанням кількості відвідувачів і необхідністю відповідати вимогам сучасної цифрової епохи. Надійність, швидкість роботи та інтерактивність таких рішень є критично важливими для покращення якості обслуговування та підвищення ефективності операційної діяльності кінотеатрів.

1.2 Опис предмета та мети дослідження

Предмет дослідження – методи та програмні засоби для купівлі квитків до кінотеатру з урахуванням сучасних вимог користувачів. Це інформаційний продукт, який дозволить спростити процес взаємодії клієнтів із сервісом, забезпечуючи автоматизоване управління бронюванням місць, оплатою квитків і замовленням додаткових послуг. Такий інструмент також оптимізує бізнес-процеси кінотеатрів, включаючи аналіз даних і управління сеансами.

Мета дослідження полягає у підвищенні швидкості та зручності персоналізованого вибору та придбання квитків до кінотеатру, а також додаткових послуг за допомогою мобільного додатка. Впровадження такого програмного забезпечення забезпечить зручний доступ користувачів до інформації про доступні сеанси, дозволить швидко купувати квитки, а також здійснювати додаткові замовлення. У результаті розробка сприятиме підвищенню рівня обслуговування клієнтів, зменшенню витрат часу на операційні процеси та оптимізації діяльності кінотеатрів.

1.3 Огляд існуючих додатків для онлайн – купівлі квитків

Для розробки програмного забезпечення було проаналізовано основні існуючі рішення в сфері автоматизації купівлі квитків для кінотеатрів. До таких рішень належать мобільні додатки Планета Кіно, Multiplex та SmartCinema. Нижче наведено коротку інформацію, переваги та недоліки кожного з них, а також порівняння з розроблюваним додатком «PopcornPalms».

1.3.1 Планета Кіно

Планета Кіно – українська мережа кінотеатрів, відома використанням сучасних технологій, таких як IMAX, 4DX, і технологія ScreenX. Додаток

дозволяє переглядати розклад, вибирати місця, купувати квитки та отримувати доступ до ексклюзивних пропозицій [1].

Переваги:

- інтеграція з програмою лояльності, яка надає бонуси за покупки;
- зручний інтерфейс додатку з можливістю перегляду трейлерів і опису фільмів;
- розширений функціонал для групових замовлень і організації приватних показів.

Недоліки:

- немає повної інтеграції з додатковими послугами, такими як замовлення їжі чи напоїв безпосередньо через додаток;
- деякі функції програми працюють повільно в години пікового навантаження;
- немає функції швидкого доступу до останніх транзакцій або історії покупок для повторних замовлень.

На рисунку 1.1 зображено інтерфейс мобільного додатку Планета Кіно [2].

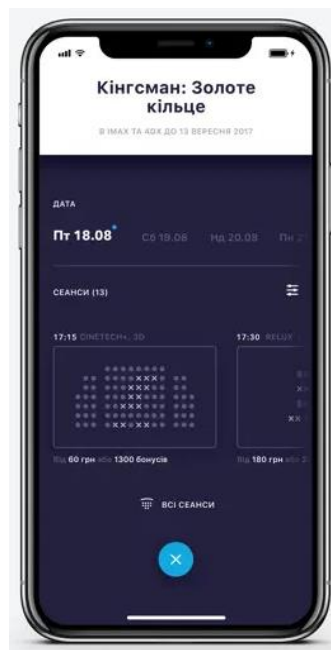


Рисунок 1.1 – Інтерфейс мобільного додатку Планета Кіно [2]

1.3.2 Multiplex

Multiplex – один із найпопулярніших сервісів купівлі квитків в Україні, що дозволяє користувачам переглядати розклад кіносеансів, вибирати місця та купувати квитки [3].

Переваги:

- оперативне оновлення інформації про доступні сеанси та зміни в розкладі;
- інтеграція з програмою лояльності для постійних клієнтів;
- можливість онлайн-замовлення та отримання електронного квитка.

Недоліки:

- додаток перевантажується під час великих прем'єр;
- обмежена підтримка додаткових функцій, таких як замовлення попкорну;
- періодичні технічні збої під час проведення платежів.

На рисунку 1.2 зображено інтерфейс мобільного додатку Multiplex [4].



Рисунок 1.2 – Інтерфейс мобільного додатку Multiplex [4]

1.3.3 SmartCinema

SmartCinema орієнтується на гнучкість і функціональність, дозволяючи не лише купувати квитки, але й переглядати трейлери, отримувати рекомендації та ділитися відгуками з іншими користувачами [5].

Переваги:

- розширений функціонал (рекомендації, відгуки);
- система персоналізованих повідомлень про новинки;
- інтеграція з платіжними системами та популярними платформами оплати.

Недоліки:

- перевантажений інтерфейс, який є складним для нових користувачів;
- відсутність можливості бронювання квитків без оплати;
- немає опції синхронізації облікового запису для одночасного використання на декількох пристроях.

На рисунку 1.3 зображено інтерфейс мобільного додатку SmartCinema [6].

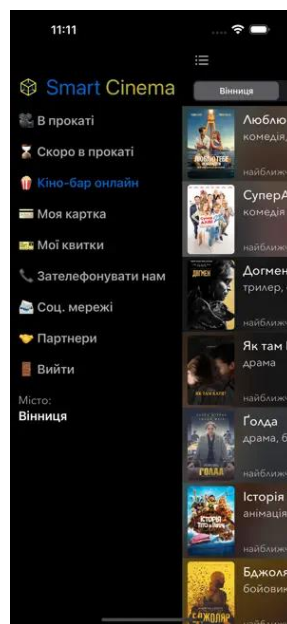


Рисунок 1.3 – Інтерфейс мобільного додатку SmartCinema [6]

У ході виконання дипломної роботи буде розроблено програмний продукт, що матиме наступні переваги порівняно з існуючими аналогами:

- можливість вибору місць в залі: інтерактивна схема залу для точного вибору бажаного місця.
- інтеграція з додатковими послугами: можливість замовлення попкорну, напоїв та інших послуг безпосередньо через додаток.
- перегляд куплених квитків: доступ до історії замовлень та квитків у зручному форматі.
- зручний та інтуїтивний інтерфейс: простота у використанні для всіх категорій користувачів.

У таблиці 1.1 зображено порівняння вже існуючих програмних продуктів та додатку, який буде розроблений в ході виконання роботи.

Таблиця 1.1 – Порівняння вже існуючих програмних продуктів та додатку, що розробляється

Критерії	Планета Кіно	Multiplex	Smart Cinema	PopcornPalms
Інтерактивний вибір місць	+	+	–	+
Інтеграція з додатковими послугами	–	–	–	+
Перегляд придбаних квитків	–	–	–	+
Програма лояльності	+	–	+	+
Доступний інтерфейс	+	+	–	+
Трейлери та детальна інформація про фільми	+	+	–	+

Таким чином, у ході аналізу аналогів програмного забезпечення для кінотеатрів було розглянуто низку популярних додатків, їхні характеристики

та функціональні можливості. В результаті визначено ключові переваги та недоліки кожного з них, що дозволило сформулювати основні вимоги до розробки мобільного додатку «PopcornPalms». Цей додаток спрямований на усунення недоліків існуючих аналогів та впровадження нових функціональних можливостей для забезпечення максимального комфорту користувачів і оптимізації процесів обслуговування кінотеатрів.

1.4 Формальна постановка задачі

У рамках розробки програмного продукту (додатку) для автоматизації процесу замовлення квитків до кінотеатру, необхідно визначити вхідні та вихідні дані, бажані результати, критерії оцінювання якості роботи додатку та обмеження, що будуть враховані під час реалізації.

Вхідні дані:

- логін та пароль для авторизації користувача в системі;
- особисті дані користувача: ім'я, прізвище, контактна інформація.

Вихідні дані:

- назва фільму, жанр, тривалість, рік випуску, режисер, актори та інші деталі;
- опис фільму, постер (для візуального представлення), рейтинг фільму та час показу, трейлер (для детального представлення фільму);
- наявність фільму у розкладі кіносеансів з точними даними про дати та часи сеансів.

Дані про місця у залі:

- статус місць у залі для кожного сеансу (вільне, заброньоване, продане);
- можливість вибору місць за допомогою інтерактивної карти залу.

Додаткові послуги:

- дані про доступні послуги кінотеатру, такі як попкорн, напої;
- ціни на додаткові послуги та можливість їх замовлення разом із

квитками;

- система лояльності накопичення бонусів (за кожен придбаний квиток користувачу надходить 10% бонусів від суми квитка).

Бажані результати:

- перегляд доступних фільмів та інформації про сеанси;
- можливість бронювання та покупки квитків, а також вибору місць у залі;
- інтерфейс для замовлення додаткових послуг, таких як попкорн та напої, з автоматичним оновленням статусу замовлення;
- додавати, редагувати та видаляти фільми та анонси з бази даних;
- редагувати та оновлювати інформацію про сеанси, доступні місця та змінювати ціни;
- взаємодія з користувачами, надання відповідей на запитання користувачів;
- підтримка електронних квитків із qr-кодами для перевірки вхідних квитків.

Критерії оцінювання якості результатів:

- додаток має працювати з мінімальною затримкою, забезпечуючи швидке завантаження та обробку запитів користувачів (перегляд фільмів, бронювання місць, оформлення замовлень);
- інтерфейс додатку повинен бути інтуїтивно зрозумілим, з чітким поділом функцій та простотою в навігації, що дозволить користувачам легко здійснювати операції без потреби в додаткових інструкціях;
- додаток має забезпечувати безпечний процес оплати квитків, з перевіркою правильності введених даних та успішною обробкою платіжних транзакцій;
- забезпечення стабільної роботи програми в умовах високих навантажень, наприклад, під час проведення акцій чи великих прем'єр.

Обмеження:

- додаток повинен бути сумісним з мобільними операційними

системами Android та IOS, що дозволяє забезпечити доступність для більшості користувачів;

- у якості бази даних для зберігання інформації про фільми, сеанси, користувачів та замовлення буде обрано MongoDB, що дозволить ефективно зберігати та обробляти великі обсяги структурованої та неструктурованої інформації.

Ці етапи постановки задачі будуть основою для подальшої розробки додатку, який повинен забезпечити безперебійну, швидку та зручну взаємодію користувачів з платформою «PopcornPalms».

1.5 Висновки за розділом 1

У першому розділі було здійснено детальний аналіз предметної області та сформульовано основні аспекти дослідження, що стосуються автоматизації процесів замовлення квитків до кінотеатру.

Аналіз предметної області дозволив визначити важливість цифровізації в кіноіндустрії. Мобільні додатки для кінотеатрів стають основними інструментами для задоволення потреб глядачів. Вони дозволяють значно полегшити процес вибору фільму, бронювання місць і замовлення квитків, а також інтегрують додаткові послуги, такі як замовлення попкорну та напоїв. Це дозволяє покращити досвід користувачів і оптимізувати внутрішні процеси кінотеатрів.

Опис предмета та мети дослідження чітко окреслили, що основною метою є підвищення швидкості та зручності персоналізованого вибору та придбання квитків до кінотеатру, а також додаткових послуг за допомогою мобільного додатка. Цей додаток має забезпечити зручність у виборі фільмів, бронюванні місць і замовлення квитків, а також дозволяти користувачам замовляти додаткові послуги через платформу.

Аналіз і порівняльна характеристика існуючих методів і засобів вирішення задачі показав, що на ринку вже існують популярні мобільні

додатки, такі як Планета Кіно, Multiplex та SmartCinema. Проте жоден з цих додатків не відповідає всім сучасним вимогам, таким як повна інтеграція з додатковими послугами, зручність користувацького інтерфейсу та ефективне управління піковими навантаженнями. Це створює простір для покращення функціональності в рамках нового програмного продукту.

Формальна постановка задачі визначила основні вимоги до майбутнього додатку. Важливими аспектами є інтеграція з додатковими послугами, забезпечення зручного інтерфейсу, стабільна робота під час пікових навантажень і можливість ефективно обробляти великі обсяги даних. Урахування цих факторів дозволить створити програмне забезпечення, яке відповідатиме вимогам сучасних користувачів і забезпечить конкурентні переваги для кінотеатрів.

Таким чином, перший розділ роботи визначив основні напрямки для розробки мобільного додатку, що має значно покращити користувацький досвід та внутрішні процеси кінотеатрів, і забезпечити переваги в порівнянні з існуючими аналогами.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір середовища розробки програмного забезпечення

2.1.1 Обране середовище: Visual Studio

Visual Studio – це інтегроване середовище розробки (IDE), розроблене компанією Microsoft. Воно забезпечує широкий набір інструментів для створення, тестування та налагодження додатків, орієнтованих на платформи .NET. Visual Studio є стандартним вибором для розробників, які працюють із мовами програмування, такими як C#, VB.NET та інші, а також з технологіями, такими як WPF і WinForms [7].

Особливості Visual Studio:

- Visual Studio пропонує потужний редактор xaml, який підтримує інструменти для інтерактивного редагування ui. режим «дизайнер» дозволяє розробникам переглядати в реальному часі, як виглядає інтерфейс, і вносити зміни без необхідності запуску програми;
- вбудована підтримка git дозволяє зручно виконувати такі операції, як створення репозиторіїв, коміти, перегляд змін і виконання злиття гілок безпосередньо з інтерфейсу Visual Studio;
- Visual Studio Marketplace надає доступ до сотень розширень, які покращують продуктивність розробки. наприклад, розширення для роботи з mongodb дозволяє легко інтегрувати базу даних у додаток і керувати нею.

2.1.2 Альтернативи середовища розробки

2.1.2.1 JetBrains Rider

JetBrains Rider – це комерційна IDE, орієнтована на розробку .NET-додатків. Вона забезпечує легкий і швидкий інтерфейс, підтримує всі популярні платформи (Windows, macOS, Linux) і має потужні інструменти для аналізу коду [8].

Переваги:

- підтримка кросплатформенності;
- висока швидкість роботи навіть на великих проєктах;
- інтеграція з іншими продуктами JetBrains, наприклад ReSharper.

Недоліки:

- платна ліцензія, яка може бути недоступною для студентів;
- обмеження в роботі з деякими розширеннями для WPF.

2.1.2.2 Eclipse

Eclipse – це безкоштовне середовище розробки з відкритим вихідним кодом, яке підтримує різноманітні мови програмування. Воно більше орієнтоване на Java-розробку [9].

Переваги:

- повністю безкоштовне;
- широкий набір розширень і плагінів.

Недоліки:

- складніший інтерфейс порівняно з Visual Studio;
- менша підтримка інструментів для WPF.

2.1.2.3 NetBeans

NetBeans – це ще одне середовище розробки з відкритим кодом, яке спеціалізується на Java-розробці. Хоча воно може бути використане для C# через плагіни, основна увага приділяється JavaFX і Swing [10].

Переваги:

- простота роботи з JavaFX;
- підтримка розширень для базових операцій із C#.

Недоліки:

- обмежена підтримка WPF;
- повільніше працює з великими проєктами.

У таблиці 2.1 зображено порівняння середовищ розробки.

Таблиця 2.1 – Порівняння середовищ розробки

Критерії	JetBrains Rider	Eclipse	NetBeans	Visual Studio
Підтримка WPF	+	—	—	+
Зручна робота з XAML	+	+	—	+
Підтримка .NET	+	—	—	+
Робота з базами даних	+	—	—	+
Підтримка багатомовності	+	—	+	+
Інтеграція з іншими технологіями .NET	+	—	—	+
Простота використання	—	—	—	+
Розширюваність через плагіни	+	—	+	+

Отже, Visual Studio є найкращим вибором для розробки WPF-додатків, оскільки воно забезпечує повну підтримку технологій .NET, включаючи XAML і WPF, має інтеграцію з Git, інструменти для роботи з базами даних та широкий набір можливостей для налагодження.

JetBrains Rider також є потужним середовищем для розробки .NET-додатків, але воно більше підходить для досвідчених розробників і не пропонує такого зручного інтерфейсу для роботи з XAML, як Visual Studio.

Eclipse і NetBeans, хоч і мають хорошу багатомовну підтримку та базові інструменти налагодження, не призначені для розробки WPF-додатків. Відсутність інтеграції з технологіями .NET значно обмежує їх застосування в цьому контексті.

Таким чином, Visual Studio для розробки проєкту буде обґрунтованим і забезпечить максимально комфортний процес створення додатка.

2.2 Вибір шаблону архітектури проєктування

2.2.1 Обрана архітектура проєктування: Model-View-ViewModel (MVVM)

Model-View-ViewModel (MVVM) – це архітектурний патерн, розроблений для спрощення взаємодії між інтерфейсом користувача (UI) і бізнес-логікою програми. MVVM активно використовується в додатках на базі Windows Presentation Foundation (WPF) завдяки можливості розділення коду на незалежні частини [11].

Особливості архітектури MVVM:

- Model відповідає за управління даними та бізнес-логікою програми. Він забезпечує взаємодію з базою даних MongoDB через драйвер MongoDB C#, дозволяючи зберігати та обробляти інформацію. Model є незалежним від View і ViewModel, що забезпечує його повторне використання;
- View реалізує інтерфейс користувача, розроблений за допомогою XAML у WPF. Цей шар відповідає лише за відображення даних і виклики команд через ViewModel. Завдяки чітко визначеній ролі View не має прямого доступу до Model, що забезпечує чистоту архітектури;
- ViewModel є ключовим посередником між Model і View. Він містить логіку відображення даних і обробки подій, використовуючи прив'язку даних для двостороннього зв'язку між View та Model. Ця структура гарантує незалежність UI від бізнес-логіки.

2.2.2 Альтернативи архітектури проєктування

2.2.2.1 MVC (Model-View-Controller)

MVC – це популярна архітектура, що використовується переважно в веброботці. Вона організовує програму в три компоненти: Model, що відповідає за управління даними; View, яка відображає ці дані користувачеві;

та Controller, який обробляє дії користувача й оновлює Model і View [12].

Переваги:

- добре підходить для вебдодатків;
- забезпечує чіткий розподіл відповідальностей;
- полегшує підтримку й тестування.

Недоліки:

- складна в реалізації для настільних і мобільних додатків;
- залежність між компонентами View і Controller.

2.2.2.2 MVP (Model-View-Presenter)

MVP є варіацією MVC, яка фокусується на Presenter, що керує взаємодією між View і Model. View відповідає за відображення, але Presenter бере на себе всю логіку, звільняючи View від зайвих обов’язків [13].

Переваги:

- View залишається максимально «тонким» і легко тестованим;
- зручна для додатків із складним інтерфейсом.

Недоліки:

- Presenter може стати «вузьким місцем», якщо його функціонал стає надто великим;
- потребує більше зусиль для впровадження в простих проєктах.

2.2.2.3 Layered Architecture

Ця архітектура організовує додаток у вигляді шарів: даних, бізнес-логіки та представлення. Кожен шар взаємодіє лише зі своїм сусіднім шаром, що забезпечує високу модульність і гнучкість [14].

Переваги:

- добре масштабується для великих систем;
- чітке розділення обов’язків між шарами;

- легко інтегрується з різними інструментами.

Недоліки:

- може бути громіздкою для невеликих проєктів;
- взаємодія між шарами може впливати на продуктивність.

У таблиці 2.2 зображено порівняння архітектур проєктування.

Таблиця 2.2 – Порівняння архітектур проєктування

Критерії	MVC	MVP	Layered Architecture	MVVM
Підходить для складних UI	+	+	–	+
Простота тестування	–	+	–	+
Легка інтеграція в мобільні додатки	–	+	–	+
Відсутня складність для простих проєктів	+	–	+	+
Масштабованість	+	–	+	+

Архітектурний патерн Model-View-ViewModel (MVVM) є найбільш вдалим для розробки додатку купівлі квитків до кінотеатру, оскільки він забезпечує чітке розділення обов'язків між компонентами системи, зокрема, між інтерфейсом користувача та бізнес-логікою. Це дозволить спростити розробку, підтримку та тестування додатку.

Порівняно з альтернативами, такими як MVC та MVP, MVVM виглядає більш оптимальним для мобільних та настільних додатків, де потрібно забезпечити чітке розмежування відповідальностей, а також підтримувати масштабованість і простоту тестування. Хоча Layered Architecture має свої переваги для великих систем, її застосування може бути зайвим для проєктів середнього розміру, де MVVM забезпечує оптимальний баланс між складністю та ефективністю.

Таким чином, для розробки додатку «PopcornPalms» MVVM є ідеальним вибором, оскільки дозволить створити підтримувану, тестовану і легко розширювану систему.

2.3 Вибір бази даних

2.3.1 Обрана база даних: MongoDB

MongoDB – це документоорієнтована база даних, яка зберігає інформацію у вигляді JSON-подібних документів (BSON), що дозволяє ефективно зберігати і обробляти дані зі складною структурою. Вона є популярним вибором для сучасних веб та мобільних додатків завдяки своїй масштабованості та гнучкості [15].

Особливості бази даних MongoDB:

- MongoDB може обробляти великі об'єми даних, що є важливим для додатків з великим обсягом інформації;
- структура даних може бути змінена без значних зусиль, що дозволяє адаптуватися до змін у бізнес-логіці;
- MongoDB підтримує JSON-подібні документи, що легко інтегруються з мовами програмування, зокрема C#.

2.3.2 Альтернативи бази даних

2.3.2.1 MySQL

MySQL – це реляційна база даних, яка зберігає дані у вигляді таблиць, використовуючи SQL (Structured Query Language). Вона є одним з найбільш популярних виборів для багатьох вебдодатків завдяки стабільності та підтримці транзакцій [16].

Переваги:

- підтримка транзакцій і цілісності даних;
- ідеальна для додатків, де важливо зберігати дані в чітко

визначених схемах;

- велика кількість інструментів і документації.

Недоліки:

- зміна структури бази даних може бути складною, особливо при великих обсягах даних;
- може виникнути складність зі масштабуванням при значному зростанні обсягів даних.

2.3.2.2 PostgreSQL

PostgreSQL – це потужна об'єктно-реляційна база даних, яка також підтримує складні типи даних і розширення. Вона забезпечує високий рівень консистентності та підтримує розширені функції, такі як гео-просторові дані та індекси [17].

Переваги:

- підтримка складних запитів і об'єктно-реляційних типів даних;
- висока надійність при роботі з транзакціями;
- хороша підтримка для великих і складних систем.

Недоліки:

- може вимагати більш складної конфігурації і адміністрування в порівнянні з іншими СУБД;
- не завжди є найшвидшим варіантом для додатків з високим рівнем запису.

2.3.2.3 SQLite

SQLite – це легка реляційна база даних, яка не потребує окремого серверного програмного забезпечення. Вона зберігає всі дані в одному файлі, що робить її дуже зручною для локальних додатків і мобільних пристроїв [18].

Переваги:

- легка в налаштуванні і використанні, без необхідності в серверному програмному забезпеченні;
- ідеальна для невеликих додатків із невеликим обсягом даних;
- підходить для додатків, де зберігання даних на стороні клієнта є важливим.

Недоліки:

- не підходить для великих або складних додатків з великою кількістю даних;
- менше можливостей для роботи з складними запитами і транзакціями.

У таблиці 2.3 зображено порівняння баз даних.

Таблиця 2.3 – Порівняння баз даних

Критерії	MySQL	PostgreSQL	SQLite	MongoDB
1	2	3	4	5
Масштабованість	–	+	–	+
Гнучкість	–	+	+	+
Продуктивність	+	–	+	+
Надійність	+	+	–	+
Легкість в налаштуванні	+	–	+	+
Підтримка складних запитів	–	+	–	+

Порівнюючи бази даних, можна зробити висновок, що MongoDB є найкращим вибором для додатків, які вимагають масштабованості, гнучкості та високої продуктивності при роботі з великими обсягами даних, таких як система купівлі квитків до кінотеатру. Вона добре підходить для гнучкої структури даних і швидкої обробки інформації.

MySQL та PostgreSQL є хорошими варіантами для додатків, що потребують надійності транзакцій і стабільної структури даних, але мають

обмеження в масштабуванні.

SQLite, в свою чергу, підходить для легких, локальних додатків, але не підходить для великих чи масштабованих систем.

Таким чином, для додатку «PopcornPalms» MongoDB є найкращим вибором, враховуючи вимоги до ефективної обробки та зберігання даних.

2.4 Вибір мови програмування

2.4.1 Обрана мова програмування: C#

C# – це сучасна, об'єктно-орієнтована мова програмування, що широко використовується для розробки додатків на платформі .NET, зокрема для створення настільних додатків на базі Windows Presentation Foundation (WPF). C# є однією з найпопулярніших мов для розробки корпоративних рішень завдяки своїй простоті, високій продуктивності і потужним інструментам для розробки [19].

Особливості мови програмування C#:

- C# надає інтуїтивно зрозумілий синтаксис, який дозволяє швидко розпочати роботу, навіть для новачків у програмуванні. Це робить мову дуже популярною серед розробників різних рівнів досвіду;
- C# має велику кількість бібліотек і фреймворків, таких як WPF, ASP.NET, та Entity Framework, що значно спрощує розробку складних додатків, включаючи як настільні, так і вебрішення;
- C# підтримує сучасні можливості багатозадачності, асинхронного програмування, що дозволяє створювати високопродуктивні та масштабовані додатки.

2.4.2 Альтернативи мови програмування

2.4.2.1 Java

Java – популярна мова програмування, яка підтримує розробку кросплатформних додатків. Для створення настільних додатків на Java часто використовується бібліотека JavaFX, яка є наступником застарілої Swing [20].

Переваги:

- Java забезпечує роботу додатків на будь-якій ОС із встановленою віртуальною машиною Java (JVM);
- завдяки вбудованим механізмам управління пам'яттю та обробки виключень, Java є безпечною мовою для розробки;
- Java підтримує багатопоточність, що дозволяє створювати продуктивні й масштабовані додатки.

Недоліки:

- через використання JVM продуктивність може бути нижчою порівняно з мовами, що компілюються безпосередньо в машинний код (наприклад, C#);
- JavaFX має обмежені можливості для створення складних інтерфейсів у порівнянні з WPF;
- Java гірше взаємодіє з елементами Windows порівняно з C#.

2.4.2.2 Python

Python є гнучкою мовою програмування, яка використовується для швидкої розробки прототипів і невеликих додатків. Для створення настільних інтерфейсів зазвичай застосовуються бібліотеки Tkinter або PyQt [21].

Переваги:

- Python підходить для новачків і має читабельний синтаксис;
- Python дозволяє створювати додатки швидше завдяки великій кількості готових бібліотек;

- програми на Python можуть запускатися на Windows, macOS і Linux.

Недоліки:

- Python інтерпретується, а не компілюється, тому він повільніший за Java та C#;
- бібліотеки для створення інтерфейсів мають застарілий вигляд і обмежену функціональність;
- Python підходить більше для невеликих проєктів і погано масштабується для складних додатків.

2.4.2.3 C++

C++ – це високопродуктивна мова програмування, яка активно використовується для створення настільних додатків. Для розробки графічних інтерфейсів часто застосовується фреймворк Qt, що підтримує кросплатформеність і багатий функціонал [22].

Переваги:

- C++ є компільованою мовою, що забезпечує максимальну швидкість виконання програм;
- Qt дозволяє створювати додатки, які працюють на Windows, macOS і Linux без значних змін у коді;
- дає можливість точного управління пам'яттю й оптимізації ресурсів.

Недоліки:

- поріг входу в C++ вищий порівняно з іншими мовами, такими як Python або C#;
- потребує ретельного контролю над виділенням і звільненням ресурсів, що може призвести до помилок, таких як витоки пам'яті;
- через складність мови й необхідність додаткового налаштування середовища розробки процес створення додатків може бути довшим.

У таблиці 2.4 зображено порівняння мов програмування.

Таблиця 2.4 – Порівняння мов програмування

Критерії	Java	Python	C++	C# (WPF)
Продуктивність	+	—	+	+
Простота вивчення	+	+	—	+
Підтримка інтерфейсу (UI)	+	—	+	+
Розробка великих додатків	+	—	+	+
Підтримка архітектури MVVM	—	—	—	+
Швидкість розробки	—	+	—	+

Порівнюючи мови програмування, можна зробити висновок, що C# з WPF є найкращим вибором для створення настільних додатків, які вимагають високої продуктивності, зручного користувацького інтерфейсу та підтримки архітектури MVVM. Завдяки інтеграції з Windows API, простоті вивчення та потужним інструментам розробки, таким як Visual Studio, C# забезпечує ефективність і масштабованість для реалізації складних функцій.

Java добре підходить для кросплатформених рішень, але її продуктивність і можливості роботи з UI поступаються C#.

Python є зручним вибором для швидкого створення прототипів і невеликих додатків, проте має обмеження у продуктивності та гнучкості інтерфейсу.

C++ з Qt демонструє високу продуктивність і підтримку кросплатформеності, але потребує більше часу на розробку та має складніший синтаксис.

Таким чином, для розробки настільного додатка з сучасним дизайном, інтеграцією з базами даних і підтримкою архітектури MVVM, найкращим

вибором є C# з WPF.

2.5 Висновки за розділом 2

У процесі розробки програмного забезпечення для автоматизації купівлі квитків до кінотеатру було проведено детальний аналіз інструментів та методів, що використовуються для створення подібних систем.

Було обране середовище розробки Visual Studio, оскільки воно надає широкий спектр можливостей для роботи з мовою C# та підтримує зручну інтеграцію з базами даних і шаблонами архітектури. Розглянуто альтернативи, але Visual Studio виявилось найбільш оптимальним варіантом.

Для архітектури проєкту обрано Model-View-ViewModel (MVVM), що забезпечує розділення логіки програми, підвищує масштабованість і полегшує тестування. Також були проаналізовані інші підходи, але MVVM найбільш відповідає вимогам до зручності розробки та підтримки.

В якості сховища даних обрана база даних MongoDB, оскільки вона є гнучкою, швидкою та добре підходить для роботи з великими обсягами неструктурованих даних. Альтернативи розглянуті, але реляційні бази даних виявилися менш ефективними для цього проєкту.

Основною мовою програмування стала C#, що забезпечує високу продуктивність, сумісність із обраним середовищем розробки та підтримку необхідного функціоналу. Вивчені альтернативи, однак саме C# найкраще відповідає вимогам проєкту.

Загалом, вибір технологій та методів розробки ґрунтувався на їхній сумісності, продуктивності та зручності використання, що дозволило створити ефективну та гнучку систему для автоматизації купівлі квитків до кінотеатру.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Функціональні можливості програми

Програма буде надавати наступні функціональні можливості:

а) для користувачів:

- 1) авторизація та реєстрація нового користувача;
- 2) пошук фільму;
- 3) вибір зацікавленого фільму;
- 4) перегляд розкладу кіносеансів;
- 5) вибір місць у залі та придбання квитків;
- 6) отримання бонусів за успішне замовлення квитка;
- 7) збереження електронного квитка;
- 8) історія попередніх замовлень;
- 9) збереження банківських карт користувачів;
- 10) можливість редагування профілю;
- 11) перегляд місцезнаходження кінотеатру;
- 12) перегляд накопичених бонусів за успішні транзакції;
- 13) перехід на соціальні мережі кінотеатру зі сторінки «Contact»;
- 14) звернення по допомогу до адміністрації кінотеатру;

б) для адміністратора:

- 1) вхід до акаунту;
- 2) додавання нових фільмів та анонсів;
- 3) видалення фільмів, які завершили показ;
- 4) видалення анонсів;
- 5) надавати відповіді на поставлені запитання користувачів.

3.2 Структурна схема програми

Програма побудована за архітектурою MVVM (Model-View-ViewModel), яка передбачає розділення логіки програми, даних та інтерфейсу користувача. Основні компоненти:

а) View:

- 1) відображає інформацію користувачу;
- 2) забезпечує можливість взаємодії користувача з програмою через інтерфейс (наприклад, вибір фільмів, вибір місць у залі);
- 3) впроваджено за допомогою хaml;

б) ViewModel:

- 1) зв'язує View з Model;
- 2) містить бізнес-логіку та обробку дій користувача;
- 3) відповідає за динамічне оновлення View на основі змін у Model;

в) Model:

- 1) представляє дані програми та їх бізнес-логіку;
- 2) містить класи для обробки інформації про фільми, користувачів, замовлення, місця у залі тощо;
- 3) використовує MongoDB для зберігання даних.

На рисунку 3.1 схематично зображено архітектуру MVVM програми.

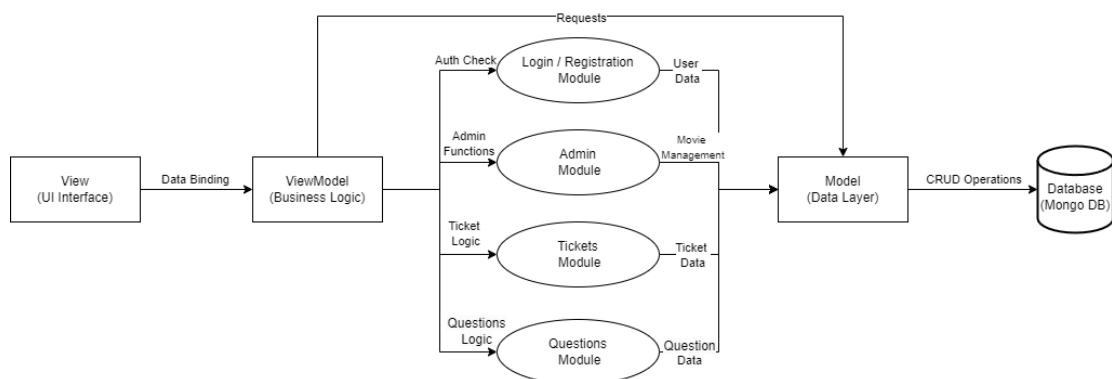


Рисунок 3.1 – Архітектура MVVM програми

3.3 Алгоритм функціонування розробленої програми

Основні алгоритми, які можуть виконуватися при роботі користувачем із розроблюваним додатком «PopcornPalms» зображені на рисунку 3.2.

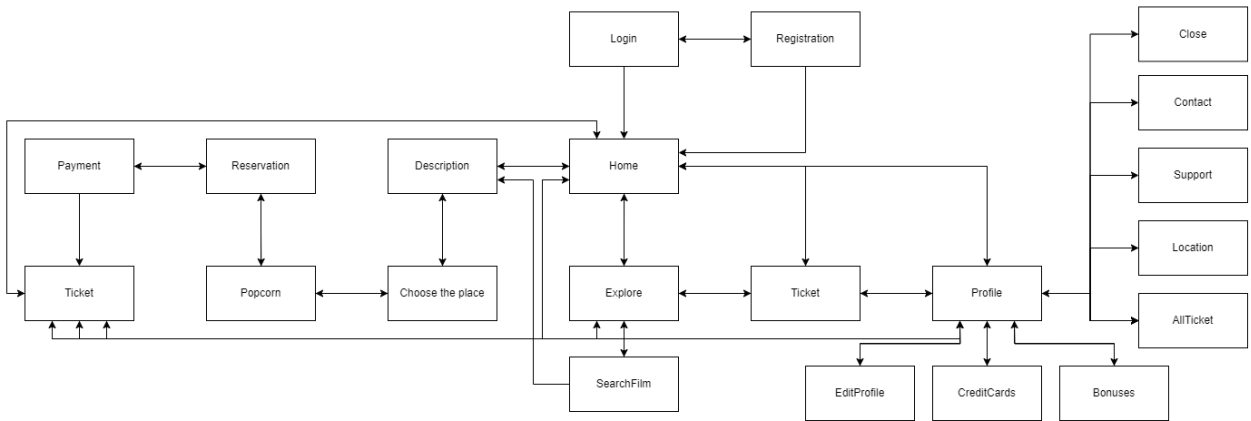


Рисунок 3.2 – Функціональна схема розроблюваного додатку «PopcornPalms»

У таблиці 3.1 наведено опис кожного етапу алгоритму користувача, які зображені на рисунку 3.2.

Таблиця 3.1 – Опис етапів алгоритмів користувача

Алгоритм	Опис
Registration	Реєстрація нового користувача за іменем та прізвищем, логіном, номером телефону та паролем.
Login	Вхід користувача за логіном та паролем.
Home	Головна, на якій відображаються фільми та анонси. З головної форми можна перейти до навігаційної панелі: Explore, Ticket, Profile.
Description	Зображення фільму, трейлер, назва, рейтинг, час відтворення, дата виходу, жанр, режисери, актори та опис фільму. З форми опису про фільм можна перейти до навігаційної панелі: Movie, Explore, Ticket, Profile.

Продовження таблиці 3.1

Алгоритм	Опис
Choose the place	Вибір дати, часу та місць у кінозалі перед придбанням квитка.
Popcorn	Вибір попкорну та напоїв для замовлення.
Reservation	Вибір способу оплати.
Payment	Введення даних банківської картки для оплати .
Ticket	Перегляд останнього придбаного квитка. З форми придбаного квитка можна перейти до навігаційної панелі: Movie, Explore, Profile.
Profile	Перехід на сторінки та інформація про користувача: редагування профілю, бонуси, квитки, підтримка, тощо.
EditProfile	Редагування профілю користувача: ім'я, прізвище та номер телефону. З форми редагування профілю можна перейти до навігаційної панелі: Movie, Explore, Ticket, Profile.
CreditCards	Перегляд збережених банківських карт користувача. З форми банківських карт можна перейти до навігаційної панелі: Movie, Explore, Ticket, Profile.
Bonuses	Відображення накопичених бонусів користувача. З форми накопичених бонусів можна перейти до навігаційної панелі: Movie, Explore, Ticket, Profile.
AllTickets	Перелік усіх квитків, придбаних користувачем. З форми придбаних квитків можна перейти до навігаційної панелі: Movie, Explore, Ticket, Profile.
Location	Карта місцезнаходження кінотеатру. З форми місцезнаходження кінотеатру можна перейти до навігаційної панелі: Movie, Explore, Ticket, Profile.

Продовження таблиці 3.1

Алгоритм	Опис
Support	З форми підтримки кінотеатру можна перейти до навігаційної панелі: Movie, Explore, Ticket, Profile.
Contact	Контактна інформація кінотеатру, включаючи адреси, соціальні мережі та телефони. З форми контактної інформації можна перейти до навігаційної панелі: Movie, Explore, Ticket, Profile.
Explore	Пошук фільмів за назвою або жанром. З форми пошуку фільму можна перейти до навігаційної панелі: Movie, Ticket, Profile.
Close	Закриття додатку.

Основні алгоритми, які можуть виконуватися при роботі адміністратора із розроблюваним додатком «PopcornPalms» зображені на рисунку 3.3.

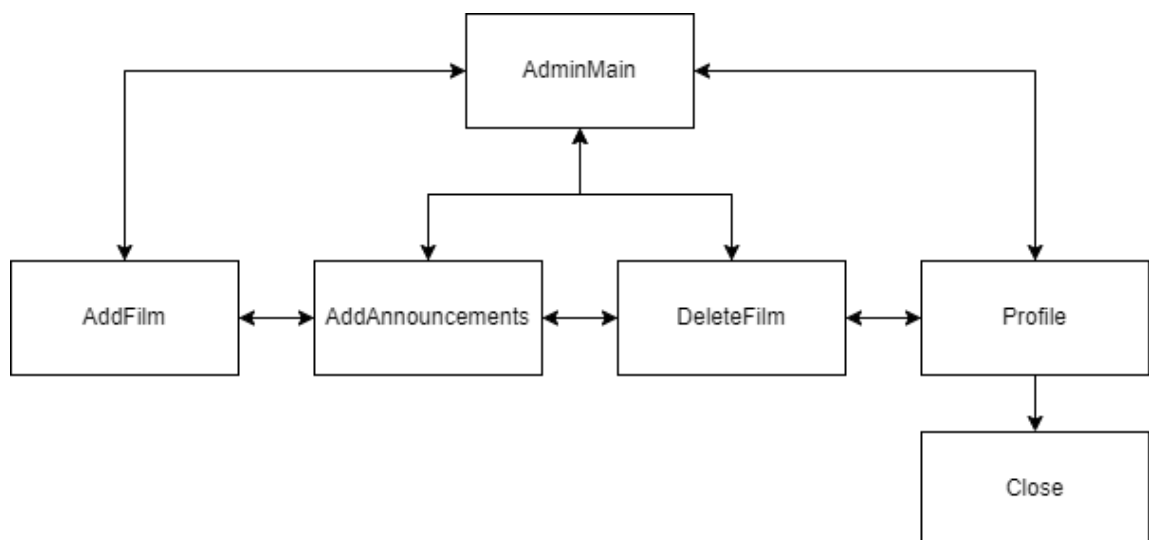


Рисунок 3.3 – Функціональна схема розроблюваного додатку «PopcornPalms»

У таблиці 3.2 наведено опис кожного етапу алгоритму адміністратора, які зображені на рисунку 3.3.

Таблиця 3.2 – Опис етапів алгоритмів адміністратора

Алгоритм	Опис
AdminMain	Перелік запитань від користувачів. З форми переліку запитань від користувачів можна перейти до навігаційної панелі: Movie (Add Film), Add (Announcements), Delete film, Profile
AddFilm	Додавання нових фільмів, які будуть доступні користувачам на головній сторінці. З форми додавання фільму можна перейти до навігаційної панелі: Add (Announcements), Delete film, Answer (Questions), Profile.
AddAnnouncements	Додавання нових анонсів, які будуть відображатися на головній сторінці користувача у вкладці Announcements. З форми додавання анонсу можна перейти до навігаційної панелі: Movie (Add Film), Delete film, Answer (Questions), Profile.
DeleteFilm	Видалення фільмів з бази даних. Видалені фільми більше не відображатимуться на головній сторінці користувачів. З форми видалення фільму можна перейти до навігаційної панелі: Movie (Add Film), Add (Announcements), Answer (Questions), Profile
Profile	Профіль адміністратора з базовою інформацією. З форми профілю адміністратора можна перейти до навігаційної панелі: Movie (Add Film), Add (Announcements), Delete film, Answer (Questions)
Close	Закриття додатку

3.4 Вибір шаблону проєктування

Розробка програмного забезпечення потребує ретельного вибору архітектурного шаблону, який забезпечує структурованість, гнучкість та легкість підтримки додатка. У цьому розділі наведено кілька ключових діаграм, які демонструють вибір та реалізацію шаблону проєктування.

3.4.1 Діаграма класів

Діаграма класів демонструє структуру програми через класи, їхні властивості, методи та зв'язки між ними. На рисунку 3.4 зображено діаграму, яка є основою об'єктно-орієнтованого програмування, оскільки вона відображає логіку та взаємодію об'єктів у системі.

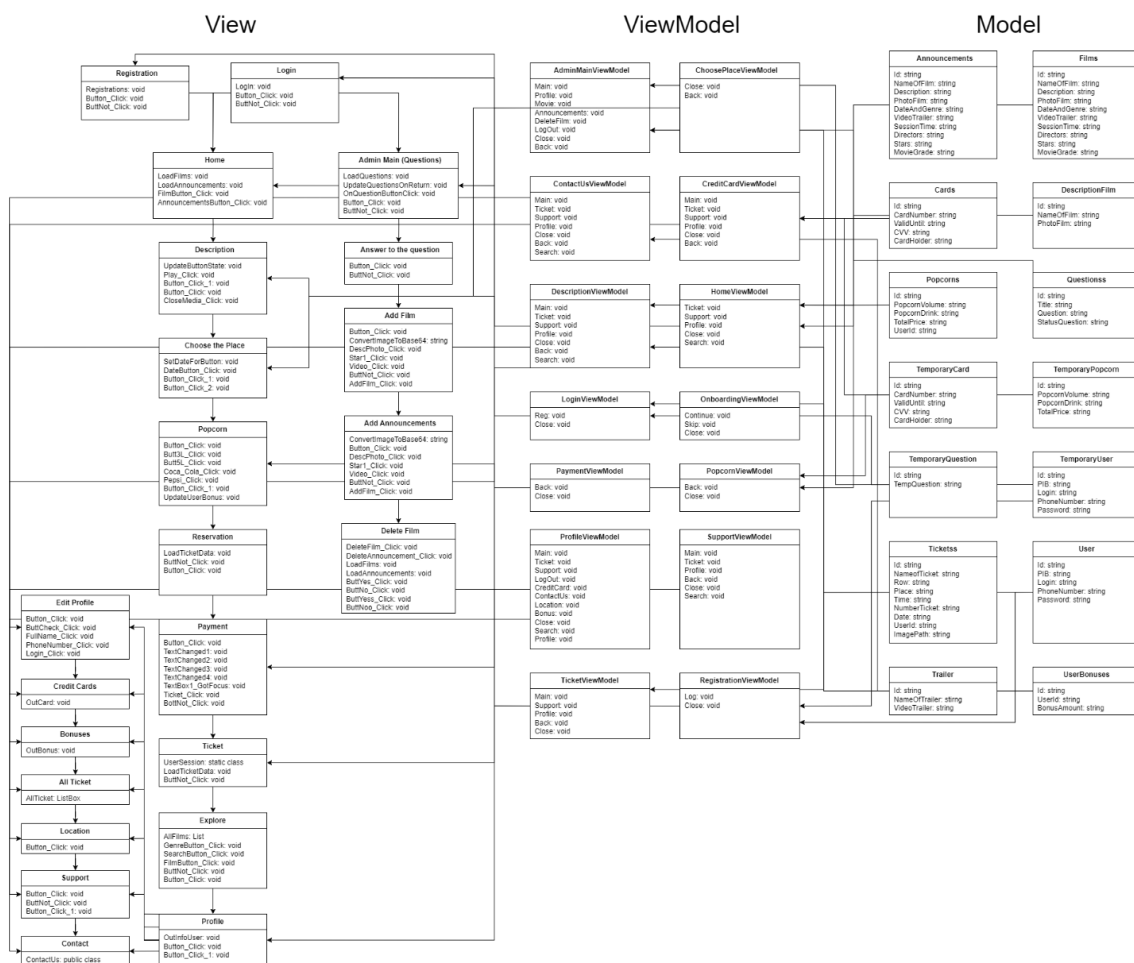


Рисунок 3.4 – Діаграма класів системи, що проєктується

На діаграмі зображені всі ключові класи, які використовуються в програмі, їх функції, змінні та залежності. Вона дозволяє деталізувати внутрішню логіку системи та служить орієнтиром для реалізації функціональності.

Опис діаграми класів:

- адміністративна частина: Класи, пов'язані з управлінням фільмами, анонсами та взаємодією з користувачами. Наприклад, AdminMain, AddFilms, AddAnnouncements, AdminAnswerQuestion реалізують функціонал для адміністратора;
- користувацька частина: Класи, які відповідають за функції для користувачів, такі як Home, Profile, Tickets, UserBonuses. Вони забезпечують взаємодію з додатком, включаючи перегляд фільмів, вибір місць, придбання квитків і накопичення бонусів;
- моделі даних: Класи на зразок Films, Announcements, Tickets, User відповідають за структуру збережених даних у базі. Вони забезпечують інтеграцію між інтерфейсом користувача та сховищем даних;
- допоміжні класи: Класи для обробки тимчасових даних (TemporaryCard, TemporaryPopcorn, TemporaryQuestion) і забезпечення зручності користувацького досвіду (Support, Contact, CreditCards).

3.4.2 Діаграма компонентів

Діаграма компонентів демонструє модульну архітектуру програми через основні компоненти, їхню структуру та взаємозв'язки між ними. На рисунку 3.5 зображено діаграму, яка є основою побудови архітектури системи.

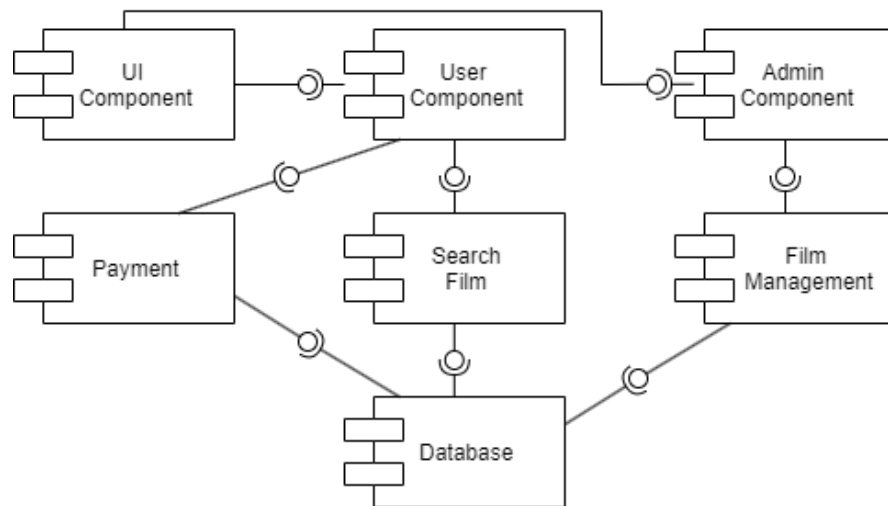


Рисунок 3.5 – Діаграма компонентів системи, що проєктується

На діаграмі компонентів зображено основні модулі системи та їх взаємодію. Кожен компонент представляє собою окрему функціональну частину програми, що відповідає за виконання певних завдань. Ось детальний опис кожного компонента:

- UI Component: відповідає за інтерфейс користувача. цей компонент забезпечує відображення всіх візуальних елементів програми, включаючи кнопки, поля вводу та відгуки на взаємодію з користувачем;
- User Component: модуль, що реалізує функціонал для користувачів програми. він відповідає за авторизацію, профілі користувачів, збережені дані (наприклад, карти чи бонуси), перегляд квитків тощо;
- Admin Component: включає функціонал для адміністрування програми. цей компонент дозволяє додавати/видаляти фільми, відповідати на запити користувачів та керувати анонсами;
- Search Film: компонент, що надає можливість пошуку фільмів за назвою, жанром або іншими критеріями. він працює з базою даних, забезпечуючи швидкий і зручний пошук;
- Film Management: відповідає за управління даними фільмів. цей модуль дозволяє додавати нові фільми, редагувати існуючі та видаляти їх з бази даних. взаємодіє з адмінським модулем та базою даних;

- Payment: реалізує функції обробки оплат. включає введення даних карток, перевірку валідності платежів та обробку фінансових транзакцій;
- Database: центральний компонент, що зберігає всі дані програми, включаючи інформацію про користувачів, фільми, анонси, транзакції тощо. забезпечує доступ до даних для інших модулів через відповідні запити.

3.4.3 Діаграма кооперації основного процесу системи

Діаграма кооперації основного процесу системи демонструє, як взаємодіють різні компоненти системи під час виконання основних функцій. На рисунку 3.6 зображено послідовність дій, що виконуються користувачем та системою, а також зв'язки між цими діями.

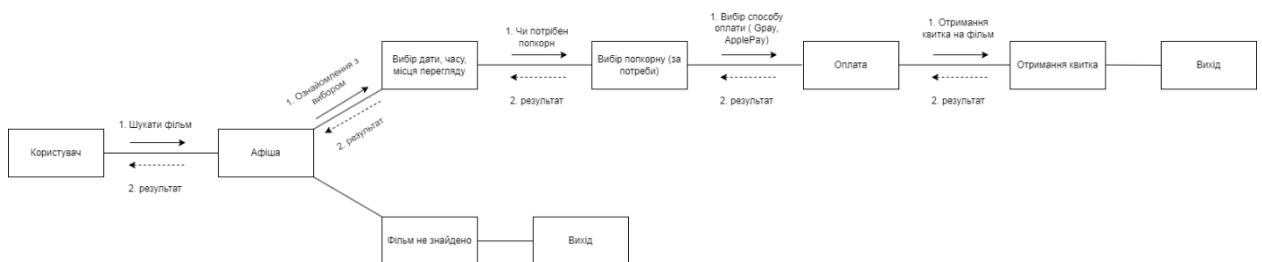


Рисунок 3.6 – Діаграма кооперації системи, що проєктується

Опис діаграми:

а) пошук фільму:

- 1) користувач розпочинає процес із запиту на пошук фільму;
- 2) система перевіряє наявність фільмів у базі даних та повертає результати;
- 3) якщо фільм знайдено, користувач переходить до наступного етапу; у разі відсутності — виводиться повідомлення "фільм не знайдено";

б) ознайомлення з афішею:

- 1) користувач переглядає доступні дати, час та місця перегляду фільму;

- 2) обирає відповідний варіант для продовження;
- в) вибір додаткових послуг (попкорн):
 - 1) система запитує, чи потрібен користувачеві попкорн;
 - 2) у разі позитивної відповіді — здійснюється вибір потрібного обсягу та типу попкорну;
- г) оплата:
 - 1) користувач обирає спосіб оплати (наприклад, через карту, Apple Pay тощо);
 - 2) система обробляє платіж та повертає результат транзакції;
- д) отримання квитка:
 - 1) після успішної оплати користувач отримує електронний квиток;
 - 2) квиток можна переглянути в особистому кабінеті;
- е) завершення:
 - 1) після завершення всіх необхідних дій користувач може вийти із системи.

3.5 Концептуальна модель бази даних

Концептуальна модель бази даних, яка зображена на рисунку 3.7 є основою для проєктування системи зберігання даних, необхідної для функціонування програми. Вона відображає основні сутності, зв'язки між ними та атрибути, що використовуються для забезпечення цілісності та ефективності роботи бази даних. На основі цієї моделі розробляється структура бази даних, яка дозволяє зберігати та обробляти дані відповідно до потреб програми.

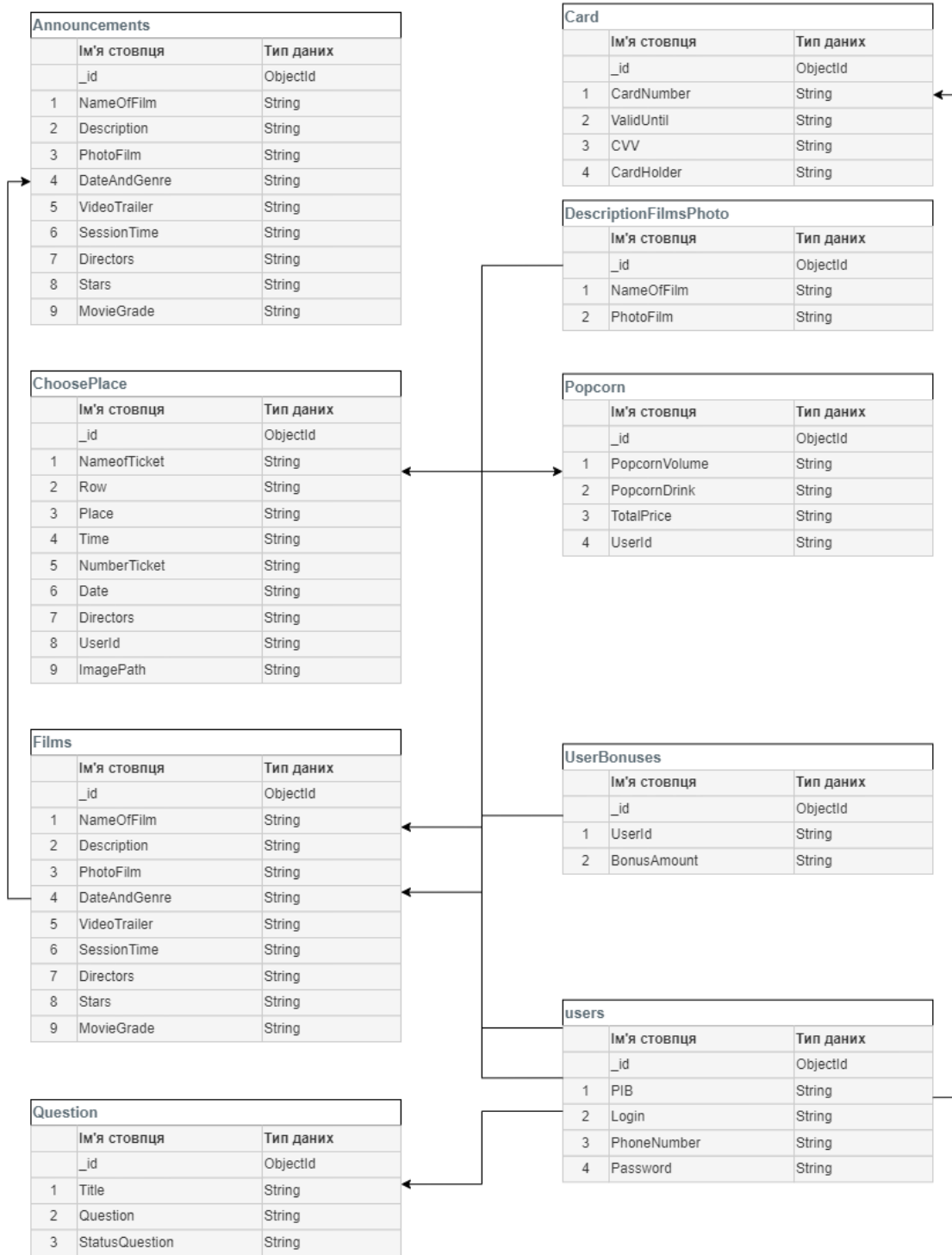


Рисунок 3.7 – Концептуальна модель бази даних

Сутність Announcements (id, NameofFilm, Description, PhotoFilm, DateAndGenre, VideoTrailer, SessionTime, Directors, Stars, MovieGrade) – Опис полів сутностей та їх типу наведено у таблиці 3.3.

Таблиця 3.3 – Опис полів сутності Announcements

№	Назва поля	Тип даних	Розмір
	_id	Objectid	—
1	NameofFilm	String	—
2	Description	String	—
3	PhotoFilm	String	—
4	DateAndGenre	String	—
5	VideoTrailer	String	—
6	SessionTime	String	—
7	Directors	String	—
8	Stars	String	—
9	MovieGrade	String	—

Сутність ChoosePlace (id, NameofTicket, Row, Place, Time, NumberTicket, Date, Directors, Userid, ImagePath) – Опис полів сутностей та їх типу наведено у таблиці 3.4.

Таблиця 3.4 – Опис полів сутності ChoosePlace

№	Назва поля	Тип даних	Розмір
	_id	Objectid	—
1	NameofTicket	String	—
2	Row	String	—
3	Place	String	—
4	Time	String	—
5	NumberTicket	String	—
6	Date	String	—
7	Directors	String	—
8	Userid	String	—
9	ImagePath	String	—

Сутність Films (id, NameOfFilm, Description, PhotoFilm, DateAndGenre, VideoTrailer, SessionTime, Directors, Stars, MovieGrade) – Опис полів сутностей та їх типу наведено у таблиці 3.5.

Таблиця 3.5 – Опис полів сутності Film

№	Назва поля	Тип даних	Розмір
	_id	Objectid	—
1	NameOfFilm	String	—
2	Description	String	—
3	PhotoFilm	String	—
4	DateAndGenre	String	—
5	VideoTrailer	String	—
6	SessionTime	String	—
7	Directors	String	—
8	Stars	String	—
9	MovieGrade	String	—

Сутність Question (id, Title, Question, StatusQuestion) – Опис полів сутностей та їх типу наведено у таблиці 3.6.

Таблиця 3.6 – Опис полів сутності Question

№	Назва поля	Тип даних	Розмір
	_id	Objectid	—
1	Title	String	—
2	Question	String	—
3	StatusQuestion	String	—

Сутність Card (id, CardNumber, ValidUntil, CVV, CardHolder) – Опис полів сутностей та їх типу наведено у таблиці 3.7.

Таблиця 3.7 – Опис полів сутності Card

№	Назва поля	Тип даних	Розмір
	_id	Objectid	—
1	CardNumber	String	—
2	ValidUntil	String	—
3	CVV	String	—
4	CardHolder	String	—

Сутність DescriptionFilmsPhoto (id, NameOfFilm, PhotoFilm) – Опис полів сутностей та їх типу наведено у таблиці 3.8.

Таблиця 3.8 – Опис полів сутності DescriptionFilmsPhoto

№	Назва поля	Тип даних	Розмір
	_id	Objectid	—
1	NameOfFilm	String	—
2	PhotoFilm	String	—

Сутність Popcorn (id, PopcornVolume, PopcornDrink, TotalPrice, Userid) – Опис полів сутностей та їх типу наведено у таблиці 3.9.

Таблиця 3.9 – Опис полів сутності Popcorn

№	Назва поля	Тип даних	Розмір
	_id	Objectid	—
1	PopcornVolume	String	—
2	PopcornDrink	String	—
3	TotalPrice	String	—
4	Userid	String	—

Сутність UserBonuses (id, Userid, BonusAmount) – Опис полів сутностей та їх типу наведено у таблиці 3.10.

Таблиця 3.10 – Опис полів сутності UserBonuses

№	Назва поля	Тип даних	Розмір
	_id	Objectid	–
1	Userid	String	–
2	BonusesAmount	String	–

Сутність users (id, PIB, Login, PhoneNumber, Password) – Опис полів сутностей та їх типу наведено у таблиці 3.11.

Таблиця 3.11 – Опис полів сутності users

№	Назва поля	Тип даних	Розмір
	_id	Objectid	–
1	PIB	String	–
2	Login	String	–
3	PhoneNumber	String	–
4	Password	String	–

3.6 Взаємодія компонентів системи

Для забезпечення ефективної роботи програмного забезпечення важливим аспектом є визначення взаємодії між основними компонентами системи. Взаємодія забезпечує виконання основних функцій програми, таких як пошук фільмів, вибір місць, оплата, обробка запитів користувачів та адміністрування контенту. Вона реалізується через комплексний підхід до організації процесів, який включає обмін даними між модулями, обробку запитів користувачів і адміністратора, а також збереження й оновлення даних у базі.

3.6.1 Діаграма прецедентів

На рисунку 3.8 зображено діаграму, яка показує основні сценарії використання системи, їхніх учасників (акторів) і зв'язки між ними. На основі діаграми прецедентів можна зрозуміти, які функції виконує система для користувачів і адміністратора.

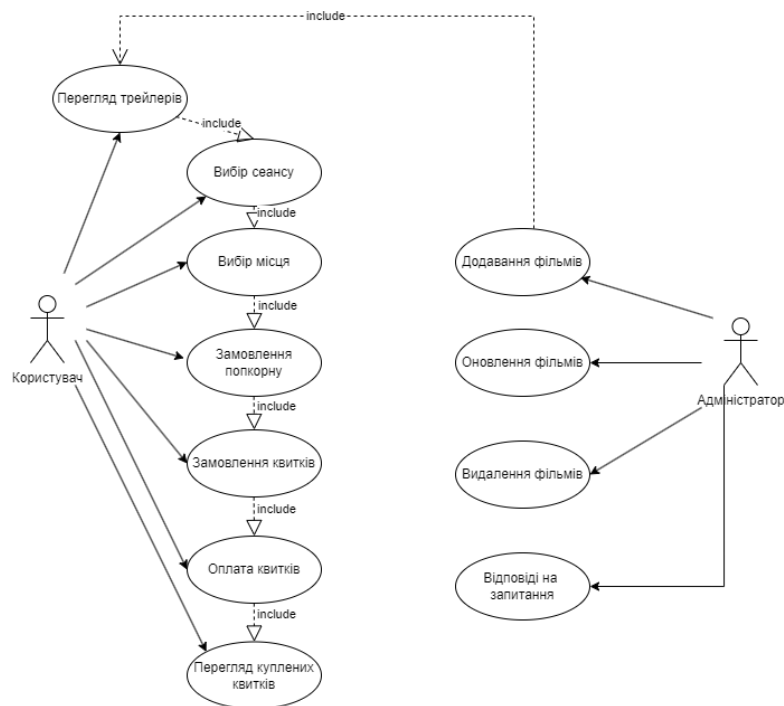


Рисунок 3.8 – Діаграма прецедентів системи, що проєктується

Опис потоків подій:

а) перегляд фільмів:

- 1) користувач відкриває застосунок;
- 2) система показує список доступних фільмів;
- 3) користувач переглядає фільми;
- 4) альтернативні потоки: 1а. Сортування фільмів за жанром або часом сеансу. 1б. Фільтрування фільмів за популярністю чи оцінками;
- 5) помилкові потоки: 1е. Немає доступних фільмів – система виводить відповідне повідомлення;

б) вибір сеансу:

- 1) користувач вибирає фільм;
 - 2) система показує список доступних сеансів;
 - 3) користувач вибирає бажаний сеанс;
 - 4) альтернативні потоки: 1а. Користувач сортує сеанси за часом;
 - 5) помилкові потоки: 1е. Немає доступних сеансів – система виводить повідомлення про відсутність сеансів;
- в) вибір місця:
- 1) після вибору сеансу користувач вибирає місце в залі через схему;
 - 2) система резервує вибране місце;
 - 3) альтернативні потоки: 1а. Користувач вибирає інший сеанс, якщо всі місця зайняті;
 - 4) помилкові потоки: 1е. Місце вже зайняте – система не дає натиснути на це місце;
- г) замовлення попкорну:
- 1) після вибору місця користувач може додати попкорн та напої до замовлення;
 - 2) система показує меню доступних товарів;
 - 3) користувач вибирає товар і за потреби додає його до замовлення;
 - 4) альтернативні потоки: 1а. Користувач може вибрати різні розміри попкорну або комбіновані пропозиції;
 - 5) помилкові потоки: 1е. Відсутність деяких товарів у наявності – система виводить відповідне повідомлення;
- д) замовлення квитків:
- 1) користувач після вибору попкорну підтверджує замовлення квитка;
 - 2) система генерує замовлення та показує підсумок;
 - 3) альтернативні потоки: 1а. Користувач може додати декілька

різних банківських карт для оплати;

4) помилкові потоки: 1е. Немає можливості додати додаткову карту, система виводить помилку додавання;

е) оплата квитків:

1) користувач переходить до оплати;

2) система показує способи оплати;

3) користувач вибирає спосіб оплати та проводить транзакцію;

4) система підтверджує оплату;

5) альтернативні потоки: 1а. Користувач вибирає інший спосіб оплати;

6) помилкові потоки: 1е. Помилка при оплаті – система виводить повідомлення та пропонує спробувати знову;

ж) перегляд куплених квитків:

1) користувач переходить до меню навігації та натискає на кнопку «ticket»;

2) система показує останній придбаний квиток;

3) альтернативні потоки: 1а. Користувач перегляд повну інформацію про квиток;

4) помилкові потоки: 1е. У користувача немає куплених квитків – система виводить відповідне повідомлення;

з) додавання фільмів (Адмін):

1) адмін переходить на форму додавання фільмів;

2) система показує форму додавання нового фільму;

3) адмін заповнює форму відповідними даними;

4) альтернативні потоки: 1а. Адмін заповнив всі поля необхідними даними, система показує повідомлення про успіх операції;

5) помилкові потоки: 1е. Помилка при додаванні фільму – система виводить повідомлення про невдалу спробу додавання та пропонує спробувати ще раз;

и) видалення фільмів (Адмін):

- 1) адмін переходить на форму видалення фільмів;
 - 2) система відображає список фільмів, які є наразі у прокаті та список анонсів;
 - 3) адмін обирає необхідні фільми та анонси, які будуть видалені;
 - 4) альтернативні потоки: 1а. Адмін обрав та натиснув на необхідний фільм або анонс для видалення, система виводить повідомлення з підтвердження операції;
 - 5) помилкові потоки: 1е. Помилка при видаленні фільму – система виводить повідомлення про невдалу спробу видалити фільм через відсутність цього фільму у прокаті;
- к) відповіді на запитання поставленні користувачами (Адмін):
- 1) адміністратор входить у свій акаунт;
 - 2) адміністратор обирає запитання на які будуть дані відповіді;
 - 3) система відкриває нову форму з полем для відповідей на запитання;
 - 4) адмін відповідає на запитання та зберігає зміни;
 - 5) альтернативні потоки: 1а. Адміністратор може відповісти на зацікавлені запитання;
 - 6) помилкові потоки: 1е. Помилка при відповіді на питання. система може змінити статус питання, та вивести його у список запитань без відповіді.

3.6.2 Діаграма послідовності

На рисунку 3.9 зображена діаграма, яка ілюструє порядок виконання дій у межах окремих сценаріїв використання. Вона відображає, як різні компоненти системи взаємодіють між собою для досягнення поставленої мети, а також порядок обміну повідомленнями між ними.

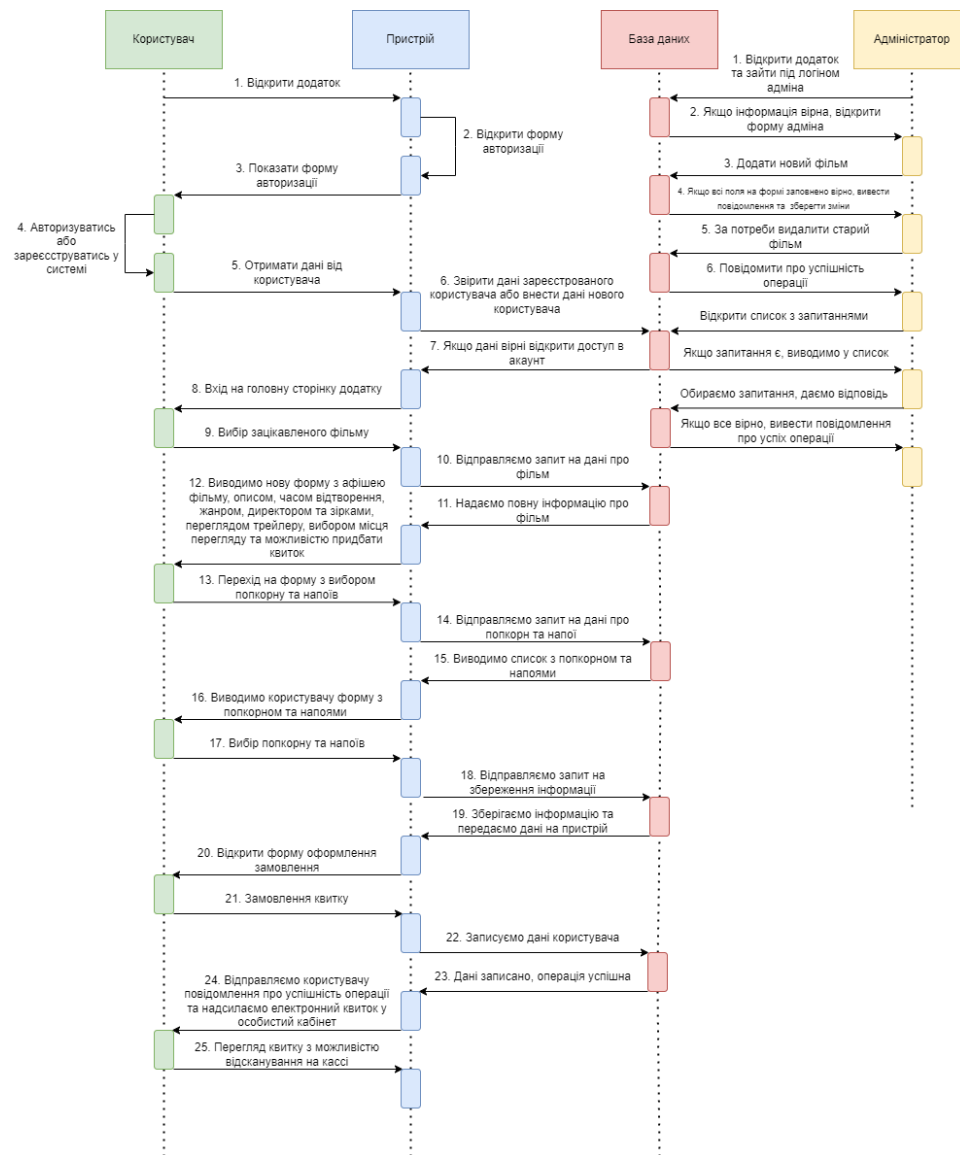


Рисунок 3.9 – Діаграма послідовності системи, що проєктується

Опис етапів послідовності:

- користувач починає взаємодію, наприклад, обираючи фільм, купуючи квиток або авторизуючись у системі;
- пристрій приймає запит від користувача та надсилає його для обробки;
- пристрій надсилає запит до бази даних для отримання потрібної інформації (наприклад, список фільмів, доступні місця, дані квитків тощо);
- база даних обробляє запит та повертає відповідь пристрою;
- пристрій показує результати (список фільмів, підтвердження купівлі тощо) користувачу;

— адміністратор має можливість додавати, видаляти чи оновлювати інформацію в базі даних, наприклад, додавання нових фільмів.

3.7 Висновки за розділом 3

У третьому розділі було розглянуто основні рішення, що стосуються реалізації компонентів системи. Визначено функціональні можливості програми, які включають перегляд розкладу сеансів, вибір місць, купівлю квитків, замовлення попкорну та перегляд інформації про користувача.

Розроблено структурну схему програми, що дозволила візуалізувати взаємозв'язки між основними компонентами системи.

Описано алгоритм функціонування програми, що пояснює логіку роботи системи, включаючи обробку запитів користувачів та взаємодію між модулями. Для ефективної реалізації застосовано шаблон проєктування MVVM, який покращує підтримку, масштабованість та тестування програми.

Виконано розробку діаграми класів, діаграми компонентів та діаграми кооперації, що відображають архітектурні особливості системи. Створено концептуальну модель бази даних, яка забезпечує ефективне зберігання інформації про користувачів, фільми, квитки та додаткові послуги.

Також проведено аналіз взаємодії компонентів системи через діаграми прецедентів та діаграми послідовності, що дозволяє простежити логіку користувацьких сценаріїв та основні бізнес-процеси системи.

Розглянуті рішення забезпечують високу продуктивність, гнучкість та масштабованість програми, що робить її зручною для користувачів і легкою в подальшій підтримці та розширенні.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення й умови застосування програми

Розроблений додаток «PopcornPalms» призначений для автоматизації процесу купівлі квитків до кінотеатру. Він надає користувачам можливість зручно обирати фільми, дати та місця перегляду, здійснювати оплату, а також додатково замовляти попкорн чи напої. Програма орієнтована на широке коло користувачів, включаючи як кінцевих споживачів (гостей кінотеатру), так і адміністратора, які можуть управляти репертуаром фільмів і анонсами.

Умови до апаратної частини сервера:

- потужний сервер: для забезпечення стабільної роботи додатку необхідно мати потужний сервер з високою продуктивністю та достатньою кількістю ресурсів;
- безперебійне забезпечення живлення: сервер повинен мати надійне живлення, щоб уникнути випадків вимикання або перебоїв у роботі системи.

Умови до апаратної частини клієнта:

- сучасний смартфон або планшет: клієнтська частина додатку буде працювати на мобільних пристроях, тому важливо, щоб клієнт мав сучасний пристрій з достатньою продуктивністю для запуску додатку та забезпечення зручного користування.

Умови до програмної частини:

- підтримка мобільних операційних систем: додаток повинен бути сумісний з популярними мобільними операційними системами, такими як Android версією не нижче 8.1 та iOS версією не нижче 15.0, і мати оптимізований інтерфейс для роботи на цих платформах;
- ефективне використання ресурсів: програмний код додатку повинен бути оптимізований для ефективного використання ресурсів мобільного пристрою, таких як процесор, пам'ять та енергія.

4.2 Характеристики програми

Функціональні можливості:

- а) реєстрація та авторизація користувачів із захистом даних;
- б) пошук і перегляд фільмів за назвою, жанром або датою;
- в) вибір місць у залі та оформлення квитків;
- г) замовлення попкорну та напоїв;
- д) оплата через інтегровані платіжні системи (Apple Pay, Google Pay, кредитні картки);
- е) збереження банківських карт користувачів;
- ж) перегляд нещодавно придбаного квитка;
- з) перегляд історії придбаних квитків за весь час;
- и) перегляд накопичених бонусів;
- к) перегляд місцезнаходження кінотеатру;
- л) перехід на соц.мережі кінотеатру зі сторінки «Contact»;
- м) звернення по допомогу до адміністрації кінотеатру;
- н) управління репертуаром фільмів для адміністратора, включаючи додавання та видалення фільмів;
- о) надання адміністратором відповідей користувачам на поставлені запитання;

Нефункціональні характеристики:

- а) висока продуктивність: обробка запитів не перевищує 2 секунд;
- б) безпека даних: авторизація користувачів з використанням хешування паролів;
- в) зручний інтерфейс з адаптивним дизайном.

4.3 Звертання до програми

Запуск програми здійснюється через головний файл, який є виконуваним додатком у середовищі Windows, у директорії Project.

Програма компілюється за допомогою exe-файлу «PopcornPalms.exe».

4.4 Початкові та вихідні дані

Початкові дані:

- дані про фільми (назва фільму, жанр, тривалість, рік випуску, режисер, актори, опис, постер, рейтинг та трейлер);
- дані користувача (ПІБ, логін, пароль та номер телефону);
- замовлення квитків (вибір місця, додаткові замовлення, спосіб оплати).

Вихідні дані:

- електронний квиток із деталями про сеанс та штрих-кодом;
- сповіщення користувача про успішну оплату;
- інформація про наявність фільмів чи місць;
- дані, внесені адміністратором у базу (нові фільми та анонси).

4.5 Повідомлення

У цьому розділі описано текстові повідомлення, які можуть бути виведені програмісту чи оператору під час виконання програми, а також дії, які необхідно виконати при виникненні цих повідомлень.

а) помилка при вході до акаунту:

- 1) текст повідомлення: «Користувача не знайдено. Спробуйте ще раз.»;
- 2) дія: Користувачеві надається можливість повторити спробу входу;

б) помилка при пошуку фільму:

- 1) текст повідомлення: «Фільм не знайдено. Перевірте введені дані та спробуйте знову.»;
- 2) дія: Програма повинна дозволити користувачеві повторити

пошук або ввести інші критерії;

в) помилка додавання банківської карти:

1) текст повідомлення: «Некоректне заповнення полів. Спробуйте знову.»;

2) текст повідомлення: «Не всі поля заповнені. Слід заповнити всі поля.»;

3) дія: Програма повинна дозволити користувачеві повторити спробу додати карту;

г) успішне замовлення квитка:

1) текст повідомлення: «Ваш квиток успішно придбано! Вам нараховано 10% від суми на ваш бонусний рахунок;

2) дія: Після цього повідомлення користувач може перейти до перегляду придбаного квитка у навігаційній панелі у розділі «Ticket»;

д) успішне редагування профілю користувача:

1) текст повідомлення: «Дані профілю успішно змінені.»;

2) дія: Після успішної зміни даних користувача перенаправляє на сторінку редагування профілю вже з оновленими даними;

е) успішне додавання фільму адміністратором:

1) текст повідомлення: "Фільм успішно додано!";

2) дія: Після цього повідомлення новий фільм має з'явитися у списку доступних для користувачів;

ж) успішне відправлення запитання до адміністрації кінотеатру:

1) текст повідомлення: «Ваше запитання успішно відправлено.»;

2) дія: Після успішного відправлення повідомлення користувача залишить на поточній формі з можливістю задати ще одне або декілька запитань;

з) помилка при додаванні фільму:

1) текст повідомлення: «Не вдалося додати фільм. Не всі поля заповнені, спробуйте знову.»;

2) дія: Адміністратор може повторити спробу додавання фільму;

и) помилка при видаленні фільму:

1) текст повідомлення: «Не вдалося видалити фільм. Даного фільму немає у базі, перезавантажте додаток.»;

к) успішна відповідь на запитання від користувача:

1) текст повідомлення: «Відповідь успішно відправлена користувачу.»;

2) дія: Після успішної відповіді на запитання від користувача адміністратора автоматично перенаправляє на головну сторінку з запитаннями, які очікують відповіді.

4.6 Висновки за розділом 4

У цьому розділі було надано інформацію, необхідну для програмістів, які працюють із розробленою системою. Визначено основні аспекти, пов'язані з функціональними можливостями та особливостями програми, включаючи її призначення, умови застосування та характеристики.

Описано процедури звертання до програми, зокрема способи запуску та налаштування середовища виконання. Визначено початкові та вихідні дані, що дозволяють правильно організувати обмін інформацією між компонентами системи та користувачем.

У розділі також висвітлено тексти повідомлень, які формує програма, та дії, які необхідно виконати програмісту чи оператору у разі їхнього виникнення. Це забезпечує ефективний процес діагностики та усунення можливих недоліків.

Таким чином, керівництво програміста охоплює всі ключові аспекти, що сприяють якісному супроводу програми, її налаштуванню та розвитку. Це дозволяє створити зручні умови для розширення функціональності програми та підтримки її коректної роботи.

5 КЕРІВНИЦТВО ОПЕРАТОРА

5.1 Призначення програми

Розроблений додаток призначений для полегшення процесу купівлі квитків у кінотеатрі, вибору місць, оформлення замовлення на напої та попкорн, а також забезпечення інтерактивного зв'язку між користувачами та адміністрацією. Оператору (користувачу) забезпечено простий та зручний інтерфейс для швидкого виконання всіх зазначених функцій.

Додаток включає модулі для реєстрації, авторизації, перегляду фільмів, вибору місць, вибору попкорну та напоїв, оформлення платежів, отримання бонусів, перегляду профілю та відправлення запитань до адміністрації. Його головною метою є забезпечення ефективного управління процесами бронювання квитків та обслуговування користувачів.

5.2 Умови виконання програми

5.2.1 Апаратні вимоги програми

Для коректної роботи додатка «PopcornPalms» необхідно:

- операційна система: Android версією не нижче 8.1 та iOS версією не нижче 15.0;
- обсяг пам'яті: мінімум 2 ГБ оперативної пам'яті;
- процесор: 4-ядерний або вище з тактовою частотою від 1,5 ГГц;
- місце на пристрої: не менше 200 МБ вільного простору для встановлення та збереження тимчасових даних;
- доступ до інтернету: Wi-Fi або мобільний інтернет зі стабільним з'єднанням для синхронізації з базою даних і завантаження контенту.

5.2.2 Вимоги до користувача

- а) рівень підготовки: базові знання роботи зі смартфоном та мобільними додатками;
- б) акаунт користувача: для повноцінного використання функцій додатка необхідно створити або увійти в особистий акаунт за його наявності;
- в) налаштування пристрою:
 - 1) дозвіл на використання інтернет-з'єднання;
 - 2) увімкнення push-сповіщень для отримання інформації;
 - 3) доступ до GPS для роботи функції визначення місцезнаходження кінотеатру.

5.3 Виконання програми

При запуску додатку користувач спершу бачить стартовий екран, який представлено на рисунку 5.1 із логотипом кінотеатру, який відображається протягом кількох секунд. Цей екран виконує роль завантажувальної сторінки, створюючи перше враження про додаток та підкреслюючи його брендованість.

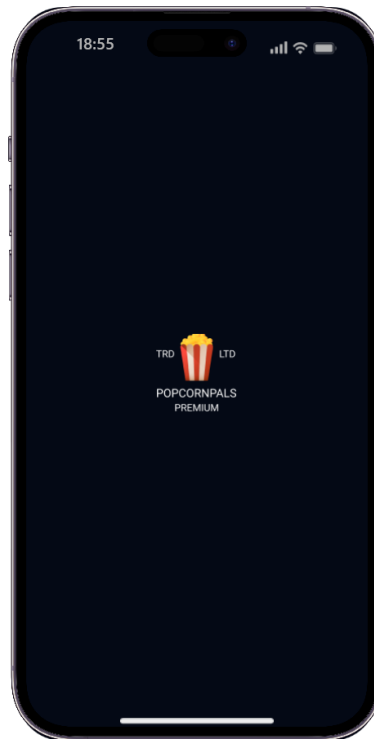


Рисунок 5.1 – Стартова сторінка додатку «PopcornPalms»

Сторінки Onboarding.

Після стартової сторінки користувач переходить до onboarding-процесу, який складається з трьох послідовних екранів. Цей процес дає змогу новим користувачам ознайомитися з основними можливостями програми.

Welcome!

Перший екран, який представлено на рисунку 5.2 вітає користувача із запуском програми. На ньому відображається повідомлення: «Buy Tickets and watch trailers quickly», що акцентує увагу на швидкості та зручності функцій програми.

У нижній частині розташована кнопка «Continue», яка дозволяє перейти до наступного екрану.

У верхній частині знаходиться кнопка «Skip», яка дає змогу пропустити onboarding і відразу перейти до вікна входу (логіну).

Quick purchase.

Другий екран, який представлено на рисунку 5.3 інформує користувача про простоту покупки квитків. На екрані відображається текст: «View the

schedule and choose the best seats», який підкреслює можливість вибору місця та перегляду розкладу сеансів.

Як і на попередньому екрані, в нижній частині є кнопка «Continue», а у верхній – «Skip» для переходу до вікна логіну.

Food without queues.

Останній екран onboarding, який представлено на рисунку 5.4 акцентує увагу на можливості замовлення їжі. Повідомлення звучить як: «Order popcorn and drinks right from your phone», що наголошує на економії часу завдяки можливості замовлення їжі без черг.

У нижній частині знову розташована кнопка «Continue», а у верхній – «Skip», що дозволяє завершити onboarding достроково.

Кнопка «Continue» після останнього екрану автоматично переводить користувача на вікно входу (логіну), забезпечуючи плавний перехід до основного функціоналу програми.

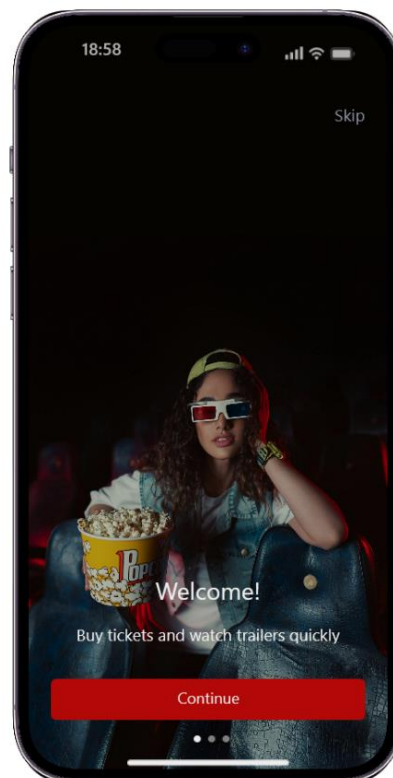


Рисунок 5.2 – Onboarding 1 додатку «PopcornPalms»

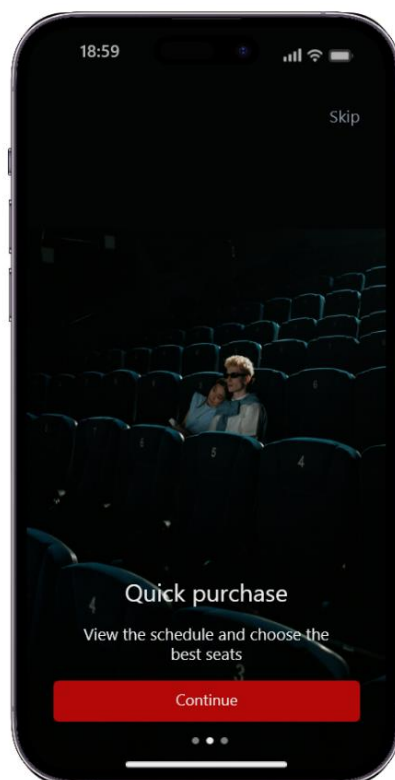


Рисунок 5.3 – Onboarding 2 додатку «PopcornPalms»



Рисунок 5.4 – Onboarding 3 додатку «PopcornPalms»

Сторінка Login.

Після завершення onboarding-процесу або натискання кнопки «Skip», користувач автоматично переходить на сторінку Login (рисунок 5.5), яка є основним вікном для входу в додаток, на якій представлено: текстовий напис з привітанням, поле для вводу «Login», поле для вводу «Password», кнопку «Auth», кнопку «Register», кнопки «Google», «Facebook» та «Apple».

Текстовий напис з привітанням «Welcome! Log in to the system».

Поле для введення «Login» призначене для введення індивідуального логіну користувача.

Поле для введення «Password» призначене для введення індивідуального паролю користувача.

Кнопка «Auth» підтверджує введені дані та здійснює вхід у систему.

Кнопка «Register» відкриває нову сторінку для створення нового облікового запису користувача, який ще не має акаунту.

Кнопки соціальних мереж «Google», «Facebook» та «Apple» призначені для швидкого входу в акаунт без заповнення полів вручну.

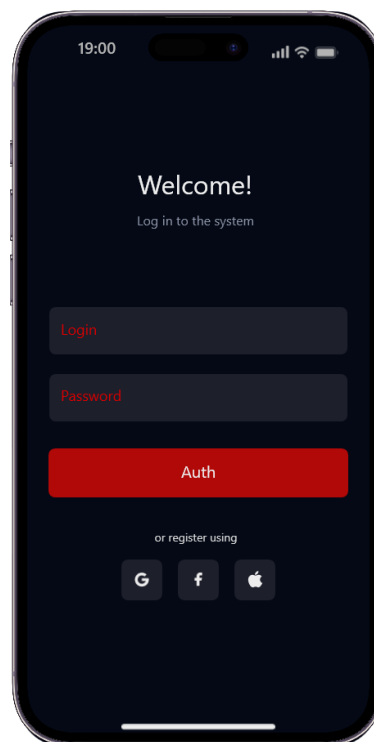


Рисунок 5.5 – Сторінка Login додатку «PopcornPalms»

Сторінка Register.

Після вибору кнопки «Register» відкривається сторінка реєстрації, яка дозволяє новим користувачам створити обліковий запис (рисунок 5.6), на якій представлено: текстовий напис з вказівками, поле для вводу «Full name», поле для вводу «Login», поле для вводу «Phone number», поле для вводу «Password», кнопку «Register», кнопку «Login», кнопки «Google», «Facebook» та «Apple».

Текстовий напис з вказівками «Welcome! Enter your full name, login, phone number and password to register»

Поле «Full Name» користувач заповнює своїм ПІБ.

Поле «Login» користувач заповнює унікальним логіном.

Поле «Phone Number» користувач заповнює контактним номером телефону.

Поле «Password» користувач заповнює унікальним паролем.

Кнопка «Register» завершує процес реєстрації та створює новий обліковий запис у системі.

Кнопка «Login» відкриває сторінку авторизації при наявності акаунту у користувача.

Кнопки соціальних мереж «Google», «Facebook» та «Apple» призначені для швидкої реєстрації акаунту без заповнення полів вручну.

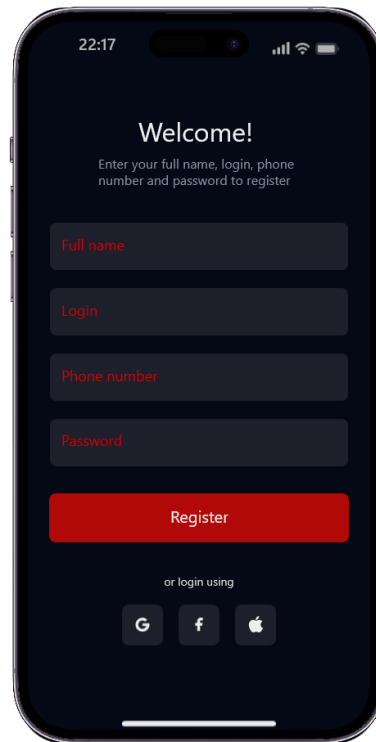


Рисунок 5.6 – Сторінка Register додатку «PopcornPalms»

Сторінка Home.

Після успішної авторизації або реєстрації користувач потрапляє на сторінку Home (головний екран програми) (рисунок 5.7), який є основною точкою доступу до всіх функцій додатка, на якій представлено: місцезнаходження кінотеатру, фільм тижня, список «Now at the cinema», список «Announcements», меню навігації.

Місцезнаходження кінотеатру містить у собі точне розташування кінотеатру у ТРЦ Lavina Mall міста Kyiv.

Фільм тижня містить у собі зображення найпопулярнішого фільму тижня за кількістю переглядів.

«Now at the cinema» містить у собі список фільмів, які вже доступні для придбання квитка. Список складається з: зображення фільму, року випуску та жанру, назви.

«Announcements» містить у собі список фільмів, які незабаром з'являться у прокаті. Список складається з: зображення фільму, року випуску та жанру, назви.

Навігаційна панель містить у собі 4 кнопки: «Movie» – поточна сторінка, «Explore» – сторінка пошуку фільму, «Ticket» – сторінка останньо–придбаного квитка, «Profile» – сторінка особистого кабінету користувача.

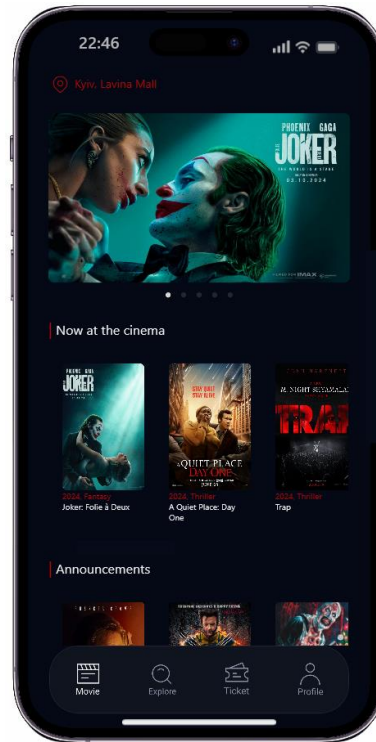


Рисунок 5.7 – Сторінка Home додатку «PopcornPalms»

Сторінка Explore.

Після переходу на сторінку Explore (рисунок 5.8) користувач потрапляє на сторінку, призначену для пошуку фільмів та їхнього сортування за жанрами, на якому представлено: кнопку «Back», рядок пошуку, список «Categories» із прокруткою жанрів, список усіх доступних фільмів для перегляду, також цей список відповідає за виведення на екран фільму, який шукав користувач та меню навігації.

Кнопка «Back» повертає користувача на попередню сторінку.

Рядок пошуку представляє собою поле для вводу назви фільму та дві кнопки: «Search» та «Delete».

«Categories» містить у собі список жанрів, за яким можна здійснити пошук.

Навігаційна панель містить у собі 4 кнопки: «Movie» – головна сторінка, «Explore» – поточна сторінка, «Ticket» – сторінка останньо-придбаного квитка, «Profile» – сторінка особистого кабінету користувача.

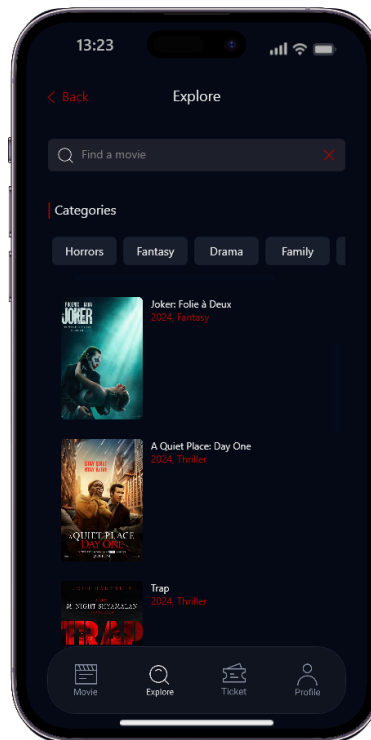


Рисунок 5.8 – Сторінка Explore додатку «PopcornPalms»

Сторінка Description.

Після вибору фільму на сторінці Explore або на сторінці Home, користувач переходить на сторінку Description (рисунок 5.9), яка забезпечує доступ до детальної інформації про обраний фільм, на якій представлено: зображення фільму, кнопку «Watch trailer», назву фільму, зірки рейтингу, час відтворення фільму, рік випуску та жанр, режисерів, акторів, опис фільму, кнопку «Select places» та навігаційну панель.

Зображення фільму знаходиться у верхній частині екрану для привернення уваги користувача.

При натисканні на кнопку «Watch trailer» поле з детальним описом фільму замінюється на відео-плеєр (рисунок 5.10), на якому користувач може переглянути трейлер фільму. При відкритті трейлеру користувачу також

доступні дві кнопки: перша кнопка «Play-Stop», друга закриває відео-плеєр та повертає детальний опис фільму.

Кнопка «Select places» відкриває нову сторінку з вибором місць для придбання квитка.

Навігаційна панель містить у собі 4 кнопки: «Movie» – головна сторінка, «Explore» – сторінка пошуку фільму, «Ticket» – сторінка останньо-придбаного квитка, «Profile» – сторінка особистого кабінету користувача.

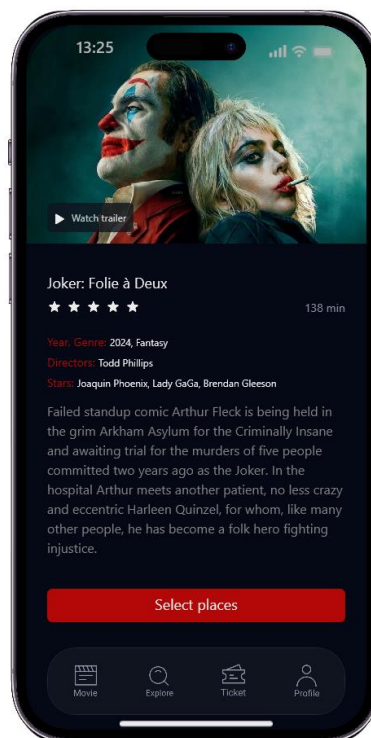


Рисунок 5.9 – Сторінка Description додатку «PopcornPalms»

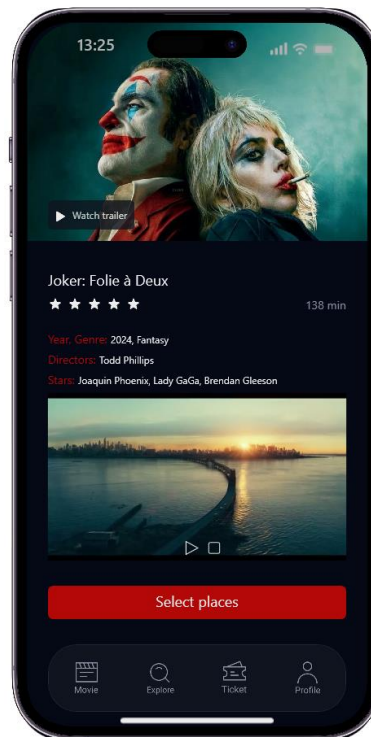


Рисунок 5.10 – Перегляд трейлеру до фільму додатку «PopcornPalms»

Сторінка Choose place.

Після натискання кнопки «Select places» на сторінці Description користувач переходить на сторінку Choose place (рисунок 5.11), на якій представлено: кнопку «Back», назву фільму, розклад на тиждень, кнопки вибору «Session time», схема залу, рядок «Total» та кнопку «Next».

Кнопка «Back» повертає користувача на попередню сторінку.

Розклад на тиждень містить у собі всього 7 днів, серед яких перший день є поточним на момент вибору місць.

Кнопки «Session time» містять у собі час та формат сеансу.

Схема залу містить у собі два типи місць: червоні та сині, серед яких червоні – вільні місця, сині – придбані.

Рядок «Total» відображає ціну, яка корегується за рахунок обраних місць.

Кнопка «Next» відкриває нову сторінку з вибором сніків.

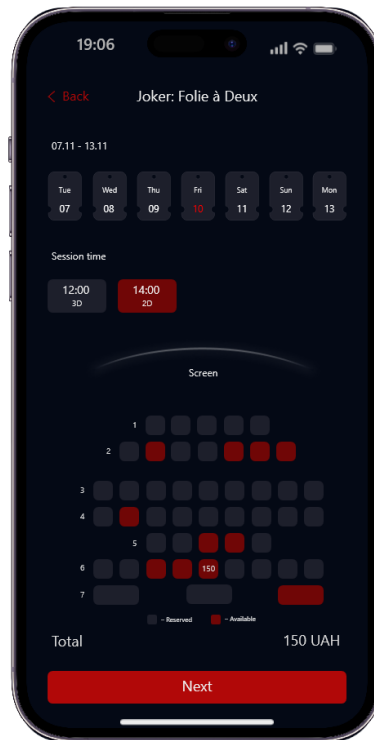


Рисунок 5.11 – Сторінка Choose place додатку «PopcornPalms»

Сторінка Snacks.

Після завершення вибору місць на сторінці Choose place користувач переходить на сторінку Snacks (рисунок 5.12). Ця сторінка дозволяє додати до замовлення попкорн та напої, забезпечуючи повний комфорт під час перегляду фільму, на якій представлено: кнопку «Back», три кнопки вибору попкорну, дві кнопки вибору напоїв, рядок «Total» та кнопку «Go to the payment».

Кнопка «Back» повертає користувача на попередню сторінку.

Кнопки вибору попкорну дають змогу користувачу обрати попкорн в залежності від літражу. Самі кнопки містять у собі: зображення попкорну, смак, літраж (1.5L, 3L, 5L) та ціну, яка є індивідуальною для кожного літражу попкорну.

Кнопки вибору напоїв дають змогу користувача обрати напої в залежності від смаку. Самі кнопки містять у собі: зображення напою, назву, літраж та ціну.

Рядок «Total» відображає ціну, яка корегується за рахунок обраних снєків.

Кнопка «Go to the payment» відкриває нову сторінку з методом оплати КВИТКА.

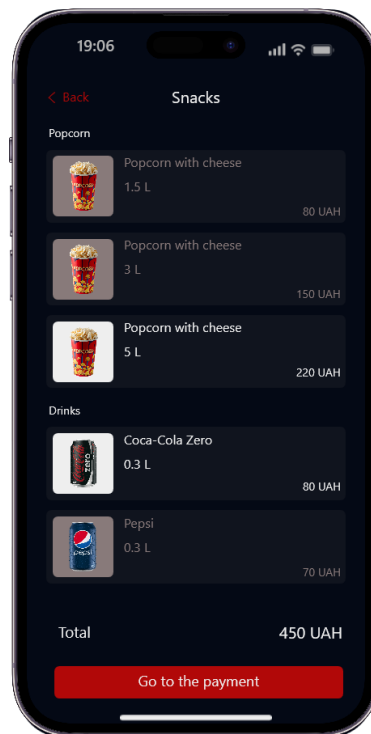


Рисунок 5.12 – Сторінка Snacks додатку «PopcornPalms»

Сторінка Reservation.

Після вибору снєків на сторінці Snacks користувач переходить на фінальну сторінку Reservation (рисунок 5.13). Ця сторінка дозволяє завершити оформлення замовлення та здійснити оплату за обрані квитки й снєки за умови вибору вже збереженої карти або вибору Apple Pay. На сторінці представлено: кнопку «Back», назву фільму, дату перегляду, час, ряд та місце, блок «Secure payment», рядок «Total», «Checkbox» та кнопку «Payment».

Кнопка «Back» повертає користувача на попередню сторінку.

Блок «Secure payment» містить у собі три вибори оплати, а саме: оплата вже збереженою картою, оплата Apple Pay та додавання нової карти у разі відсутності оплати двома іншими методами.

Рядок «Total» відображає кінцеву суму замовлення, яка сформована на основі обраних місць та снєків.

«Checkbox» містить у собі обов'язкове підтвердження користувачем згоди з умовами користування.

Кнопка «Payment» працює лише за умови вибору перших двох умов оплати.

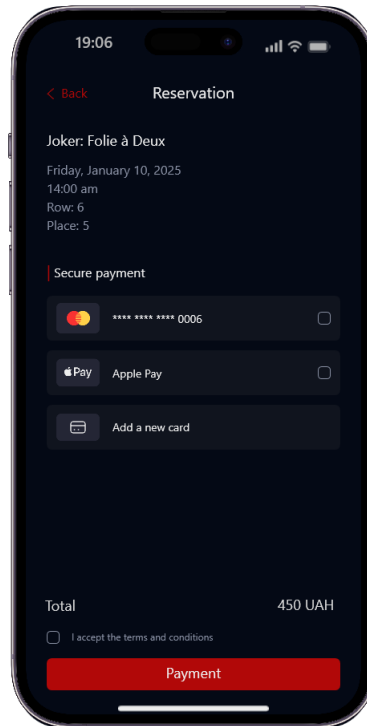


Рисунок 5.13 – Сторінка Reservation додатку «PopcornPalms»

Сторінка Add card.

Сторінка Add card дозволяє користувачу додати нову платіжну карту для здійснення оплати (рисунок 5.14), на якому представлено: кнопку «Back», зображення карти, поля для вводу даних: «Card number», «Valid until», «CVV», «Card holder», «Checkbox» та кнопку «Pay».

Кнопка «Back» повертає користувача на попередню сторінку.

Зображення карти представляє собою динамічне зображення, яке оновлюється в режимі реального часу відповідно до введених користувачем даних.

«Checkbox» містить у собі підтвердження користувачем згоди на збереження даних картки для майбутніх оплат.

Кнопка «Pay» завершує процес додавання нової карти та здійснює оплату квитка.

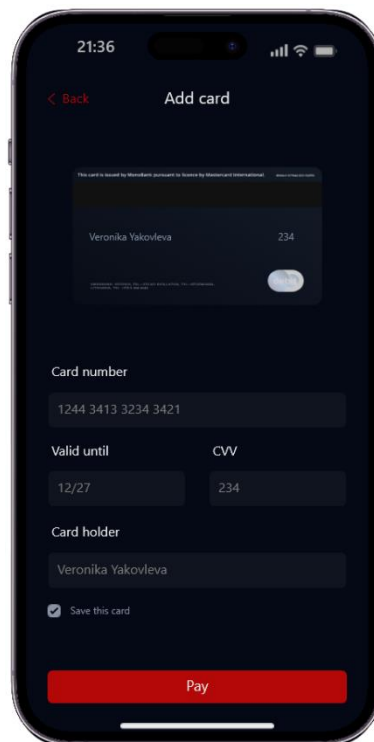


Рисунок 5.14 – Сторінка Add card додатку «PopcornPalms»

Сторінка Ticket.

Після успішної оплати квитка користувач автоматично перенаправляється на сторінку з відображенням придбаного квитка (рисунок 5.15). Ця сторінка виконує функцію електронного квитка, який може бути використаний для входу до кінотеатру, на якому представлено: кнопку «Back», зображення фільму, назву фільму, назву кінотеатру «PopcornPalms», ідентифікатор квитка «id», дату сеансу та час, ряд, місце, ціну, обрані снеки, штрих-код та навігаційну панель.

Кнопка «Back» повертає користувача на головну сторінку.

Зображення фільму знаходиться у верхній частині квитка для привернення уваги користувача.

Ідентифікатор квитка «id» містить у собі унікальний порядковий номер, який не повторюється.

Штрих-код призначений для сканування та підтвердження квитка при вході до зали працівниками кінотеатру.

Навігаційна панель містить у собі 4 кнопки: «Movie» – головна сторінка, «Explore» – сторінка пошуку фільму, «Ticket» – поточна сторінка, «Profile» – сторінка особистого кабінету користувача.

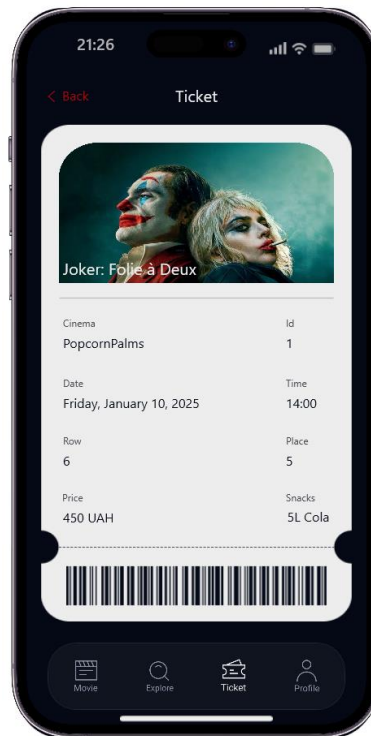


Рисунок 5.15 – Сторінка Ticket додатку «PopcornPalms»

Сторінка Profile.

Сторінка Profile (рисунок 5.16) є персоналізованим розділом додатку, де користувач може переглядати та редагувати свої дані, а також взаємодіяти з іншими функціями програми, на якій представлено: місцезнаходження кінотеатру, блок з особистою інформацією та кнопкою «Edit profile», кнопку «Credit cards», кнопку «Bonuses», кнопку «All tickets», кнопку «Location», кнопку «Support», кнопку «Contact», кнопку «Log out of account» та навігаційну панель.

Місцезнаходження кінотеатру містить у собі точне розташування кінотеатру у ТРЦ Lavina Mall міста Kyiv.

Блок з особистою інформацією містить у собі ім'я та прізвище користувача, його номер телефону та кнопку «Edit profile».

Навігаційна панель містить у собі 4 кнопки: «Movie» – головна сторінка, «Explore» – сторінка пошуку фільму, «Ticket» – поточна сторінка, «Profile» – сторінка особистого кабінету користувача.

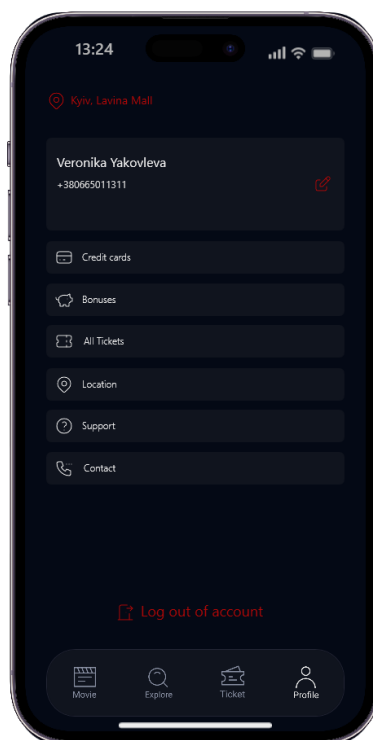


Рисунок 5.16 – Сторінка Profile додатку «PopcornPalms»

Сторінка Login (Admin).

При запуску додатку адміністратор спершу бачить стартовий екран, onboarding, згодом його перенаправляє на сторінку Login (рисунок 5.5), де він вводить свій унікальний «Login», «Password».

Після входу в акаунт адміністратор потрапляє на сторінку Answer (рисунок 5.17) для відповідей на запитання користувачів, на якій представлено: місцезнаходження кінотеатру, список заголовків запитань від користувачів та навігаційну панель.

Місцезнаходження кінотеатру містить у собі точне розташування кінотеатру у ТРЦ Lavina Mall міста Kyiv.

Після натискання на один із заголовків списку відкривається нова сторінка Answer to the question (рисунок 5.25), на якій представлено: кнопку «Back», поля заголовку та основного тексту запитання, поле для надання відповіді користувачу, кнопку Send та навігаційну панель.

Навігаційна панель містить у собі 5 кнопок: «Add movie» – сторінка додавання фільмів, «Add» – сторінка додавання анонсів, «Delete» – сторінка видалення фільмів, «Answer» – поточна сторінка, «Profile» – сторінка профілю адміністратора.

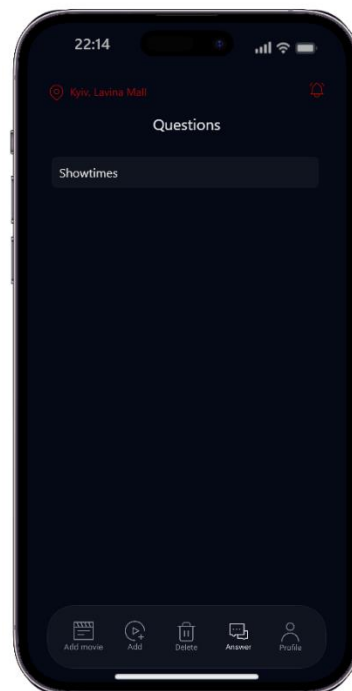


Рисунок 5.17 – Сторінка Answer (Admin) додатку «PopcornPalms»

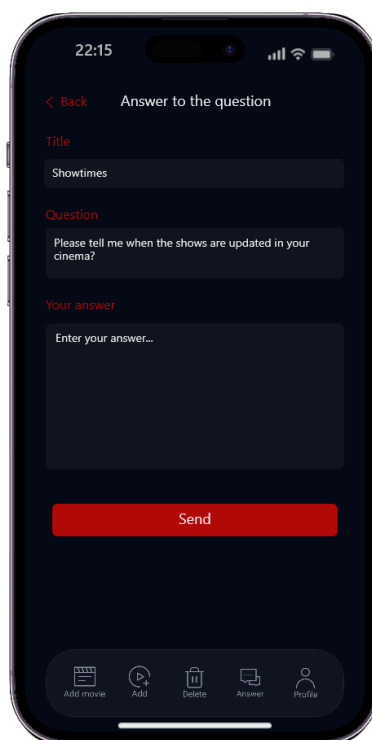


Рисунок 5.18 – Сторінка Answer to the question (Admin) додатку
«PopcornPalms»

Сторінка Add movie (Admin).

Сторінка Add movie (Admin) (рисунок 5.19) є спеціальним інтерфейсом для адміністратора кінотеатру, який дозволяє додавати фільми, які будуть доступними для перегляду в додатку користувачам. На сторінці представлено: блоки додавання зображень фільму, блок додавання трейлеру, поле для вводу назви фільму, кнопки (зірки) вибору рейтингу, поле для вводу часу відтворення фільму, поля для вводу «Year, Genre, Directors, Stars», поле для вводу детального опису фільму, кнопку «Add film» та навігаційну панель.

Блок «Add a movie image for main page» додає головне зображення до фільму.

Блок «Add a movie image for description page» додає зображення до сторінки з описом фільму.

Блок «Add a trailer to the movie» додає трейлер до сторінки з описом фільму.

Кнопка «Add film» додає фільм до бази даних для відображення

користувачам у додатку.

Навігаційна панель містить у собі 5 кнопок: «Add movie» – поточна сторінка, «Add» – сторінка додавання анонсів, «Delete» – сторінка видалення фільмів, «Answer» – сторінка для відображення запитань від користувачів, «Profile» – сторінка профілю адміністратора.

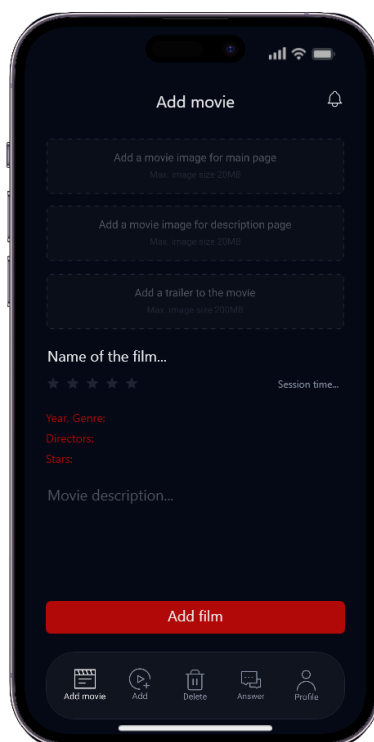


Рисунок 5.19 – Сторінка Add movie (Admin) додатку «PopcornPalms»

Сторінка Add announcement (Admin).

Сторінка Add announcements (Admin) (рисунок 5.20) призначена для адміністратора кінотеатру, який відповідають за управління майбутніми прем'єрами. Цей інтерфейс дозволяє додавати фільми, які ще не вийшли у прокат, але вже є запланованими до показу. На сторінці представлено: блоки додавання зображень фільму, блок додавання трейлеру, поле для вводу назви фільму, кнопки (зірки) вибору рейтингу, поле для вводу часу відтворення фільму, поля для вводу «Year, Genre, Directors, Stars», поле для вводу детального опису фільму, кнопку «Add film» та навігаційну панель.

Блок «Add a movie image for main page» додає головне зображення до

фільму.

Блок «Add a movie image for description page» додає зображення до сторінки з описом фільму.

Блок «Add a trailer to the movie» додає трейлер до сторінки з описом фільму.

Кнопка «Add announcement» додає анонс до бази даних для відображення користувачам у додатку.

Навігаційна панель містить у собі 5 кнопок: «Add movie» – сторінка додавання фільмів, «Add» – поточна сторінка, «Delete» – сторінка видалення фільмів, «Answer» – сторінка для відображення запитань від користувачів, «Profile» – сторінка профілю адміністратора.

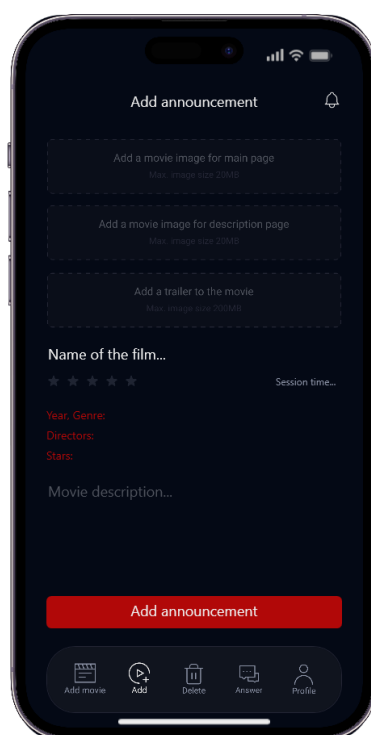


Рисунок 5.20 – Сторінка Add announcement (Admin) додатку «PopcornPalms»

Сторінка Delete (Admin).

Сторінка видалення фільму (Admin) (рисунок 5.21) призначена для адміністратора, який має повноваження видаляти фільми з бази даних кінотеатру. Цей інструмент необхідний для актуалізації списку доступних

сеансів і прем'єр. На сторінці представлено: місцезнаходження кінотеатру, список «Now at the cinema», список «Announcements», меню навігації.

Місцезнаходження кінотеатру містить у собі точне розташування кінотеатру у ТРЦ Lavina Mall міста Київ.

Після натискання на один фільм із списку «Now at the cinema» адміністратору виведеться діалогове вікно (рисунки 5.29) з підтвердженням дії: «Are you sure you want to delete?» та двома кнопками: «Yes», «No».

Після натискання на один фільм із списку «Announcements» адміністратору виведеться діалогове вікно з підтвердженням дії: «Are you sure you want to delete?» та двома кнопками: «Yes», «No».

Навігаційна панель містить у собі 5 кнопок: «Add movie» – сторінка додавання фільмів, «Add» – сторінка додавання анонсів, «Delete» – поточна сторінка, «Answer» – сторінка для відображення запитань від користувачів, «Profile» – сторінка профілю адміністратора.

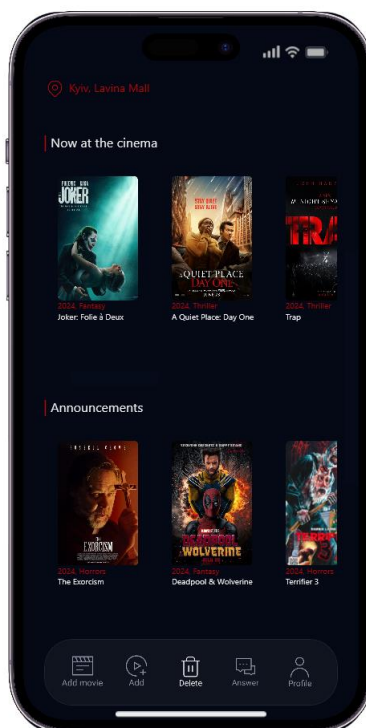


Рисунок 5.21 – Сторінка Delete (Admin) додатку «PopcornPalms»

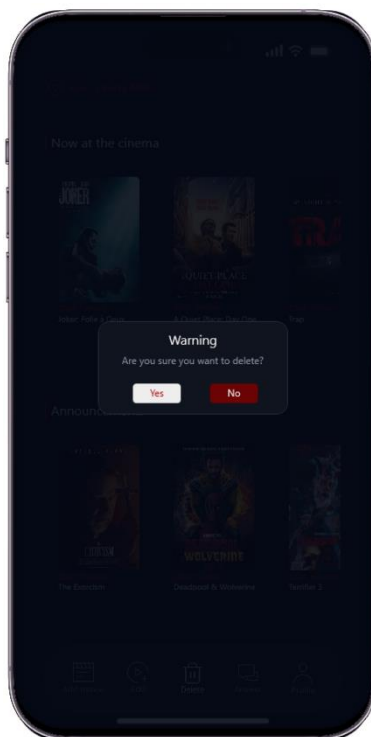


Рисунок 5.22 – Діалогове вікно з підтвердженням дії додатку «PopcornPalms»

Сторінка Profile (Admin).

Сторінка Profile (рисунок 5.23) є персоналізованим розділом додатку, де адміністратор може переглядати та редагувати свої дані. На сторінці представлено: місцезнаходження кінотеатру, блок з особистою інформацією та кнопкою «Edit profile», кнопку «Log out of account» та навігаційну панель.

Місцезнаходження кінотеатру містить у собі точне розташування кінотеатру у ТРЦ Lavina Mall міста Kyiv.

Кнопка «Log out of account» здійснює вихід з акаунту та перенаправляє на сторінку Login.

Навігаційна панель містить у собі 5 кнопок: «Add movie» – сторінка додавання фільмів, «Add» – сторінка додавання анонсів, «Delete» – сторінка видалення фільмів, «Answer» – сторінка для відображення запитань від користувачів, «Profile» – поточна сторінка.

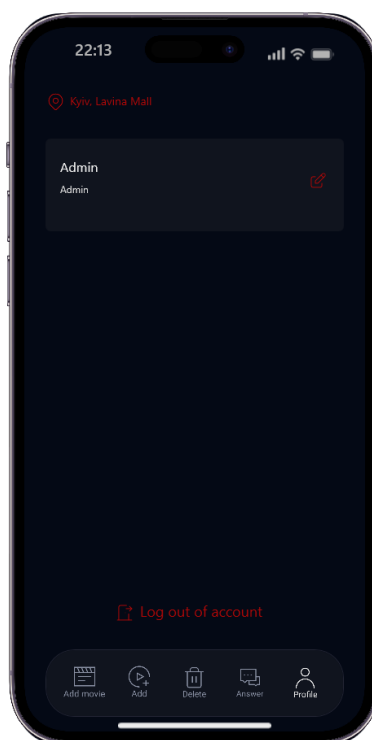


Рисунок 5.23 – Сторінка Profile (Admin) додатку «PopcornPalms»

5.4 Повідомлення оператора

У процесі роботи мобільного додатка «PopcornPalms» оператор (користувач) може отримувати різноманітні повідомлення, які надають інформацію про успішність виконання дій та можливі помилки. Ці повідомлення допомагають забезпечити зручну навігацію та уникнути непорозумінь під час використання програми.

а) інформаційні повідомлення:

1) «Welcome to PopcornPalms!» – повідомлення, яке вітає користувача при вході до додатка;

2) «Thank you for your purchase. You have been awarded 10% bonuses.» – повідомлення, яке підтверджує успішну оплату квитка та нараховує 10% бонусів від кінцевої суми замовлення;

3) «User details updated» – повідомлення, яке з'являється після успішного редагування профілю;

4) «Message Sent» – повідомлення, яке з'являється після

відправлення запитання адміністратору;

5) «Film added» – повідомлення, яке з'являється після успішного додавання фільму адміністратором;

б) попереджувальні повідомлення:

1) «Not all fields are filled in» – нагадування заповнити всі обов'язкові поля у формах (авторизація, реєстрація, додавання карти тощо);

2) «This user is already registered» – повідомлення, яке з'являється на сторінці реєстрації коли користувач вводить вже існуючий логін;

3) «Are you sure you want to delete?» – підтвердження перед видаленням фільму в акаунті адміністратора;

в) помилки:

1) «User not found» – повідомлення, яке з'являється на сторінці авторизації, коли користувач вводить помилковий логін або пароль;

2) «You don't have purchased tickets yet» – повідомлення, яке з'являється на сторінці усіх придбаних квитків, коли користувач ще не придбав жодного квитка;

3) «Movie not found» – повідомлення, яке з'являється на сторінці пошуку, коли введений фільм користувачем не знайдено;

4) «Photo not added» – повідомлення, яке з'являється на сторінці додавання фільму чи анонсів адміністратором при спробі додавання невідтриманого формату файлу;

5) «Video trailer not added» – повідомлення, яке з'являється на сторінці додавання фільму чи анонсів адміністратором при спробі додавання невідтриманого формату файлу або файлу великого розміру.

5.5 Висновки за розділом 5

У п'ятому розділі представлено всі необхідні інструкції для ефективного використання мобільного додатка «PopcornPalms», який призначений для спрощення процесу вибору та придбання квитків, замовлення попкорну та напоїв. Описано мету програми, яка полягає у наданні зручного інструменту для взаємодії з користувачем, та умови її виконання. Визначено технічні та користувацькі вимоги, що гарантують стабільну роботу програми на мобільних пристроях.

Розглянуто послідовність дій користувача, починаючи від запуску програми та авторизації, до вибору сеансу, місць та здійснення оплати. Детально описано функціонал кожного екрану додатка, що допомагає користувачу орієнтуватися у додатку та виконувати необхідні дії.

Увагу приділено текстам повідомлень, які інформують користувача про стан процесів у програмі, включаючи можливі помилки або підтвердження дій. Такі повідомлення спрямовані на підвищення зручності взаємодії з додатком та забезпечують зрозумілість процесів для оператора.

Цей розділ слугує основним орієнтиром для користувача, дозволяючи максимально ефективно використовувати можливості додатка та уникати труднощів у процесі його експлуатації.

6 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

6.1 Загальні положення

Цей розділ описує основні принципи експлуатації, тестування та дослідження мобільного додатка для покупки квитків у кінотеатрі «PopcornPalms». Мета експлуатаційних і тестувальних заходів – забезпечити високу якість роботи програми, відповідність її функціоналу вимогам користувачів та стабільність роботи на різних пристроях.

6.2 Підготовка до експлуатації

Перед початком використання програми необхідно:

- встановити додаток на смартфон з операційною системою Android або iOS через відповідні магазини додатків (Google Play, App Store);
- переконатися у відповідності пристрою мінімальним апаратним вимогам;
- виконати початкові налаштування програми, включаючи вибір мови та дозволів (доступ до інтернету, місцезнаходження).

6.3 Створення чек-листа для тестування програми

Для перевірки функціональності додатка було розроблено чек-лист (таблиця 6.1), який дозволяє детально протестувати всі основні компоненти програми на пристроях Android та IOS. Чек-лист включає перевірку:

- завантаження та запуску додатка;
- роботи onboarding, авторизації та реєстрації;
- функціональності головного екрану;
- пошуку фільмів, вибору місць та оплати квитків;
- профілю користувача та підменю.

Таблиця 6.1 – Чек-лист тестування додатку «PopcornPalms»

Перевірка	Android 14.0, Samsung	IOS 18.0, Iphone
Програма успішно завантажується	Passed	Passed
Відображається логотип при запуску	Passed	Passed
Onboarding працює без збоїв	Passed	Passed
Кнопка «Skip» перенаправляє на сторінку логін	Passed	Passed
Успішна авторизація	Passed	Passed
Помилка при введенні некоректного логіну/паролю	Passed	Passed
Реєстрація працює за умов заповнення всіх полів	Passed	Passed
Відображається локація кінотеатру	Passed	Passed
Список «Now at the cinema» відображається коректно	Passed	Passed
Список «Announcements» відображається коректно	Passed	Passed
Поле пошуку працює	Passed	Passed
Категорії жанрів доступні	Passed	Passed
Всі фільми відображаються	Passed	Passed
Вибір методу оплати працює	Passed	Passed
Поля введення даних картки	Passed	Passed
Після натискання кнопки «Pay» відкривається сторінка придбаного квитка	Passed	Passed
Можливість змінювати особисті дані	Passed	Passed
Переходи до підменю працюють	Passed	Passed

Продовження таблиці 6.1

Перевірка	Android 14.0, Samsung	IOS 18.0, Iphone
Кнопка виходу з акаунта працює	Passed	Passed
Список запитань від користувачів виводиться	Passed	Passed
Додавання фільмів працює коректно (Адмін)	Passed	Passed
Додавання анонсів працює коректно (Адмін)	Passed	Passed
Фільм видаляється без збоїв (Адмін)	Passed	Passed

Чек-лист значно спростив процес тестування, забезпечивши перевірку всіх ключових функцій додатка. Завдяки цьому вдалося усунути всі виявлені недоліки, що забезпечило розробку стабільного та якісного продукту без помилок.

6.4 Методика тестування

Методика тестування описує підхід до проведення перевірки додатка на відповідність вимогам та виявлення потенційних помилок. Для тестування додатка «PopcornPalms» були використані різні методи та техніки, що дозволили забезпечити всебічне покриття тестування, ефективно перевірити всі аспекти функціональності і стабільності системи. Ось основні етапи і підходи, що були застосовані під час тестування:

а) підготовка до тестування:

1) огляд вимог та специфікацій: Перед початком тестування було проведено детальний аналіз технічних вимог до додатка, а також документації, що описує основні функціональні можливості додатка;

2) створення тестового середовища: Для проведення тестування були налаштовані середовища на різних пристроях (смартфони з різними операційними системами), що дозволило здійснити тестування на реальних пристроях.

б) типи тестування:

1) функціональне тестування: Перевірка відповідності функціональних вимог додатка. Це включає в себе перевірку всіх основних функцій додатка, таких як вибір фільмів, процес придбання квитків, вибір місць, додавання попкорну та напоїв до замовлення, перевірка профілю користувача тощо;

2) тестування сумісності: Перевірка роботи додатка на різних пристроях і операційних системах (Android, iOS). Враховувались різні версії ОС, роздільна здатність екранів, а також перевірка адаптивності інтерфейсу до різних розмірів екранів;

3) тестування зручності користування (Usability testing): Оцінка інтерфейсу додатка, перевірка його зручності, інтуїтивності та простоти використання. Для цього залучались тестові користувачі, які оцінювали інтерфейс без попередньої підготовки;

4) тестування безпеки: Перевірка наявності уразливостей, таких як можливість несанкціонованого доступу до особистих даних користувачів або бази даних. Для цього застосовувались методи тестування на проникнення і перевірка захисту даних;

5) тестування продуктивності: Перевірка додатка на ефективність його роботи під час високих навантажень. Це включило в себе перевірку швидкості відгуку додатка, часу завантаження сторінок, стабільності роботи за великої кількості одночасних користувачів;

6) тестування на помилки та баги: Виявлення можливих помилок, багів і недоліків у коді та інтерфейсі, а також перевірка їх впливу на основну функціональність додатка.

в) методи тестування:

1) ручне тестування: Для перевірки інтерфейсу, зручності використання та виявлення помилок, що не можуть бути легко автоматизовані, використовувалося ручне тестування;

2) автоматизоване тестування: Для тестування функціональних можливостей і перевірки повторюваних сценаріїв, що не змінюються від версії до версії, використовувались автоматизовані тести.

6.5 Результати тестування

Під час проведеного тестування додатка «PopcornPlams» було перевірено ряд ключових аспектів роботи системи, щоб підтвердити її надійність та відповідність вимогам. За результатами тестування було підтверджено наступне:

а) стабільність роботи додатка на різних пристроях:

1) тестування проводилося на різноманітних пристроях з різними операційними системами (Android, iOS) і версіями ОС;

2) перевірено, як додаток працює на смартфонах з різними характеристиками (процесор, обсяг пам'яті, роздільна здатність екрана);

3) виявлено, що додаток стабільно працює на пристроях середнього і високого класу, не з'являються критичні помилки під час запуску або використання основних функцій;

б) відсутність критичних помилок:

1) під час тестування не було виявлено критичних помилок, які б могли призвести до збоїв роботи додатка або втрати даних;

2) протестовано всі основні функціональні модулі додатка, зокрема процес придбання квитків, вибір місць, додавання попкорну та напоїв до замовлення;

3) помилки, що виникали, були незначними (наприклад, некоректне відображення елементів інтерфейсу на деяких пристроях), але не впливали на основну функціональність;

в) зручність використання інтерфейсу:

1) протестовано інтерфейс на зручність і інтуїтивність використання;

2) тестувались різні варіанти взаємодії користувача з додатком, включаючи пошук фільмів, вибір місць, придбання квитків та оформлення замовлення;

3) інтерфейс було оцінено як зручний і зрозумілий, без явних бар'єрів для користувача;

4) не було виявлено значних складнощів у користувацькому досвіді, всі дії були виконані без помилок і з мінімальними затримками;

г) відповідність функціоналу заявленим вимогам:

1) під час тестування функціонал додатка був перевірений на відповідність технічним вимогам і специфікаціям, визначеним на етапі розробки;

2) усі основні функції додатка (перегляд фільмів, вибір квитків, придбання попкорну, підтримка користувачів, відображення квитків) працюють відповідно до вимог;

3) жодних відхилень від вимог виявлено не було. Усі заявлені функції працюють стабільно та коректно.

6.6 Висновки за розділом 6

У розділі було детально розглянуто процес підготовки до експлуатації, методику тестування та результати перевірки програми. Основною метою тестування було забезпечення стабільної роботи додатку, виявлення можливих помилок і підтвердження відповідності функціоналу заявленим вимогам.

Під час підготовки до експлуатації здійснено налаштування тестового середовища, що дозволило забезпечити коректну роботу програми на різних пристроях і операційних системах. Розроблено чек-лист для перевірки ключових функцій програми, що дало змогу систематично підходити до тестування.

Методика тестування включала різні види перевірок, серед яких: функціональне, сумісне, тестування зручності, тестування безпеки та продуктивності. Такий підхід забезпечив всебічну перевірку всіх аспектів програми. Результати тестування підтвердили стабільність роботи програми, відсутність критичних помилок і зручність використання інтерфейсу.

Загалом, виконане тестування засвідчило, що програма відповідає технічним вимогам, є надійною у роботі та готовою до використання кінцевими користувачами.

ВИСНОВКИ

У процесі виконання роботи було проведено комплексне дослідження предметної області, спроектовано та реалізовано програмний додаток для придбання квитків до кінотеатру. Розробка охопила всі необхідні етапи – від аналізу задачі до тестування, підготовки програми до експлуатації та розробки відповідної документації.

На початковому етапі було здійснено аналіз предметної області, визначено мету й завдання розробки, а також проаналізовано існуючі рішення для виявлення їх переваг і недоліків. На основі отриманих даних сформульовано чітку постановку задачі, що забезпечило міцну основу для подальшого проектування.

Для реалізації додатка було обрано оптимальні технології, зокрема інструменти розробки, архітектурний підхід, базу даних та мову програмування. Це дозволило створити масштабовану, надійну та функціональну систему. У процесі проектування архітектури було визначено ключові модулі системи, розроблено структурні схеми, діаграми та концептуальну модель бази даних, що забезпечило чіткий поділ функціональності та узгодженість між компонентами програми.

У рамках розробки керівництва програміста детально описано основні технічні аспекти функціонування програми. Ця частина роботи включила опис умов застосування, характеристики програми, порядок звертання до неї, а також формати початкових і вихідних даних. Це дозволяє спростити підтримку та модернізацію додатка.

Керівництво оператора містить детальні інструкції для кінцевих користувачів і адміністраторів щодо використання програми. Включено опис апаратних вимог, сценарії виконання ключових функцій, а також повідомлення, які допомагають оператору в разі виникнення помилок або запитань. Ця документація робить використання програми максимально простим і зрозумілим.

Важливою частиною роботи стало тестування. Для цього було розроблено чек-лист, визначено методику перевірки та проведено всебічне тестування додатка. Результати тестування підтвердили стабільність роботи програми, відсутність критичних помилок, зручність використання інтерфейсу та відповідність функціоналу заявленим вимогам.

Завдяки реалізації інтуїтивно зрозумілого інтерфейсу, користувачі мають можливість швидко та зручно обирати фільми, місця, а також додавати попкорн і напої до замовлення. Адміністративний функціонал забезпечує просте додавання, видалення фільмів та анонсів, а також надання відповідей на запитання від користувачів .

Таким чином, розроблений додаток відповідає сучасним вимогам до програмного забезпечення, демонструє високу якість, має зручну документацію для різних користувачів і готовий до впровадження в реальну експлуатацію.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Планета Кіно. [Електрон. ресурс]. – Режим доступу: <https://planetakino.ua/>.
2. Планета Кіно. Інтерфейс додатку. [Електрон. ресурс]. – Режим доступу: <https://appadvice.com/app/d0-bf-d0-bb-d0-b0-d0-bd-d0-b5-d1-82-d0-b0-d0-ba-d1-96-d0-bd-d0-be/593841464>.
3. Multiplex. [Електрон. ресурс]. – Режим доступу: <https://multiplex.ua/>.
4. Multiplex. Application interface. [Electronic resource]. – Access mode: <https://multiplex.en.aptoide.com/app>.
5. SmartCinema. [Електрон. ресурс]. – Режим доступу: <https://smartcinema.ua/>.
6. SmartCinema. Application interface. [Electronic resource]. – Access mode: <https://appadvice.com/app/smartcinema/1186208172>.
7. Visual Studio. [Electronic resource]. – Access mode: <https://code.visualstudio.com/>.
8. JetBrains Rider. [Electronic resource]. – Access mode: <https://www.jetbrains.com/>.
9. Eclipse IDE. [Electronic resource]. – Access mode: <https://www.eclipse.org/>.
10. Apache NetBeans. [Electronic resource]. – Access mode: <https://netbeans.apache.org/front/main/index.html>.
11. MVVM (Model-View-ViewModel). [Electronic resource]. – Access mode: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>.
12. MVC (Model-View-Controller). [Electronic resource]. – Access mode: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>.
13. MVP (Model-View-Presenter). [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Model-View-Presenter>.

14. Layered Architecture. [Electronic resource]. – Access mode:
<https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>.
15. MongoDB. [Electronic resource]. – Access mode:
<https://www.mongodb.com/>.
16. MySQL. [Electronic resource]. – Access mode:
<https://www.mysql.com/>.
17. PostgreSQL. [Electronic resource]. – Access mode:
<https://www.postgresql.org/>.
18. SQLite. [Electronic resource]. – Access mode: <https://www.sqlite.org/>.
19. C#. [Электрон. ресурс]. – Режим доступа:
https://uk.wikipedia.org/wiki/C_Sharp.
20. Java. [Электрон. ресурс]. – Режим доступа:
<https://uk.wikipedia.org/wiki/Java>.
21. Python. [Электрон. ресурс]. – Режим доступа:
<https://uk.wikipedia.org/wiki/Python>.
22. C++. [Электрон. ресурс]. – Режим доступа:
<https://uk.wikipedia.org/wiki/C%2B%2B>.

ДОДАТОК А
Технічне завдання

A.1 Вступ

У сучасному світі мобільні додатки відіграють важливу роль у спрощенні повсякденного життя людей, надаючи зручні засоби для вирішення різноманітних завдань. Кіноіндустрія є однією з найбільш популярних сфер дозвілля, що потребує сучасних рішень для поліпшення взаємодії з глядачами.

Цей технічний проєкт передбачає розробку мобільного додатка «PopcornPalms», який забезпечить швидкий доступ до розкладу кіносеансів, зручний процес придбання квитків та замовлення попкорну й напоїв. Додаток дозволить користувачам заощаджувати час, уникаючи черг у касах та автоматизуючи процес вибору місць.

Розробка спрямована на створення інтуїтивно зрозумілого інтерфейсу, що відповідає сучасним стандартам користувацького досвіду, з підтримкою платформ Android та iOS. Додаток також надасть адміністраторам можливість ефективно управляти контентом, включаючи додавання та видалення фільмів із прокату.

Проєкт розробляється відповідно до потреб кінотеатру «PopcornPalms», із врахуванням специфіки його діяльності, що робить додаток унікальним інструментом для покращення взаємодії з клієнтами.

A.2 Підстави для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему: Мобільний додаток купівлі квитків до кінотеатру «PopcornPalms», затверджене наказом Національного університету «Запорізька Політехніка» № 209 від 28 квітня 2025 р.

A.3 Призначення розробки

Мобільний додаток «PopcornPalms» призначений для автоматизації процесів обслуговування клієнтів кінотеатру та спрощення управління адміністративними завданнями.

Основні цілі розробки:

а) забезпечення зручного користувацького досвіду:

1) надання користувачам можливості переглядати актуальний розклад сеансів та трейлери фільмів;

2) забезпечення швидкого процесу придбання квитків із вибором місць у залі;

3) можливість замовлення попкорну та напоїв онлайн для отримання без черг;

б) цифровізація кінотеатру:

1) впровадження інноваційного способу взаємодії з клієнтами за допомогою мобільного додатка;

2) оптимізація роботи касирів та операторів шляхом скорочення кількості офлайн-продажів;

в) підвищення рівня сервісу:

1) надання користувачам персоналізованих пропозицій (наприклад, бонуси чи акції);

2) доступ до профілю з історією придбаних квитків та персональною інформацією;

г) покращення адміністративних процесів:

1) можливість додавання та видалення фільмів та анонсів адміністраторами;

2) надання відповідей на запитання від користувачів;

д) розширення функціоналу:

1) забезпечення доступу до програми для користувачів Android та iOS;

2) інтеграція з популярними платіжними системами (Google Pay, Apple Pay, банківські картки).

A.4 Вимоги до програми чи програмного виробу

Розробка мобільного додатка для придбання квитків до кінотеатру «PorchPalms» базується на забезпеченні чітко визначених вимог, які гарантують його функціональність, сумісність, продуктивність та зручність використання.

Вимоги до програми поділяються на три основні категорії:

- функціональні вимоги, які визначають основні завдання та функції, що виконує додаток. Вони забезпечують можливість користувачам виконувати всі необхідні дії, включаючи авторизацію та реєстрацію, вибір фільмів, місць, сніків, оплату квитків та керування особистим профілем;
- нефункціональні вимоги, що регламентують якість роботи програми. Ці вимоги враховують продуктивність, стабільність, безпеку даних та сумісність додатка з різними платформами й пристроями;
- обмеження, які окреслюють технічні рамки розробки, включаючи вимоги до розміру додатка, необхідності інтернет-з'єднання та інтеграції сторонніх сервісів

A.4.1 Функціональні вимоги

а) робота з користувачами:

1) реєстрація та авторизація користувачів за допомогою email, номеру телефону або соціальних мереж (Google, Facebook, Apple);

б) основний функціонал:

1) перегляд розкладу фільмів у кінотеатрі;

2) вибір місць у залі та замовлення попкорну, напоїв із зазначенням обсягу та типу;

- 3) придбання квитків;
- 4) збереження історії придбаних квитків у профілі користувача;
- в) адміністративний функціонал:
 - 1) додавання та видалення фільмів та анонсів;
 - 2) відповіді на запитання від користувачів;
- г) додаткові можливості:
 - 1) відображення фільму тижня та анонсів майбутніх релізів;
 - 2) система бонусів для залучення клієнтів;
 - 3) відправлення повідомлень адміністратору у випадку проблем.

A.4.2 Нефункціональні вимоги

- а) продуктивність:
 - 1) час відгуку додатка не повинен перевищувати 2 секунд на будь-якій операції;
 - 2) стабільна робота при одночасній взаємодії до 1000 користувачів;
- б) сумісність:
 - 1) підтримка операційних систем Android (версія 8.0 і вище) та iOS (версія 13.0 і вище);
- в) інтерфейс:
 - 1) додаток повинен мати інтуїтивно зрозумілий та адаптивний дизайн;
 - 2) локалізація додатка англійською мовою;
- г) безпека:
 - 1) захист особистих даних користувачів за допомогою шифрування;

д) стабільність:

1) додаток має зберігати роботу у разі відключення інтернет-з'єднання та синхронізувати дані після його відновлення.

A.4.3 Обмеження

- розмір додатка: Загальний розмір файлів для завантаження не повинен перевищувати 100 МБ;
- підключення: Для роботи додатка необхідне стабільне інтернет-з'єднання;
- інтеграція: Використання сторонніх API (наприклад, Google Maps для відображення локації кінотеатру).

A.5 Вимоги до програмної документації

Програмна документація є невід'ємною частиною розробки мобільного додатка для придбання квитків у кінотеатрі «PopcornPalms». Її основне призначення — забезпечити розробників, тестувальників, адміністраторів та кінцевих користувачів необхідною інформацією про функціонування програми, її структуру, вимоги та порядок експлуатації.

Документація повинна відповідати наступним вимогам:

- а) повнота та структурованість: Уся документація має бути чітко структурована, з логічно організованими розділами, що забезпечують повне висвітлення всіх аспектів роботи програми;
- б) актуальність: Вся інформація повинна відповідати актуальному стану програми, включаючи всі внесені зміни чи оновлення;
- в) ясність і зрозумілість: Викладена інформація має бути зрозумілою для кожної цільової аудиторії:
 - 1) для розробників: деталі архітектури, коду, алгоритмів;
 - 2) для тестувальників: чек-листи, тест-кейси, сценарії

тестування;

3) для адміністраторів: інструкції щодо підтримки програми;

4) для користувачів: керівництво з експлуатації;

г) відповідність стандартам: Документація повинна відповідати чинним державним та міжнародним стандартам з програмної інженерії;

д) перелік обов'язкових документів: До комплексу програмної документації належать:

1) технічне завдання: описує мету, вимоги до програми та обмеження;

2) керівництво користувача: інструкція для кінцевих користувачів щодо встановлення та використання програми;

3) опис архітектури програми: висвітлює основні технічні аспекти та структуру додатка;

4) звіт про тестування: результати тестування програми, зокрема функціонального та навантажувального;

е) доступність та інтерактивність: Документація має бути доступна у зручному цифровому форматі (Word, PDF) та інтерактивній формі (вбудовані підказки у додатку).

Виконання цих вимог забезпечує не тільки якісне впровадження та експлуатацію додатку, а й полегшує її обслуговування та розвиток у майбутньому.

A.6 Техніко-економічні показники

Розробка мобільного додатка для придбання квитків до кінотеатру «PopcornPalms» базується на досягненні високих техніко-економічних показників, які забезпечують ефективність, рентабельність та зручність використання програми для кінцевих користувачів і адміністрації кінотеатру. Основні техніко-економічні показники:

а) ефективність розробки:

- 1) термін розробки: мінімізовано завдяки використанню сучасних фреймворків та бібліотек для кроссплатформеного програмування;
- 2) модульна архітектура: дозволяє легко розширювати функціонал програми;

б) економічна ефективність:

- 1) скорочення витрат кінотеатру: автоматизація процесів продажу квитків, вибору місць, вибору снеків, обробки платежів зменшує потребу в залученні додаткового персоналу;
- 2) зростання доходів: завдяки зручності використання додатку прогнозується збільшення кількості придбання квитків і супутніх товарів (попкорн, напої);
- 3) доступність інструментів: використання безкоштовних або доступних ліцензій на технології, що зменшило вартість розробки;

в) технічні показники:

- 1) продуктивність програми: час завантаження додатка не перевищує 2 секунд, а час відповіді на запити — до 500 мс;
- 2) сумісність: додаток працює на пристроях з операційними системами Android (версія 8.0 і вище) та iOS (версія 13.0 і вище);
- 3) енергоефективність: програма оптимізована для мінімального споживання заряду акумулятора;

г) зручність і доступність для користувачів:

- 1) інтуїтивний інтерфейс: мінімізує час навчання користувача;

2) мультимовність: забезпечує доступність програми для широкого кола користувачів;

3) додаткові функції: такі як вибір попкорну, вибір напоїв і можливість перегляду трейлерів, підвищують цінність додатка;

д) потенційні показники доходності:

1) збільшення відвідуваності кінотеатру: очікується приріст продажів квитків до 20% протягом першого року використання додатку;

2) супутні товари: прогнозоване збільшення продажів попкорну та напоїв до 15% через інтеграцію цих функцій у додаток;

е) соціальні показники:

1) зручність для користувачів: скорочення часу на придбання квитків та вибору місць;

2) скорочення черг: завдяки автоматизації процесів продажу;

3) доступність кінотеатру: можливість користування додатком людьми з обмеженими можливостями завдяки інтеграції функцій доступності.

Таким чином, розробка додатку забезпечує баланс між технічними характеристиками, зручністю для користувачів та економічною вигодою для кінотеатру.

A.7 Порядок контролю та приймання

Порядок контролю та приймання мобільного додатка для придбання квитків до кінотеатру «PopcornPalms» передбачає комплексну перевірку його відповідності технічним вимогам, функціональним і нефункціональним характеристикам, а також стандартам якості програмного забезпечення.

A.7.1 Етапи контролю

а) первинне тестування:

На цьому етапі проводиться внутрішнє тестування розробниками для перевірки основних функцій програми. Використовується створений чек-лист, який охоплює всі аспекти функціонування, включаючи:

- 1) авторизацію та реєстрацію користувачів;
- 2) перегляд фільмів та трейлерів;
- 3) вибір місць, вибір додаткових послуг , придбання квитків;
- 4) інтеграцію платіжних систем;
- 5) роботу програми на різних платформах (Android, iOS).

б) кросплатформене тестування:

Додаток перевіряється на сумісність із різними пристроями та версіями операційних систем. Особлива увага приділяється правильності відображення інтерфейсу, продуктивності та коректній роботі на пристроях із різними технічними характеристиками.

в) тестування навантаження:

Аналізується продуктивність додатку під час одночасного використання великою кількістю користувачів. Вимірюється час відгуку серверів, стабільність додатку та можливість обробки великої кількості запитів.

г) фінальне тестування:

Перед прийняттям програми проводиться заключне тестування, яке включає демонстрацію всіх основних функцій замовнику.

A.7.2 Критерії приймання

— повна відповідність функціональним і нефункціональним вимогам, зазначеним у технічному завданні;

- відсутність критичних помилок, які можуть вплинути на функціонування програми;
- стабільна робота додатку в умовах реального навантаження;
- зручність використання та позитивний відгук тестових користувачів.

A.7.3 Документація для приймання

Під час здачі проєкту замовнику передається наступна документація:

- інсталяційні файли програми для Android та iOS;
- керівництво користувача, яке описує функціональні можливості програми;
- тестова документація, включаючи чек-лист із результатами перевірок;
- технічна документація, що містить опис архітектури та ключових компонентів програми.

A.7.4 Процедура приймання

Замовник проводить незалежну перевірку роботи додатку відповідно до наданої документації. У разі успішного проходження всіх тестів і відсутності зауважень здійснюється підписання акту приймання.

Цей порядок контролю та приймання дозволяє забезпечити високу якість програмного продукту, відповідність його функціоналу заявленим вимогам та готовність до використання кінцевими користувачами.

ДОДАТОК Б
Текст програми

Б.1 – Логіка роботи адміністратора

Б.1.1 – AdminProfile.xaml.cs

```

using Project.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Project.Views
{
    /// <summary>
    /// Interaction logic for AdminProfile.xaml
    /// </summary>

    public partial class AdminProfile : Page
    {
        DataAccess db;
        public AdminProfile()
        {
            InitializeComponent();
            db = new DataAccess();
            OutInfoUser();

            //Time
            DateTime currentTime = DateTime.Now;
            string formattedTime = currentTime.ToString("HH:mm");
            Time.Content = formattedTime;
        }

        private void OutInfoUser()
        {
            profileName.Content = "Admin";
            profileNumber.Content = "Admin";
        }
    }
}

```

Б.2 – Логіка роботи користувача

Б.2.1 – AllTicketUser.xaml.cs

```

using Project.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using static Project.Tickets;

namespace Project.Views
{
    /// <summary>
    /// Interaction logic for AllTicketUser.xaml
    /// </summary>
    public partial class AllTicketUser : Page
    {
        DataAccess db;
        public AllTicketUser()
        {
            InitializeComponent();
            db = new DataAccess();
            LoadUserTickets();
        }
        private async void LoadUserTickets()
        {
            string userId = UserSession.UserId;

            var userTickets = await db.FindAllTicketsByUserId(userId);

            if (userTickets.Any())
            {
                FilmsListBox.ItemsSource = userTickets;
            }
            else
            {
                TextBlock1Not.Text = "Warning";
                TextBlock2Not.Text = "You don't have purchased tickets yet";
            }
        }
    }
}

```

```

        ButtonNot.Content = "Ok";

        BorderNotification.Visibility = Visibility.Visible;
        GridNot.Visibility = Visibility.Visible;
    }
}

private void FilmsListBox_SelectionChanged(object sender,
SelectionChangedEventArgs e)
{

}

private void ButtNot_Click(object sender, RoutedEventArgs e)
{
    BorderNotification.Visibility = Visibility.Hidden;
    GridNot.Visibility = Visibility.Hidden;

    Profile profile = new Profile();
    Frame mainFrame = Application.Current.MainWindow.Content as Frame;
    mainFrame.NavigationService.Navigate(profile);
}
}
}

```

B.2.2 – ContactUs.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Project
{
    /// <summary>
    /// Interaction logic for ContactUs.xaml
    /// </summary>
    public partial class ContactUs : Page
    {
        public ContactUs()
        {
            InitializeComponent();

```

```

        //Time
        DateTime currentTime = DateTime.Now;
        string formattedTime = currentTime.ToString("HH:mm");
        Time.Content = formattedTime;
    }
}
}

```

B.2.3 – Description.xaml.cs

```

using Project.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Project
{
    /// <summary>
    /// Interaction logic for Description.xaml
    /// </summary>
    public partial class Description : Page
    {
        private bool isMediaOpened = false;
        private bool isFirstImage = true;
        DataAccess db;

        public Description(string imagePath, string filmDescription, string nameofFilm,
string trailerPath,
        string yearGenre, string directors, string stars, string sessionTime, string
movieGrade)
        {
            InitializeComponent();
            db = new DataAccess();

            //Time
            DateTime currentTime = DateTime.Now;
            string formattedTime = currentTime.ToString("HH:mm");
            Time.Content = formattedTime;

```

```

        mediaElement.Source = new Uri(trailerPath, UriKind.RelativeOrAbsolute);
        Image.Source = new BitmapImage(new Uri("D:/Универ/4 курс/1
семестр/PopcornPalms/Project/Project/Photos/PlayerPlay.png", UriKind.Absolute));

        photoFilm.Source = new BitmapImage(new Uri(imagePath,
UriKind.RelativeOrAbsolute));
        FilmDescriptionTextBlock.Text = filmDescription;
        NameOfFilmDesc.Text = nameofFilm;
        outYearGenre.Text = yearGenre;
        outDirectors.Text = directors;
        outStars.Text = stars;
        outSessionTime.Text = sessionTime;
        outMovieGrade.Source = outMovieGrade.Source = new BitmapImage(new
Uri(movieGrade, UriKind.RelativeOrAbsolute));
    }

    public void UpdateButtonState(string newText, bool isEnabled)
    {
        BuyTicketButt.Content = newText;
        BuyTicketButt.IsEnabled = isEnabled;
    }

    private void Play_Click(object sender, RoutedEventArgs e)
    {
        if (isFirstImage)
        {
            Image.Source = new BitmapImage(new Uri("D:/Универ/4 курс/1
семестр/PopcornPalms/Project/Project/Photos/PlayerPause.png", UriKind.Absolute));
            mediaElement.Play();
        }
        else
        {
            Image.Source = new BitmapImage(new Uri("D:/Универ/4 курс/1
семестр/PopcornPalms/Project/Project/Photos/PlayerPlay.png", UriKind.Absolute));

            mediaElement.Pause();
        }
        isFirstImage = !isFirstImage;
    }

    private void Button_Click_1(object sender, RoutedEventArgs e)
    {
        ChoosePlace choosePlace = new ChoosePlace(NameOfFilmDesc.Text);
        Frame mainFrame = Application.Current.MainWindow.Content as Frame;
        mainFrame.NavigationService.Navigate(choosePlace);
    }

    private void Button_Click(object sender, RoutedEventArgs e)
    {
        FilmDescriptionTextBlock.Visibility = Visibility.Hidden;
        mediaElement.Visibility = Visibility.Visible;
    }

```

```

        PlayvsPause.Visibility = Visibility.Visible;
        CloseMedia.Visibility = Visibility.Visible;
    }

    private void CloseMedia_Click(object sender, RoutedEventArgs e)
    {
        FilmDescriptionTextBlock.Visibility = Visibility.Visible;
        mediaElement.Visibility = Visibility.Hidden;
        PlayvsPause.Visibility = Visibility.Hidden;
        CloseMedia.Visibility = Visibility.Hidden;
    }
}

```

B.2.4 – Home.xaml.cs

```

using Microsoft.Win32;
using Project.Models;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using static System.Net.WebRequestMethods;
using MongoDB.Driver;

namespace Project
{
    /// <summary>
    /// Interaction logic for Home.xaml
    /// </summary>
    public partial class Home : Page
    {
        DataAccess db;
        public Home()
        {
            InitializeComponent();

            //Time

```

```

DateTime currentTime = DateTime.Now;
string formattedTime = currentTime.ToString("HH:mm");
Time.Content = formattedTime;

db = new DataAccess();
LoadFilms();
LoadAnnouncements();
}
private async void LoadFilms()
{
    List<Films> films = await db.GetAllFilms();

    FilmsListBox.ItemsSource = films;
}
private async void LoadAnnouncements()
{
    List<Announcements> announcements = await db.GetAllAnnouncements();

    AnnouncementsListBox.ItemsSource = announcements;
}
private async void FilmButton_Click(object sender, RoutedEventArgs e)
{
    var selectedFilm = (Films)((Button)sender).Tag;

    var dataAccess = new DataAccess();
    var descriptionFilm = await
dataAccess.FindDescriptionFilmByName(selectedFilm.NameOfFilm);

    Button clickedButton = sender as Button;
    if (clickedButton != null && clickedButton.Tag is Films selectedddFilm)
    {
        var nameFilm = selectedddFilm.NameOfFilm;
        ChoosePlace choose = new ChoosePlace(nameFilm);
        ChoosePlace placeSelectionForm = new
ChoosePlace(selectedFilm.NameOfFilm.ToString());
        NavigationService.Navigate(new Description(descriptionFilm.PhotoFilm,
selectedddFilm.Description, selectedFilm.NameOfFilm, selectedddFilm.VideoTrailer,
selectedddFilm.DateAndGenre, selectedddFilm.Directors, selectedddFilm.Stars,
selectedddFilm.SessionTime, selectedddFilm.MovieGrade));
    }
}
private async void AnnouncementsButton_Click(object sender, RoutedEventArgs e)
{
    var selectedFilm = (Announcements)((Button)sender).Tag;

    var dataAccess = new DataAccess();
    var descriptionFilm = await
dataAccess.FindDescriptionFilmByName(selectedFilm.NameOfFilm);

    Button clickedButton = sender as Button;
    if (clickedButton != null && clickedButton.Tag is Announcements
selectedddFilm)

```

```

        {
            var nameFilm = selectedddFilm.NameOfFilm;
            ChoosePlace choose = new ChoosePlace(nameFilm);
            ChoosePlace placeSelectionForm = new
ChoosePlace(selectedFilm.NameOfFilm.ToString());
            NavigationService.Navigate(new Description(descriptionFilm.PhotoFilm,
selectedddFilm.Description, selectedFilm.NameOfFilm, selectedddFilm.VideoTrailer,
selectedddFilm.DateAndGenre, selectedddFilm.Directors,
selectedddFilm.Stars, selectedddFilm.SessionTime, selectedddFilm.MovieGrade));
            Description descriptionPage = new Description(descriptionFilm.PhotoFilm,
selectedddFilm.Description, selectedFilm.NameOfFilm, selectedddFilm.VideoTrailer,
selectedddFilm.DateAndGenre, selectedddFilm.Directors,
selectedddFilm.Stars, selectedddFilm.SessionTime, selectedddFilm.MovieGrade);
            NavigationService.Navigate(descriptionPage);

            descriptionPage.UpdateButtonState("Release date: 26.06.2025", false);
        }
    }
}

```

B.2.5 – Login.xaml.cs

```

using Project.Models;
using Project.Views;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using static Project.Tickets;

namespace Project
{
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Page
    {
        DataAccess db;
        public Login()
    }
}

```

```

{
    InitializeComponent();
    db = new DataAccess();

    //Time
    DateTime currentTime = DateTime.Now;
    string formattedTime = currentTime.ToString("HH:mm");
    Time.Content = formattedTime;
}
private async void LogIn()
{
    try
    {
        var user = await db.FindUser(input_login.Text, input_password.Text);
        if (input_login.Text == "admin" || input_password.Text == "Admin")
        {
            AdminMain adminMain = new AdminMain();
            Frame mainFrame = Application.Current.MainWindow.Content as Frame;
            mainFrame.NavigatonService.Navigate(adminMain);
        }
        else if(user != null)
        {
            UserSession.UserId = user.Id;
            TemporaryUser.Id = user.Id;
            TemporaryUser.PIB = user.PIB;
            TemporaryUser.Password = user.Password;

            Home home = new Home();
            Frame mainFrame = Application.Current.MainWindow.Content as Frame;
            mainFrame.NavigatonService.Navigate(home);
        }
        else
        {
            TextBlock1Not.Text = "Warning";
            TextBlock2Not.Text = "User not found";
            BorderNotification.Visibility = Visibility.Visible;
            GridNot.Visibility = Visibility.Visible;
        }
    }
    catch (Exception er)
    {
        MessageBox.Show(er.Message, "Warning", MessageBoxButton.OK,
        MessageBoxImage.Warning);
    }
}
private void Button_Click(object sender, RoutedEventArgs e)
{
    LogIn();
}
private void ButtNot_Click(object sender, RoutedEventArgs e)
{
    BorderNotification.Visibility = Visibility.Hidden;
}

```

```
GridNot.Visibility = Visibility.Hidden;  
  
input_login.Text = "";  
input_password.Text = "";  
}}
```

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Слайд №1

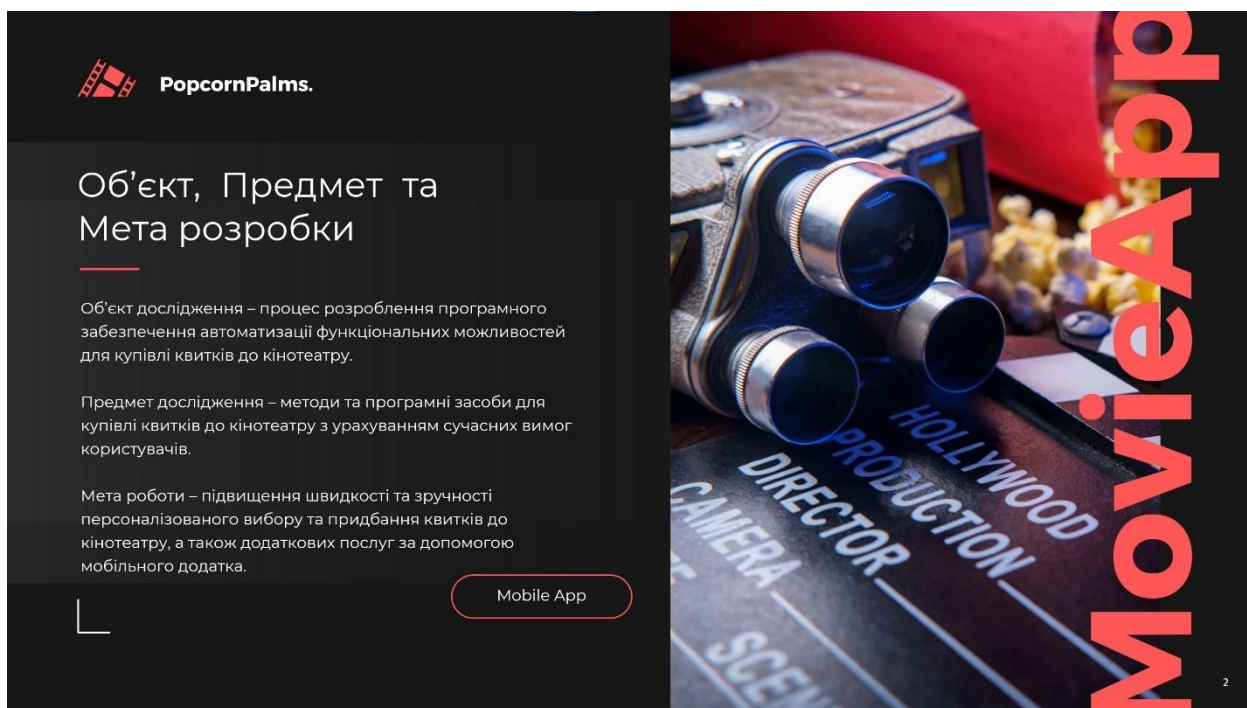


Рисунок В.2 – Слайд №2

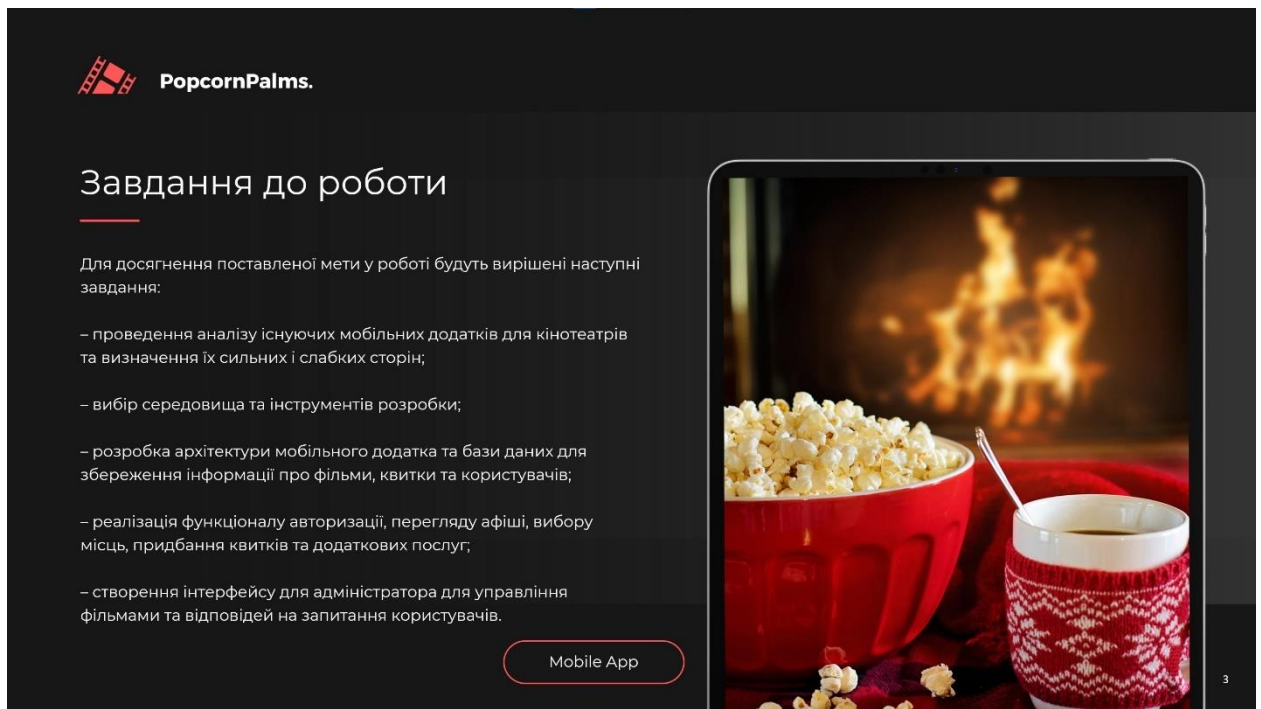


Рисунок В.3 – Слайд №3

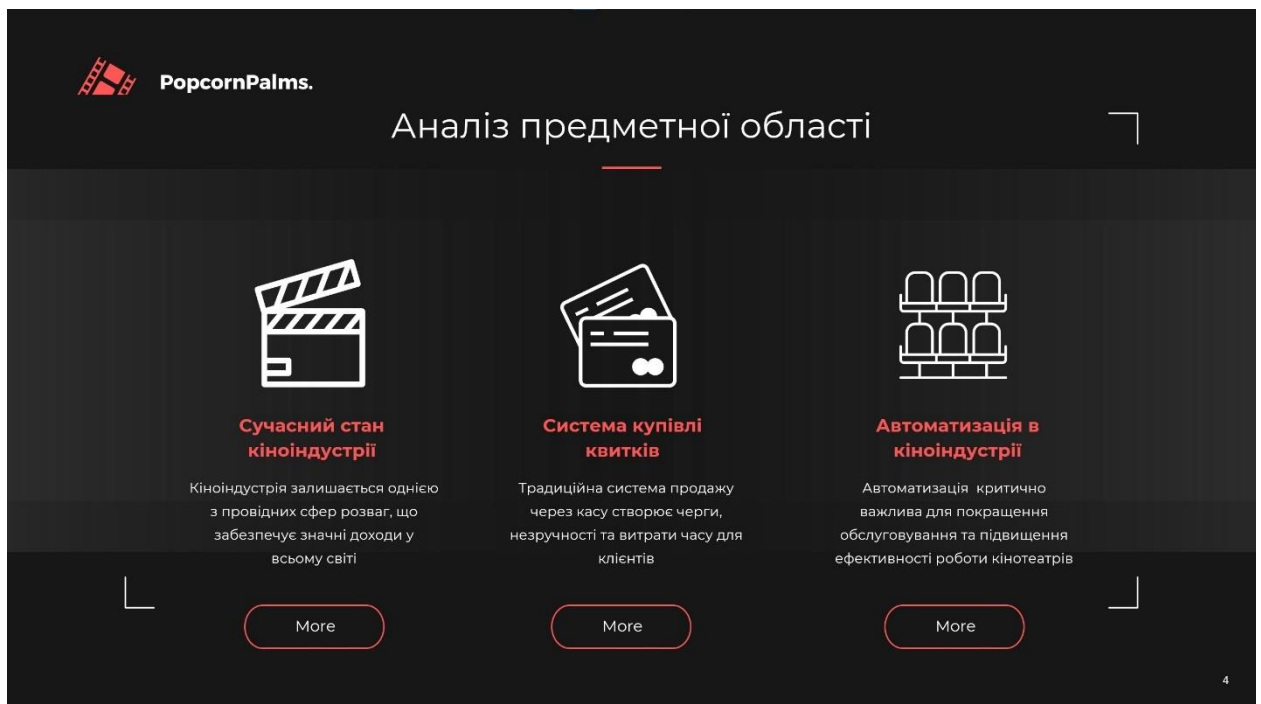


Рисунок В.4 – Слайд №4



PopcornPalms.

Порівняння існуючих аналогів ПЗ

Критерії	Планета Кіно	Multiplex	Smart Cinema	Popcorn Palms
Інтерактивний вибір місць	+	+	–	+
Інтеграція з додатковими послугами	–	–	–	+
Перегляд придбаних квитків	–	–	–	+
Програма лояльності	+	–	+	+
Доступний інтерфейс	+	+	–	+
Трейлери та детальна інформація про фільми	+	+	–	+

5

Рисунок В.5 – Слайд №5



PopcornPalms.

Порівняння середовищ розробки

Критерії	JetBrains Rider	Eclipse	NetBeans	Visual Studio
Підтримка WPF	+	–	–	+
Зручна робота з XAML	+	+	–	+
Підтримка .NET	+	–	–	+
Робота з базами даних	+	–	–	+
Підтримка багатомовності	+	–	+	+
Інтеграція з іншими технологіями .NET	+	–	–	+
Простота використання	–	–	–	+
Розширюваність через плагіни	+	–	+	+

6

Рисунок В.6 – Слайд №6



PopcornPalms.

Порівняння архітектур проектування

Критерії	MVC	MVP	Layered Architecture	MVVM
Підходить для складних UI	+	+	–	+
Простота тестування	–	+	–	+
Легка інтеграція в мобільні додатки	–	+	–	+
Відсутня складність для простих проєктів	+	–	+	+
Масштабованість	+	–	+	+

7

Рисунок В.7 – Слайд №7



PopcornPalms.

Порівняння баз даних

Критерії	MySQL	PostgreSQL	SQLite	MongoDB
Масштабованість	–	+	–	+
Гнучкість	–	+	+	+
Продуктивність	+	–	+	+
Надійність	+	+	–	+
Легкість в налаштуванні	+	–	+	+
Підтримка складних запитів	–	+	–	+

8

Рисунок В.8 – Слайд №8



PopcornPalms.

Порівняння мов програмування

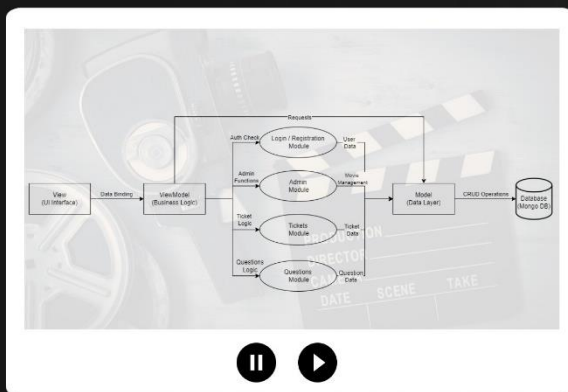
Критерії	Java (JavaFX)	Python	C++ (Qt)	C# (WPF)
Продуктивність	+	–	+	+
Простота використання	+	+	–	+
Підтримка інтерфейсу (UI)	+	–	+	+
Розробка великих додатків	+	–	+	+
Підтримка архітектури MVVM	–	–	–	+
Швидкість розробки	–	+	–	+

9

Рисунок В.9 – Слайд №9



PopcornPalms.



Структурна схема програми

Програма побудована за архітектурою MVVM (Model-View-ViewModel), яка передбачає розділення логіки програми, даних та інтерфейсу користувача. Основні компоненти:

1. View
2. ViewModel
3. Model

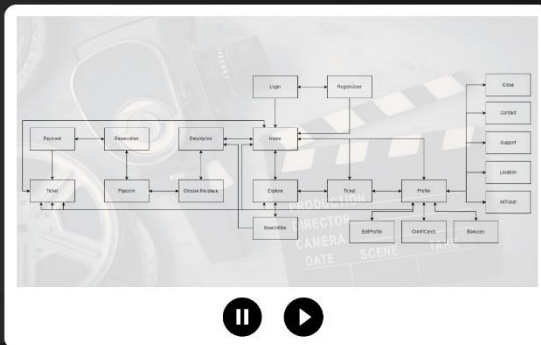
10

Рисунок В.10 – Слайд №10

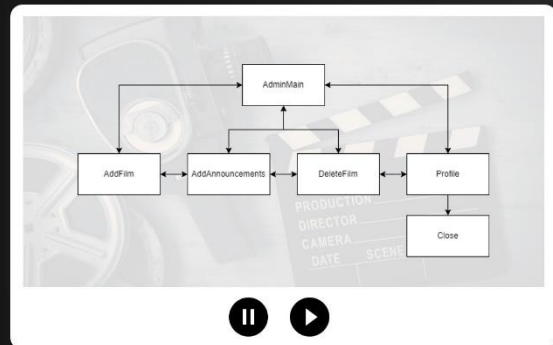


PopcornPalms.

Схеми функціонування програми



Алгоритм роботи користувача



Алгоритм роботи адміністратора

Рисунок В.11 – Слайд №11



PopcornPalms.

Діаграма класів

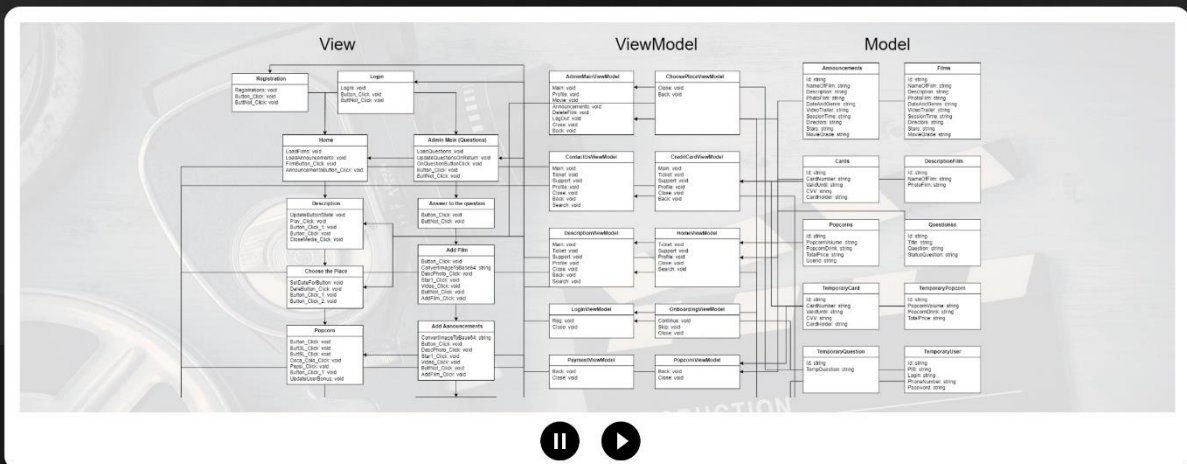
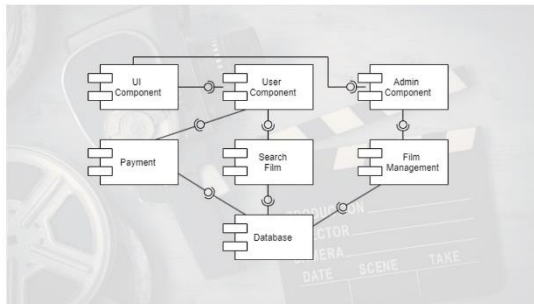


Рисунок В.12 – Слайд №12



PopcornPalms.



Діаграма компонентів

Діаграма компонентів демонструє модульну архітектуру програми через основні компоненти, їхню структуру та взаємозв'язки між ними. Основні компоненти:

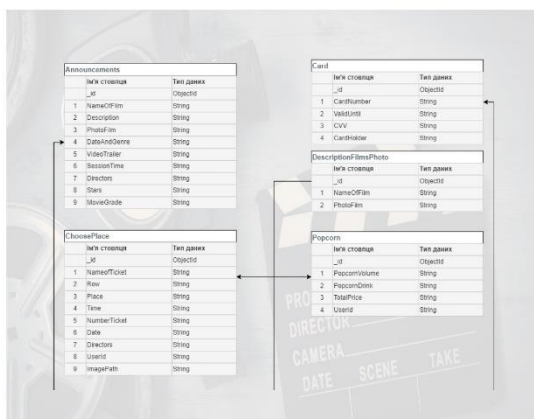
1. UI Component
2. User Component
3. Admin Component
4. Search Film
5. Film Management
6. Payment
7. Database

13

Рисунок В.13 – Слайд №13



PopcornPalms.



Концептуальна модель БД

Концептуальна модель бази даних є основою для проектування системи зберігання даних, необхідної для функціонування програми. Основні сутності:

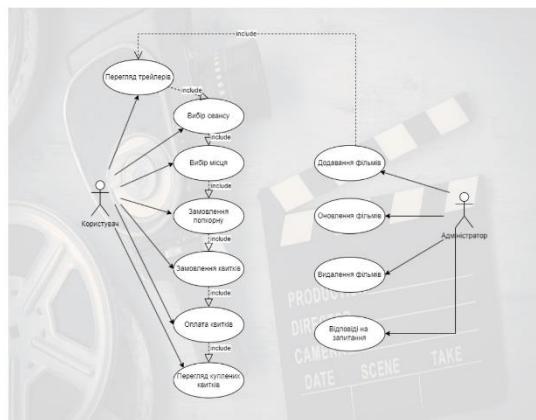
1. Announcements
2. ChoosePlace
3. Card
4. DescriptionFilmsPhoto
5. Popcorn

14

Рисунок В.14 – Слайд №14



PopcornPalms.



Діаграма прецедентів

На основі діаграми прецедентів можна зрозуміти, які функції виконує система для користувачів і адміністратора. Потік подій:

1. Основний потік
2. Помилковий потік

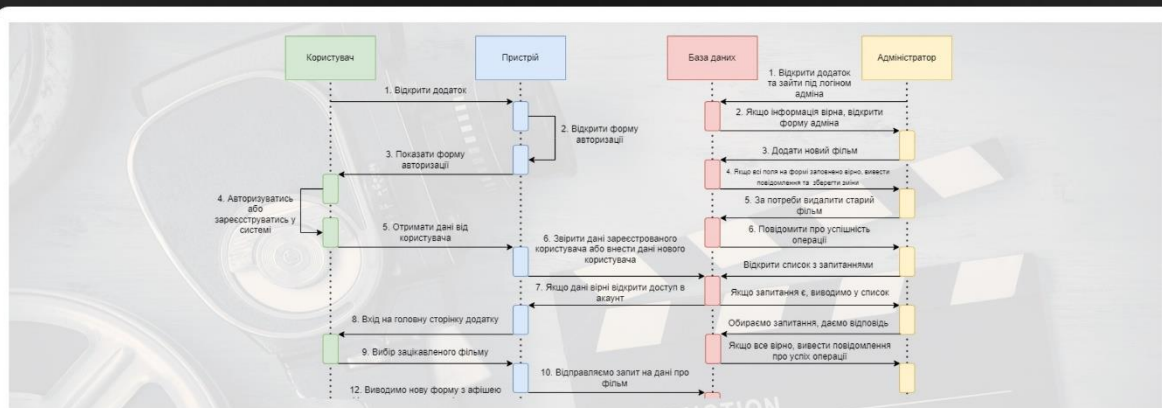
15

Рисунок В.15 – Слайд №15



PopcornPalms.

Діаграма послідовності



16

Рисунок В.16 – Слайд №16

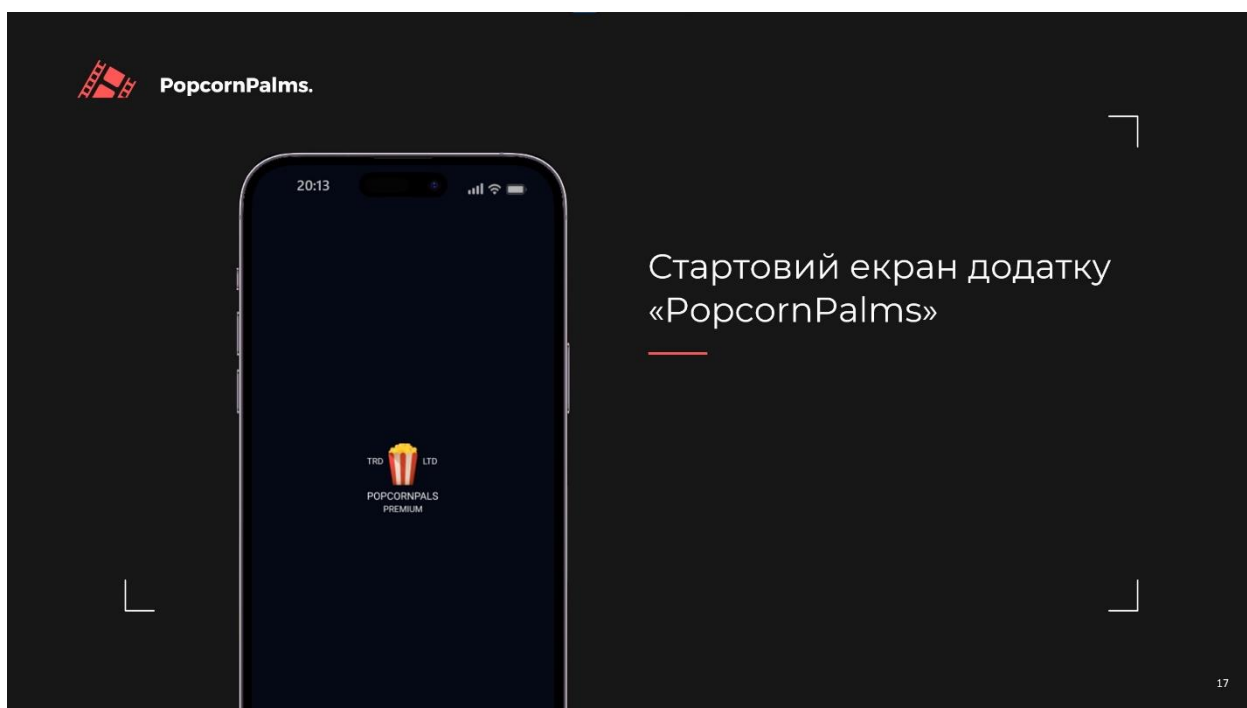


Рисунок В.17 – Слайд №17

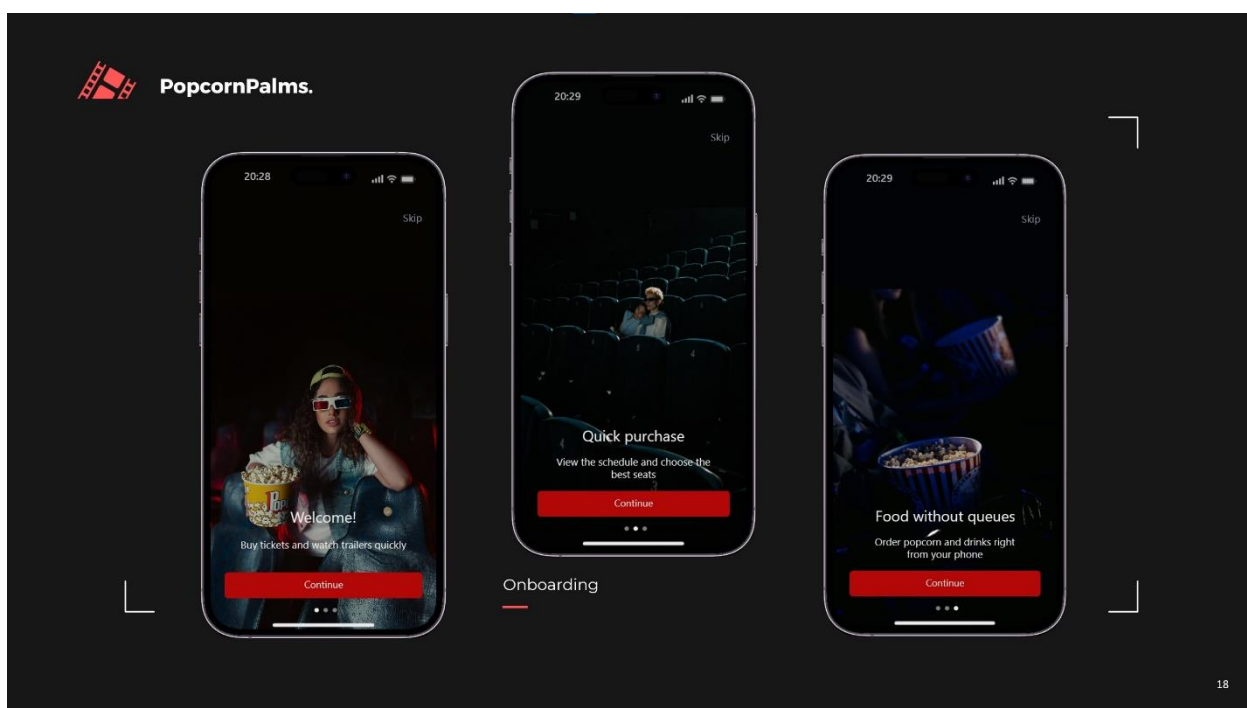


Рисунок В.18 – Слайд №18

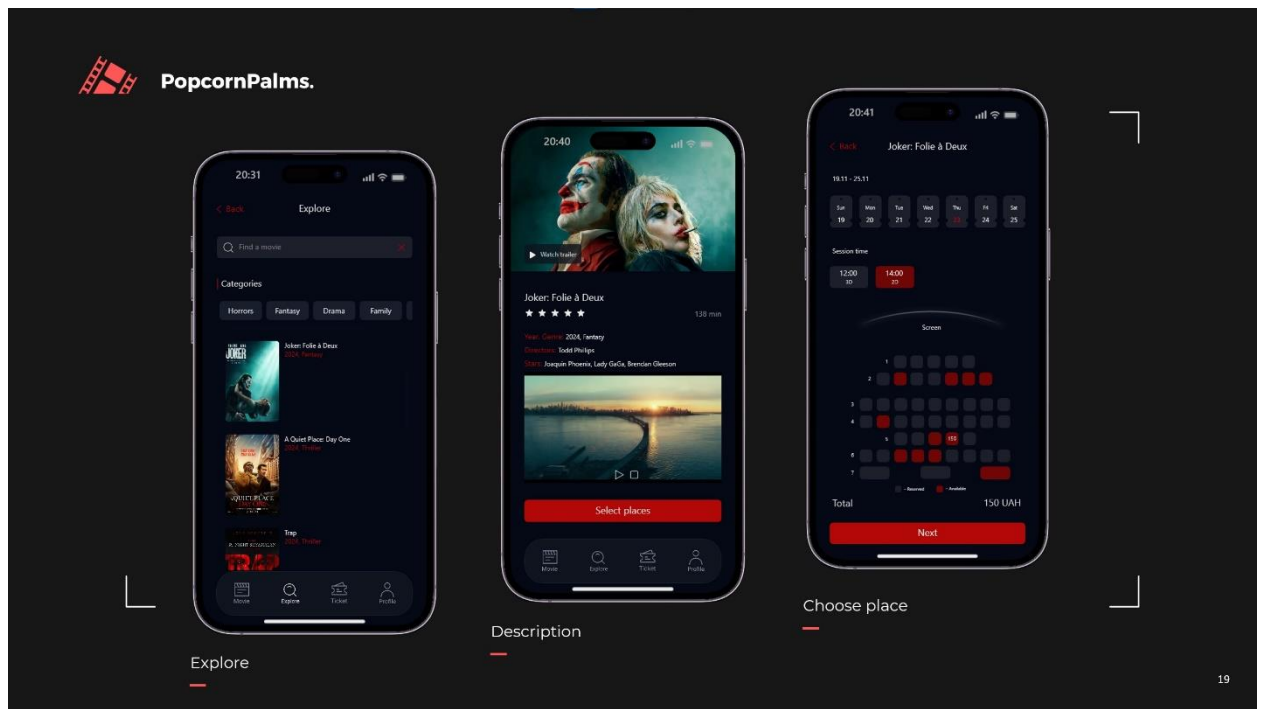


Рисунок В.19 – Слайд №19

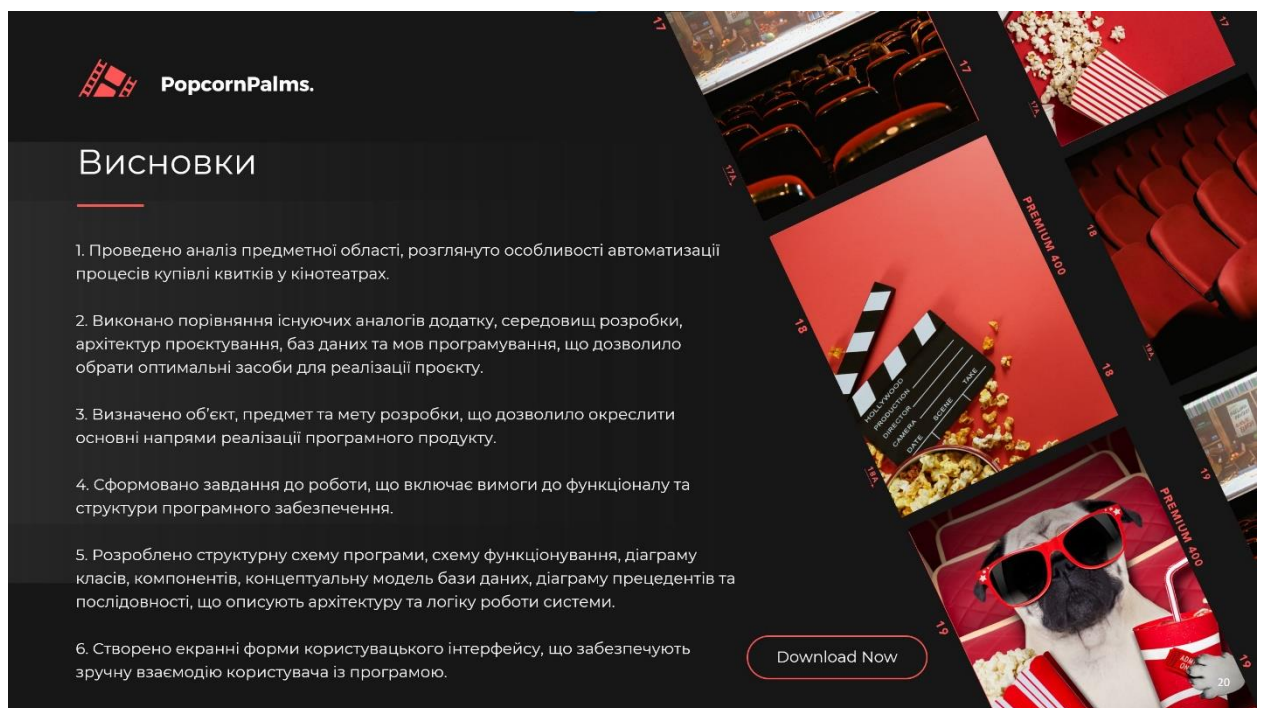


Рисунок В.20 – Слайд №20