

Міністерство освіти і науки України
Одеський національний технологічний університет
Кафедра комп'ютерної інженерії



**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему Розробка мобільного додатку для заказу
(назва кваліфікаційної роботи згідно наказу ОНТУ)
театральних квитків

Здобувача Казимірова Д.С.
(прізвище, ініціали)

4 курсу 542a групи

Керівники: ст. викл. Слушина Н.В.
(посада, прізвище та ініціали)

д.т.н. проф. Артеменко С.В.
(посада, прізвище та ініціали)

Консультанти: _____
(посада, прізвище та ініціали)

д.е.н., проф. Басюркіна Н.Й.
(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від 10.06 2023 р., протокол № 8

Завідувач кафедри комп. інженерії _____ Сергій АРТЕМЕНКО
(назва кафедри) (підпис) (Ім'я ПРІЗВИЩЕ)

Одеса - 2023 рік

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерної інженерії, програмування та кіберзахисту
Кафедра комп'ютерної інженерії
Ступінь вищої освіти бакалавр
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма Розробка ігор та інтерактивних медіа у віртуальній реальності

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної інженерії
Сергій АРТЕМЕНКО
« 10 » серпня 2022 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Казимірова Дениса Сергійовича

1. Тема роботи Розробка мобільного додатку для заказу театральних квитків

Затверджена наказом університету від « 10 » серпня 2022 р., наказ № 440-03

2 Термін здачі здобувачем закінченої роботи 5 червня 2023 р.

3. Вихідні дані
роботи

1. Редактор коду з інтегрованим середовищем Android Studio

2. Вбудований емулятор/Мобільний телефон 3. Power Point 4. Microsoft Word

4. Перелік питань, які потрібно розробити

1. Вступ. 2. Дослідження предметної області. 3. Проектування додатку.

4. Моделювання мобільного додатку 5. Створення програми

6. Економічні розрахунки. 7. Охорона праці. 8. Загальні висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайд 1. Титульний слайд Слайд 2. Постановка задачі. Слайд 3. Актуальність Слайд 4.

Аналіз предметної області. Слайд 5. Аналіз існуючих аналогів. Слайд 6. Проектування

Додатку Слайд 7,8,9,10,11,12. Розробка компонентів системи Слайд 13,14. Висновки

Слайд 15. Дякую за увагу

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Економіка</i>	<i>Басюркіна Н.Й., д.е.н., проф.</i>		
<i>Охорона праці</i>	<i>Жуковецька С.Л., ст. викладач</i>		
<i>Нормоконтроль</i>	<i>Жуковецька С.Л., ст. викладач</i>		

7. Дата видачі завдання 30.09.2022

Керівники _____ *Наталя СЛУШНА*
_____ *Сергій АРТЕМЕНКО*
Завдання прийняв до виконання _____ *Денис КАЗИМІРОВ*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	<i>Аналіз предметної області</i>	<i>13.02.2023</i>	
2.	<i>Аналіз існуючих аналогів</i>	<i>05.03.2023</i>	
3.	<i>Проектування системи</i>	<i>20.03.2023</i>	
4.	<i>Розробка компонентів системи</i>	<i>15.04.2023</i>	
5.	<i>Розробка призначеного для користувача інтерфейсу</i>	<i>11.05.2023</i>	
6.	<i>Розробка економічної частини</i>	<i>25.05.2023</i>	
7.	<i>Розробка розділу охорони праці</i>	<i>29.05.2023</i>	
8.	<i>Оформлення пояснювальної записки</i>	<i>05.06.2023</i>	
9.	<i>Оформлення презентації</i>	<i>05.06.2023</i>	

Керівники роботи _____ *Наталя СЛУШНА*
_____ *Сергій АРТЕМЕНКО*

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач - дипломник _____ *Денис КАЗИМІРОВ*

АНОТАЦІЯ

Роботу присвячено проектуванню та розробці мобільного додатку для заказу театральних квитків. Проаналізовано предметну область, розглянуто існуючі сучасні аналоги мобільних додатків. Висвітлено їх переваги та недоліки. Розроблено усі компоненти системи. Розроблено демонстраційний варіант додатку для заказу театральних квитків та проведено його тестування.

У розділі пов'язаному з економічною частиною обґрунтовано економічну вигоду та ефективність застосування додатка.

У розділі щодо охорони праці досліджено питання безпеки роботи та охорони праці.

Результатом даної роботи є створений додаток для заказу театральних квитків для користувачів

Ключові слова: мобільні додатки, *Android Studio*, база даних.

ABSTRACT

The work is devoted to the design and development of a mobile application for ordering theater tickets. The subject area was analyzed, existing modern analogues of mobile applications were considered. Their advantages and disadvantages are highlighted. All system components have been developed. A demo version of the application for ordering theater tickets was developed and tested.

In the section related to the economic part, the economic benefits and effectiveness of the application are substantiated.

In the section on occupational health and safety, the issue of occupational safety and health is investigated.

The result of this work is the created application for ordering theater tickets for users

Keywords: *mobile applications, Android Studio, database.*

ЗМІСТ

	Стор.
ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ДОСЛІДЖЕННЯ ІСНУЮЧИХ АНАЛОГІВ	12
1.1 Аналіз предметної області.....	12
1.1.1 Методи вирішення аналогічних завдань	12
1.2 Існуючі аналоги	13
1.2.1 <i>Ticketmaster</i>	13
1.2.2 <i>StubHub</i>	14
1.2.3 <i>TodayTix</i>	15
1.2.4 <i>Bilet.ua</i>	17
1.3 Постановка завдання.....	18
Висновок до першого розділу	18
РОЗДІЛ 2 ПРОЕКТУВАННЯ ДОДАТКУ	19
2.1 Вибір і обґрунтування базового варіанту побудови системи.....	19
2.1.1 Технології та мови програмування	19
2.1.2 Архітектура системи.....	25
2.2 Загальна характеристика інформаційних масивів	26
2.2.1 Типи даних.....	26
2.2.2 Джерела даних.....	26
2.2.3 Структура даних.....	26
2.2.4 Обсяг даних	27
2.3 Діаграма інформаційних подій	27

					КРБ.КІ.1.440-03.1.16			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка мобільного додатку для заказу театральних квитків	Літ.	Арк.	Аркушів
<i>Розробив</i>		<i>Денис КАЗИМІРОВ</i>					6	102
<i>Перевірів</i>		<i>Наталія СЛУШНА</i>						
<i>Рецензент</i>		<i>Марина БУЛЮК</i>						
<i>Нормоконтроль</i>		<i>Світлана ЖУКОВЕЦЬКА</i>						
<i>Затвердив</i>		<i>Сергій АРТЕМЕНКО</i>				ар. 542а, ОНТУ		

2.4	Проектування структури системи	28
	Висновок до другого розділу	31
РОЗДІЛ 3 РОЗРОБКА КОМПОНЕНТІВ СИСТЕМИ		32
3.1	Використання засобів розробки	32
3.2	Розробка системи	33
3.2.3	Розробка статичної частини системи	34
3.2.4	Створення баз даних і таблиць	36
3.2.5	Створення екранних форм	40
3.2.6	Розробка динамічної частини системи	45
	Висновок до третього розділу	49
РОЗДІЛ 4 ОПИС РОБОТИ ЗАСТОСУНКУ		50
4.1	Реєстрація та авторизація користувача у системі	50
4.2	Бічне меню додатка та основні його сторінки	53
4.3	Замовлення та перегляд заброньованого театрального білету	55
4.4	Сторінка театрів	60
4.5	Налаштування додатку	62
	Висновок до четвертого розділу	63
РОЗДІЛ 5 ЕКОНОМІЧНІ РОЗРАХУНКИ		64
5.1	Аналіз економічної частини	64
5.2	Маркетинговий ефект	66
5.3	Науково-технічний, соціальний та економічний вплив	67
5.4	Розрахунок коефіцієнта оцінки науково-технічної ефективності	68
	Висновок до п'ятого розділу	73
РОЗДІЛ 6 ОХОРОНА ПРАЦІ		74
6.1	Робота за комп'ютером	74
6.2	Ергономіка робочого місця	74
6.3	Електробезпека	75
6.4	Класифікація виробництв за ступенем вибухової, вибухопожежної і пожежної небезпеки	77

6.4.1	Визначення понять.....	77
6.4.2	Класифікація виробництв.....	77
6.4.3	Пожежна профілактика	78
6.4.4	Рекомендації	78
6.5	Профілактика промислової санітарії.....	79
6.5.1	Санітарно-гігієнічні умови праці	79
6.5.2	Ергономіка робочого місця.....	79
6.5.3	Запобігання шуму	80
6.5.4	Профілактика втоми	80
6.5.5	Виконання вимог Закону України.....	80
	Висновок до шостого розділу	80
	ЗАГАЛЬНІ ВИСНОВКИ	82
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83
	ДОДАТКИ.....	86
	Додаток Б Код додатку	86
	Додаток А Презентація.....	95

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дат		

ВСТУП

В сучасному світі майже у кожної людини є мобільний телефон, отже створення мобільного застосунку будь-якого призначення та масштабу є дуже актуальною темою. Хоча для ПК існує більша кількість програм, ніж на мобільні засоби, та ще й більшість телефонних застосунків є спрощеними варіантами таких самих програм, але створених для ПК платформ, однак задля простих, побутових, розважальних та інших програм, які полегшують життя користувачів, створення мобільного застосунка є найбільш коректним варіантом вирішення вимоги користувача. Доцільність використання мобільних застосунків у порівнянні з програмами створеними тільки для ПК платформ полягає у простоті використання.

Яскравими прикладами таких застосунків є:

- банківські застосунки, завдяки яким користувачі можуть з легкістю проводити грошові операції;
- різні навігатори, месенджери і т.д.;
- поштові застосунки, а також застосунки щодо замовлення їжі, кіно- та театральних квитків.

Головна мета при створенні мобільного застосунку типу застосунку для заказу театральних квитків полягає в простоті *UI* та в зрозумілому функціоналі програми, так як даного типу ПЗ перш за все має виконувати свою головну мету – допомагати користувачам найпростішим, найшвидшим та найзрозумілішим способом знаходити необхідну театральну постанову та замовити на неї білети.

Серед великої кількості застосунків даного призначення, є невелика кількість прикладів, в яких можна знайти лише незначні недоліки. В основному, такі застосунки мають доволі тривіальні незручності та недоліки:

- погано пропрацьований каркас програми (маються на увазі функції, які подаються користувачу);

					КРБ.КІ.1.440-03.1.16	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

- не *user-friendly* інтерфейс програми: перше враження користувача має велику цінність для того, щоб таким застосунком взагалі користувались.

Наприклад, якщо користувачу дуже не сподобається візуальна частина застосунку, то він навіть не буде заморочуватися спробувати функціонал та замовити квитки, а просто знайде іншу альтернативу такому застосунку.

1. Проблеми з оплатою: деякі застосунки або часто, або постійно мають проблеми з оплатою користувачами квитків. Це може бути пов'язано з проблемами та помилками на стороні програми, що стосується передачі даних між банківською системою та базою даних застосунку. Також можуть виникнути незручності з оплатою, наприклад якщо користувач бажає зробити оплату завдяки банку, яким він користується все життя, а у застосунку не має в варіантах оплати цього банку.

2. Недостатня кількість наданої інформації користувачу щодо події: деякі застосунки описують доволі мало деталей, а іноді навіть й деяких деталей не описують. Тобто користувач має сам дізнатися, наприклад, щодо групи, яка буде виступати або передісторії розвитку театру і т.д. З цієї проблеми витікає останній недолік мобільних застосунків такої тематики.

3. Недовіра користувача щодо застосунку (Інформаційна безпека): у Інтернет-просторі є безмежна кількість шахраїв, які в кожний момент часу готові обманути та вкрасти гроші людей, або особисті дані.

Для вирішення даних проблем або уникнення їх у майбутньому, необхідно спочатку архітектурно описувати застосунок, а також мати на увазі все основні проблематичні моменти при його створенні, тому що для того, щоб застосунок мав місце для життя у просторі користувацьких мобільних програм, необхідно догоджати клієнтів.

Зазвичай, у таких застосунках клієнти шукають простоти інтерфейсу (в ідеалі – тематичного оформлення щодо окремих театральних постанов при переході на сторінку опису події), зрозумілі та чіткі шляхи для переходу між

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дат		

сторінками та частинами застосунку, а також опис цих самих подій, на які люди мають бажання заказать квитки.

Щодо технічної складової таких застосунків, можна вирізнити основні та найбільш популярні інструментарії, бібліотеки та програми, які використовуються для створення систем заказу театральних квитків:

1. *IOS* та *Android*: новітні та найбільш клієнто-орієнтовані застосунки розробляються для двох основних платформ *IOS* та *Android*. Якщо користувач використовує систему *Android*, а застосунок був створений лише для використання на системі *IOS*, то *Android* – користувачі просто не зможуть використати програму, а це дуже велика доля користувачів.

2. Обробка даних та робота з базами даних: для *Backend* частини програми, зазвичай, й навіть в більшості випадків використовують такі технології, як *Node.JS*, *.NET*, *PHP*, *Java*, *Python*. *Backend* частини деяких випадків програм можуть бути написані, загалом, на будь-якій мові програмування, але ці технології є найбільш зручними. Щодо роботи з базами даних для зберігання усього необхідного пласту інформації, майже завжди використовують одну або (досить рідко) декілька таких систем баз даних, як *MySQL*, *PostgreSQL*, *MongoDB*. Як у всесвіті веб-сайтів у 99% випадків використовують *HTML* та *CSS*, так і у всесвіті баз даних усюди використовується *SQL*.

3. Штучний інтелект (*AI*): більшість сучасних застосунків намагаються використовувати технології машинного навчання та штучного інтелекту для аналізу тенденцій, інтелектуального та автоматичного підбору релевантних та персоналізованих рекомендацій, а також для прогнозування вимог користувачів.

					КРБ.КІ.1.440-03.1.16	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ДОСЛІДЖЕННЯ ІСНУЮЧИХ АНАЛОГІВ

1.1 Аналіз предметної області

Людам завжди були потрібні розважальні події, а тема театру – дуже довгий час була популярною, ще з часів стародавньої Греції. З плином часу, театральні події перетворилися скоріш на духовну подію, аніж на розважальну.

1.1.1 Методи вирішення аналогічних завдань

Тому першим методом вирішення цієї проблеми став традиційний метод замовлення квитків: здійснення замовлення квитків за допомогою телефонного дзвінка, особистого візиту до каси або використовуючи веб-сайт театального підприємства. Ці методи й досі є актуальними, але, легко прийти до висновку, що такий метод доволі незручний і займає більше часу у порівнянні з сучасними методами.

Щодо другого методу, то це – спеціальні онлайн-сервіси для замовлення квитків. Вони собою представляють онлайн-платформи, які дозволяють замовляти квитки на різноманітні події, звісно ж, включаючи театральні вистави. Такі сервіси забезпечують дещо більшу зручність та легкість у замовленні квитків.

Третій і найбільш популярний та розповсюджений метод вирішення проблеми – мобільні додатки. Завдяки використанню таких додатків, користувачам стає набагато легше проводити оплату, отримання та повідомлення-нагадування щодо нових або майбутніх подій. Головна та загальна особливість такого методу замовлення квитків полягає в можливих

					КРБ.КІ.1.440-03.1.16	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дат		

додаткових функціоналах у цих додатках. Створюються вони, звичайно, для полегшення життя користувачів та збереження їх дорогоцінного часу.

1.2 Існуючі аналоги

1.2.1 *Ticketmaster*

Ticketmaster – це один з найбільших світових сервісів по продажу квитків на різні події, включаючи театральні вистави. Цей сервіс був заснований в 1976 році в США. Він був створений компанією *Ticketmaster Entertainment, Inc.*, яка належить *Live Nation Entertainment*, однією з провідних компаній у галузі організації концертів та інших масових заходів.

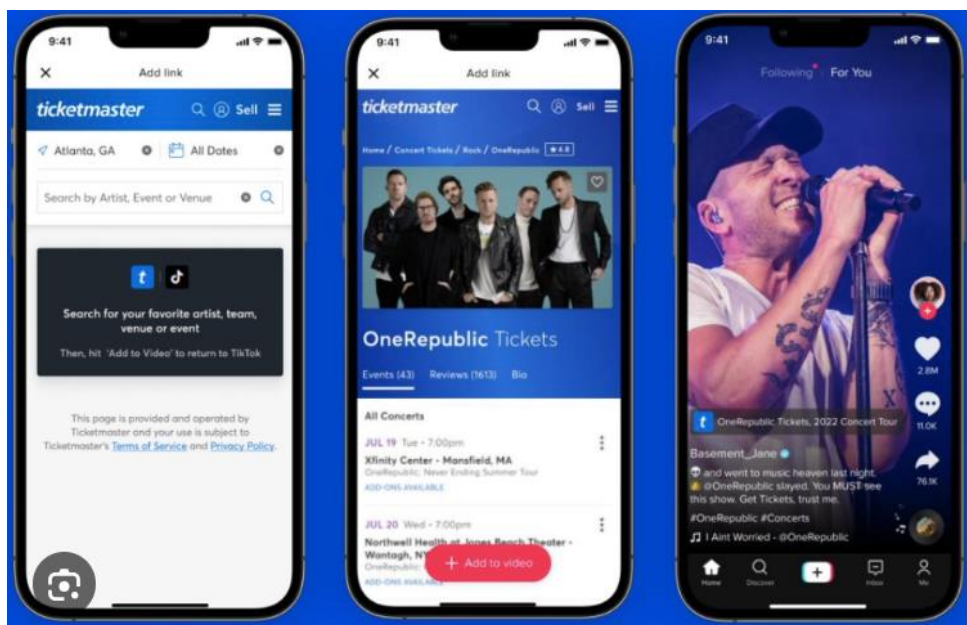


Рис. 1.1 – Скріншот застосунку «Ticketmaster»

Ticketmaster спрямований на широку аудиторію, включаючи людей різного віку та інтересів. Платформа пропонує квитки на різні заходи, включаючи театральні вистави, концерти, спортивні змагання та інші події. Вона охоплює широкий спектр країн та регіонів у своїй діяльності, включаючи Україну.

Переваги:

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дат		

1. Широкий вибір подій: Додаток *Ticketmaster* надає можливість замовляти квитки на різноманітні театральні події, концерти, спортивні змагання та інші події по всьому світу.

2. Простий інтерфейс: Додаток має зручний та простий інтерфейс, що дозволяє легко знаходити та замовляти квитки на різні події.

3. Відгуки та рейтинги: Користувачі можуть читати відгуки і рейтинги інших учасників про події, що допомагає приймати інформовані рішення при замовленні квитків.

Недоліки:

1. Висока комісія: Додаток може стягувати високу комісію за кожну покупку квитка, що може підвищити загальну вартість замовлення.

2. Обмеженість розрахункових систем: Додаток може обмежувати доступні способи оплати, що може бути незручним для деяких користувачів.

3. Можливість підробки: Існує ризик придбання підроблених квитків через додаток, оскільки він не завжди забезпечує повну гарантію валідності квитків.

1.2.2 *StubHub*

StubHub – це сервіс з перепродажу квитків на різні події, включаючи театральні вистави. Компанія була заснована у 2000 році в США та стала одним з найбільших платформ для купівлі та продажу квитків.

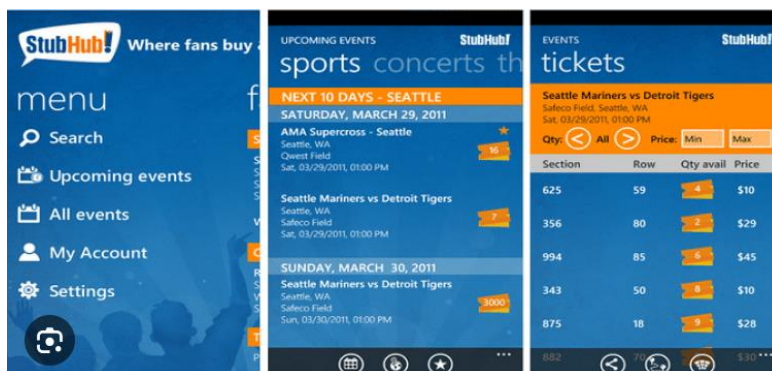


Рис. 1.2 – Скріншот застосунку «*StubHub*»

					КРБ.КІ.1.440-03.1.16	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дат		

StubHub націлено на широку аудиторію, включаючи людей, які шукають квитки на різні події, включаючи театральні вистави. Платформа діє у різних країнах, включаючи США, Канаду, Великобританію та інші.

Переваги:

1. Великий вибір подій: *StubHub* пропонує широкий вибір театральних подій у різних містах і країнах, що дозволяє знайти підходящу подію для будь-якого смаку.

2. Функція "*Last Minute Services*": Додаток надає можливість придбати квитки на останній хвилині перед подією, що дозволяє використовувати можливості залишених невикористаними квитків.

3. Захист покупців: *StubHub* має систему захисту покупців, яка забезпечує повернення коштів у разі недійсності квитка або скасування події.

Недоліки:

1. Висока вартість: Додаток може бути дещо дорогим для деяких користувачів, оскільки комісія *StubHub* може бути високою.

2. Ризик купівлі підроблених квитків: Існує ризик придбання підроблених квитків через додаток, хоча *StubHub* має систему гарантії валідності квитків, але вона не є 100% надійною.

3. Обмежена географічна доступність: Деякі театральні події можуть бути обмежені за межами певних географічних регіонів.

1.2.3 *TodayTix*

TodayTix – це мобільний додаток для замовлення квитків на театральні вистави. Додаток був створений в 2013 році в США. *TodayTix* пропонує зручну платформу для вибору та замовлення квитків на різні театральні події.

					КРБ.КІ.1.440-03.1.16	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дат		

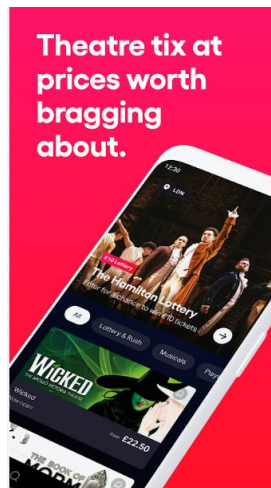


Рис. 1.3 – Скріншот застосунку «TodayTix»

TodayTix націлено на широку аудиторію театральних улюбленців, включаючи людей різного віку та інтересів. Додаток працює у різних країнах, включаючи США, Великобританію, Канаду та інші.

Переваги:

1. Зручний пошук подій: *TodayTix* надає зручний пошуковий інструмент, що дозволяє швидко знаходити доступні театральні події у вашому регіоні.
2. Ексклюзивні пропозиції: Додаток пропонує ексклюзивні знижки та спеціальні пропозиції на театральні квитки, що робить його привабливим для користувачів.
3. Простий інтерфейс та навігація: *TodayTix* має зручний та простий інтерфейс, що дозволяє легко знайти і придбати квитки на бажану подію.

Недоліки:

1. Обмежена географічна доступність: *TodayTix* може бути обмеженим за межами певних географічних регіонів, що обмежує доступ до театральних подій у деяких місцях.
2. Обмежений вибір: В порівнянні з іншими додатками, *TodayTix* може мати обмежений вибір театральних подій, особливо в менш розповсюджених регіонах.
3. Залежність від наявності квитків: У деяких випадках доступність квитків може бути обмеженою залежно від популярності театральних подій.

					КРБ.КІ.1.440-03.1.16	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дат		

1.2.4 Bilet.ua

Bilet.ua – це український сервіс для замовлення квитків на різні події, включаючи театральні вистави. Платформа була створена в 2009 році і є однією з провідних платформ для купівлі та продажу квитків в Україні.



Рис. 1.4 – Скріншот застосунку «*Bilet.ua*»

Bilet.ua спрямований на українську аудиторію, яка шукає квитки на театральні вистави та інші події в Україні. Платформа пропонує квитки на різноманітні події у різних містах країни.

Переваги:

1. Широкий вибір театральних подій: *Bilet.ua* пропонує великий вибір театральних вистав у різних містах України.
2. Простий процес замовлення: Додаток має зручний інтерфейс, що дозволяє легко знайти та замовити квитки на бажану виставу.
3. Знижки та спеціальні пропозиції: *Bilet.ua* часто пропонує знижки та спеціальні пропозиції на квитки, що дозволяє зекономити кошти.

Недоліки:

1. Обмежена географічна доступність: Додаток може бути обмеженим за межами України, що обмежує доступ до театральних подій в інших країнах.

					КРБ.КІ.1.440-03.1.16	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дат		

2. Відсутність рейтингів та відгуків: *Bilet.ua* не надає детальні рейтингита відгуки про вистави, що ускладнює прийняття рішення при замовленні квитків.

1.3 Постановка завдання

Метою даної дипломної роботи є створення мобільного додатку для заказу театральних квитків.

Об'єктом дослідження є розробка мобільних додатків для заказу театральних квитків.

Предметом дослідження є методи та засоби розробки мобільних додатків тематики заказу театральних квитків.

Основними задачами, які необхідно вирішити в ході роботи, є:

1. Створення унікального застосунку для заказу квитків.
2. Створення макетів відображення сторінок застосунку.
3. Налаштування та створення логіки побудови застосунку.
4. Компіляція та тестування демонстраційної версії застосунку.

Новизна роботи: мобільний додаток має декілька стилів візуального відображення на розсуд користувача.

Висновок до першого розділу

1. Під час аналізу предметної області, були виявлені загальні риси мобільних застосунків для замовлення театральних квитків.

2. Проведено аналіз сучасних мобільних застосунків для замовлення театральних квитків. Відображено їх переваги та недоліки.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 2

ПРОЕКТУВАННЯ ДОДАТКУ

2.1 Вибір і обґрунтування базового варіанту побудови системи

Застосунок необхідний для заказу театральних квитків, отже, головна його задача-надати можливість користувачеві заказати театральні квитки на один з доступних йому подій. Інтерфейс для користувача повинен бути зрозумілий та легкий заради того, щоб він зміг швидко і доволі зрозумілим способом знайти потрібну собі подію, отримати базову інформацію щодо неї та зробити замовлення.

Деякі особливості застосунку включають в себе легкий доступ до будь-якого розділу застосунку з будь-якої його сторінки, можливість вибирати тему оформлення застосунку (світла або темна), та дивитись інформацію щодо театрів, які присутні у місті, яке доступне зі списку обрання міст у навігаційному меню.

2.1.1 Технології та мови програмування

Серед багатьох мов програмування, які використовуються у світі задля будь-яких потреб, можна виділити декілька, які найчастіше використовуються для розробки мобільних застосунків. Це:

1. *Java* – є однією з найпопулярніших мов програмування для розробки мобільних додатків. Вона широко використовується для розробки додатків на платформі *Android*.

2. *Kotlin* – є новою мовою програмування, яка стала офіційною мовою розробки для платформи *Android*. Вона базується на *Java*, але пропонує більш сучасний синтаксис та додаткові можливості.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дат		

3. *Swift* – це мова програмування, розроблена компанією *Apple* для створення додатків для *iOS* та *macOS*. Вона стала основною мовою розробки додатків для пристроїв *Apple*.

4. *Objective-C* – є старішою мовою програмування, яка також використовується для розробки додатків для пристроїв *Apple*. Хоча *Swift* тепер є переважаючою мовою для нових проектів, багато існуючих додатків все ще написані на *Objective-C*.

5. *C#* – (*C Sharp*) - це мова програмування, розроблена компанією *Microsoft*, яка використовується для створення додатків для платформи *Windows* і платформи *Xamarin*, яка дозволяє розробляти крос-платформові мобільні додатки.

6. *JavaScript* – це мова програмування, яку широко використовують веб-розробники. Вона також може бути використана за допомогою фреймворків та інструментів, таких як *React Native* та *NativeScript*, для розробки крос-платформових мобільних додатків.

Серед усіх цих мов програмування, я обрав саме *Kotlin*. Мій вибір обумовлений тим, що мій досвід у розробці мобільних застосунків є доволі невеликим, але з мовою *Java* я ознайомився під час навчання в інституті, тому ця мова мені відома. Основні переваги використання мови *Kotlin* для розробки мобільного застосунку, включають у себе:

1. Взаємодія з *Java*: *Kotlin* повністю сумісний з *Java*, що дозволяє використовувати існуючий *Java*-код і бібліотеки в проектах *Kotlin*. Це забезпечує плавний перехід з *Java* на *Kotlin* і надає більшу гнучкість при виборі мови для проекту.

2. Покращена безпека: *Kotlin* розроблено з урахуванням безпеки, включаючи механізми статичної типізації, які дозволяють виявляти багато помилок на етапі компіляції. Це зменшує ймовірність помилок під час виконання програми, таких як винятки *NullPointerException*.

3. Компактний і виразний синтаксис: *Kotlin* має чистий і лаконічний синтаксис, що робить код більш зрозумілим і читабельним. Скорочений запис,

					КРБ.КІ.1.440-03.1.16	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дат		

функції-розширення, оператори безпечного виклику та інші конструкції дозволяють писати код ефективніше і економити час розробки.

4. Підтримка нульових значень: *Kotlin* включає систему типів, яка дозволяє вказувати, чи може змінна мати значення *null* або ні. Це допомагає уникнути багатьох помилок, пов'язаних з нульовими значеннями, і спрощує обробку їх у коді.

5. Функціональне програмування: *Kotlin* пропонує широкий набір функціональних можливостей, включаючи лямбда-вирази, функції вищого порядку, анонімні функції та інші. Вони дозволяють писати більш компактний і гнучкий код, особливо при роботі з колекціями даних.

6. Корутини: *Kotlin* надає вбудовану підтримку корутин для управління асинхронними операціями. Корутини полегшують написання асинхронного коду, надаючи просту та зрозумілу модель програмування без використання зворотних викликів або багатьох потоків.

7. Розширюваність: *Kotlin* дозволяє розширювати функціональність існуючих класів за допомогою механізму розширень. Це дозволяє додавати нові функції та методи до класів сторонніх бібліотек без зміни вихідного коду цих класів.

8. Розширюваність: *Kotlin* дозволяє розширювати функціональність існуючих класів за допомогою механізму розширень. Це дозволяє додавати нові функції та методи до класів сторонніх бібліотек без зміни вихідного коду цих класів.

9. Підтримка особливостей *Android*: *Kotlin* тісно інтегрований з *Android SDK* і надає додаткові можливості, специфічні для платформи *Android*. Він спрощує взаємодію з різними *API*, забезпечує кращу підтримку анотацій та генерацію коду.

10. Підтримка особливостей *Android*: *Kotlin* тісно інтегрований з *Android SDK* і надає додаткові можливості, специфічні для платформи *Android*. Він спрощує взаємодію з різними *API*, забезпечує кращу підтримку анотацій та генерацію коду.

					КРБ.КІ.1.440-03.1.16	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дат		

Android Studio, яку я використав для розробки застосунку є доволі зручною та зрозумілою програмою для розробки таких застосунків, яка має вбудовані ассети, методи та системи керуваннями залежностями, яка дуже спрощує розробку.

Android Studio – це інтегроване середовище розробки (*IDE*), спеціально призначене для створення мобільних додатків для платформи *Android*. Воно має численні переваги, які сприяють ефективному розробленню додатків. Ось деякі з них:

1. Комплексне середовище: *Android Studio* надає повноцінне середовище розробки, яке включає в себе всі необхідні інструменти для проектування, розробки, налагодження і тестування мобільних додатків. Воно забезпечує зручний і одностайний робочий процес для розробників.

2. Інтеграція з *Android SDK*: *Android Studio* постачається з офіційним *Android SDK*, що дозволяє розробникам використовувати всі найновіші можливості платформи *Android*. Вона також надає доступ до багатьох корисних інструментів, таких як емулятор *Android*, монітор профілювання та інші.

3. Легка налагодження: *Android Studio* має потужний вбудований інструмент для налагодження додатків, який дозволяє розробникам легко виявляти й усувати помилки. Воно підтримує крок за кроком виконання коду, точки зупину, перегляд значень змінних і багато іншого.

4. Розширені можливості дизайну: *Android Studio* має вбудований редактор макетів, що дозволяє розробникам швидко створювати і налаштувати інтерфейс користувача. Він підтримує перегляд в реальному часі, перетягування елементів, попередній перегляд на різних пристроях та інші зручні функції.

5. Велике співтовариство: *Android Studio* є стандартним інструментом для розробки мобільних додатків для *Android*, тому має велике співтовариство розробників, яке надає підтримку, документацію, статті та готові рішення. Це забезпечує швидкий доступ до ресурсів і допомоги при необхідності.

					КРБ.КІ.1.440-03.1.16	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дат		

6. Підтримка мов програмування: *Android Studio* підтримує мови програмування, такі як *Java* і *Kotlin*, що дозволяє розробникам використовувати ту, з якою вони знайомі або вважають зручною. Це розширює можливості вибору для розробки додатків.

Android Studio є потужним інструментом для розробки мобільних додатків для платформи *Android*. Воно надає розробникам зручність, продуктивність та широкий спектр можливостей, що сприяють швидкому та якісному процесу розробки.

Щодо вибору операційної системи для розробки мобільного застосунку – я схилиюсь до використання таких операційних систем, як *Windows 10* та *Windows 11*. Частково це зумовлено тим, що для мене ця система найбільш зручна у використанні, тому що впродовж всього свого життя я використовував переважно її, але ось деякі моменти щодо голосів «за» використання цієї операційної системи:

1. Підтримка *Android Studio*: *Windows 11* забезпечує повну підтримку для *Android Studio* – інтегрованої розробки середовища для створення мобільних додатків на платформі *Android*. Це дозволяє розробникам використовувати всі функціональні можливості *Android Studio* на платформі *Windows 11* для швидкого та ефективного створення додатків.

2. Компатибельність з *Kotlin*: *Windows 11* повністю сумісний з мовою програмування *Kotlin*, яка стала офіційною мовою розробки на платформі *Android*. Це дозволяє розробникам використовувати потужні можливості *Kotlin* для створення мобільних додатків на платформі *Android* у комбінації з *Android Studio*.

3. Оптимізована продуктивність: *Windows 11* пропонує покращену продуктивність та оптимізацію, що дозволяє розробникам працювати швидше та ефективніше. Це особливо важливо при використанні потужних інструментів розробки, таких як *Android Studio*, які вимагають великої кількості ресурсів.

4. Інтеграція з *Microsoft*-екосистемою: *Windows 11* має глибоку інтеграцію з іншими продуктами та сервісами *Microsoft*, такими як *Azure*, *Visual Studio*,

					КРБ.КІ.1.440-03.1.16	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дат		

GitHub тощо. Це дозволяє розробникам зручно працювати з цими інструментами, інтегрувати їх у свої проекти та ефективно керувати розробкою мобільних додатків.

5. Зручна розробка та налагодження: *Windows 11* надає розробникам широкий набір інструментів для розробки та налагодження мобільних додатків. Включаючи ефективні інструменти для налагодження коду, інструменти для профілювання додатків та інші засоби, які спрощують процес розробки та допомагають виявляти та виправляти помилки.

6. Інтеграція з екосистемою *Android*: *Windows 11* дозволяє зручно тестувати та відлагоджувати мобільні додатки безпосередньо на віртуальних пристроях з *Android*, що підтримуються *Android Studio*. Це спрощує процес розробки, оскільки розробники можуть безпосередньо перевіряти та тестувати додатки на різних конфігураціях пристроїв.

7. Широкі можливості для розробки інтерфейсу користувача: *Windows 11* пропонує різноманітні можливості для розробки привабливого та функціонального інтерфейсу користувача для мобільних додатків. Розробники можуть використовувати різні компоненти та бібліотеки для створення естетичного та інтуїтивно зрозумілого інтерфейсу для своїх додатків.

8. Широке співтовариство розробників: *Windows 11* має велике та активне співтовариство розробників, що працюють з платформою *Android* та *Kotlin*. Це означає, що розробники можуть отримати підтримку, поради та допомогу від інших фахівців у галузі, що сприяє поліпшенню якості та ефективності розробки додатків.

9. Можливості розгортання та публікації: *Windows 11* дозволяє розробникам зручно розгортати та публікувати свої мобільні додатки на платформі *Android*. Він надає інструменти для створення пакетів додатків та відповідних файлів налаштувань для успішного розгортання додатків на пристроях з *Android*.

Вибір операційної системи *Windows 11* для розробки мобільних додатків з використанням *Android Studio* на *Kotlin* та бази даних *Room* може забезпечити

					КРБ.КІ.1.440-03.1.16	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дат		

розробнику зручну та ефективну робочу середу, підтримку сучасних інструментів розробки, високу продуктивність та можливість легкого розгортання додатків на платформі *Android*.

2.1.2 Архітектура системи

Для розробки застосунку, звісно ж, потрібно обрати одну з архітектур, за якою буде проводитися його розробка. Основні з них вирізняють:

1. Нативна архітектура: Нативні додатки розробляються спеціально для однієї платформи (*Android* або *iOS*) з використанням мов програмування, які підтримуються цією платформою (*Java* або *Kotlin* для *Android*, *Swift* або *Objective-C* для *iOS*). Нативні додатки мають високу продуктивність, оптимальну інтеграцію з платформою та можуть повноцінно використовувати всі можливості пристрою.

2. Гібридна архітектура: Гібридні додатки є комбінацією нативних та веб-додатків. Вони розробляються за допомогою *HTML*, *CSS* та *JavaScript*, а потім "обгортаються" в нативну оболонку, яка дозволяє їм встановлюватися на пристрої та використовувати нативні *API*. Вони зазвичай менш ефективні, ніж нативні додатки, але дозволяють розробникам використовувати один кодовий базис для різних платформ.

3. Крос-платформна архітектура: Крос-платформні додатки розробляються за допомогою спеціалізованих фреймворків, таких як

4. *React Native*, *Xamarin* або *Flutter*, які дозволяють використовувати одну кодову базу для створення додатків для різних платформ. Вони зазвичай менш продуктивні, ніж нативні додатки, але можуть забезпечити більш послідовний користувацький досвід на різних пристроях.

Мною було обрано нативну архітектуру, так як цей спосіб розробки є найбільш придатним для розробки на *Android Studio*. Серед її переваг можна виділити високу продуктивність (нативні додатки виконуються набагато швидше і ефективніше, ніж гібридні або крос-платформні, тому що вони прямо

					КРБ.КІ.1.440-03.1.16	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дат		

взаємодіють з операційною системою без додаткових шарів абстракції), доступ до повного функціоналу пристрою (такі додатки можуть безпосередньо взаємодіяти з апаратним забезпеченням пристрою), кращий користувацький досвід (оскільки нативні додатки спеціально розроблені для певної платформи, вони можуть надати користувачам більш гладкий, зрозумілий і приємний досвід), безпека (нативні додатки також можуть надати вищий рівень безпеки, ніж не-нативні додатки, оскільки вони можуть безпосередньо використовувати вбудовані механізми безпеки платформи).

2.2 Загальна характеристика інформаційних масивів

2.2.1 Типи даних

Мій застосунок буде містити у собі різні типи даних, такі як текстові (назви театрів, міст), числові та дати (дати початку подій), зображення (різні малюнки, такі як малюнки театрів, подій, пунктів меню та ін.).

2.2.2 Джерела даних

Дані, які знаходяться у моєму застосунку, взяті з різних джерел, таких як офіційні сторінки театрів, де також знаходиться інформація про події та квитки, які можна замовляти. Також, щодо інформації, яку вводить користувач-це дані про самого користувача, якими він буде заповнювати сторінку з інформацією про користувача.

2.2.3 Структура даних

Дані зберігаються на локальній середі (на моєму ПК). Цими даними я оперую саме у деяких класах мого коду, де я отримую ці дані, передаю між сторінками, де потрібно-вставляю ці дані у потрібно поля (наприклад, при

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дат		

виборі користувачем міста, по якому він хоче знайти доступні йому події, інформація про вибране місто передається на сторінку Театри, де, в залежності від вибраного міста, у користувача відображаються театри цього міста).

2.2.4 Обсяг даних

В залежності від кількості занесеної мною, як адміністратором застосунку, інформації, обсяг даних може варуватися від невеликого до середнього. Але, взагалом, цей обсяг даних є доволі невеликим, так як дана модель застосунку є прототипною, яка включає в себе демонстраційний варіант роботи програми, завдяки якому користувачі матимуть можливість замовити театральні квитки на потрібну їм подію.

2.3 Діаграма інформаційних подій

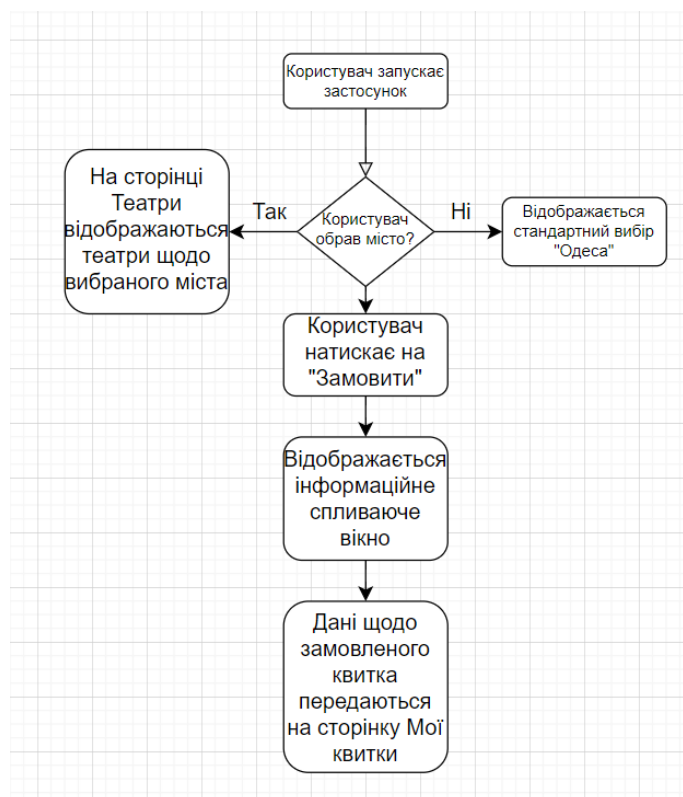


Рис. 2.1 – Діаграма інформаційних подій

2.4 Проектування структури системи

Система мого застосунку складається з багатьох модулів, які включають у себе:

1. Основний модуль, у якому створюється список міст, ініціалізуються основні дані та виконуються деякі функції програми, наприклад передача даних з .xml файлів до всіх сторінок системи, де відбувається їх відмалювання.

2. Модуль бази даних. В цьому модулі, я ініціалізую свою локальну базу даних, передаю в неї та зберігаю дані, які передаються з різних методів: вибір міста, або наприклад замовлення квитка на театральну подію. Також, в цьому модулі додається квиток до бази даних, який потім передається до потрібної сторінки, а саме до сторінки Мої квитки.

3. Модуль роботи з усіма сторінками застосунку, а це – сторінки «Домашня сторінка», «Мої квитки», «Налаштування», «Театри», «Мій аккаунт».

4. Модуль відмалювання усіх сторінок, спливаючих вікон та іншого, що відображається у візуалі користувача. Цей модуль включає у себе .xml файли для відмалювання «Домашньої сторінки», «Сторінки налаштувань», «Сторінки театрів», «Сторінки Мій аккаунт». Це – основні сторінки мого застосунку, але також є й інші .xml-файли для відмалювання деяких візуальних об'єктів, що включають в себе: плашки квитків, навігаційне меню, плашки театрів та інші.

Майже кожний модуль взаємодіє з основним файлом системи, що забезпечує легкість адміністрування та налаштування масштабних подій застосунку.

Модуль, який відповідає за роботу з базами даних взаємодіє з модулями театральних подій, сторінкою «Мої квитки», мікро-модулем для роботи з обраним містом.

Взагалом, модель роботи усіх модулів мого проекту можна відобразити на діаграмі:

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дат		

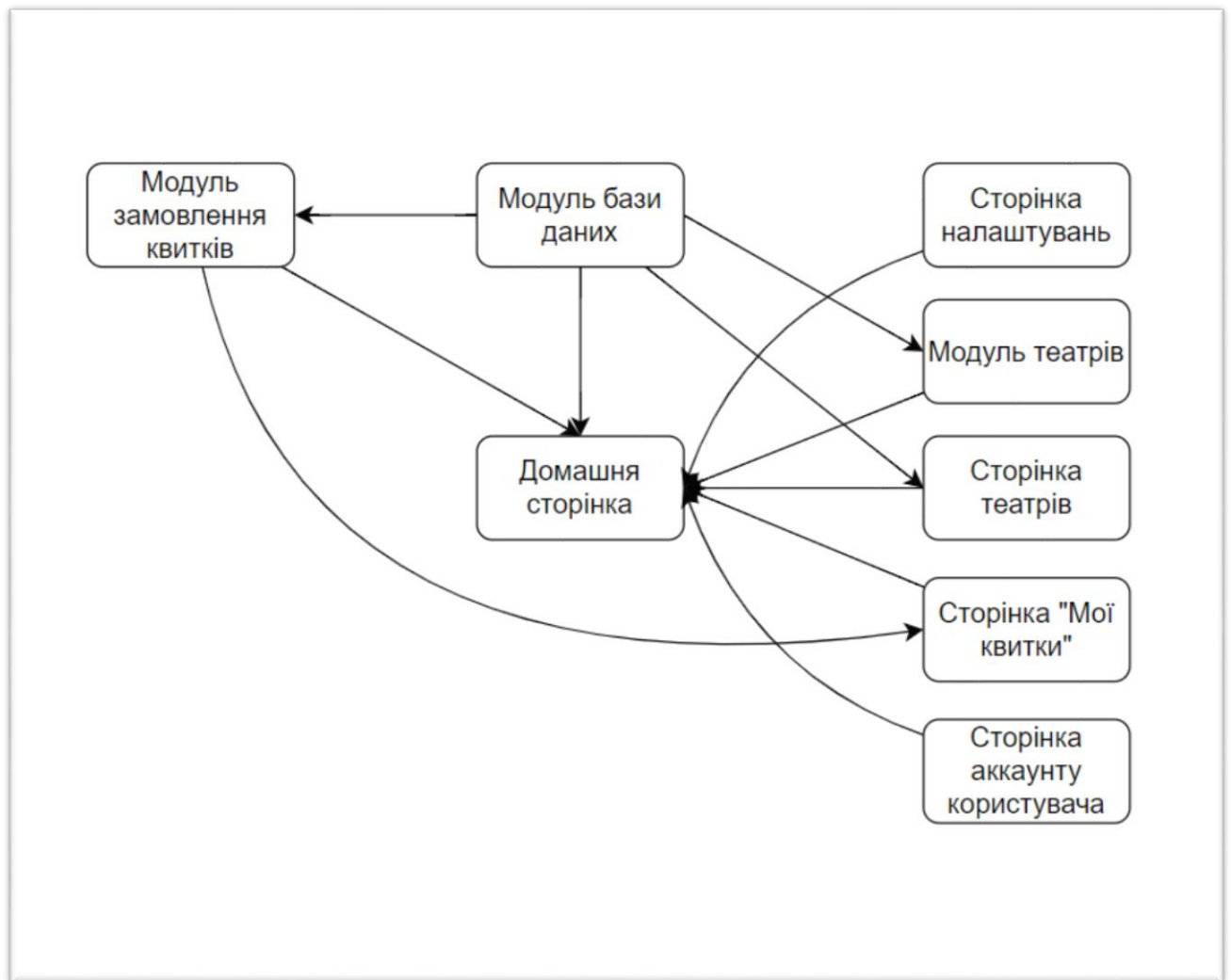


Рис. 2.2 – Модель роботи взаємодії модулів проекту

Щодо використання бази даних – в застосунку використовується база даних *Room*. База даних *Room* є реляційною базою даних, яка використовується в розробці мобільних додатків на платформі *Android* з метою зберігання та управління структурованими даними. Вона є вбудованою і легкового використання базою даних, яка надає можливості зберігання та обробки великого обсягу даних з високою швидкістю.

Основні переваги використання бази даних *Room* в розробці мобільних додатків на *Kotlin* включають:

1. Легкість використання: *Room* проста у використанні і не вимагає складного налаштування. Це вбудована база даних в платформі *Android*, що

полегшує її розгортання і використання. Ви можете легко створювати, модифікувати та взаємодіяти з базою даних *Room* у своєму додатку.

2. Низькі вимоги до ресурсів: *Room* має низькі вимоги до пам'яті та обчислювальних ресурсів пристрою. Це особливо важливо для мобільних пристроїв з обмеженими ресурсами, такими як мобільні телефони. Використання *Room* дозволяє ефективно використовувати ресурси пристрою і забезпечує оптимальну продуктивність додатка.

3. Швидкість та продуктивність: *Room* відома своєю високою швидкістю обробки даних. Вона працює локально на пристрої і забезпечує швидкий доступ до даних. *Room* використовує компактний формат файлів бази даних, що сприяє швидкому виконанню операцій зчитування та запису.

4. Підтримка *SQL*: *Room* повністю підтримує мову *SQL (Structured Query Language)*, що дозволяє виконувати потужні та складні запити до бази даних. Ви можете використовувати різні типи запитів, такі як *SELECT*, *INSERT*, *UPDATE* та *DELETE*, для маніпулювання даними у базі даних *SQLite*.

5. Транзакційна безпека: *Room* забезпечує транзакційну безпеку для збереження цілісності даних. Ви можете використовувати транзакції для групування операцій над даними і забезпечення консистентного стану бази даних. Це важливо для запобігання пошкодженню даних при випадкових збоях або помилках.

6. Гнучкість: *Room* надає гнучкість у структуруванні даних та виконанні запитів. Ви можете створювати таблиці з різними типами даних, встановлювати обмеження на поля, використовувати індекси для покращення швидкодії запитів та встановлювати зв'язки між таблицями.

Інтерфейси користувачів

Для користувача буде відкриватись велика кількість інтерфейсів для взаємодії, які включають в себе:

1. Цікаві, кольорові та різноманітні зображення, такі як зображення подій, театрів, пунктів бокового меню (зрозумілий та популярний дизайн бокового меню).

					КРБ.КІ.1.440-03.1.16	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дат		

2. Прості та зрозумілі елементи навігації.

3. Зрозумілий інтерфейс користувача, який забезпечує добрий *User Experience*, що дозволяє будь-якому користувачеві, без будь-яких великих зусиль зрозуміти, що він може робити у застосунку, та яка ціль цього застосунку. Також, всі натискання на діючі елементи застосунку є доволі докладними та простими, що дає користувачеві максимальну інформацію щодо його дій у застосунку.

4. *User-friendly* інтерфейс, який забезпечує найбільш доцільний й якісний досвід користувача у використанні застосунку.

Використання *Room* в якості рішення для роботи з базою даних може допомогти мені створити більш стабільний, ефективний і легкий у розробці мобільний додаток.

Висновок до другого розділу

1. У процесі проектування було спроектовано архітектурно проект мого застосунку, загальні рішення щодо використання певних технологій та їх взаємодія.

2. Також було обрано базу даних, з якою буде найбільш легко й швидко працювати, та яка надає найбільш великі можливості у створенні необхідного мені процесу роботи з даними.

3. Обрано такі технології для розробки додатку, як: програма *Android Studio*, мова програмування *Kotlin*, база даних *Room*.

					КРБ.КІ.1.440-03.1.16	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 3

РОЗРОБКА КОМПОНЕНТІВ СИСТЕМИ

3.1 Використання засобів розробки

Android Studio було розроблено як офіційне середовище розробки для *Android*, щоб забезпечити розробникам зручну і потужну платформу для створення високоякісних додатків для мобільних пристроїв.

У меню створення проекту в *Android Studio* можна обрати один з декількох шаблонів для створення проекту, який потрібен заради полегшення розробки розробником:

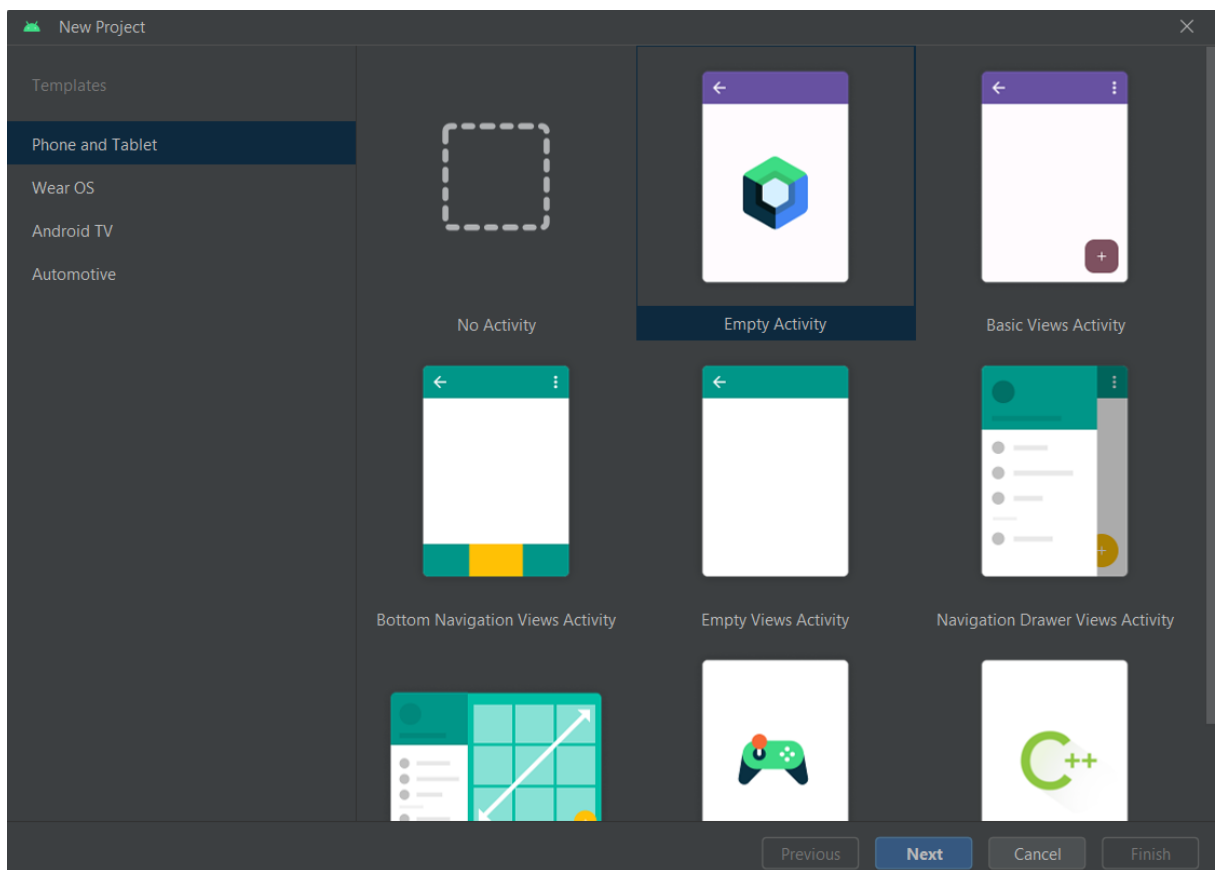


Рис. 3.1 – Меню створення проекту в *Android Studio*

					КРБ.КІ.1.440-03.1.16	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дат		

3.2 Розробка системи

Система мого проекту складається з деякої кількості модулів, кожний з яких відповідає за певні задачі у застосунку. Загалом, можна вирізнити такі модулі:

1. Модуль Аутентифікації (*Authentication*): Включає в себе класи *AuthActivity*, *LoginFragment*, *RegistrationFragment* та *AuthRepository*, а також *AuthDao*. Цей модуль відповідає за реєстрацію нових користувачів і вхід в систему існуючих.

2. Модуль Користувацького Профілю (*User Profile*): Цей модуль включає в себе класи *User*, *Profile*, *ProfileRepository* та *ProfileDao*. Він відповідає за зберігання та обробку даних користувача.

3. Модуль Квитків (*Ticketing*): Модуль включає в себе класи *Ticket*, *TicketViewModel*, *TicketRepository*, *TicketDao*, *MyTicketsAdapter*, *MyTicketsFragment*, та *PickTicketFragment*. Цей модуль керує всіма операціями, пов'язаними з квитками.

4. Модуль Театрів (*Theatres*): Цей модуль включає в себе класи *Theatre*, *TheatreAdapter*, *TheatreDetailFragment*, *TheatreFragment*, та *TheatresActivity*. Він управляє переглядом інформації про театри.

5. Модуль Подій (*Events*): Модуль включає в себе класи *EventDetailFragment*, *TheatreEvent*, та *TheatreEventAdapter*. Він відповідає за керування подіями в театрах.

6. Модуль Налаштувань (*Settings*): Цей модуль складається з класу *SettingsFragment* та допоміжного класу *NightMode*, що відповідає за переключення теми додатка.

7. Модуль Головного Інтерфейсу (*Main UI*): Модуль містить *MainActivity*, *HomeFragment*, та зв'язані з ними ресурси. Він відповідає за основний інтерфейс програми.

8. Модуль Даних (*Data*): Цей модуль містить *AppDatabase*, яка відповідає за зберігання даних у локальній базі даних.

					КРБ.КІ.1.440-03.1.16	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дат		

9. Модуль Додатку (*Application*): Модуль складається з класу `MyApplication`, який відповідає за ініціалізацію додатка.

10. Модуль Ресурсів (*Resources*): Цей модуль містить всі ресурси додатка: *strings.xml*, *dimens.xml*, *colors.xml*, *arrays.xml*, *themes.xml*, все що в папці *drawable*, та інші ресурси, які використовуються для створення інтерфейсу користувача.

Модуль *res* або ресурси є важливою частиною будь-якого *Android*-проекту. Він включає в себе всі не кодові ресурси, які використовуються в додатку, такі як макети, зображення, рядки, стилі, анімації, кольори, меню тощо.

Модуль *res*:



Рис. 3.2 – Модуль «res»

3.2.3 Розробка статичної частини системи

Моя система має деякі статичні компоненти, які не змінюються протягом виконання програми та відіграють важливу роль у функціонуванні системи. Розглянемо деякі з цих статичних частин та їхні функції.

Конфігураційні файли: Моя система має різні конфігураційні файли, такі як *AndroidManifest.xml*, *arrays.xml* та *colors.xml*. У цих файлах зберігаються

					КРБ.КІ.1.440-03.1.16	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дат		

різноманітні налаштування, ресурси та інші дані, які використовуються системою під час роботи. Наприклад, *AndroidManifest.xml* містить інформацію про додаток, його дозволи та компоненти. Файли *arrays.xml* та *colors.xml* містять набори значень, які можуть використовуватись у системі для забезпечення єдності стилю та локалізації.

Ресурси: Моя система використовує статичні ресурси, такі як зображення, рядки, макети та інші файли, які використовуються для відображення інтерфейсу користувача. Ці ресурси зберігаються в папці *res* та можуть бути використані в різних частинах системи, наприклад, для відображення зображень в адаптерах, встановлення тексту в текстових полях та інших візуальних компонентах.

Навігаційний граф (*nav_graph.xml*): Цей статичний файл містить інформацію про різні екрани та переходи між ними. Він визначає структуру навігації у системі та допомагає керувати взаємодією користувача з додатком. Навігаційний граф містить фрагменти, діяльності, дії та інші компоненти, які використовуються для навігації між екранами.

					КРБ.КІ.1.440-03.1.16	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дат		

```

<fragment
  android:id="@+id/homeFragment"
  android:name="com.example.theatretickets2.ui.HomeFragment"
  android:label="HomeFragment"
  tools:layout="@layout/fragment_home">
  <action
    android:id="@+id/action_homeFragment_to_theatreFragment"
    app:destination="@id/theatreFragment" >
    <argument
      android:name="city"
      app:argType="string" />
    </action>
  <action>
    android:id="@+id/action_homeFragment_to_settingsFragment"
    app:destination="@id/settingsFragment" />
  <action
    android:id="@+id/action_homeFragment_to_myTicketsFragment"
    app:destination="@id/myTicketsFragment" />
  <action
    android:id="@+id/action_homeFragment_to_pickTicket"
    app:destination="@id/pickTicket" />
</fragment>

```

Рис. 3.3 – Фрагмент коду навігаційного графу

Статичні класи та методи: система містить статичні класи та методи, які не залежать від стану об'єктів і можуть бути викликані без створення екземплярів класів. Ці статичні компоненти можуть виконувати різні функції, такі як утиліти для роботи з датами, математичні операції, утиліти для роботи зі строками тощо. Вони забезпечують зручний спосіб доступу до певних функціональних можливостей без необхідності створення об'єктів класів.

Мій модуль *com.example.theatretickets2* складається з різних статичних компонентів, які спільно працюють для забезпечення функціональності системи. Конфігураційні файли визначають основні налаштування системи, ресурси забезпечують відображення інтерфейсу користувача, навігаційний граф керує переміщенням користувача між екранами, а статичні класи та методи надають додаткові функціональні можливості. Використання цих статичних компонентів дозволяє покращити ефективність, стабільність та зручність розробки моєї системи.

3.2.4 Створення баз даних і таблиць

У моєму проєкті використовується база даних *Room*.

Базі даних *Room* використовується в розробці мобільних додатків на платформі *Android* з метою зберігання та управління структурованими даними. Вона є вбудованою і легковаговою для використання реляційною базою даних,

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дат		

яка надає можливості зберігання та обробки великого обсягу даних з високою швидкістю.

Основні компоненти бази даних *Room* включають наступне:

1. База даних (*Database*): База даних *Room* – це основний контейнер для зберігання даних. Вона може містити одну або кілька таблиць, які організовані у вигляді реляційних структур зі стовпцями та рядками.

2. Таблиці (*Tables*): Таблиці в *Room* представляють собою структуровані набори даних, які містять стовпці і рядки. Кожен стовпець в таблиці має тип даних, який визначає вид даних, які можуть бути збережені в цьому стовпці.

3. Запити (*Queries*): *Room* надає різноманітні можливості для виконання запитів до бази даних, таких як вставка (*INSERT*), вибірка (*SELECT*), оновлення (*UPDATE*) та видалення (*DELETE*). Запити дають змогу отримати, модифікувати або видалити дані з таблиць в базі даних. Індeksi (*Indexes*):

4. Індeksi в *Room* використовуються для покращення швидкодії операцій пошуку і сортування даних. Вони дозволяють ефективніше здійснювати запити до бази даних шляхом створення додаткових структур даних, які прискорюють пошук інформації. Транзакції (*Transactions*):

5. Транзакції в *Room* дозволяють групувати декілька операцій бази даних в один блок, що виконується як атомарна операція. Вони забезпечують консистентність даних та забезпечують відновлення бази даних до стану перед початком транзакції у разі виникнення помилки.

Щодо архітектури роботи з базою даних в моєму проекті:

Для реалізації необхідного мені функціоналу, я створив декілька файлів, які допоможуть мені працювати з базою даних, а саме:

У моделях даних у додатку є опис сутностей або об'єктів, які використовуються у додатку.

Profile.kt – модель профілю користувача. Містить таку інформацію, як ім'я користувача, прізвище, дату народження та власник. Власник може бути використаний для зв'язку профілю з конкретним користувачем або акаунтом.

					КРБ.КІ.1.440-03.1.16	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дат		

Ticket.kt – модель квитка на захід. Містить таку інформацію, як назва заходу, власник квитка та номер квитка. Власник може бути використаний для зв'язку квитка з конкретним користувачем або акаунтом.

User.kt – модель користувача, що містить дані для автентифікації користувача (логін та пароль).

Всі ці класи анотовані як *@Entity*, що означає, що вони є таблицею в базі даних під час використання бібліотеки *Room* для роботи з базами даних *SQL* на *Android*. Анотація *@Entity* використовується для вказівки того, що цей клас є сутністю бази даних.

@PrimaryKey анотує поле, яке буде використовуватися як первинний ключ у таблиці бази даних.

@ColumnInfo використовується для визначення імені стовпця у таблиці бази даних, якому відповідатиме поле у класі.

Таким чином, у кожному з цих класів визначено поля, які відповідають стовпцям таблиці бази даних. Ці класи надають структуру для моїх даних і дозволяють моєму додатку взаємодіяти з базою даних.

AppDatabase.kt – це абстрактний клас, який служить контейнером для бази даних та асоціює ваші *DAO* (об'єкти доступу до даних) з вашою базою даних. Я визначає тут всі сутності, які будуть використовуватися в базі даних (в даному випадку, *Ticket*, *User*, *Profile*). Він також містить методи, що повертають інстанції *DAO*. Всередині *companion object* я створюю *singleton* інстанцію бази даних, що гарантує, що все додаток використовує одну і ту ж базу даних, що запобігає одночасному доступу до ресурсів.

AuthDao.kt, *ProfileDao.kt*, *TicketDao.kt* – це інтерфейси, які визначають всі методи для доступу до даних, які ви плануєте використовувати.

AuthDao використовується для автентифікації користувачів. Він має метод *insert*, що додає користувача до бази даних, і метод *getUser*, що отримує користувача з бази даних за логіном.

					КРБ.КІ.1.440-03.1.16	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дат		

ProfileDao використовується для роботи з профілями користувачів. Він має метод *insert*, що додає профіль до бази даних, і метод *getProfile*, що отримує профіль з бази даних за власником.

TicketDao використовується для роботи з квитками. Він має методи для отримання всіх квитків, додавання квитків до бази даних та видалення квитків з бази даних.

Кожен *DAO* використовує анотацію *@Dao* для ідентифікації його як *DAO* у Room. Методи, що використовуються для взаємодії з базою даних, використовують анотації, такі як *@Query* та *@Insert*.

Також приведено діаграму взаємодії файлів бази даних у моїй системі (рис. 3.4).

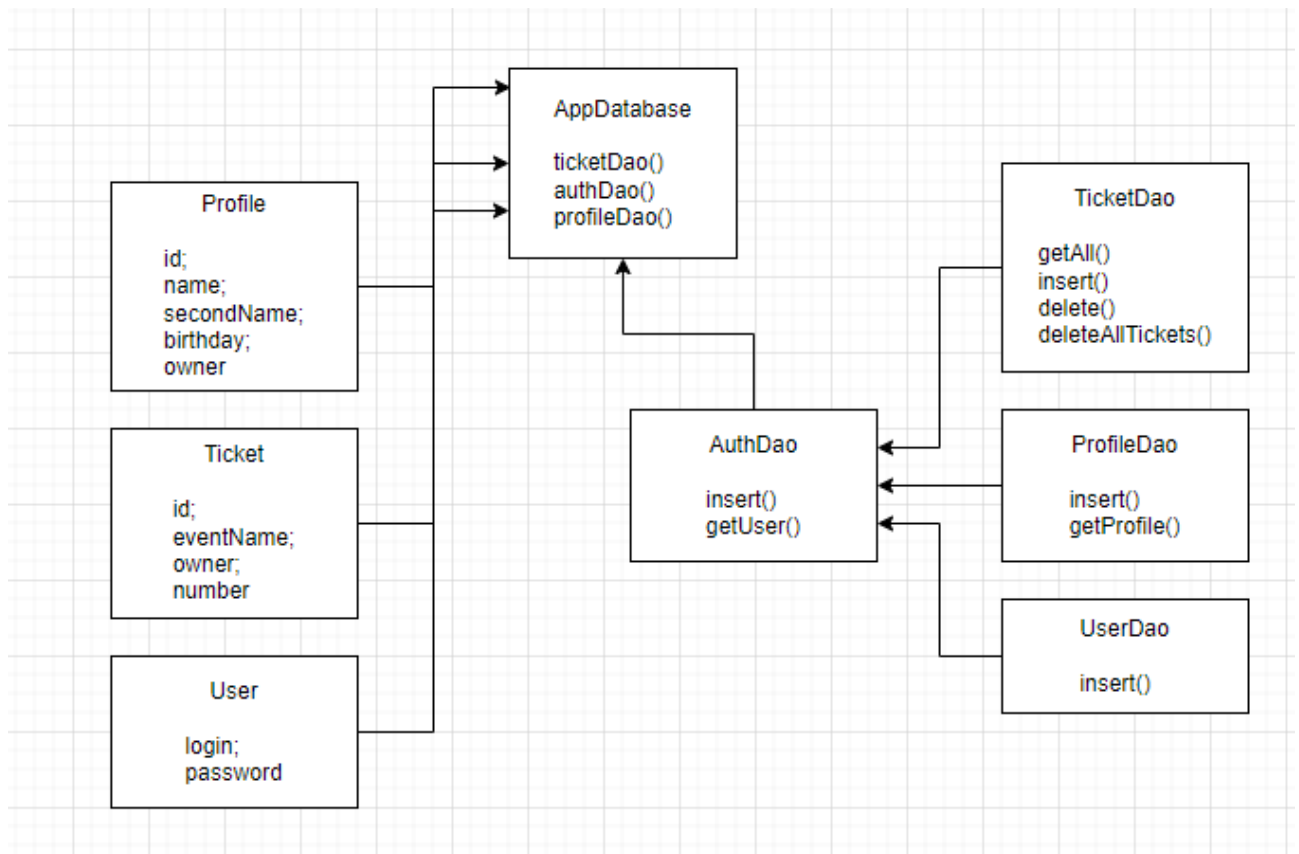


Рис. 3.4 – Діаграма взаємодії файлів бази даних у додатку

3.2.5 Створення екранних форм

В моїй системі "створення екранних форм" відіграє важливу роль у створенні інтерфейсу користувача та взаємодії з моїм додатком. Цей процес включає розробку та реалізацію різних екранів та форм, що відображаються на екрані пристрою користувача та забезпечують йому зручність взаємодії з додатком.

Створення екранних форм включає наступні етапи:

1. Розробка макетів: У моєму проекті розробка екранних форм починається з створення макетів, які визначають загальний вигляд та компоненти кожної форми. Макети можуть бути розроблені за допомогою інструментів графічного дизайну, таких як *Adobe XD*, *Sketch* або *Figma*. Вони включають в себе розміщення елементів інтерфейсу, таких як кнопки, поля введення, списки тощо. Щодо мого проекту-я використовую звичайні зображення, ресурси яких я взяв з інтернету, а також з базових асетів *Android Studio*.

2. Використання *XML*-ресурсів: у розробці застосунку використовуються *XML*-файли ресурсів для опису компонентів екранних форм. Це дозволяє розділити вигляд елементів інтерфейсу від логіки додатку. *XML*-ресурси містять опис компонентів, таких як текстові поля, кнопки, зображення тощо, а також їх атрибути, такі як розмір, кольори, стиль тощо. Це дозволяє зручно змінювати вигляд елементів інтерфейсу без необхідності змінювати логіку програми.

3. Використання *Layout*-компонентів: Для розміщення компонентів екранних форм використовуються *Layout*-компоненти, такі як *LinearLayout*, *RelativeLayout* або *ConstraintLayout*. Ці компоненти дозволяють визначати правила розміщення елементів на екрані, наприклад, вирівнювання до верху, лівої частини екрана, встановлення відступів між елементами тощо. Вони забезпечують гнучкість у створенні вигляду екранних форм та адаптацію до різних розмірів екранів.

					КРБ.КІ.1.440-03.1.16	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дат		

4. Додавання функціональності: Крім вигляду, екранні форми також містять функціональність, яка забезпечує взаємодію з користувачем. У проекті цю функціональність я реалізую за допомогою коду на мові програмування *Kotlin*. Наприклад, обробники подій можуть бути призначені для кнопок, текстових полів тощо, щоб відповідати на взаємодію користувача із цими елементами. Це дозволяє реагувати на дії користувача та виконувати певні дії, такі як перехід на інший екран, збереження даних тощо.

5. Тестування екранних форм: Під час розробки застосунку я надаю велику увагу тестуванню екранних форм, щоб переконатися, що вони працюють коректно та забезпечують зручну взаємодію з користувачем. Для цього використовуються інструменти тестування *Android Studio*.

Майже усі візуальні елементи промальовані в *XML*-ресурсах, наприклад:

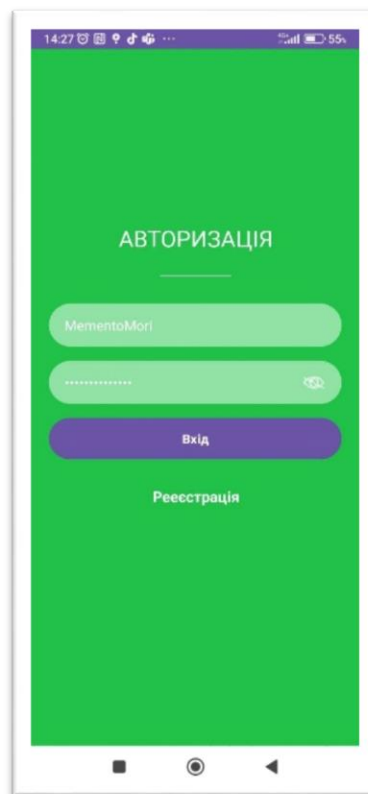


Рис. 3.5 – Відображення сторінки логіну

					КРБ.КІ.1.440-03.1.16	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дат		



Рис. 3.6 – Відображення сторінки опису події

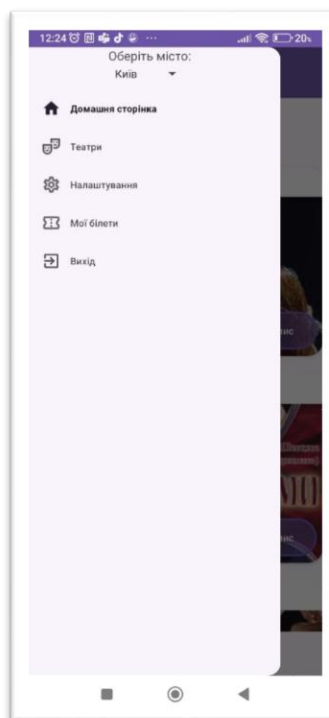


Рис. 3.7 – Відображення бічного меню додатку

					КРБ.КІ.1.440-03.1.16	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дат		

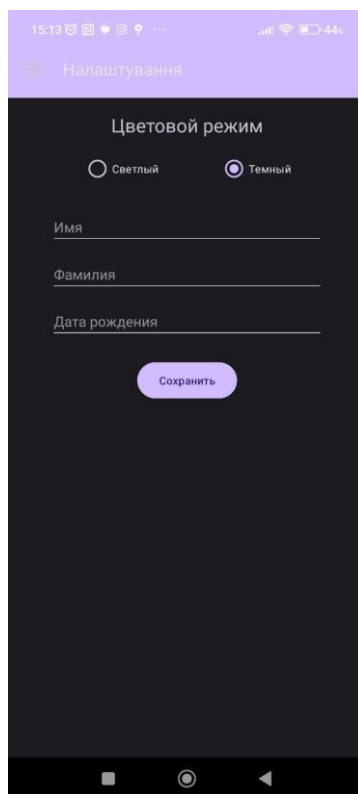


Рис. 3.8 – Сторінка налаштувань у системі

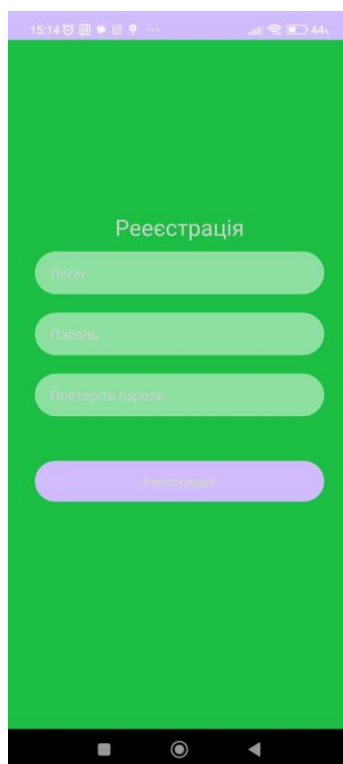


Рис. 3.9 – Сторінка рееєстрації користувача у застосунку

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дат		

Коди цих макетів реалізовано, взагалом, основним функціоналом відображення елементів на сторінці, їх розташування, вигляд та інше. Можна виділити основні з них на прикладі реалізації відображення сторінки логіну користувача:

1. `<EditText`: Визначає, що цей елемент є полем введення тексту.
2. `android:id="@+id/etPassword"`: Встановлює унікальний ідентифікатор для цього елемента, який можна використовувати для посилання на нього в коді.
3. `android:layout_width="0dp"`: Встановлює ширину елемента на 0 "пікселів" із використанням системи вимірювання відносності.
4. `android:layout_height="50dp"`: Встановлює висоту елемента на 50 "пікселів".
5. `android:layout_marginStart="30dp"` та `android:layout_marginLeft="30dp"`: Встановлюють відступ зліва від елемента до його батьківського контейнера (у цьому випадку відступ зліва становить 30 "пікселів").
6. `android:layout_marginTop="15dp"`: Встановлює відступ зверху від елемента до його батьківського контейнера (у цьому випадку відступ зверху становить 15 "пікселів").
7. `android:layout_marginEnd="30dp"` та `android:layout_marginRight="30dp"`: Встановлюють відступ справа від елемента до його батьківського контейнера (у цьому випадку відступ справа становить 30 "пікселів").
8. `android:background="@drawable/et_border"`: Встановлює фоновий ресурс (зображення або колір) для елемента.
9. `android:hint="@string/password"`: Встановлює текст-підказку для поля введення, яка вказує користувачеві, який тип інформації слід вводити (у цьому випадку текст підказки отримується з ресурсу строкового ресурсу з ідентифікатором "password").
10. `android:inputType="textPassword"`: Встановлює тип введення для поля введення як текстовий пароль. Це приховує введений текст, щоб ніхто не міг його побачити.

					КРБ.КІ.1.440-03.1.16	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дат		

11. `android:textColor="@color/colorWhite"`: Встановлює колір тексту у білий.

12. `android:paddingLeft="18dp"`: Встановлює відступ зліва від тексту до межі поля введення (у цьому випадку відступ зліва становить 18 "пікселів").

13. `android:textColorHint="@color/colorWhite"`: Встановлює колір тексту підказки у білий.

14. `app:layout_constraintEnd_toEndOf="parent"`: Встановлює обмеження, що правий край елемента повинен бути прив'язаний до правого краю батьківського контейнера.

15. `app:layout_constraintHorizontal_bias="0.0"`: Встановлює горизонтальну зсувність елемента (0.0 означає, що елемент буде вирівняний зліва).

16. `app:layout_constraintStart_toStartOf="parent"`: Встановлює обмеження, що лівий край елемента повинен бути прив'язаний до лівого краю батьківського контейнера.

17. `app:layout_constraintTop_toBottomOf="@+id/etLogin"`: Встановлює обмеження, що верхній край елемента повинен бути прив'язаний до нижнього краю елемента з ідентифікатором "etLogin".

Загалом, з використанням *XML*-ресурсів, *Layout*-компонентів та функціональності, моя система дозволяє розміщувати та організовувати різні елементи інтерфейсу на екрані та взаємодіяти з ними.

3.2.6 Розробка динамічної частини системи

Одним з ключових елементів "розробки динамічної частини системи" є можливість написання скриптів на мовах програмування, таких як *Kotlin* або *Java*, які використовуються для взаємодії з іншими компонентами системи та виконання певних завдань. Скрипти можуть містити логіку обробки подій, розрахунків, маніпуляцій з даними та інші дії, які розширюють функціональність додатку.

					КРБ.КІ.1.440-03.1.16	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дат		

Динамічна частина системи також включає розробку сценаріїв, які виконуються на певних подіях або відповідно до певного розкладу. Це може бути автоматичне оновлення даних, планування завдань, відтворення відео або аудіо, анімації та багато іншого. Сценарії дозволяють автоматизувати певні процеси та забезпечити більш зручне та ефективне використання додатку.

Розробка динамічної частини системи в моєму проекті дозволяє розширити функціональність додатку та забезпечити більш гнучкі можливості для користувачів. Вона дозволяє писати скрипти та сценарії для автоматизації рутинних завдань, взаємодії зі сторонніми сервісами, відтворення мультимедійного контенту та багато іншого. Це допомагає забезпечити більш персоналізований та динамічний досвід використання додатку для кожного користувача.

У моєму проекті переважна кількість коду відповідає за динамічну частину системи, серед них можна вирізнити:

1. *AuthRepository.kt*, *ProfileRepository.kt*, *TicketRepository.kt* - це репозиторії, які взаємодіють з базою даних через *DAO*. Це динамічна частина системи, оскільки вони реагують на запити від користувача і повертають результати з бази даних. Клас *TicketRepository* є частиною шару даних в моїй архітектурі програми. Його завдання – забезпечувати мост між різними джерелами даних (в даному випадку – база даних через *TicketDao*) та бізнес-логікою програми.

Специфічні методи цього класу виконують такі дії:

val allTickets: Flow<List<Ticket>> = ticketDao.getAll() – цей метод повертає потік (*Flow*) зі списком усіх квитків з бази даних. Це дозволяє програмі реагувати на зміни в базі даних в реальному часі.

suspend fun insert(ticket: Ticket) та *suspend fun insert(ticket: List<Ticket>)* – ці методи додають один квиток або список квитків в базу даних відповідно. Ключове слово *suspend* вказує, що ці функції можуть бути призупинені та відновлені, що дозволяє ефективно працювати з асинхронними операціями, такими як взаємодія з базою даних.

					КРБ.КІ.1.440-03.1.16	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дат		

suspend fun deleteAllTickets() – цей метод видаляє всі квитки з бази даних.

suspend fun delete(ticket: Ticket) – цей метод видаляє конкретний квиток з бази даних.

Закоментований метод *suspend fun findTicketByName(eventName: String): Ticket?* може бути використаний для пошуку квитка за назвою події.

Усі ці методи дозволяють інкапсулювати логіку взаємодії з базою даних, що спрощує подальшу розробку та тестування програми.

2. *AuthActivity.kt*, *LoginActivity.kt*, *RegistrationFragment.kt*,
EventDetailFragment.kt, *SettingsFragment.kt*, *TheatreDetailFragment.kt*,
TheatreFragment.kt, *TheatresActivity.kt*, *MyTicketsFragment.kt*, *MainActivity.kt*,
SplashActivity.kt, *HomeFragment.kt*, *PickTicketFragment.kt* - це файли, які керують роботою інтерфейсу користувача. Вони відповідають за відображення відповідних вікон, обробку дій користувача (кліки по кнопкам, введення тексту тощо) та динамічну зміну вмісту вікон.

Клас *MainActivity* є головною активністю в моїй програмі.

Вона є вхідною точкою для користувача після аутентифікації. У ньому визначається конфігурація для *App Bar*, також в цьому класі є зв'язка між *XML* – макетом та кодом програми, контролер навігації, який управляє переходами між фрагментами в рамках цієї активності, *ViewModel* для управління даними про квитки, а також для управління даними які діляться між різними компонентами системи. Також тут знаходиться масив міст, який використовується для налаштування випадаючого списку в *Navigation Drawer*, а також присутній список який містить дані про театри.

В методі *onCreate()*, викликається при створенні активності, відбувається ініціалізація зв'язування, налаштування навігаційного контролера, налаштування *App Bar* та налаштування *Navigation Drawer*.

onItemSelected – слухач для випадаючого списку міст в *Navigation Drawer*. Коли користувач вибирає місце зі списку, відповідне значення оновлюється в *ViewModel*.

					КРБ.КІ.1.440-03.1.16	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дат		

setNavigationItemSelectedListener – слухач для обробки вибору елементів меню в *Navigation Drawer*. Коли користувач вибирає елемент меню, відбувається відповідна навігація або, у випадку виходу, закривається програма.

Метод *onSupportNavigateUp()* використовується для обробки навігації вгору в *App Bar*.

Метод *onBackPressed()* обробляє натискання кнопки "Назад" користувачем. Якщо *Navigation Drawer* відкритий, він закривається; в іншому випадку, обробка натискання кнопки "Назад" передається до системи.

TheatreEvent.kt, *Theatre.kt* – ці файли, використовуються для управління об'єктами "Театральний захід" та "Театр", які змінюються в реальному часі в залежності від даних, отриманих з бази даних.

MyApplication.kt – цей файл використовується для ініціалізації і контролю життєвого циклу додатка, що робить його частиною динамічної частини системи. Клас *MyApplication* є спеціалізованою версією класу *Application*. Цей клас використовується для створення глобального контексту програми.

lateinit var database: AppDatabase – це об'єкт бази даних, що зберігається в статичному полі та доступний в усій програмі.

override fun onCreate() – цей метод викликається при створенні програми. В ньому створюється база даних за допомогою *Room*, використовуючи *Room.databaseBuilder(...)*. Отриманий об'єкт бази даних зберігається в *database* для подальшого використання. Також тут викликається метод *DarkLightTheme.getMode(this)*, що отримує поточний режим теми (темна чи світла).

Основною метою цього класу є ініціалізація об'єктів, які мають бути доступні на рівні всієї програми. У цьому випадку, ми ініціалізуємо базу даних та визначаємо поточний режим теми.

TheatreEventAdapter.kt, *TheatreAdapter.kt*, *MyTicketsAdapter.kt* - адаптери використовуються для управління динамічними списками та сітками в

					КРБ.КІ.1.440-03.1.16	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дат		

інтерфейсі користувача. Вони динамічно змінюють вміст, опираючись на дані, що надходять з бази даних.

NightMode.kt – цей файл використовується для управління нічним режимом в додатку, який може динамічно вмикатися та вимикатися.

Це лише кілька прикладів використання динамічної частини системи у моєму проекті. Ці функції та класи необхідні для забезпечення правильного функціонування мого додатку, а також можуть бути використані як приклади для розробки подібних функціональностей у інших проектах.

Висновок до третього розділу

1. У ході проведеної роботи було розроблено програмну та візуальну структуру для реалізації застосунку для замовлення театральних квитків.

2. Розроблено базу даних для роботи з даними, які користувачі передають, та які система отримує.

3. Мобільний застосунок був запущений на віртуальному середовищі, у ході його роботи та тестування не було виявлено помилок.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 4

ОПИС РОБОТИ ЗАСТОСУНКУ

У даному розділі я опишу як користувач може користуватися додатком, та який функціонал йому доступний. Насамперед, додаток розроблювався з метою надання користувачеві можливості замовити театральні квитки, тому цей блок є найважливішим й обов'язковим.

Щодо інших блоків – таких, як, наприклад, функціонал зміни теми додатку (на темний чи світлий), то даний функціонал у моєму додатку необхідний для того, щоб користувач отримав найкращий *user-experience* від використання додатку.

Також користувачеві необхідно було надати можливість отримати деяку обов'язкову інформацію про саму подію, заради того щоб йому було легше та чіткіше вирішити, яку подію він все-ж таки хоче обрати.

Але почати необхідно з самого початку, а саме з реєстрації користувача та його входу у систему.

4.1 Реєстрація та авторизація користувача у системі

При запуску додатку, користувач одразу потрапляє на сторінку логіну у випадку, якщо він ще не входив в систему, або одразу ж на домашню сторінку зі списком театральних подій по обраному місту.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дат		

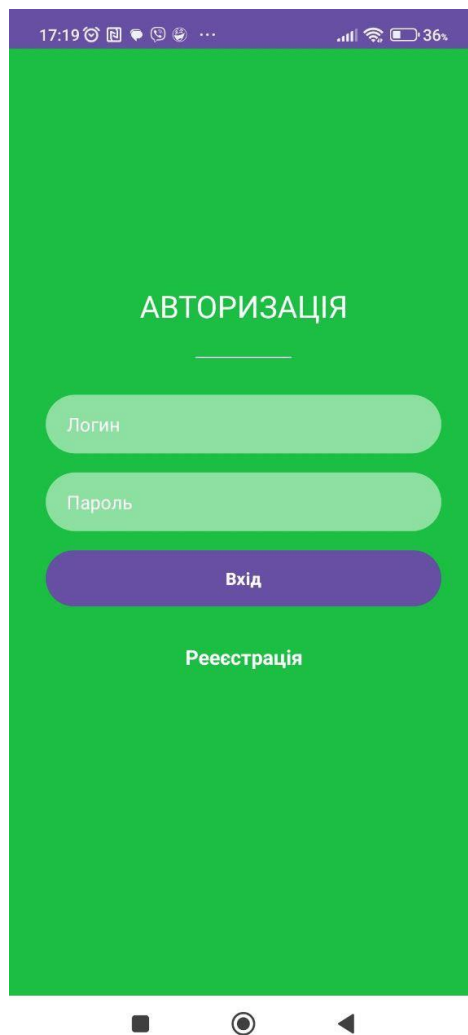


Рис. 4.1 – Сторінка авторизації користувача

На цій сторінці користувачеві необхідно ввести свій логін та пароль (пароль при вводі відображається зірочками), та натиснути «Вхід», після чого користувач зареєструється у систему.

Але, у випадку, якщо у користувача немає облікового запису, йому необхідно натиснути на «Реєстрація» на сторінці авторизації, після чого він потрапить на сторінку, де зможе зареєструвати користувача.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дат		

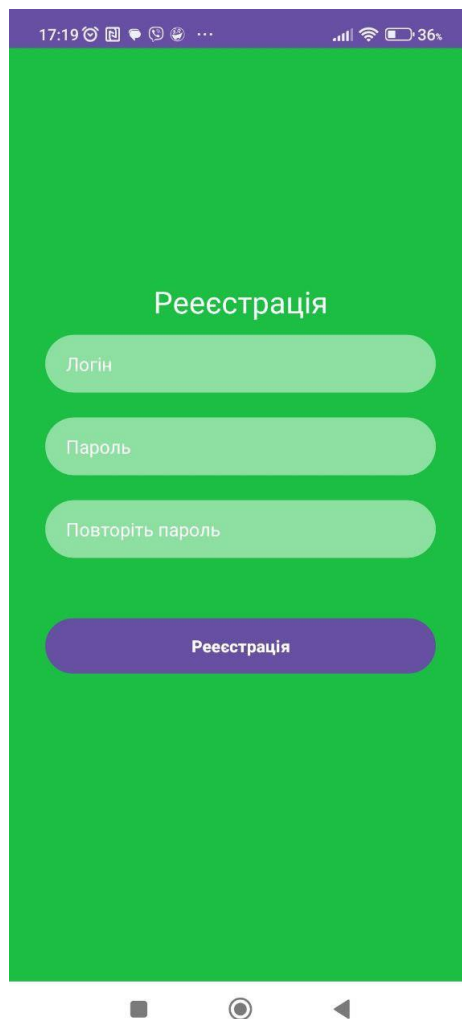


Рис. 4.2 – Сторінка реєстрації облікового запису

У полі «Логін» необхідно ввести бажаний користувацький логін, за яким буде здійснюватися вхід до системи, а у полях «Пароль» та «Повторіть пароль» необхідно ввести бажаний пароль, який теж буде використовуватися для входу до системи. Символи в обох полях з паролями для користувача будуть відображатися у вигляді зірочок. Але для комфорту та зручності, останній введений символ у цих полях буде відображатися на декілька секунд, щоб користувач не загубився під час створення облікового запису.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дат		

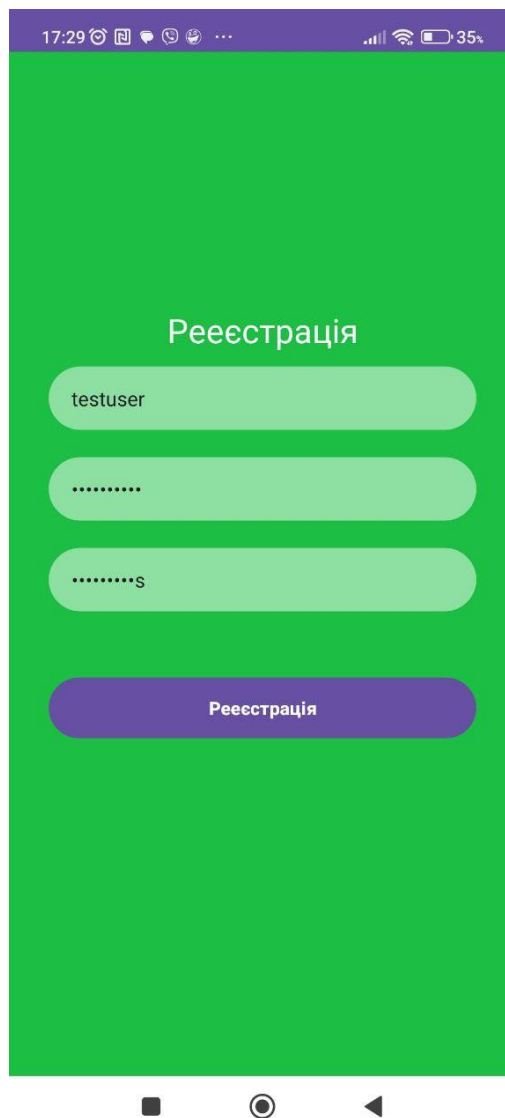


Рис. 4.3 – Введення даних для реєстрації користувача

Після реєстрації, користувач може спокійно зайти до системи, використовуючи свої облікові дані.

4.2 Бічне меню додатка та основні його сторінки

У бічному меню додатка, користувач може переглянути сторінки, які йому доступні, а саме:

1. «Оберіть місто» - даний функціонал відповідає за обрання користувачем міста, яке необхідне для того, щоб по обраному місту

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дат		

відображалися його події, а також надалі на сторінці «Театри» відображалися театри цього міста.

2. «Домашня сторінка» – на цій сторінці користувач може передивлятися всі театральні події обраного міста, бронювати квитки та дивитися детальну інформацію щодо події.

3. «Театри» – тут відображається інформація про театри обраного міста

4. «Налаштування» – на цій сторінці є можливість обрання кольорового режиму а також занесення особистих даних, таких як «Ім'я», «Прізвище» та «Дата народження».

5. «Мої білети» – тут користувач може переглянути заброньовані ним білети та невеличку інформацію по ним.

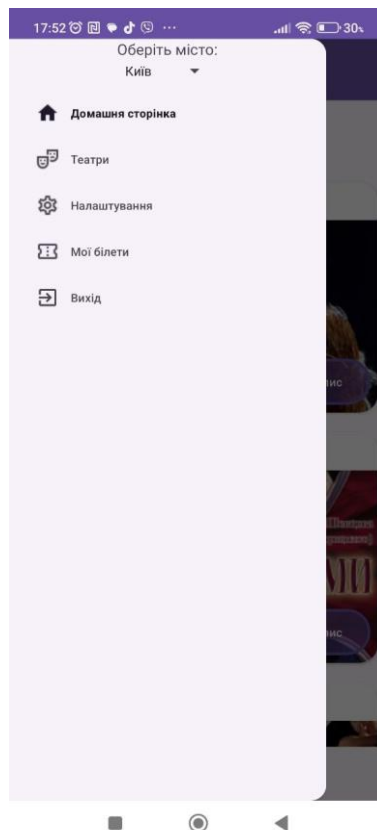


Рис. 4.4 – Бічне меню додатку

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дат		

4.3 Заовлення та перегляд заброньованого театрального білету

Після входу до системи, користувач одразу потрапляє на «Домашню сторінку», на якій він може побачити список театральних подій, доступних для обраного міста, а також назву обраного міста у нижній частині екрану.

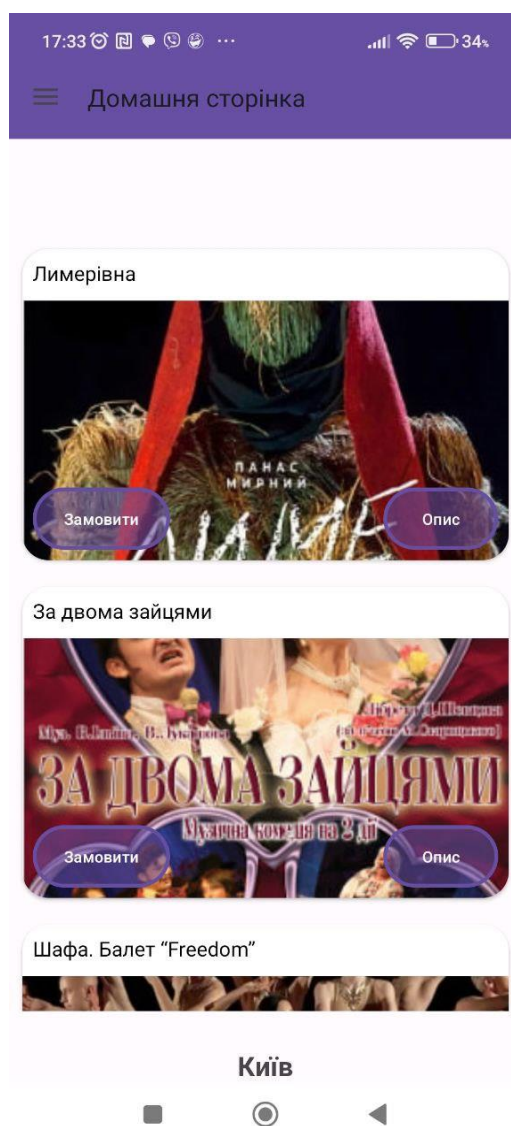


Рис. 4.5 – Домашня сторінка користувача

Також, на плашці кожної з подій на цій сторінці користувач може натиснути на кнопку «Замовити», яка перенесе його на сторінку замовлення білету, або на «Опис», після натискання якої користувачеві відкриється сторінка з описом події.

Після натискання на кнопку «Опис», користувач може передивитися детальну інформацію щодо події, а саме:

- назва події;
- дата початку;
- постер до події;
- опис або передісторія;
- дійові особи, якщо вони присутні.

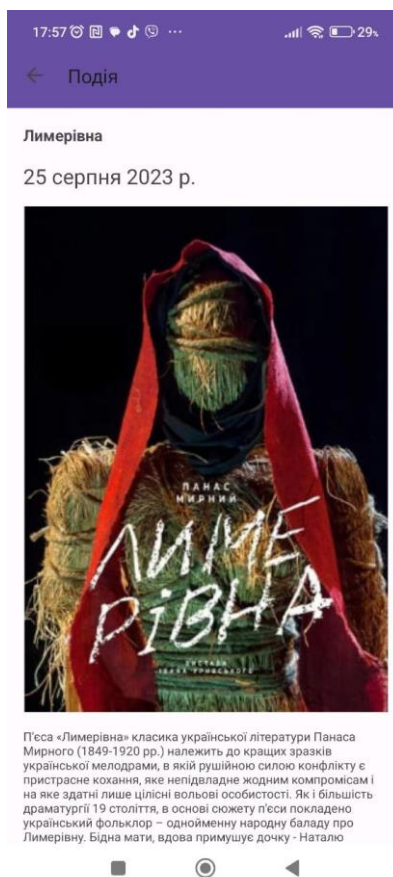


Рис. 4.6 – Сторінка опису події

Після натискання на кнопку «Замовити», користувач може передивитися доступні для нього місця та отримати інформацію про доступні.

					КРБ.КІ.1.440-03.1.16	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дат		



Рис. 4.7 – Сторінка замовлення білетів

На даній сторінці, вже зайняті місця відображаються прямокутником з темно-синім фоновим кольором, а вільні місця – з білим фоновим кольором. Таким чином, користувачеві доволі легко зорієнтуватися, які місця вільні, а які-зайняті.

Натиснувши на одне з вільних місць, система запитає у користувача підтвердження вибору місця з варіантами вибору «Так» та «Ні».

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дат		

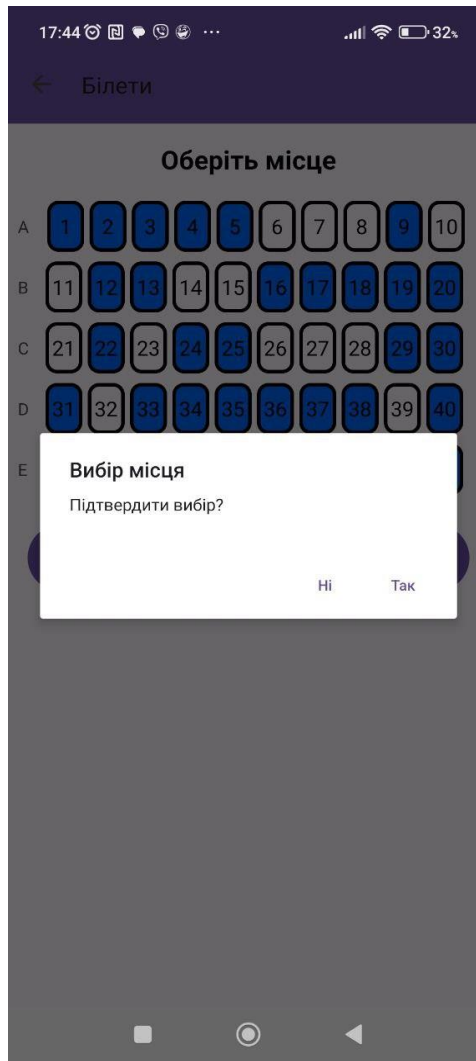


Рис. 4.8 – Сторінка підтвердження вибору місця

Якщо користувач натискає «Ні», то спливаюче вікно закривається й користувач повертається до вибору місць, а якщо натискає «Так», то обране місце помічається жовтим фоновим кольором. Також користувач має можливість обрати декілька місць для бронювання.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дат		

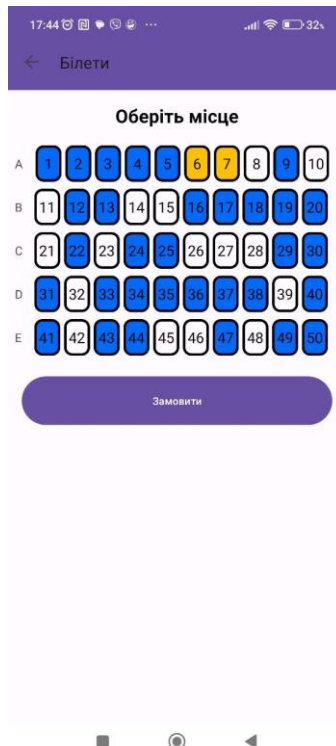


Рис. 4.9 – Обрані користувачем місця для бронювання

Після натискання на кнопку «Замовити», білети бронюються а користувача переносить до «Домашньої сторінки», а також відображається спливаюче повідомлення про те, що замовлення прийнято.

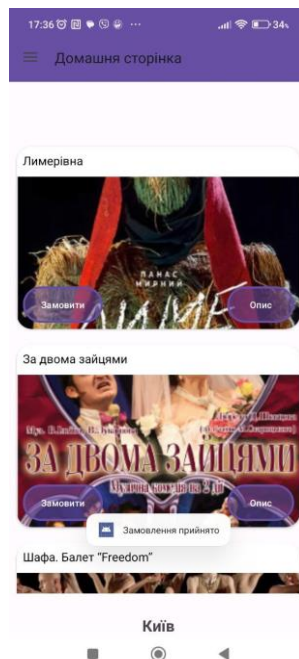


Рис. 4.10 – Відображення повідомлення про прийняття замовлення на домашній сторінці

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дат		

Для того, щоб подивитися усі заброньовані користувачем білети, він може перейти у бічне меню та натиснути на пункт меню «Мої білети», де він побачить свої білети, а також інформацію про їх назви та зайняті місця.

Також, на цій сторінці користувач може відмінити бронь, натиснувши на «Відмінити резервування», якщо він змінив свою думку щодо відвідування цієї події.

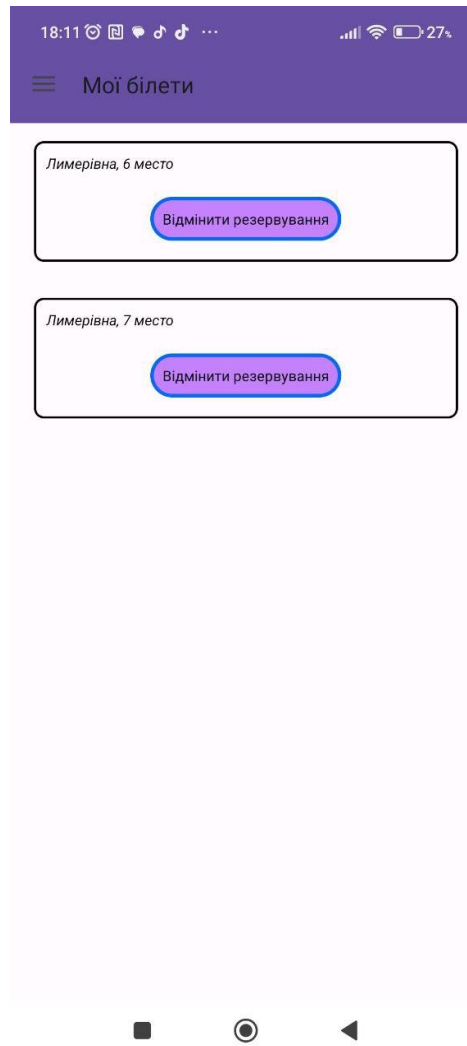


Рис. 4.11 – Сторінка Мої білети

4.4 Сторінка театрів

Для отримання більш детальної інформації щодо театрів обраного міста, користувач може перейти до сторінки «Театри» з бічного меню, де він одразу

					КРБ.КІ.1.440-03.1.16	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дат		

побачить інформацію про назву, дату та деталі заснування театру, фото театру, а також деяку історичну частину життя театру.



Рис. 4.12 – Перша частина сторінки Театри



Рис. 4.13 – Друга частина сторінки Театри

					КРБ.КІ.1.440-03.1.16	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дат		

В залежності від обраного міста, на цій сторінці може відображатися різна інформація про театри.

4.5 Налаштування додатку

Якщо користувачеві більш до вподоби темний кольоровий режим, то він може його обрати на сторінці «Налаштування» (за замовчуванням, кольоровий режим застосунку – світлий). Також, для того, щоб ввести особисті дані, користувач вводить їх у поля «Ім'я», «Прізвище» та «Дата народження» та натиснути «Зберегти», щоб особисті дані та налаштування додатку були збережені для цього облікового запису.

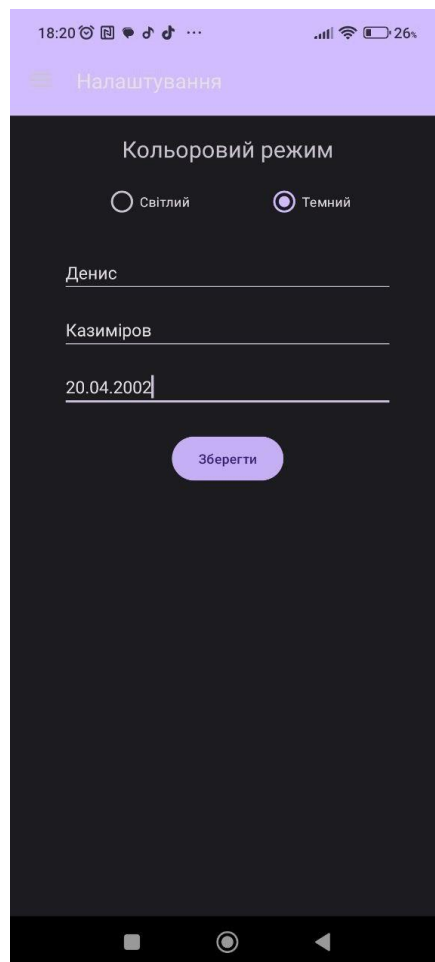


Рис. 4.14 – Сторінка налаштувань додатку

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дат		

Тепер, при переході на будь-які сторінку додатку, при обраному темному кольоровому режимі, кольорова гамма застосунку дещо зміниться та дійсно буде у темному стилі.

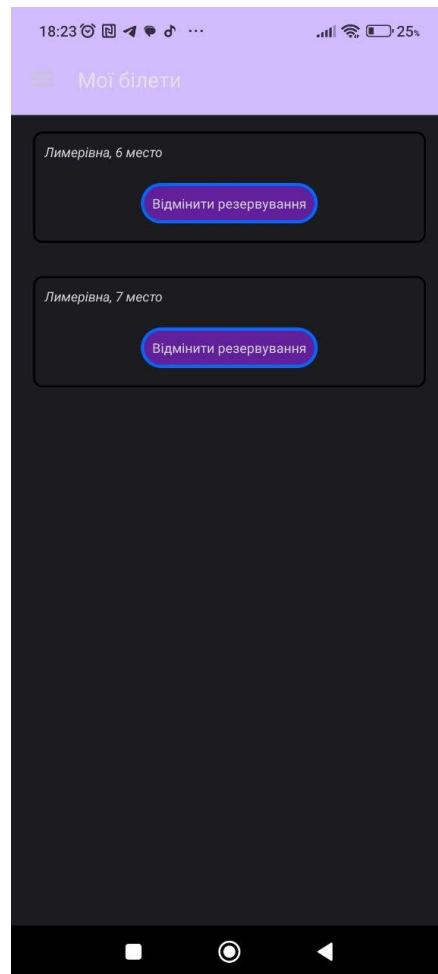


Рис. 4.15 – Сторінка білетів користувача у темному стилі

Висновок до четвертого розділу

1. Продемонстровано роботу та функціонал сторінок реєстрації та авторизації.
2. Відображено процес заказу театрального білета та його перегляду на сторінці білетів.
3. Показано додатковий функціонал застосунку, такий як зміна кольорової теми та перегляд театрів міста.

					КРБ.КІ.1.440-03.1.16	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 5

ЕКОНОМІЧНІ РОЗРАХУНКИ

5.1 Аналіз економічної частини

В організаційно-економічній частині дипломної роботи проводиться техніко-економічне обґрунтування (ТЕО) проектування та розробки мобільного застосунку для замовлення театральних квитків.

Метою даного проекту є створення функціонального та ефективного мобільного додатку, що дозволить користувачам зручно та швидко замовляти квитки на театральні вистави.

В сучасному світі майже в кожному домі є комп'ютери, які використовуються для розваги, роботи, відпочинку: хтось просто слухає музику, хтось користується додатками для замовлення їжі, або театральних квитків, інші працюють, або дивляться відео. Технологічний прогрес рухається гігантськими кроками – двадцять років тому люди користувалися простими мобільними додатками як *Fandango*, а зараз в нас є такі продвинуті додатки як: *TodayTix*, *TKTS*, *StubHub*. Сучасність надає нам можливості доповненої та віртуальної реальності, раніше функції сучасних мобільних додатків здавалися нам як чудо, а зараз будь яка людина може навчитися програмуванню і розробляти власні додатки. Тож, скільки можна заробити на цьому? У цьому розділі ми і дізнаємось скільки коштує розробка мобільного додатка. Проведемо оцінку ефективності проекту, який був реалізований під час виконання дипломної роботи.

Оцінка ефективності є важливою складовою управління проектом, оскільки дозволяє визначити ступінь досягнення поставленої мети та оцінити успішність виконання проекту.

Також буде розглянуто вплив розробки додатка на подальше життя користувачів, а також розрахована вартість затрат на цей проект. У контексті

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дат		

відкритої ринкової економіки спостерігається розширення сфери оцінки ефективності науково-технічного розвитку та поява нових видів основних ефектів НДДКР, які потребують оцінки.

Ці ефекти охоплюють:

1. Ефект маркетингу, який відображає комерційний успіх у ринковій сфері. Він проявляється у зростанні попиту, продажів та прибутку з мобільного додатка для замовлення театральних квитків. Цей ефект є відображенням ефективних маркетингових стратегій, привабливості додатка для користувачів та його конкурентоспроможності на ринку.

2. Науково-технічний ефект, який демонструє досягнення та результати наукових та технічних досліджень, використаних під час розробки додатка. Цей ефект виявляється у підвищенні ефективності, зниженні витрат, збільшенні якості та стимулюванні інновацій. Він сприяє прогресу та розвитку відповідної сфери, а також забезпечує конкурентну перевагу на ринку.

3. Соціальний ефект відображає позитивний вплив додатка на суспільство. Він проявляється у покращенні доступності та комфорту покупки театральних квитків, сприянні культурному розвитку, розвитку освіти, та інших аспектах, які підвищують соціальне благополуччя та рівень культурного розвитку.

4. Економічний ефект показує прибутковість та фінансову вигоду від додатка. Він проявляється у збільшенні прибутку, зниженні витрат, підвищенні ефективності, створенні нових робочих місць, стимулюванні економічного зростання та розвитку, привабливості для інвестицій та поліпшенні конкурентоспроможності розробника чи регіону.

Ці ефекти є ключовими при оцінюванні ефективності науково-технічного розвитку і допомагають визначити вплив на результати та досягнення проектів у даній сфері. Тепер можна провести оцінку ефективності розробленого додатку.

					КРБ.КІ.1.440-03.1.16	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дат		

5.2 Маркетинговий ефект

Маркетинговий ефект дипломної роботи може проявлятися з різних позицій. З одного боку, якщо використати дану якісну розробку мобільного додатка для замовлення театральних квитків як частину свого портфоліо при наданні послуг у сфері програмування, це може привернути увагу потенційних замовників, як стабільних клієнтів, так і тих, хто замовляє послуги одноразово. З іншого боку, оскільки розробка проводилася в якості проекту для кафедри, його буде достатньо часто бачити аудиторія, що служить додатковим засобом підвищення репутації як розробника додатків. Будучи створеним в якості реклами, цей проект буде привертати людей до нашого навчального закладу, а також залучати більше студентів та ресурсів для подальшого розвідку освітніх програм у галузях кафедри комп'ютерної інженерії.

Ринок розробки мобільних додатків є дуже конкурентним, але, як показує практика, людей, здатних створити дійсно якісний продукт, не так багато. Велику частину ринку захопили люди з початковим рівнем розробки, а люди, які користуються цими послугами, часто не знають, що можливо краще та дешевше.

Визначення ключових гравців на ринку розробки мобільних додатків для покупки театральних квитків може бути складним, але деякі країни можна виділити:

1. США: Величезний ринок цифрових продуктів США включає значну кількість компаній, що спеціалізуються на розробці мобільних додатків, включаючи додатки для покупки театральних квитків. Серед них – *TodayTix* та *BroadwayHD*.

2. Китай: Китай відомий своєю масштабною ІТ-індустрією. Міста, як Пекін та Шанхай, є важливими центрами розробки мобільних додатків. Проте важливо відзначити, що застосунки для покупки театральних квитків в Китаї зазвичай є частиною більших платформ, таких як *WeChat* або *Alibaba*.

					КРБ.КІ.1.440-03.1.16	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дат		

3. Індія: Індія є одним з найбільших ринків розробки ПЗ у світі. Вони мають значну кількість талановитих розробників додатків, що створюють програми для різних сфер, включаючи продаж театральних квитків.

4. Ізраїль: Ізраїль став відомим як "стартап-нація", завдяки великій кількості інноваційних компаній, що базуються в цій країні. Він вирізняється високотехнологічними рішеннями в області мобільних додатків, включаючи додатки для покупки театральних квитків.

Незважаючи на географічне розташування цих країн, інтернет та віддалена робота надають можливість розробникам та компаніям працювати на глобальному рівні, обслуговуючи клієнтів по всьому світу. Тому якісні мобільні додатки для покупки театральних квитків доступні в різних країнах, незалежно від їхнього географічного розташування.

5.3 Науково-технічний, соціальний та економічний вплив

Науково-технічний вплив - протягом процесу розробки мобільного додатка для замовлення театральних квитків було накопичено велику кількість знань в цій сфері. Це, в свою чергу, забезпечить більш ефективний підхід до подібних проєктів в майбутньому, зменшуючи витрати часу та збільшуючи коефіцієнт корисної дії. Буде більше часу на освоєння нових технологій і засобів розробки додатків, що сприятиме покращенню якості кінцевого продукту.

Соціальний вплив – цей мобільний додаток надасть користувачам можливість швидко знайти та замовити квитки на театральні вистави, економлячи час. Це збільшує їхній інтерес до театрального мистецтва та сприяє його розвитку в суспільстві. Додаток, якщо він зацікавить користувачів, може дати їм докладну інформацію про вистави та театри, що посилює їхнє включення в культурне життя.

Економічний вплив додатка розглядається з точки зору потенційного ринку, можливих доходів, розвитку навичок, залучення інвестицій та

					КРБ.КІ.1.440-03.1.16	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дат		

потенціалу співпраці. Цей проект принесе користь театрам та іншим культурним установам, збільшуючи їхню аудиторію, а для розробника він може стати джерелом прибутку в довготривалій перспективі. Він може також привернути увагу потенційних роботодавців в області розробки мобільних додатків.

5.4 Розрахунок коефіцієнта оцінки науково-технічної ефективності

За проведеним орієнтовним розрахунком витрати на розробку складатимуть 55593 грн. Детальніші статті витрат наведені в таблиці 5.1.

Таблиця 5.1

Статті витрат на розробку

Статті витрат	Кількість	Одиниця виміру	Ціна (грн)	Вартість (грн)
Зарплата студента	140 годин	годин	100	14000
Зарплата керівника	20 годин	годин	150	3000
Електроенергія	1 місяць	місяць	93	93
Опалення	1 місяць	місяць	950	950
Інтернет	1 місяць	місяць	450	450
Комп'ютерна техніка	1 шт	штука	35000	35000
Статті витрат	Кількість	Одиниця виміру	Ціна (грн)	Вартість (грн)
Програмне забезпечення	3 шт	штука	700	2100
Загалом:				55593

Розрахунок витрачання грошей на електроенергію (140 годин роботи):

Ціна за світло – 2,64 грн за 1кВт

Споживання комп'ютером: 250 Вт/год

$$\frac{250 \cdot 140}{1000} \cdot 2,64 = 92,4$$

Науково-технологічну ефективність (НТЕ) висновків прикладних досліджень встановлюють відповідно до критеріїв науково-технічного стандарту.

Визначення науково-технологічної продуктивності НДР виконується на основі коефіцієнта (ОНТЕ), який відображає міру наближення до найвищого потенціального рівня, величина якого становить 1 (одиниця):

$$O_{\text{НТЕ}} = \frac{K^{\Phi}_{\text{НТЕ}}}{K^{\Pi}_{\text{НТЕ}}} \quad (5.1)$$

де $K^{\Phi}_{\text{НТЕ}}$ – показник (коефіцієнт) фактичного рівня науково-технічної ефективності;

$K^{\Pi}_{\text{НТЕ}}$ – показник (коефіцієнт) потенціально можливого рівня науково-технічної ефективності (дорівнює одиниці).

Значення показника $K^{\Phi}_{\text{НТЕ}}$ визначають на основі шкали експертних оцінок (табл. 5.2)

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дат		

**Шкала експертних оцінок для виміру рівня
науково-технічної ефективності проектів**

№	Групи показників	Характеристика показників	Інтервал рейтингового числа	Коефіцієнт значущості показників
1	Науково-технічний рівень	Перевищує кращі світові аналоги	10	0,35
		Відповідає світовому рівню	7 – 9	
		Нижче кращих світових аналогів	5 – 6	
		Перевищує кращі вітчизняні аналоги	3 – 4	
		Відповідає вітчизняному рівню	1 – 2	
		Нижче вітчизняного рівня	0	
2	Перспективність	Першочергова значущість	8 – 10	0,35
		Значущий	5 – 7	
		Корисний	1 – 4	
3	Потенційний масштаб практичного використання	Світовий ринок	10	0,20
		Галузі національної економіки	7 – 9	
		Галузь (регіон)	3 – 6	
		Окремі підприємства (об'єднання)	1 – 2	
4	Ступінь вірогідності досягнення позитивних результатів	Великий	10	0,10
		Середній	5 – 9	
		Малий	1 – 4	

Примітка: об'єкт оцінки і аналог(и), які порівнюють за однаковими показниками, наведеними у співставленому вигляді відхилення в значеннях кожного з показників, мають бути однаковими для варіантів, що порівнюються.

Проведення оцінки

Визначають $K^{\Phi}_{НТЕ}$ на основі експертної оцінки науково-технічного рівня розробки.

					КРБ.КІ.1.440-03.1.16	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дат		

З цією метою:

1. Розроблюють перелік специфічних показників, необхідних для виміру науково технічного рівня розробки.

2. Формують групу аналогів, які реалізовані на світовому і вітчизняному ринках.

3. Здійснюють відповідні розрахунки для співставлення показників і визначення балів по табл. 5.2.

До числа специфічних показників відносять:

1. **Для нової техніки:** продуктивність, споживання інженерних ресурсів на виробітку одиниці продукції, потреба в робочих, які обслуговують обладнання, експлуатаційні витрати на одиницю продукції.

2. **Для нових матеріалів і речовин:** вміст корисних речовин для виробітки готової продукції, питома вага відходів у загальному обсязі переробленої сировини, вартість одиниці ... нового матеріалу.

3. **Для нових технологій:** якість виробленої продукції, енергоємність і трудомісткість продукції, собівартість одиниці продукції.[27]

З метою спрощення визначення $K_{НТЕ}^Ф$ у табл. 5.3 не введено показника витрат на одиницю продукції.”

Таблиця 5.3

Порівняльні показники для виконання оцінки НТЕ

ПОКАЗНИКИ	Варіанти технології	
	розробленої	співвідносної (аналога)
Рівень новизни	високий	світовий
Якість продукції	вища	вища
Використання ресурсів пристрою	висока	висока
Дизайн	середній	високий
Зв'язок з сервером	середньої якості	високоякісний

“На основі співставлення даних таблиці встановлюють бали по характеристиках чотирьох груп і на цій основі розраховують значення інтегрального показника НТЕ:

$$НТЕ = \sum B_i \times K_i^3 \quad (5.2)$$

де $i = 1 \div 4$,

B_i – бали (рейтингове число),

K – коефіцієнт значущості показників.

Рівень науково-технічної ефективності НДДКР розраховано на основі наведених даних прикладу” (табл. 5.4).

Таблиця 5.4

Експертна оцінка і розрахунок величини інтегрального показника НТЕ

№	Групи показників	Рейтинг експертів			Середня за експертними оцінками	НТЕ
		1	2	3		
1	Науково-технічний рівень	6	7	8	7	2,45(7 x 0,35)
2	Перспективність	8	7	9	8	2,8(8 x 0,35)
3	Потенційний масштаб практичного використання	8	9	7	8	1,6(8 x 0,20)
4	Ступінь вірогідності досягнення позитивних результатів	6	7	6	6,33	0,633(6,33x0,10)
В С Ь О Г О						7,483

$$НТЕ = 7 \cdot 0,35 + 8 \cdot 0,35 + 8 \cdot 0,20 + 6,33 \cdot 0,10 = 2,45 + 2,8 + 1,6 + 0,633 = 7,483$$

Отже, потрібно зіставити отриманий результат з максимальним показником, який дорівнює 10 балам ($10 \cdot 0,35 + 10 \cdot 0,35 + 10 \cdot 0,2 + 10 \cdot 0,1$).

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дат		

Таким чином, оцінювання рівня науково-технічної ефективності (НТЕ) можна провести за допомогою інтегрального коефіцієнта оцінки науково-технічної ефективності (НТЕ) ($K_{НТЕ}$):

$$K_{НТЕ} = \frac{НТЕ}{10} * 100 \% \quad (5.3)$$

На основі даних табл. 6.4 можна дійти до висновку, що $K_{НТЕ}$ відповідає 74,83%, тобто:

$$\frac{7,483}{10} * 100\% = 74,83 \%$$

В тому випадку, коли значення $K_{НТЕ}$ перевищує середнє значення, яке дорівнює 5,0, має бути зроблено висновок про достатній рівень НТЕ:

- цілком достатній 5,0 – 6,0;
- достатній 6,1 – 8,0;
- достатньо високий 8,1 – 9,0;
- високий 9,1 – 10.

Таким чином рівень НТЕ технології можна визначити достатньо високим. Отже, розроблену технологію пропонується впроваджувати у виробництво.”

Висновок до п'ятого розділу

1. Розробка мобільного додатка для замовлення театральних квитків є вигідним проектом з швидкою окупністю та потенціалом для додаткового прибутку у майбутньому. Розробка додатків для театральних подій є доцільним бізнесом, оскільки попит на цю послугу зростає з кожним роком.

2. Також можна зробити висновок, що розроблений мною застосунок є економічно вигідним та окупним.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 6 ОХОРОНА ПРАЦІ

6.1 Робота за комп'ютером

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів, спрямованих на забезпечення безпеки, збереження здоров'я та працездатності людини в процесі трудової діяльності.

Розробка мобільних додатків включає доволі тривале робоче місце за комп'ютером, що може призвести до ряду проблем зі здоров'ям. У зв'язку з цим, з метою забезпечення безпеки працівника, важливо враховувати такі аспекти, як ергономіка робочого місця, відповідність електробезпеки, запобігання вибухопожежній і пожежній небезпеці, дотримання правил санітарії.

Згідно з Законом України "Про охорону праці", роботодавець зобов'язаний створити на робочому місці безпечні умови праці, забезпечити обладнання та засоби індивідуального захисту, а також проводити навчання з питань охорони праці.

Робота з комп'ютером відноситься до 3 класу виробництва з незначною шкідливістю. Проте, необхідно враховувати деякі особливості цього виду діяльності.

6.2 Ергономіка робочого місця

Ергономіка є важливою складовою створення безпечних і продуктивних умов праці. Вона включає в себе дослідження і оптимізацію таких параметрів робочого місця, як розташування обладнання, освітлення, шум, вологість, температура, а також взаємодія людини з робочими процесами і технологіями:

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дат		

1. Розташування обладнання: При розробці мобільного додатку основним обладнанням є комп'ютер та периферійні пристрої. Монітор комп'ютера має бути розташований на відстані витягнутої руки (десь 50-70 см), а верхня його кромка – на рівні очей або трохи нижче. Клавіатура і мишка мають бути розташовані таким чином, щоб руки при роботі лежали на столі під кутом 90 градусів.

2. Освітлення: Освітлення має забезпечувати комфортну роботу за комп'ютером. Важливо уникнути прямого сонячного світла на моніторі, що може викликати блиск. Рекомендується використовувати джерела світла з розсіювачами та регулюванням інтенсивності світла.

3. Шум: Шум в робочій кімнаті може відволікати від роботи та підвищувати рівень стресу. Тому необхідно забезпечити достатній рівень тиші, а також використовувати навушники для зосередженості на роботі, якщо потрібно.

4. Вологість та температура: Зручні температурні умови зазвичай становлять від 20 до 24 градусів за Цельсієм, а вологість повітря повинна бути в межах 40-60%. Для контролю цих параметрів можна використовувати кліматичні пристрої та гігрометри.

5. Взаємодія людини з робочими процесами та технологіями: Щоб зменшити ризик перевантаження очей та забезпечити комфортну роботу, рекомендується робити короткі перерви кожні 45-60 хвилин. Відпочинок може включати вправи для очей, розтяжку та коротку ходьбу. Врахування цих ергономічних рекомендацій допоможе забезпечити здорові, безпечні та продуктивні умови праці під час розробки мобільного додатку.

6.3 Електробезпека

1. Електрообладнання: Повинно бути в гарному стані та мати надійну ізоляцію. Регулярно перевіряйте стан проводів і розеток, уникайте

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дат		

перевантаження мережі підключенням занадто багатьох пристроїв до однієї розетки.

2. Правила експлуатації: Не допускайте контакту електрообладнання з вологою, в тому числі при прибиранні приміщення. Не робіть ремонтних робіт в електромережі самостійно, якщо ви не є кваліфікованим електротехніком.

3. Використання захисних засобів: Використовуйте автоматичні вимикачі та диференціальні автомати, які швидко вимикають електрообладнання в разі виникнення несправностей, що можуть призвести до ураження електричним струмом. Згідно з Правилами будови електроустановок (ПБЕ), пункт 1.1.26, "всі приміщення, крім сухих, в яких особи постійно перебувають, включаючи житлові приміщення, приміщення для адміністративно-бюро-, науково-дослідної та проектно-конструкторської роботи, навчальні приміщення, а також санітарно-гігієнічні приміщення, ванні кімнати та приміщення для прання білизни в житлових будинках" відносяться до категорії з підвищеною небезпекою ураження електричним струмом. Згідно з пунктом 1.1.15 ПБЕ, "для захисту людей від ураження електричним струмом в приміщеннях, що відносяться до категорії з підвищеною небезпекою, потрібно використовувати автоматичні вимикачі захисного відключення".

4. Навчання та інструктажі: Навіть працюючи вдома, важливо знати основи електробезпеки. Відповідно до Закону України "Про охорону праці", стаття 16, роботодавець має забезпечити навчання та інструктаж працівників з питань охорони праці.

5. Перевірка електробезпеки: Після проведення ремонту або установки нового обладнання, а також просто періодично, варто провести перевірку стану електромережі. Ви можете замовити проведення перевірки в спеціалізованій організації або використати послуги кваліфікованого електрика.

6. Евакуаційні шляхи: У випадку виникнення пожежі через електромережу, важливо мати чітко розроблені та відомі всім мешканцям будинку евакуаційні шляхи. Згідно з вимогами ДБН В.2.5-56:2014 "Пожежна безпека об'єктів будівництва", в приміщенні повинні бути передбачені

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дат		

необхідні виходи для евакуації людей, включаючи додаткові виходи в разі блокування основного шляху евакуації.

Згідно з цими нормами, евакуаційні виходи мають бути вільними, а доступ до них – безперешкодним. Загалом, варто пам'ятати про регулярний контроль та профілактику усіх електричних пристроїв, що використовуються, виключно в сухих руках, відсутність доступу дітей до електричних приладів без нагляду дорослих.

Моя діяльність пов'язана з використанням комп'ютерної техніки, яка вимагає належного дотримання всіх норм та правил електробезпеки для забезпечення власного здоров'я та продуктивної роботи.

6.4 Класифікація виробництв за ступенем вибухової, вибухопожежної і пожежної безпеки

6.4.1 Визначення понять

Спочатку необхідно зрозуміти, що означають ці поняття.

1. Вибухова безпека — це ймовірність виникнення вибуху внаслідок взаємодії речовин з окислюючими агентами або під впливом зовнішніх джерел запалювання.

2. Вибухопожежна безпека — це ймовірність виникнення вибуху, що викликає пожежу.

3. Пожежна безпека — це ймовірність виникнення пожежі під впливом зовнішніх або внутрішніх чинників.

6.4.2 Класифікація виробництв.

Згідно з Правилами пожежної безпеки в Україні (НАПБ А.01.001-2004), виробництва класифікуються на чотири категорії за ступенем вибухової і пожежної безпеки – А, Б, В, Г.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дат		

Моє виробництво (розробка мобільного додатку для замовлення театральних квитків) відноситься до категорії "Г" – виробництва з низьким рівнем пожежної небезпеки.

Це зумовлено тим, що основні матеріали, що використовуються у моїй роботі (електронне обладнання), є негорючими, а рідина, що використовується для чищення обладнання (наприклад, спирт), зберігається в обмежених кількостях і використовується відповідно до інструкцій виробника.

6.4.3 Пожежна профілактика

Незважаючи на низьку пожежну небезпеку мого виробництва, необхідно дотримуватися основних норм пожежної безпеки:

1. Обладнання повинно перебувати в справному стані та періодично проходити технічний огляд.
2. Необхідно уникати перевантаження електричних мереж.
3. В приміщенні повинні бути засоби для гасіння пожежі - вогнегасники.
4. Всі працівники повинні знати евакуаційний план і бути в курсі дій у випадку пожежі.
5. Забороняється курити в приміщенні.
6. Виходячи з приміщення, необхідно відключати всі електроприлади від мережі.

6.4.4 Рекомендації

1. Постійно вивчати нормативні документи, що регулюють питання пожежної безпеки.
2. Проводити перевірку електромереж і електрообладнання, використовуючи електротехнічні засоби вимірювання.
3. негайно припинити роботу електрообладнання, якщо воно видає нехарактерний шум, запах горіння або дим.

					КРБ.КІ.1.440-03.1.16	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дат		

6.5 Профілактика промислової санітарії

Профілактика промислової санітарії включає в себе ряд дій, які спрямовані на підтримку гігієнічних умов праці та запобігання розвитку професійних захворювань.

Для мого робочого місця, де розробляється мобільний додаток для замовлення театральних квитків, основні аспекти промислової санітарії можуть включати в себе наступні пункти:

6.5.1 Санітарно-гігієнічні умови праці

1. Освітлення: Місце роботи повинно бути достатньо освітлене. Рекомендовано використовувати денне світло, якщо це можливо, але там, де це не досяжно, необхідне якісне штучне освітлення.

2. Вентиляція: Необхідно забезпечити достатню вентиляцію приміщення. Правильна вентиляція допомагає знизити концентрацію шкідливих речовин в повітрі, як-от пилу від електронного обладнання, та знижує рівень вологості.

3. Температура: Оптимальна температура для приміщення з комп'ютерною технікою становить 20-24°C. Це дозволить уникнути перегрівання обладнання та надмірного охолодження, що може призвести до розробки професійних захворювань.

6.5.2 Ергономіка робочого місця

1. Правильне регулювання висоти стола та стільця для роботи за комп'ютером.

2. Використання спеціальних ковриків для миші, які знижують навантаження на зап'ястя.

3. Регулярні перерви в роботі за комп'ютером для розминки та релаксації м'язів.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дат		

6.5.3 Запобігання шуму

Шум від комп'ютерного обладнання може стати причиною стресу та зниження продуктивності. Важливо використовувати тихе обладнання та забезпечити належну звукоізоляцію приміщення.

6.5.4 Профілактика втоми

- регулярні перерви протягом робочого дня;
- раціональне чергування роботи та відпочинку;
- виконання фізичних вправ для розтяжки під час перерв.

6.5.5 Виконання вимог Закону України

Виконання вимог Закону України "Про забезпечення санітарного та епідемічного благополуччя населення" та інших нормативних актів з питань охорони праці та промислової санітарії.

У загальному випадку, важливо зрозуміти, що здоров'я працівника залежить не тільки від виконання вимог безпеки на робочому місці, але і від його особистого ставлення до свого здоров'я та дотримання здорового способу життя.

Висновок до шостого розділу

У ході розробки цієї частини дипломної роботи було розглянуто важливість дотримання вимог до охорони праці при створенні мобільного додатку для замовлення театральних квитків.

Моє робоче місце, де проводиться розробка, знаходиться вдома, у кімнаті площею 6х5 метрів, що є частиною квартири площею 45 квадратних метрів. Це

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дат		

робоче місце відповідає основним вимогам ергономіки, в тому числі забезпечує комфортні умови для роботи з комп'ютером.

Було проаналізовано питання електробезпеки. При використанні електронного обладнання важливо дотримуватися правил безпечної експлуатації, зокрема, не користуватися пошкодженими електричними кабелями і вимикачами.

Враховуючи класифікацію виробництв за ступенем вибухової, вибухопожежної і пожежної небезпеки, робоче місце класифікується як клас П-Іа (низький рівень пожежної небезпеки).

Таким чином, за результатами проведеної роботи, я маю усвідомлення про важливість дотримання норм охорони праці та промислової санітарії під час виконання дипломного проекту. Впровадження цих рекомендацій допоможе забезпечити безпечні та продуктивні умови для продовження роботи над моїм проектом.

					<i>КРБ.КІ.1.440-03.1.16</i>	Арк.
						81
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>		

ЗАГАЛЬНІ ВИСНОВКИ

1. За результатами аналізу предметної області, було виявлено загальні риси мобільних застосунків для заказу театральних квитків.
2. Проаналізовані сучасні аналогічні застосунки для заказу театральних квитків з виявленням переваг та їх недоліків.
3. Здійснено постановку задачі із зазначенням бажаного результату, щодо того, що є необхідним для створення застосунку такого типу.
4. Під час проектування розроблено архітектуру застосунку.
5. В ході проведеної роботи розроблено програмну та візуальну складові застосунку.
6. Дана розробка проекту є вигідною за висновками того, що вона має перспективи прибутку та доволі швидко окупність.
7. У ході розробки цієї частини дипломної роботи було розглянуто важливість дотримання вимог до охорони праці при створенні мобільного додатку для замовлення театральних квитків.
8. Продемонстровано функціонал роботи додатку та різних його частин.

					КРБ.КІ.1.440-03.1.16	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дат		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційна документація *Android Studio* - *Android Studio* [Електронний ресурс]. – Режим доступу: <https://developer.android.com/studio/intro> (останнє звернення 04.06.2023).
2. Офіційна документація *Kotlin* – *Kotlin* [Електронний ресурс]. – Режим доступу: <https://kotlinlang.org/docs/home.html> (останнє звернення 04.06.2023).
3. Офіційна документація *SQLite* – *SQLite* [Електронний ресурс]. – Режим доступу: <https://www.sqlite.org/docs.html> (останнє звернення 04.06.2023).
4. Офіційна документація *RecyclerView* – *Android Studio* [Електронний ресурс]. – Режим доступу: <https://developer.android.com/develop/ui/views/layout/recyclerview> (останнє звернення 04.06.2023).
5. Офіційна документація для роботи зі світлом в *Android - Environment sensors* [Електронний ресурс]. – Режим доступу: https://developer.android.com/guide/topics/sensors/sensors_environment#java (останнє звернення 04.06.2023).
6. *SQLite* – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/SQLite> (останнє звернення 04.06.2023).
7. *Android Studio* – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Android_Studio (останнє звернення 04.06.2023).
8. *Kotlin* – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Kotlin> (останнє звернення 04.06.2023).
9. Розробка програмного забезпечення – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Розробка_програмного_забезпечення (останнє звернення 04.06.2023).
10. *Android* (операційна система) – Вікіпедія [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Android_\(операційна_система\)](https://uk.wikipedia.org/wiki/Android_(операційна_система)) (останнє звернення 04.06.2023).

					КРБ.КІ.1.440-03.1.16	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дат		

11. Архітектура програмного забезпечення – Вікіпедія [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/Архітектура_програмного_забезпечення (останнє звернення 04.06.2023).

12. Мобільний додаток – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Мобільний_додаток (останнє звернення 04.06.2023).

13. Платформа *Android* – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Платформа_Android (останнє звернення 04.06.2023).

14. Програмне забезпечення – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Програмне_забезпечення (останнє звернення 04.06.2023).

15. Інтегроване середовище розробки – Вікіпедія [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/Інтегроване_середовище_розробки (останнє звернення 04.06.2023).

16. Мобільна розробка – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Мобільна_розробка (останнє звернення 04.06.2023).

17. Мобільний додаток для покупки квитків – Вікіпедія [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/Мобільний_додаток_для_покупки_квитків (останнє звернення 04.06.2023).

18. Система керування базами даних – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Система_керування_базами_даних (останнє звернення 04.06.2023).

19. Серверна база даних – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Серверна_база_даних (останнє звернення 04.06.2023).

20. Мобільний маркетинг – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Мобільний_маркетинг (останнє звернення 04.06.2023).

21. Онлайн-платформа – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Онлайн-платформа> Мобільний інтерфейс – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Мобільний_інтерфейс (останнє звернення 04.06.2023).

22. Авторизація – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Авторизація> Інтернет-технології – Вікіпедія [Електронний

					КРБ.КІ.1.440-03.1.16	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дат		

ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Інтернет-технології> (останнє звернення 04.06.2023).

23. Мобільний комерційний додаток – Вікіпедія [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/Мобільний_комерційний_додаток (останнє звернення 04.06.2023).

24. Інтерфейс користувача - Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Інтерфейс_користувача (останнє звернення 04.06.2023).

25. Мобільна платформа - Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Мобільна_платформа (останнє звернення 04.06.2023).

26. Жива черга – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Жива_черга (останнє звернення 04.06.2023).

27. Методичні вказівки до оцінки науково-технічної ефективності розробки нової технології, нового обладнання та інших інновацій. Для студентів всіх спеціальностей СВО «бакалавр» і «магістр» денної і заочної форм навчання. Укладачі Басюркіна Н.Й., Свистун Т.В. Одеса: ОНТУ, 2022 р. 18 с.

28. Методичні вказівки до дипломної роботи бакалавра. Укладачі Князева Н.О., Шестоपालов С.В., Жуковецька С.Л. Одеська національна академія харчових технологій, 2016 р. 46с.

					КРБ.КІ.1.440-03.1.16	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дат		

ДОДАТКИ

Додаток Б Код додатку

Код файлу *MainActivity.kt*:

```
package com.example.theatretickets2.ui.main

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.view.MenuItem
import android.view.View
import android.widget.AdapterView
import android.widget.AdapterView.OnItemClickListener
import android.widget.ArrayAdapter
import android.widget.Spinner
import androidx.activity.viewModels
import androidx.appcompat.app.AppCompatActivity
import androidx.appcompat.widget.Toolbar
import androidx.core.view.GravityCompat
import androidx.navigation.NavController
import androidx.navigation.NavOptions
import androidx.navigation.fragment.NavHostFragment
import androidx.navigation.ui.AppBarConfiguration
import androidx.navigation.ui.NavigationUI
import com.example.theatretickets2.databinding.ActivityMainBinding
import com.example.theatretickets2.R
import com.example.theatretickets2.ui.auth.AuthActivity
import com.example.theatretickets2.viewModels.SharedViewModel
import com.example.theatretickets2.ui.main.theatre.Theatre
import com.example.theatretickets2.viewModels.TicketViewModel

class MainActivity : AppCompatActivity() {

    private lateinit var appBarConfiguration: AppBarConfiguration
    private lateinit var binding: ActivityMainBinding
```

					КРБ.КІ.2.440-03.1.16	Арк.
						86
Змн.	Арк.	№ докум.	Підпис	Дат		

```

private lateinit var navController: NavController

// private lateinit var db: AppDatabase

private val ticketViewModel: TicketViewModel by viewModels()
private val viewModel: SharedViewModel by viewModels()
private val cities = arrayOf("Одесса", "Киев", "Львов")
val theatres = listOf(

    Theatre("Театр 1", "Описание 1", R.drawable.theatre1_image,
"Одесса"),
    Theatre("Театр 2", "Описание 2", R.drawable.theatre2_image,
"Киев"),
    Theatre("Театр 3", "Описание 3", R.drawable.theatre3_image,
"Львов"),
    // ...
)
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
// lifecycleScope.launch {
//     ticketViewModel.deleteAllTickets()
// }

    viewModel.theatres.value = theatres // Pass theatres to the
ViewModel

    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)

    val navHostFragment =
supportFragmentManager.findFragmentById(R.id.nav_host_fragment_activity_m
ain) as NavHostFragment
    navController = navHostFragment.navController

    appBarConfiguration = AppBarConfiguration(setOf(
        R.id.homeFragment,
        R.id.theatreFragment,
        R.id.settingsFragment,

```

					КРБ.КІ.2.440-03.1.16	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        R.id.myTicketsFragment,
        R.id.exit
    ), binding.drawerLayout)
    val toolbar: Toolbar = findViewById(R.id.toolbar)
    setSupportActionBar(toolbar)
    NavigationUI.setupActionBarWithNavController(this,
navController, appBarConfiguration)
    NavigationUI.setupWithNavController(binding.navView,
navController)
    val spinner =
binding.navView.getHeaderView(0).findViewById<Spinner>(R.id.spinner)

    // Setup the spinner
    val adapter = ArrayAdapter(this,
android.R.layout.simple_spinner_item, cities)

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_
item)

    spinner.adapter = adapter
    spinner.onItemSelectedListener = object :
AdapterView.OnItemSelectedListener {
        override fun onItemSelected(parent: AdapterView<*>,
view: View?, position: Int, id: Long) {
            val city =
parent.getItemAtPosition(position).toString()
            viewModel.selectedCity.value = city // Update the
selected city in the ViewModel
            Log.d("MainActivity", "Selected city: $city")
        }

        override fun onNothingSelected(parent: AdapterView<*>)
{}
    }
    binding.navView.setNavigationItemSelectedListener { item:
MenuItem ->
        if (item.itemId == R.id.exit) {
            finish()
            ticketViewModel.clear()
        }
    }

```

					КРБ.КІ.2.440-03.1.16	Арк.
						88
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        startActivity(Intent(this,
AuthActivity::class.java))
        true // return true -> we handled this navigation
event

        } else {
            val handled =
NavigationUI.onNavDestinationSelected(item, navController)
            if (handled)

binding.drawerLayout.closeDrawer(binding.navView)
            handled
        }
    }

    override fun onSupportNavigateUp(): Boolean {
        return NavigationUI.navigateUp(navController,
appBarConfiguration)
    }

    override fun onBackPressed() {
        if (binding.drawerLayout.isDrawerOpen(GravityCompat.START))
{
            binding.drawerLayout.closeDrawer(GravityCompat.START)
        } else {
            onBackPressedDispatcher.onBackPressed()
        }
    }
}

```

Код файла *HomeFragment.kt*:

```

package com.example.theatretickets2.ui

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment

```

					КРБ.КІ.2.440-03.1.16	Арк.
						89
Змн.	Арк.	№ докум.	Підпис	Дат		

```

import androidx.fragment.app.activityViewModels
import androidx.lifecycle.lifecycleScope
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.theatretickets2.databinding.FragmentHomeBinding
import androidx.lifecycle.ViewModelProvider
import android.widget.Toast
import androidx.navigation.fragment.findNavController
import com.example.theatretickets2.R
import com.example.theatretickets2.viewModels.SharedViewModel
import com.example.theatretickets2.viewModels.TicketViewModel
import com.example.theatretickets2.data.models.Ticket
import com.example.theatretickets2.ui.main.event.TheatreEvent
import
com.example.theatretickets2.ui.main.event.TheatreEventAdapter
import kotlinx.coroutines.launch

class HomeFragment : Fragment(),
TheatreEventAdapter.OnEventClickListener {
    private var _binding: FragmentHomeBinding? = null
    private val binding get() = _binding!!
    private val viewModel: SharedViewModel by activityViewModels()
    private lateinit var ticketViewModel: TicketViewModel

    override fun onResume() {
        super.onResume()
        (activity as? AppCompatActivity)?.supportActionBar?.title =
getString(R.string.home)
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        _binding = FragmentHomeBinding.inflate(inflater, container,
false)
        binding.recyclerView.layoutManager =
LinearLayoutManager(requireContext())

        val events = listOf(

```

					КРБ.КІ.2.440-03.1.16	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        TheatreEvent(getString(R.string.event_1), "Description
1", R.drawable.event1),
        TheatreEvent(getString(R.string.event_2), "Description
2", R.drawable.event2),
        TheatreEvent(getString(R.string.event_3), "Description
3", R.drawable.event3)
    )

    // Create ViewModel
    ticketViewModel =
ViewModelProvider(this).get(TicketViewModel::class.java)

    // Create and set the adapter
    val adapter = TheatreEventAdapter(events, ticketViewModel,
this) // Add this as the last argument
    binding.recyclerView.adapter = adapter
    return binding.root
}

override fun onCreateView(view: View, savedInstanceState:
Bundle?) {
    // Observe the selected city in the ViewModel
    viewModel.selectedCity.observe(viewLifecycleOwner) { city -
>
        binding.cityName.text = city
    }
}

override fun onDestroyView() {
    super.onDestroyView()
    _binding = null
}

override fun onOrderClicked(event: TheatreEvent) {
    val bundle = Bundle()
    bundle.putParcelable("event", event)
}

```

					КРБ.КІ.2.440-03.1.16	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дат		

```

findNavController().navigate(R.id.action_homeFragment_to_pickTicket,
bundle)
    }

    override fun onDetailsClicked(event: TheatreEvent) {
        Toast.makeText(requireContext(), "Подробная информация
загружается", Toast.LENGTH_SHORT).show()
        // Implement your logic here
    }
}

```

Код файла *LoginFragment.kt*:

```

package com.example.theatretickets2.ui.auth

import android.content.Intent
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.fragment.app.Fragment
import androidx.fragment.app.viewModels
import androidx.lifecycle.ViewModelProvider
import androidx.lifecycle.LifecycleScope
import androidx.navigation.fragment.findNavController
import com.example.theatretickets2.R
import com.example.theatretickets2.databinding.FragmentLoginBinding
import com.example.theatretickets2.ui.main.MainActivity
import com.example.theatretickets2.viewModels.AuthViewModel
import com.example.theatretickets2.viewModels.TicketViewModel
import kotlinx.coroutines.launch

class LoginFragment : Fragment() {
    private var _binding: FragmentLoginBinding? = null
    private val binding get() = _binding!!

    private val viewModel: AuthViewModel by viewModels()

```

					КРБ.КІ.2.440-03.1.16	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        override fun onCreateView(inflater: LayoutInflater, container:
ViewGroup?, savedInstanceState: Bundle?): View {
            _binding = FragmentLoginBinding.inflate(inflater,
container, false)

            return binding.root
        }

        override fun onViewCreated(view: View, savedInstanceState:
Bundle?) {
            viewModel.user.observe(viewLifecycleOwner) {
                it?.let {
                    if (it.password ==
binding.etPassword.text.toString()) {
                        viewModel.setLogin(it.login)
                        startActivity(Intent(context,
MainActivity::class.java))
                        activity?.finish()
                    }
                    else
                        Toast.makeText(context, "Пароль не правильный",
Toast.LENGTH_SHORT).show()
                }
            }

            binding.btnLogin.setOnClickListener {
                if (binding.etLogin.text.trim().isEmpty() &&
binding.etPassword.text.toString().length > 5)
                    lifecycleScope.launch {
                        viewModel.getUser(binding.etLogin.text.toString())
                    }
                else
                    Toast.makeText(context, "Введите коректный логин и
пароль не меньше 6 символов", Toast.LENGTH_SHORT).show()
            }

            binding.tvRegistration.setOnClickListener {

```

					КРБ.КІ.2.440-03.1.16	Арк.
						93
Змн.	Арк.	№ докум.	Підпис	Дат		

```
findNavController().navigate(R.id.login_to_registration)
    }
}

override fun onDestroyView() {
    super.onDestroyView()
    _binding = null
}
}
```

					<i>КРБ.КІ.2.440-03.1.16</i>	Арк.
						94
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>		

Додаток А Презентація



Одеський Національний Технологічний Університет

Розробка мобільного додатку для заказу театральних квитків

Виконав: студент групи 542-А Казиміров Денис Сергійович

Керівники: Сергій АРТЕМЕНКО
Наталя СЛУШНА

Рис. А.1 – Слайд 1



Постановка задачі

Мета моєї дипломної роботи - створити мобільний додаток для заказу театральних квитків.

Предметом є методи та засоби розробки мобільних додатків тематики заказу театральних квитків.

Об'єктом є розробка мобільних додатків для заказу театральних квитків

- ▶ Створення повнофункціонального застосунку
- ▶ Створення файлів логіки
- ▶ Створення макетів сторінок
- ▶ Відлагодження коду та його тестування

Рис. А.1 – Слайд 2

					КРБ.КІ.0.440-03.1.16	Арк.
						95
Змн.	Арк.	№ докум.	Підпис	Дат		



Актуальність

- ▶ Цифрова трансформація: Сучасний світ вимагає швидкої цифрової трансформації у всіх сферах нашого життя, включаючи і сферу культури. Розробка мобільного додатку для замовлення театральних квитків дозволить сучасному споживачеві зручно і оперативно планувати свій культурний дозвілля.
- ▶ Популяризація театального мистецтва: Театр як форма мистецтва потребує нових форм привернення аудиторії. Мобільний додаток для замовлення квитків може стати цікавим і привабливим інструментом для молодого покоління.
- ▶ Вищий рівень сервісу: Сучасний глядач очікує не тільки високої якості вистав, але й високого рівня обслуговування. Мобільний додаток дозволить оптимізувати процес покупки квитків, забезпечивши швидке та зручне обслуговування.
- ▶ Зручність та ефективність: Мобільний додаток дозволить кожному користувачу в будь-який час і в будь-якому місці отримати необхідну інформацію про вистави, доступність квитків, а також зробити замовлення.
- ▶ COVID-19 та нові норми: В умовах пандемії та соціальної дистанції, додаток для замовлення квитків онлайн стає ще більш актуальним. Це допомагає забезпечити безпеку глядачів, зменшуючи необхідність фізичного контакту при придбанні квитків.

Рис. А.1 – Слайд 3



Аналіз предметної області

Театральні події є істотною частиною культури та дозвілля людей вже століттями, перетворившись з розважальних зустрічей в духовні заходи.

Проблема замовлення квитків вирішувалась традиційними методами: через телефонні дзвінки, особисті візити до кас або використання веб-сайтів театрів. Проте, ці методи, хоча й досі актуальні, є досить незручними та часомісткими.

Проведши аналіз таких аналогів додатків для заказу театральних квитків, як Ticketmaster, StubHub, TodayTix та Bilet.ua, можна виділити такі моменти:

Основні переваги:

- ▶ Широкий вибір подій: Більшість програм пропонують великий вибір подій.
- ▶ Простий інтерфейс: Багато програм мають зручний та простий інтерфейс для замовлення квитків.
- ▶ Спеціальні пропозиції: Деякі з таких, як TodayTix та Bilet.ua, пропонують знижки та спеціальні пропозиції.

Основні недоліки:

- ▶ Обмежена географічна доступність: Додатки мають обмеження за географією доступу до подій.
- ▶ Висока комісія: Декілька програм стягують високу комісію за покупку квитків.
- ▶ Ризик підроблених квитків: Проаналізовані додатки мають ризик придбання підроблених квитків.
- ▶ Відсутність відгуків та рейтингів: Наприклад, Bilet.ua, не надає відгуків та рейтингів про події.

Рис. А.1 – Слайд 4

					КРБ.КІ.0.440-03.1.16	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дат		



Аналіз існуючих аналогів

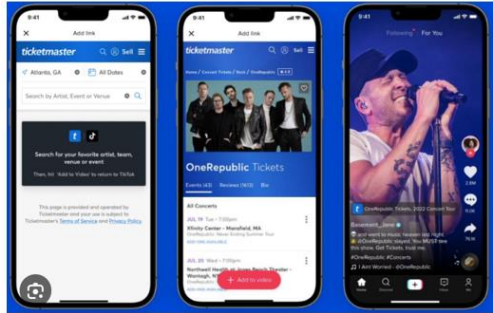
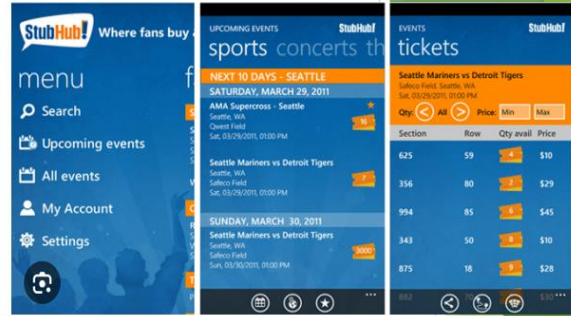
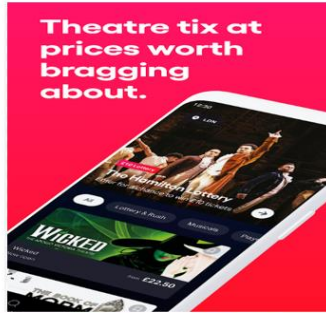


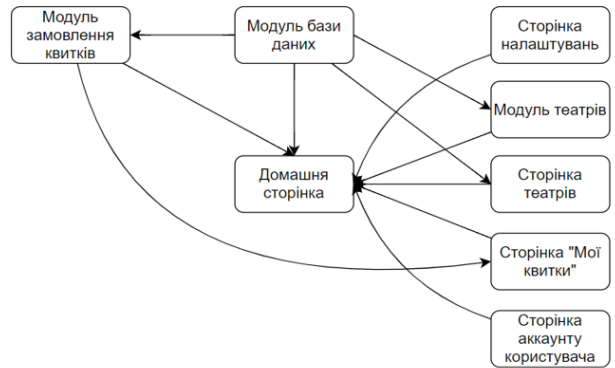
Рис. А.1 – Слайд 5



Проектування додатку

Мій застосунок складається з кількох основних модулів, зокрема основного модуля, модуля бази даних, модуля роботи зі сторінками застосунку та модуля відмалювання.

- ▶ Основний модуль створює список міст, ініціалізує дані та виконує функції програми.
- ▶ Модуль бази даних працює з локальною базою даних і зберігає різні дані.
- ▶ Модуль роботи зі сторінками включає в себе домашню сторінку, сторінку квитків, налаштування, театри та мій акаунт.
- ▶ Модуль відмалювання візуально відображає сторінки та об'єкти, використовуючи .html файли. Модулі взаємодіють з основним файлом системи для простоти адміністрування та налаштування подій. Також є модуль, що працює з базами даних театральних подій, сторінкою квитків та обраним містом.



Модель роботи взаємодії модулів проекту

Рис. А.1 – Слайд 6

Змн.	Арк.	№ докум.	Підпис	Дат



Розробка компонентів системи

Для реалізації необхідного мені функціоналу, я створив модулі та класи, які допоможуть мені працювати з базою даних, а саме:

У моделях даних у додатку є опис сутностей або об'єктів, які використовуються у додатку:

```
package com.example.theatretickets2.data.models
import ...

@Entity(tableName = "users")
data class User {
    @PrimaryKey
    @ColumnInfo(name = "login")
    val login: String,

    @ColumnInfo(name = "password")
    val password: String

    // Add other fields as necessary
}
```

User.kt

```
package com.example.theatretickets2.data.models
import ...

@Entity(tableName = "tickets")
data class Ticket {
    @PrimaryKey(autoGenerate = true)
    val id: Int = 0,

    @ColumnInfo(name = "event_name")
    val eventName: String,

    @ColumnInfo(name = "owner")
    val owner: String,

    @ColumnInfo(name = "number")
    val number: Int
}
```

Ticket.kt

```
package com.example.theatretickets2.db
import ...

@Dao
interface TicketDao {
    @Query("SELECT * FROM tickets ORDER BY event_name ASC")
    fun getAll(): Flow<List<Ticket>>

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insert(ticket: Ticket)

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insert(ticket: List<Ticket>)

    @Delete
    suspend fun delete(ticket: Ticket)

    @Query("DELETE FROM tickets")
    suspend fun deleteAllTickets()
}
```

TicketDao

```
package com.example.theatretickets2.db
import ...

@Dao
interface AuthDao {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insert(user: User)

    @Query("SELECT * FROM users WHERE login = :login")
    suspend fun getUser(login: String): User?
}
```

AuthDao

Рис. А.1 – Слайд 11



Розробка компонентів системи

На слайді приведено основні активності додатку, які надають користувачеві можливість робити активності у системи, тобто реєстрація, авторизація, бронювання білету, перегляд інформації про подію, тощо.

Дані відображення реалізовані за допомогою написання коду у файлах макетів формату .xml

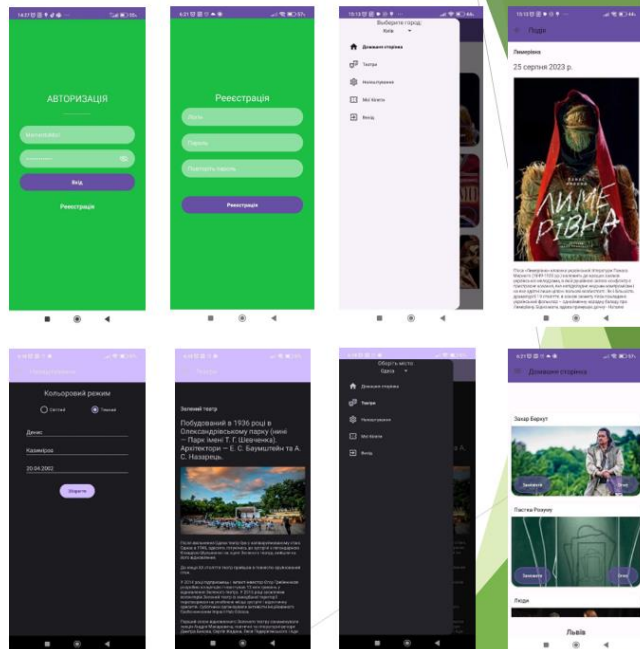


Рис. А.1 – Слайд 12

					КРБ.КІ.0.440-03.1.16	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		100



Висновки

На даній таблиці приведено результат роботи праці над економічною частиною, а саме експертну оцінку і розрахунок величини інтегрального показника НТЕ.

Я розрахував коефіцієнт науково-технічної ефективності свого продукту, який у результаті дорівнює 74,83%.

Тобто, рівень НТЕ є достатнім.

№	Групи показників	Рейтинг експертів			Середня за експертними оцінками	НТЕ
		1	2	3		
1	Науково-технічний рівень	6	7	8	7	2,45(7 x 0,35)
2	Перспективність	8	7	9	8	2,8(8 x 0,35)
3	Потенційний масштаб практичного використання	8	9	7	8	1,6(8 x 0,20)
4	Ступінь вірогідності досягнення позитивних результатів	6	7	6	6,33	0,633(6,33x0,10)
В С Ъ О Г О						7,483

Експертна оцінка і розрахунок величини інтегрального показника науково-технічної ефективності

$$\frac{7,483}{10} * 100\% = 74,83 \%$$

Коефіцієнт науково-технічної ефективності

Рис. А.1 – Слайд 13



Висновки

- ▶ Виконано аналіз предметної області мобільних застосунків для замовлення театральних квитків.
- ▶ Проведено дослідження сучасних аналогів застосунків з виявленням їхніх переваг та недоліків.
- ▶ Визначено ключові вимоги до нового застосунку та розроблено архітектуру застосунку.
- ▶ Реалізовано програмну та візуальну складові застосунку.
- ▶ Встановлено, що проект є вигідним і має перспективи прибутку.
- ▶ Враховано вимоги охорони праці при створенні додатку.

Рис. А.1 – Слайд 14

					КРБ.КІ.0.440-03.1.16	Арк.
						101
Змн.	Арк.	№ докум.	Підпис	Дат		



Дякую за увагу!

Автор: Казиміров Денис

Рис. А.1 – Слайд 15

					КРБ.КІ.0.440-03.1.16	Арк.
						102
Змн.	Арк.	№ докум.	Підпис	Дат		