

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ

ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА МУЗИЧНОГО ДОДАТКУ “SOUNDX” МОВОЮ SWIFT»**

Виконала: студентка 5 курсу, групи ППЗ– 51
спеціальності

121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

_____ Мала Ю.В.
(прізвище та ініціали)

Керівник Жебка В.В.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Нормоконтроль _____
(прізвище та ініціали)

Київ – 2022

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ НАВЧАЛЬНО-
НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення _____

Ступінь вищої освіти - «Бакалавр» _____

Спеціальність - 121 «Інженерія програмного забезпечення» _____

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

О.В. Негоденко

“ _____ ” _____ 2022 року

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Малій Юлії Володимирівні

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка музичного додатку “SoundX” мовою Swift»

Керівник роботи Жебка В.В., д.т.н., доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “18” лютого 2022 року №65.

2. Строк подання студентом роботи 03.06.2022.

3. Вихідні дані до роботи:

3.1. Аналіз аналогів програмного забезпечення;

3.2. Порівняння додатку з аналогами;

3.3. Науково-технічна література;

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1. Розробка структури додатку за допомогою мови програмування Swift та фреймворку SwiftUI

- 4.3. Програмна реалізація додатку
- 4.4. Приклади використання та тестування застосунку
- 4.5. Висновки

5. Перелік графічного матеріалу.

- 5.1. Аналоги
- 5.2. Програмні засоби реалізації
- 5.3. Інструменти використані для реалізації

6. Дата видачі завдання: 11.04.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04 – 14.04	
2	Аналіз предметної області	15.04 - 17.04	
3	Моделювання структури додатку	18.04 – 21.04	
4	Розробка дизайну	22.04 – 25.04	
5	Розробка додатку	26.04 – 05.05	
6	Тестування додатку	05.05 – 07.05	
7	Вступ, висновки, реферат	07.05 – 10.05	
8	Створення презентації	11.05 – 15.05	
9	Попередній захист	16.05 – 01.06	
10	Здача роботи	03.06	

Студент _____ Мала Ю.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Жебка В.В.
(підпис) (прізвище та ініціали)

ЗМІСТ

<i>РЕФЕРАТ</i>	7
<i>ВСТУП</i>	8
<i>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</i>	10
1.1 Аналіз предметної області і напрямків дослідження	10
1.2 Аналіз існуючих додатків.....	10
1.2.1 Аналіз музичного додатку Spotify	11
1.2.2 Аналіз музичного додатку SoundCloud.....	15
1.2.3 Аналіз музичного додатку Apple Music	17
1.2.4 Аналіз музичного додатку Deezer	21
1.3 Порівняння SoundX з аналогами	24
1.4 Моделювання UML-діаграми.....	24
<i>2 ЗАСОБИ РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ</i>	26
2.1 Мова програмування Swift	26
2.2 Фреймворок SwiftUI.....	30
2.3 Середовище розробки Xcode.....	34
<i>3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ</i>	35
3.1 Розробка інтерфейсу додатку	35
3.2 Тестування мобільного застосунку	51
3.2 Тестування “SoundX”	53
<i>ВИСНОВКИ</i>	72
<i>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</i>	73

РЕФЕРАТ

Текстова частина бакалаврської роботи 82 сторінки, включає 36 рисунків, 7 таблиць та 30 посилань.

Актуальність даної роботи полягає у різкому збільшенні медіа інформації, доступної користувачу для прослуховування онлайн, в особливості музичних записів та зручності, у вигляді мобільного додатку.

Об'єктом дослідження є процес прослуховування музики.

Предметом дослідження є методи та засоби розробки музичного додатку.

Мета – оптимізація процесу відтворення музики за допомогою створеного музичного додатку з використанням фреймворку SwiftUI.

Робота виконувалася за допомогою методів конкретизації, моделювання та системного аналізу.

Суть розробки полягає в створенні конкурентно спроможного музичного додатку для платформи IOS за допомогою мови програмування Swift та фреймворку SwiftUI, який буде задовольняти всі потреби користувача.

Наукова новизна полягає в оптимізації створення мобільного додатку для платформи IOS за допомогою фреймворку SwiftUI.

Галузь застосування: уроки музики, уроки танцю, культурні заходи тощо.

Ключові слова: музичний додаток, Swift, SwiftUI, розробка для мобільної платформи IOS.

ВСТУП

Наше суспільство прагне використовувати пристрої, які роблять наше життя простішим і зручнішим. Враховуючи цей факт, мобільна індустрія розробила один пристрій, який зробив кілька інших застарілими. Через це значно впали продажі стаціонарних телефонів, настільних комп'ютерів, цифрових фотоапаратів, mp3-плеєрів, GPS-пристроїв тощо. Хоча, не так давно був час, коли потрібно було носити всі ці речі з собою.

Раніше настільний комп'ютер був незамінною технічною частиною нашого життя, але в останні місяці його місце зайняв смартфон. Ми все ще любимо використовувати настільні комп'ютери для виконання практичних завдань, але віддаємо перевагу смартфону, через його портативність.

Наш смартфон – це наш особистий помічник, який допомагає нам оволодіти повсякденним життям.

На рис.1 представлено інфографіку, на якій зображено, як телефони впливають на життя сімей.



Рисунок 1 – Вплив телефону на життя сімей

Понад 45% опитаних сімей стверджують, що смартфон допомагає їм проводити більше часу разом

Мета – оптимізація процесу відтворення музики за допомогою створеного музичного додатку з використанням фреймворку SwiftUI.

В процесі дослідження вирішувалися наступні завдання:

- аналіз предметної галузі;
- аналіз засобів розробки програмного забезпечення;
- аналіз аналогів;
- визначення об'єкта та предмета дослідження, мети;
- постановка завдань;
- побудова діаграми станів та діаграми діяльності;
- розробка додатку;
- тестування додатку.

Об'єктом дослідження є процес прослуховування музики.

Предметом дослідження є методи та засоби розробки музичного додатку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області і напрямків дослідження

Apple завжди тримала відтворення музики одним із своїх головних пріоритетів, і сімейство iPod є одним із найбільш захоплюючих прикладів її інновацій, пов'язаних із музикою. Хоча iPod стали більш-менш застарілими, їхня філософія поширилася на iPhone та такі сервіси, як Apple Music. Але якщо ви хочете не лише відтворювати локальні музичні файли, а й мати у легкому доступі тисячі різноманітних пісень та можливість відстежувати тенденції у музичному світі, то варто звернути свою увагу на музичні додатки.

Так як музика – це невід'ємний елемент нашого життя, то на сьогоднішній день важко знайти смартфон на якому не було б встановлено музичного додатку. Вони несподівано увійшли в наше життя та стали незамінною частиною сучасного світу.

Останнім часом особливої популярності набули подкасти. Подкаст – це медіа-файл або низка таких файлів, які розповсюджуються інтернетом для відтворення на портативних медіапрогравачах або персональних комп'ютерах. За змістом вони можуть нагадувати радіошоу, звукову виставу, включати інтерв'ю, лекції тощо, що належить до усного жанру.

Слідуючи тенденціям, в музичних додатках створили окрему категорію, в якій можна знайти подкасти на будь-який смак. Тим самим, застосунки підвищили свою практичність. Тепер можна не лише слухати музику, а й дізнаватись щось нове.

1.2 Аналіз існуючих додатків

Для аналізу музичного програмного забезпечення було обрано наступні музичні додатки:

- Spotify;

- SoundCloud;
- Apple Music;
- Deezer.

1.2.1 Аналіз музичного додатку Spotify

Американський сервіс потокової музики з'явився в 2006 році. Доступний у вигляді веб-сайту, додатків для всіх операційних систем, смартфонів, розумних пристроїв та автомобільних медіасистем. Розробники обіцяють моментальний доступ до 30 мільйонів треків.

На частку, що належить понад 36% світового потокового аудіо, він доступний у країнах Північної та Південної Америки, майже у всіх країнах Європи, деяких країнах Азії та Африки, Австралії та Океанії.

За підсумками вересня 2021 року в сервісі активно понад 248 мільйонів облікових записів, 113 мільйонів з яких користуються платними послугами.

Spotify містить понад 50 мільйонів пісень, кожен день завантажується кілька десятків тисяч нових аудіо-файлів. Пошук здійснюється по пісням, артистам, альбомам, плей-листам і лейблам.

Після успішної реєстрації користувачеві покажуть плейлист, щоб вибрати треки, які подобаються. Відібрані композиції зберігаються в розділі Song. Додаток проаналізує смаки і раз в сім днів буде пропонувати оновлений список рекомендованих пісень.

Також для розваги в додатку є рубрики, наведенні в табл. 1.1.

Таблиця 1.1

Release Radar	Нові треки від виконавців, які вже є в колекції користувача. Оновлення – по п'ятницях.
Your Daily Mix	До 6 щоденних збірок, кожна з яких – в певному жанрі. Складаються навпіл з уже обраних треків і схожих з ними.
Browse	Каталог музики за жанрами, країнам, мовами, настроєм, а також чарти.
Running	Плейлисти з музикою для занять бігом з підбором композицій в темпі бігу.

За даними з App Store Spotify має рейтинг 4.8 з 166 тисяч опитуваних користувачів. Займає перше місце в категорії “Музика” та має розмір 136 МБ. Також додаток перекладено більш ніж на 57 мов світу, але, нажаль, серед них немає української, рис. 1.1.

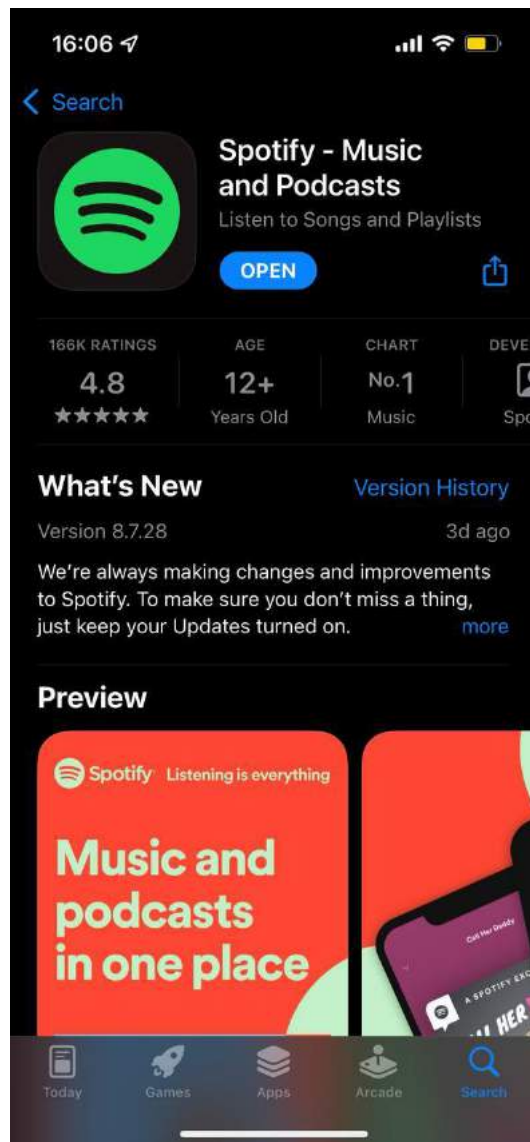


Рисунок 1.1 – Spotify в App Store

Можливості підписки

У безкоштовної версії для стаціонарних комп'ютерів обмежень немає, хіба що якість звуку становить усього 160 kb/s. А ось в мобільному застосунку платний і безкоштовний акаунти мають досить багато відмінностей, які перераховані в табл. 1.2.

Таблиця 1.2

Можливості	Безкоштовно	Преміум (10 доларів на місяць)
Якість звуку	160 kb/s	320 kb/s
Реклама	Ні	Ні
Відтворення	Тільки перемішане. Переключити трек можна тільки сім разів за годину	Перемішати можна відключити. Можна слухати конкретну пісню і перемикає треки скільки завгодно раз
Кнопка Previous	Завжди заблокована	Працює
Прослуховування в offline-режимі	На гаджет зберігаються треки, прослухані останніми. Без підключення до Мережі – не відтворюються	Завантажені композиції відтворюються без підключення до інтернету

Переваги:

1. Продумана фітнес-функція, принцип роботи якої чимось то схожий з метрономом. Перед відтворенням композиції користувачеві пропонують почати бігти чи вручну ввести темп бігу. Музичний супровід буде звучати в зазначеному темпі.
2. Різноманітні і добре продумані плейлисти на будь-який смак, включаючи підбір на основі часу доби.
3. В процесі відтворення деяких треків сервіс пояснює значення того чи іншого тексту. Іноді це історія створення композиції або подібна інформація.
4. Додаток підтримує всі платформи.

Недоліки:

1. Немає версії українською мовою.

1.2.2 Аналіз музичного додатку SoundCloud

Сервіс, створений музикантом Еріком Вальфорсом і звукорежисером Алексом Льюнг для молодих талантів музичної індустрії, спочатку був орієнтований на високоякісний контент. Завдяки цьому завоював визнання не тільки серед музикантів, які можуть завантажити на платформу до 180 хвилин власної музики, а й серед слухачів. Адже вони можуть відкрити нові, цікаві композиції, яких більше ніде немає. При цьому пошук – продуманий, дає можливість фільтрувати треки за жанрами, стилями, виконавцям, датою створення і популярністю.

У даного сервісу є власний сайт, а також мобільні додатки для Android та IOS.

За даними з App Store додаток має рейтинг 4.6 з 7 тисяч опитуваних користувачів. Займає п'яте місце в категорії “Музика” та має розмір 151 МБ. Перекладено на 8 мов світу. рис.1.2.

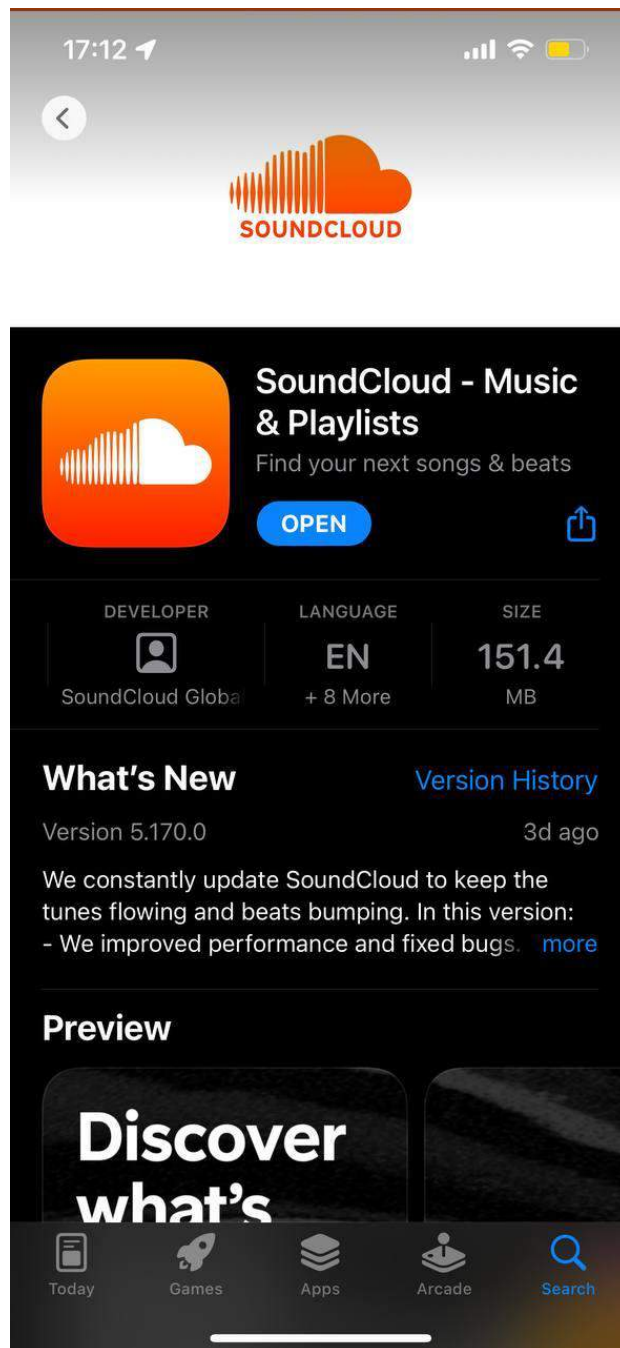


Рисунок 1.2 – SoundCloud в App Store

Реєстрація проста, досить ввести:

- адреса електронної пошти;
- пароль;
- вік.

Після сайт попросить уточнити нікнейм та мову, але, нажаль, в інтерфейсі немає української мови.

Можливості особистого облікового запису

Втім, слухати композиції можна і без реєстрації, безкоштовно. Але власний акаунт дає такі переваги:

- Наповнювати папку «вибране» треками, які сподобалися.
- Створювати особисті плейлисти.
- Підписатися на новинки від вподобаних виконавців.
- Вступати в групи.
- Ділитись треками в соцмережах з друзями.

Переваги:

1. Платформа підійде навіть тим, хто не довіряє невідомим талантам, оскільки значну частину контенту становлять аудіозаписи реміксів і римейків популярних треків.

2. Простий і зрозумілий пошук в мобільному додатку і в десктопній версії.

3. Різноманітність жанрів і стилів допоможе підібрати супровід практично для будь-якого заняття і настрою.

4. Сервіс – безкоштовний для слухача.

5. Музиканти з України також можуть поділитися своїми творіннями зі світом, завантаживши трек. Або підтримати колег, які вже там є.

Недоліки:

1. Відсутність україномовного інтерфейсу компенсується зрозумілим користувачеві оформленням.

2. Без підключення до мережі на мобільному телефоні можна запустити тільки раніше прослухані треки.

1.2.3 Аналіз музичного додатку Apple Music

«Яблуневий» сервіс пропонує прослуховувати музику в потоковому режимі. Для користувачів пристроїв бренду послуга доступна на всіх гаджетах. Розроблено

і створено програму перш за все для техніки Apple, але можна встановити її і на смартфон з Android. Головне, щоб був ідентифікатор Apple ID і включена медіатека iCloud. Незважаючи на те, що існує сервіс всього пару років, фанів у нього достатньо – понад 20 мільйонів було вже в середині 2017 року, рис. 1.3.

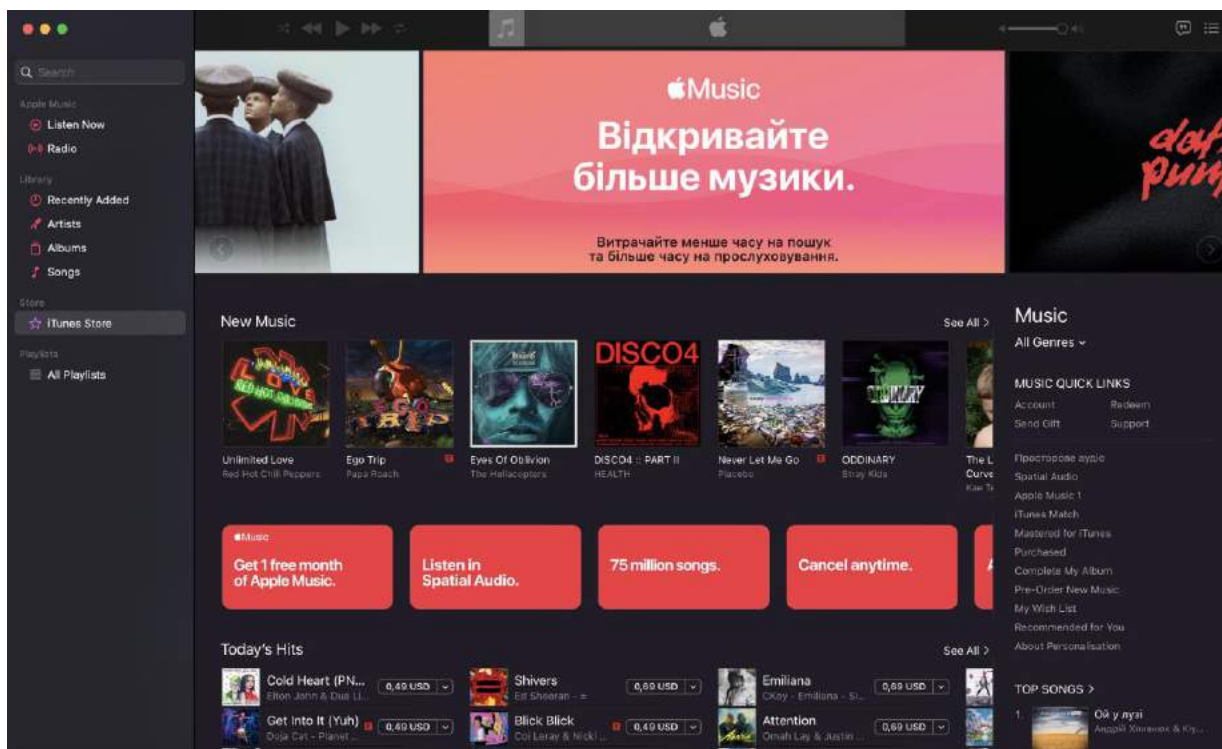


Рисунок 1.3 - Apple Music

В Україні за допомогою Apple Music можна:

1. Завантажувати музику з каталогу програми і додавати свою.
2. Слухати треки, використовуючи функції «повтор» і «перемішати».
3. Формувати плейлисти з композицій, які пропонує сервіс, або з особистої колекції.
4. Дивіться фільми, відеокліпи, програми.
5. Читати книги.

В процесі реєстрації додаток пропонує три місяці безкоштовного доступу до 45 мільйонів треків. Але необхідно ввести дані кредитки, причому на ній повинно бути достатньо коштів для оплати одного місяця користування платною послугою.

Після закінчення безкоштовного пробного доступу з карти буде зніматися оплата сервісу.

Для керування додатком призначені 5 вкладок:

- медіатека;
- для вас;
- огляд;
- радіо;
- пошук.

Для керування всією колекцією композицій використовується розділ «Медіатека». Тут можна знайти і запустити будь-який трек, незалежно від походження – закачаний самостійно або з бібліотеки Apple Music.

Підрубрики розділу:

- пісні;
- виконавці;
- жанри;
- композитори;
- відеокліпи;
- альбоми;
- завантажена музика;
- ТБ і фільми.

Є також категорія «Нещодавно додані» – вона поповнюється «автоматично».

Щоб ознайомитися з альбомами і добірками, які пропонує програма з урахуванням смаків користувача, потрібно скористатися вкладкою «Для вас». Цей розділ кожен день поповнюється свіжими треками. Крім того, тут відображаються композиції, які передплатник нещодавно прослуховував. Для уточнення переваг потрібно перейти в особистий обліковий запис.

Послухати що-небудь свіженьке вийде за допомогою розділу «Огляд». Тут музичні новинки з'являються раз в 7 днів. Кожен з треків, знайдений в цій вкладці, можна додати в медіатеку.

Радіостанція працює 24 години на добу і пропонує слухачам не тільки сучасні треки, але і радіо-шоу, а також інтерв'ю з відомими музикантами. Вибрати відповідну «хвилю» можна за жанром. Крім того, користувач може самостійно створювати радіостанцію. Реклама на радіо в додатку не транслюється.

За допомогою розділу «Пошук» вийде знайти будь-які треки, плейлисти, телешоу, радіостанції, альбоми та інші потрібні файли в особистому каталозі і в базі даних програми.

Можливості підписки

Ті, хто після трьох місяців безкоштовного користування сервісом захочуть продовжити, можуть зробити це за допомогою платної підписки. Для бажаючих є такі варіанти описані у табл. 1.3.

Таблиця 1.3

Для однієї людини	5 доларів
Сімейний доступ для 6 осіб	8 доларів
Річна підписка	49 доларів

Переваги:

1. Необмежений доступ до десятків мільйонів композицій.
2. Можливість не тільки слухати треки, але й дивитися відео.
3. Безліч різножанрових радіостанцій плюс можливість створити свою власну.
4. Працює на різних платформах і пристроях.

Недоліки:

1. Необхідність введення даних кредитки.
2. Досить висока вартість передплати для українських клієнтів.

1.2.4 Аналіз музичного додатку Deezer

Deezer був створений у Франції як сервіс прослуховування музики на різних девайсах. Зараз він працює на комп'ютерах, планшетах смартфонах з операційними системами IOS, Windows, Android. Крім того, скористатися сервісом можуть власники телевізорів з Apple TV, Smart TV і Android TV, а також смарт-годинників на платформах Apple і Android.

Реєстрація проста як з ПК, так і з будь-якого гаджета. Буде потрібно лише ввести:

- адреса електронної пошти;
- пароль;
- логін;
- мова;
- вік.

Після реєстрації потрібно буде вказати жанри і виконавців, які подобаються, а після – перейти на головну сторінку.

Додаток Deezer має наступні рубрики табл. 1.4.

Таблиця 1.4

Переглянути	пошук музики та подкастів за жанрами
Підбірка плейлистів	на основі попереднього вибору користувача редактори пропонують музичні добірки
Популярне	альбоми та збірки, у яких найбільше прослуховувань
Хіт-Паради	щоденний рейтинг популярних треків, хіт-паради музики по країнам, рейтинги трендових альбомів і виконавців
Новинки	свіжі альбоми і треки від виконавців і груп з усього світу

Розділ Моя музика дає можливість додати плейлисти, альбоми та виконавців.

Для отримання новин із соціальних мереж досить увійти в додаток зі свого облікового запису в Facebook або Twitter.

Можливості підписки

Додаток пропонує слухати музику безкоштовно або оформити одну з платних підписок – Premium + або Deezer Family, табл. 1.5.

Таблиця 1.5

Можливості	Безкоштовно	Premium + (6,99 € на місяць)	Deezer Family (10,49 € на місяць)
Музика без обмежень на комп'ютері і планшеті	Так	Так	Так
Музика без обмежень на мобільному телефоні	Тільки Flow і мікси	Так	Так
Без реклами	-	Так	Так
Музика, доступна в режимі офлайн	-	Так	Так
Якість звуку	Стандартна	Висока	Висока
6 профілів Premium +	-	-	Так
Профілі для дітей	-	-	Так

За даними з App Store додаток має рейтинг 4.6 з 3 тисяч опитуваних користувачів. Займає 17 місце в категорії “Музика” та має розмір 95 МБ. Перекладено більш ніж на 26 мов світу і серед них є українська. рис. 1.4.

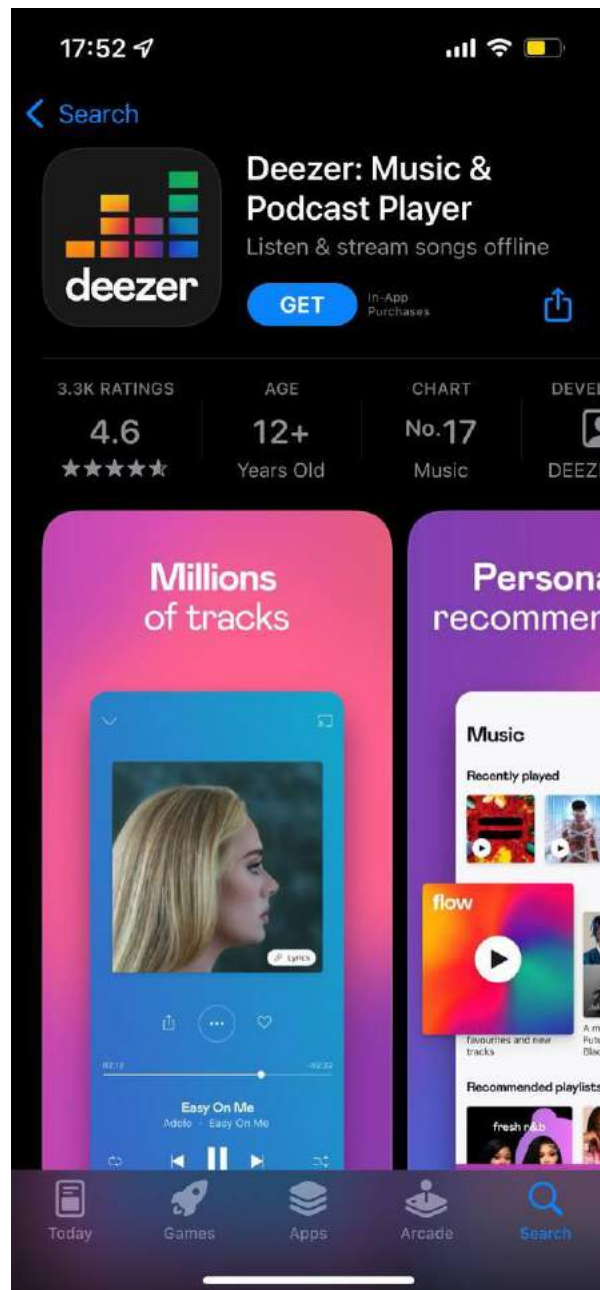


Рисунок 1.4 – Deezer в App Store

Переваги:

1. Додаток легко освоїти завдяки зрозумілому інтерфейсу.
2. Музика високої якості передається на швидкості 320 кбіт/с.
3. Каталог музики – великий, включає безліч виконавців в різних стилях і жанрах.
4. Є доступна версія без плати.

5. Є автоматизований плейлист Flow, який дозволяє слухати музику, не відволікаючись на зміну треків вручну. Це підійде тим, хто слухає музику за кермом, під час роботи або в транспорті.

Недоліки:

1. Висока вартість передплати (більше 200 гривень на місяць) середньостатистичного українського слухача.
2. Музики українських виконавців небагато, але їх кількість поступово збільшується.
3. У безкоштовному додатку якість музики становить всього 128 кбіт/с.

1.3 Порівняння SoundX з аналогами

Таблиця 1.6

	Spotify	SoundCloud	Apple Music	Deezer	SoundX
Доступна безкоштовна версія	+	+	-	+	+
Присутня реклама	-	+	-	-	-
Велика кількість українських виконавців	+	-	+	-	+
Необхідність введення кредитних даних	-	-	+	-	-
Можна завантажувати музику	-	-	+	+	-

1.4 Моделювання UML-діаграми

UML — це аббревіатура, що розшифровується як уніфікована мова моделювання. Простіше кажучи, UML — це сучасний підхід до моделювання та документування програмного забезпечення. Насправді, це один з найпопулярніших методів моделювання бізнес-процесів.

Він заснований на схематичних представленнях програмних компонентів. Як говорить старе прислів'я: «картинка вартує тисячі слів». Використовуючи візуальні уявлення, ми можемо краще зрозуміти можливі недоліки або помилки в програмному забезпеченні або бізнес-процесах.

UML-діаграма для застосунку SoundX представлена на рис. 1.5.

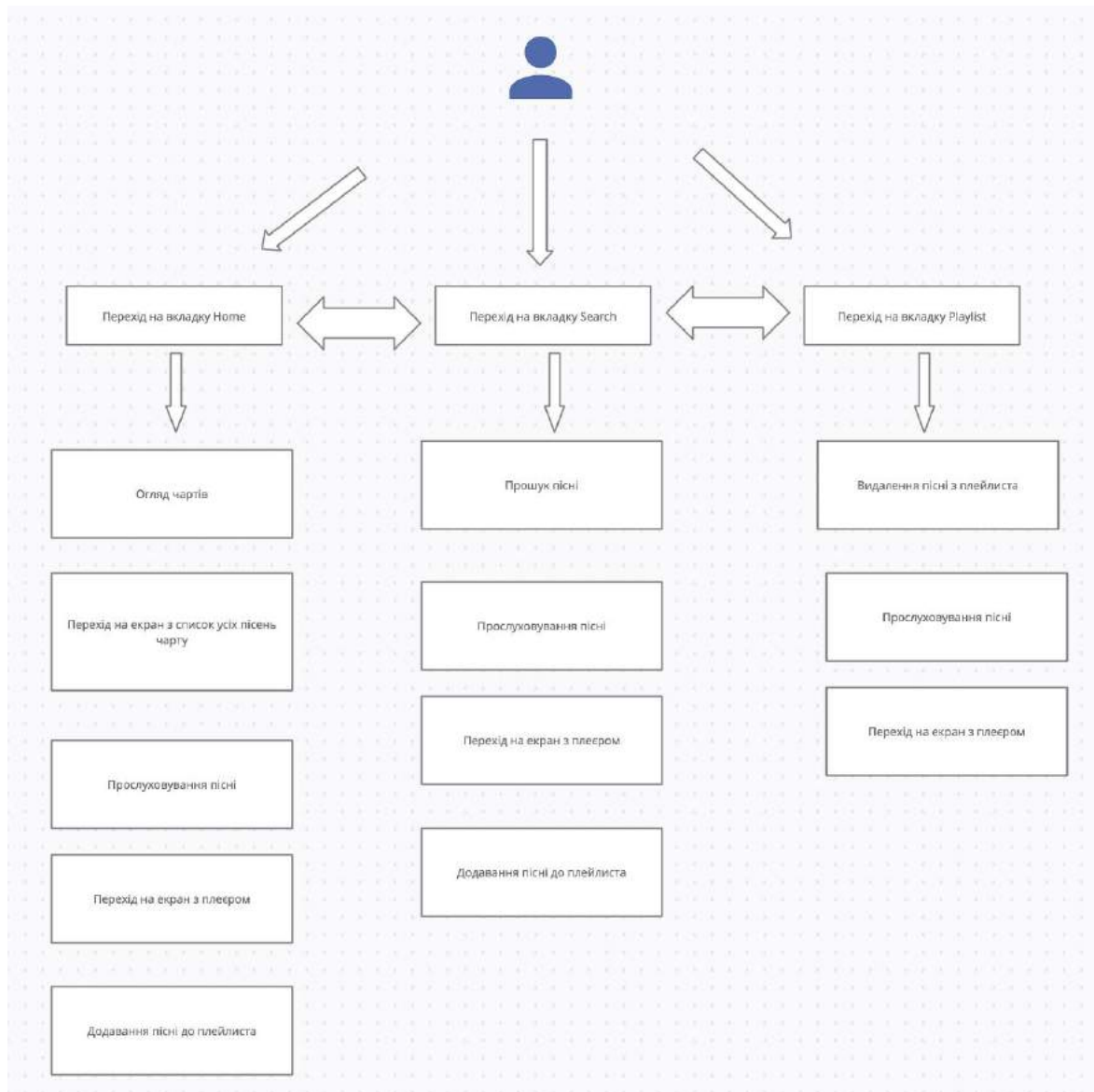


Рисунок 1.5 – UML-діаграма для застосунку SoundX

2 ЗАСОБИ РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ

2.1 Мова програмування Swift

Swift — це потужна та інтуїтивно зрозуміла мова програмування для iOS, iPadOS, macOS, tvOS та watchOS. Написання коду Swift є інтерактивним і веселим, синтаксис лаконічний, але виразний, а Swift включає сучасні функції, які люблять розробники. Swift-код безпечний за своєю конструкцією, але також створює програмне забезпечення, яке працює блискавично.

Сучасність

Swift є результатом останніх досліджень мов програмування в поєднанні з десятиліттям досвіду створення платформ Apple. Параметри виражені в чистому синтаксисі, робить API в Swift легким для читання та обслуговування. Вам навіть не потрібно вводити крапку з комою. Виведені типи роблять код чистішим і менш схильним до помилок, а модулі усувають заголовки та надають простір імен. Управління пам'яттю здійснюється автоматично за допомогою жорсткого, детермінованого підрахунку посилань, що зводить використання пам'яті до мінімуму без накладних витрат на збирання сміття.

Swift має багато функцій, які роблять ваш код більш виразним, наприклад:

- Дженерики, які є потужними та простими у використанні.
- Розширення протоколу, які спрощують написання загального коду.
- Першокласні функції та легкий синтаксис закриття.
- Швидке та стисле повторення діапазону чи колекції.
- Структури, які підтримують методи, розширення та протоколи.
- Перерахування можуть мати корисні навантаження та підтримувати відповідність шаблону.
- Шаблони функціонального програмування, наприклад, мапа та фільтр.
- Внутрішня обробка помилок за допомогою try/catch/throw.

Використання Swift на практиці представлено на рис. 2.1 – 2.4.

```

let players = getPlayers()

// Sort players, with best high scores first
let ranked = players.sorted(by: { player1, player2 in
    player1.highScore > player2.highScore
})

// Create an array with only the players' names
let rankedNames = ranked.map { $0.name }
// ["Erin", "Rosana", "Tomas"]

```

Рисунок 2.1 – Виконання потужного користувацького перетворення за допомогою спрощених замикань

```

struct Player {
    var name: String
    var highScore: Int = 0
    var history: [Int] = []

    init(_ name: String) {
        self.name = name
    }
}

var player = Player("Tomas")

```

Рисунок 2.2 – Оголошення нового типу, надання значення за замовчуванням для властивостей екземпляра та спеціальний ініціалізатор

```

extension Player {
    mutating func updateScore(_ newScore: Int) {
        history.append(newScore)
        if highScore < newScore {
            print("\(newScore)! A new high score for \(name)! 🎉")
            highScore = newScore
        }
    }
}

player.updateScore(50)

```

Рисунок 2.3 – Додання функціональності до існуючого типу за допомогою розширення

```

extension Player: Codable, Equatable {}

import Foundation
let encoder = JSONEncoder()
try encoder.encode(player)

print(player)

```

Рисунок 2.4 – Розширення власного типу, щоб скористатися перевагами автоматичного кодування та декодування JSON.

Чудова перша мова

Swift може відкрити двері у світ кодування. Насправді, вона була розроблена, щоб стати першою мовою програмування будь-кого, незалежно від того, навчаєтесь ви ще в школі чи досліджуєте нові кар'єрні шляхи. Для викладачів Apple створила безкоштовну навчальну програму, щоб навчати Swift як у класі, так і поза

ним. Початківці можуть завантажити Swift Playgrounds — додаток для iPad, який робить початок роботи з кодом Swift інтерактивним і веселим.

Починаючи розробники можуть отримати доступ до безкоштовних курсів, щоб навчитися створювати свої перші програми в Xcode. А Apple Stores по всьому світу сьогодні проводять сеанси Apple Coding & Apps, де ви можете отримати практичний досвід роботи з кодом Swift.

Відкрите джерело

Swift відкрито розробляється на Swift.org, а вихідний код, засіб відстеження помилок, форуми та регулярні збірки доступні для всіх. Ця широка спільнота розробників, як всередині Apple, так і сотні зовнішніх учасників, працюють разом, щоб зробити Swift ще більш дивовижним. Існує ще ширший спектр блогів, подкастів, конференцій та зустрічей, де розробники спільноти діляться своїм досвідом того, як реалізувати великий потенціал Swift.

Swift для сервера

Swift ідеально підходить для використання в серверних програмах, які потребують безпеки під час виконання, продуктивності компіляції та невеликого обсягу пам'яті. Щоб керувати напрямком розробки та розгортання серверних додатків Swift, спільнота сформувала робочу групу Swift Server. Першим продуктом цих зусиль був SwiftNIO, кросплатформна асинхронна мережева програма, керована подіями, для серверів і клієнтів високопродуктивних протоколів. Він служить основою для створення додаткових серверно-орієнтованих інструментів і технологій, включаючи журналювання, метрики та драйвери баз даних, які активно розробляються.

Плейграунд та Read-Eval-Print-Loop (REPL)

Подібно до Swift Playgrounds для iPad, плейграунди в Xcode роблять написання коду Swift неймовірно простим. Введіть рядок коду, і результат з'явиться відразу. Потім ви можете швидко переглянути результат збоку коду або закріпити цей результат безпосередньо нижче. Перегляд результатів може відображати графіки, списки результатів або графіки значення за певний час. Ви можете відкрити Timeline Assistant, щоб спостерігати, як складний вигляд

розвивається та анімується, що чудово підходить для експериментів з новим кодом інтерфейсу користувача або для відтворення анімованої сцени SpriteKit під час її кодування. Коли ви вдосконалите свій код за допомогою плейграунду, просто перемістіть цей код у свій проект. Swift також інтерактивний, коли ви використовуєте його в терміналі або в консолі налагодження LLDB Xcode.

Менеджер пакетів

Swift Package Manager — це єдиний міжплатформний інструмент для створення, запуску, тестування та упаковки ваших бібліотек і виконуваних файлів Swift. Пакети Swift — найкращий спосіб розповсюджувати бібліотеки та вихідний код серед спільноти Swift. Конфігурація пакетів написана на самому Swift, що дозволяє легко налаштовувати цілі, оголошувати продукти та керувати залежностями пакетів. Нова в Swift 5, команда `swift run` тепер включає можливість імпортувати бібліотеки в REPL без необхідності створювати виконуваний файл.

Сумісність з Objective-C

Swift-код співіснує разом із наявними файлами Objective-C в одному проекті з повним доступом до вашого API Objective-C, що робить його легким у застосуванні.

2.2 Фреймворок SwiftUI

SwiftUI допомагає створювати чудові програми на всіх платформах Apple за допомогою потужності Swift — і якомога менше коду. Завдяки SwiftUI ви можете надати всім користувачам ще кращий досвід на будь-якому пристрої Apple, використовуючи лише один набір інструментів та API. рис. 2.5.

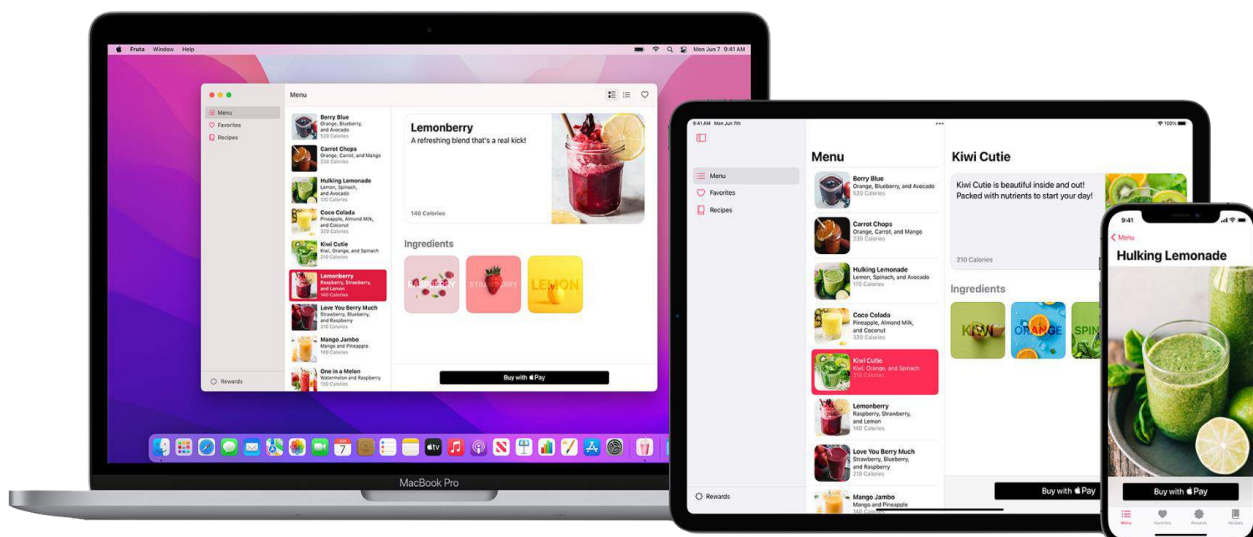


Рисунок 2.5 – Приклад дизайну, який можна створити за допомогою SwiftUI

Розширені можливості та інструменти для додатків

Покращений перегляд списків, кращий пошук та підтримка контрольних зон фокусування. Більше контролю над примітивами малювання нижнього рівня за допомогою нового API Canvas, сучасного еквіваленту drawRect з прискореним графічним процесором.

Покращення доступності

Прискорення взаємодії, показуючи найбільш релевантні елементи на екрані в простому списку за допомогою нового API Rotor. Поточний стан фокуса доступності, наприклад курсор VoiceOver, тепер можна прочитати і навіть змінити програмно. А завдяки новому API представлення доступності ваші спеціальні елементи керування легко успадковують повну підтримку доступності від існуючих стандартних елементів керування SwiftUI.

Покращення SwiftUI на macOS

Покращення продуктивності та доступності API, включаючи підтримку багатостовпцевих таблиць, роблять ваші програми macOS ще кращими.

Підтримка Always-On Retina Display

На Apple Watch Series 5 і новіших версіях дисплей Always-On Retina Display дозволяє програмам watchOS залишатися видимими, навіть коли циферблат годинника затемнений, завдяки чому ключова інформація доступна відразу.

Віджети для iPadOS

Тепер віджети можна розміщувати в будь-якому місці головного екрана та збільшити їх до нового надзвичайно великого розміру.

Декларативний синтаксис

SwiftUI використовує декларативний синтаксис, тому можна просто вказати, що має робити ваш інтерфейс. Наприклад, ви можете написати, що вам потрібен список елементів, що складається з текстових полів, а потім описати вирівнювання, шрифт і колір для кожного поля. Це робить код простішим та легшим для читання, ніж будь-коли раніше, заощаджуючи час і обслуговування. Приклад приведено на рис. 2.6.

```
import SwiftUI

struct Content : View {

    @State var model = Themes.listModel

    var body: some View {
        List(model.items, action: model.selectItem) { item in
            Image(item.image)
            VStack(alignment: .leading) {
                Text(item.title)
                Text(item.subtitle)
                    .color(.gray)
            }
        }
    }
}
```

Рисунок 2.6 – Код-сніпет з використанням SwiftUI

Цей декларативний стиль застосовується навіть до таких складних понять, як анімація. Можна легко додати анімацію майже до будь-якого елемента керування, обрати набір готових до використання ефектів лише за допомогою кількох рядків коду. З такою простою анімацією можна швидко оживити свій додаток.

Інструменти проектування

Xcode включає інтуїтивно зрозумілі інструменти проектування, які дозволяють створювати інтерфейси за допомогою SwiftUI дуже легко. Під час роботи з полотном дизайну все, що ви редагуєте, повністю синхронізується з кодом у сусідньому редакторі. Код миттєво відображається у вигляді попереднього перегляду під час введення, і будь-які зміни, які ви вносите в цей попередній перегляд, одразу з'являються у вашому коді. Xcode миттєво перекомпілює ваші зміни та вставляє їх у запущену версію вашої програми.

Drag and drop

Розташувати компоненти інтерфейса користувача, можливо просто перетягуючи їх на робоче полотно. В інспекторі можна обрати шрифт, колір, вирівнювання та інші параметри дизайну, а також легко переставити елементи керування за допомогою курсора. Багато з цих візуальних редакторів також доступні в редакторі коду, тому можна розробляти інтерфейс як за допомогою інспектора, так і за допомогою коду.

Динамічна заміна

Компілятор Swift і середовище виконання повністю вбудовані в Xcode, тому ваша програма постійно створюється і запускається. Полотно дизайну, яке ви бачите — це не просто наближення вашого інтерфейсу користувача — це ваш живий додаток. А Xcode може замінювати відредагований код безпосередньо у вашому додатку в реальному часі за допомогою «динамічної заміни», нової функції в Swift.

Попередній перегляд

В SwiftUI можливо створити один або кілька попередніх переглядів будь-яких представлень, щоб отримати зразки даних, і налаштувати майже все, що можуть бачити користувачі, наприклад великі шрифти, локалізацію або темний

режим. Попередній перегляд також може відображати інтерфейс користувача на будь-якому пристрої та в будь-якій орієнтації.

2.3 Середовище розробки Xcode

Xcode вміщає все, що потрібно розробникам для створення чудових програм для Mac, iPhone, iPad, Apple TV і Apple Watch. Xcode надає розробникам уніфікований робочий процес для проектування інтерфейсу користувача, кодування, тестування та налагодження.

Xcode включає компілятори Xcode IDE, Swift і C/C++/Objective-C, інструмент аналізу інструментів, симулятори, найновішу SDKA та сотні потужних функцій:

- перегляд налагодження показує тривимірний набір усіх шарів перегляду UI програми під час виконання;
- помилки відображають проблеми в реальному часі під час введення тексту, а виправлення покращує код лише одним клацанням миші;
- рефакторинг дозволяє легко змінювати структуру коду Swift, Objective-C, C і C++;
- порівняння продуктивності ЦП, диска, пам'яті та графічного процесора як графічні доріжки з часом;
- визначення вузьких місць продуктивності;
- створення власних інструментів з унікальними візуалізаціями для аналізу коду та фреймворків і тд.

Щоб тестувати або запускати програми на iPhone, iPad, Apple TV або Apple Watch, вам потрібен лише безкоштовний Apple ID.

3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1 Розробка інтерфейсу додатку

Після встановлення додатку, на головному екрані з'являється іконка “SoundX”, яку можемо спостерігати на рис. 3.1.

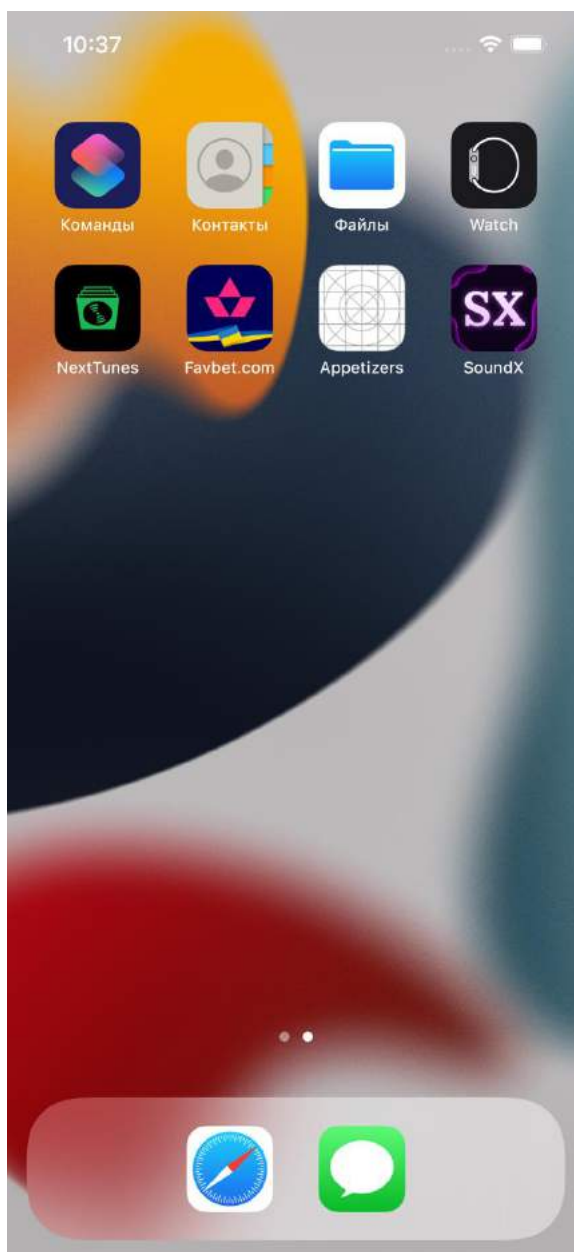


Рисунок 3.1 – Іконка додатку “SoundX”

При відкритті застосунку на декілька секунд з'являється лаунч скрін, що зображений на рис. 3.2 в світлій темі, а на рис. 3.3 у темній.

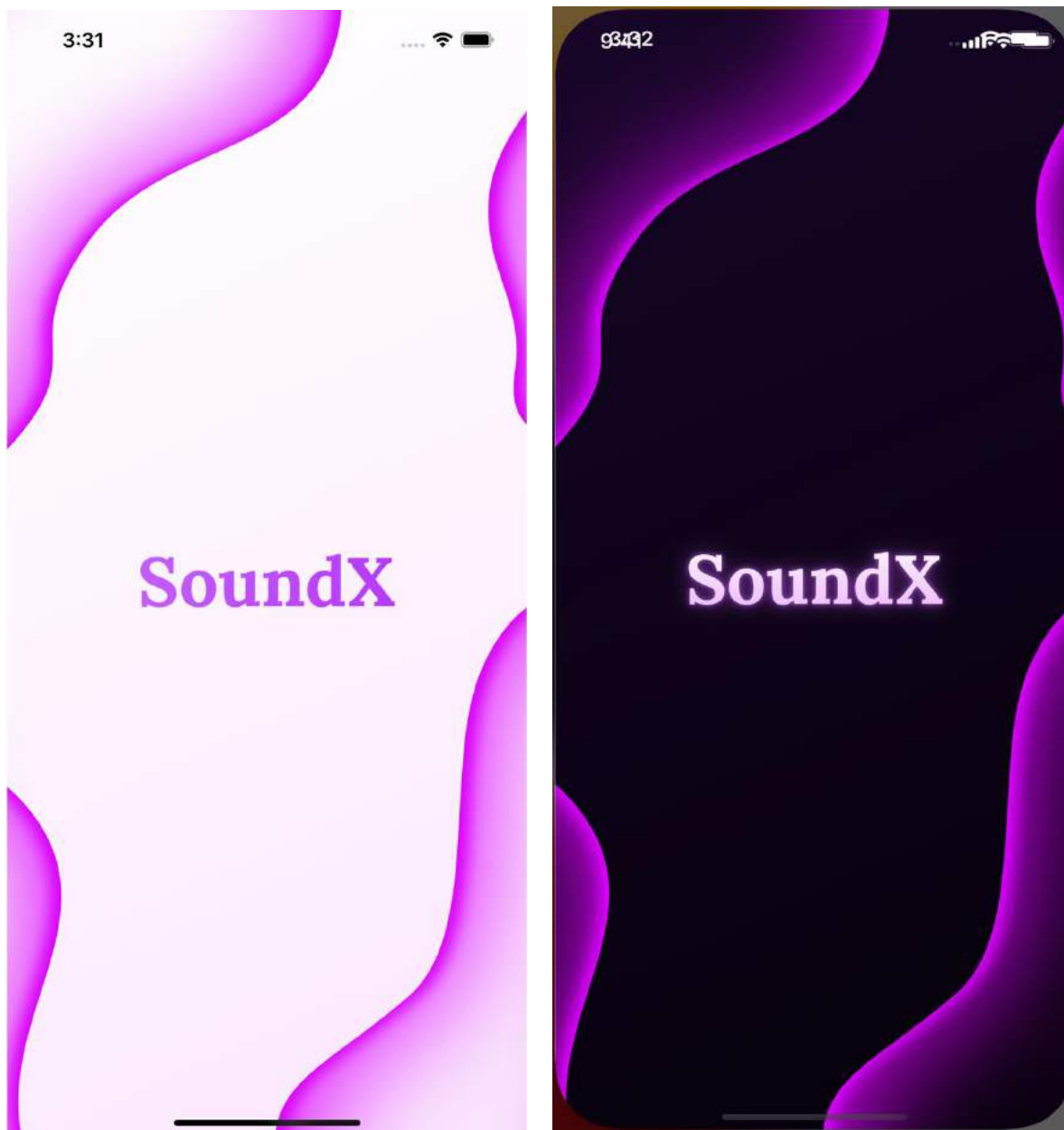


Рисунок 3.1- 3.2 – Лаунч скрін у світлій та темній темах

Після того, як зникає лаунч скрін починається завантаження даних, які будуть відображатися на головному екрані додатку. Тому поки все не завантажиться користувачеві показується скрін завантаження рис. 3.3 – 3.4. Зазвичай всі дані завантажуються протягом декількох секунд, тому чекати довго не доведеться.

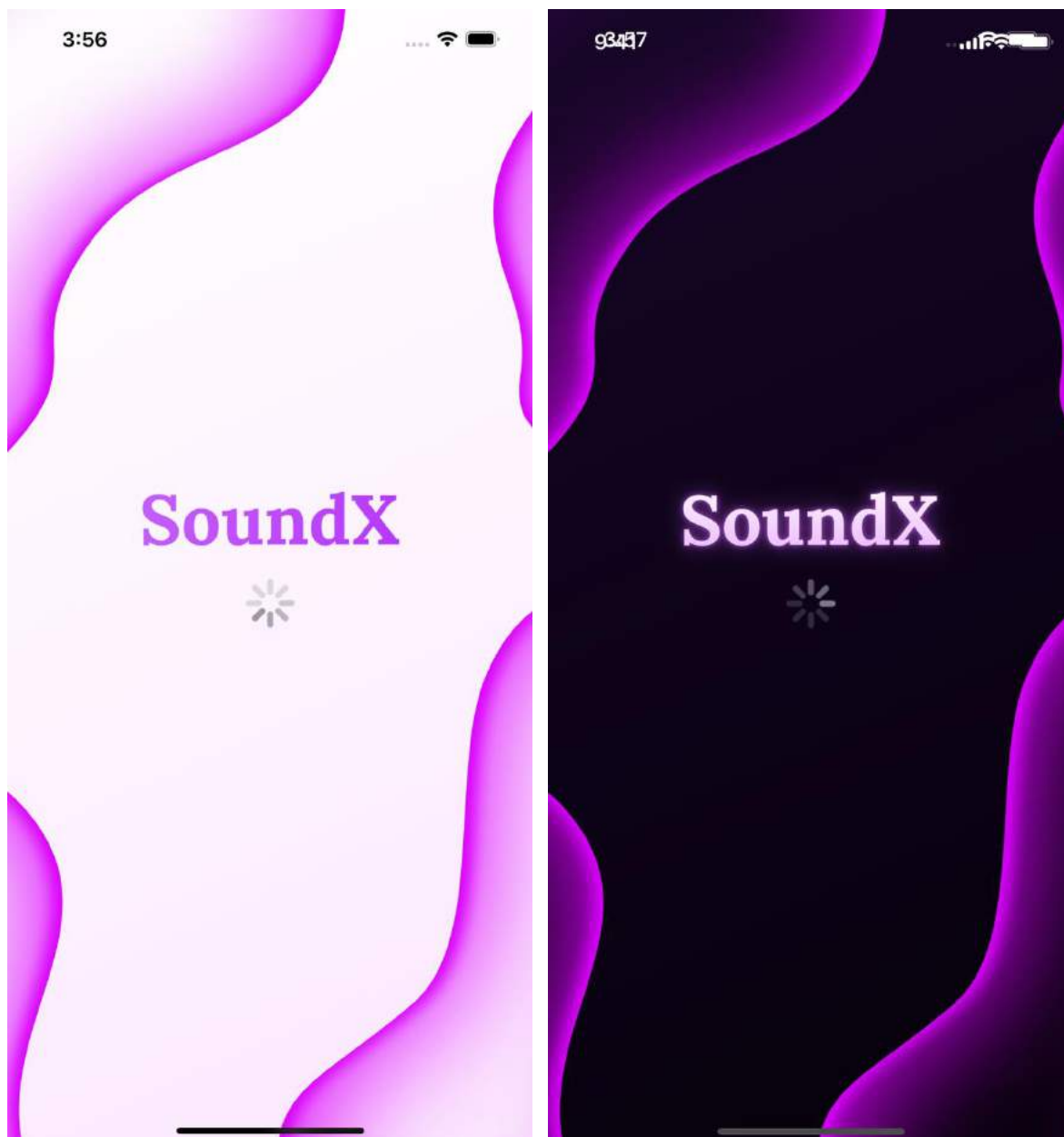


Рисунок 3.3- 3.4 – Екран завантаження у світлій та темній темах

Коли зникає екран завантаження, з'являється головна вкладка додатку - "Home", представлена на рис. 3.5 – 3.6.

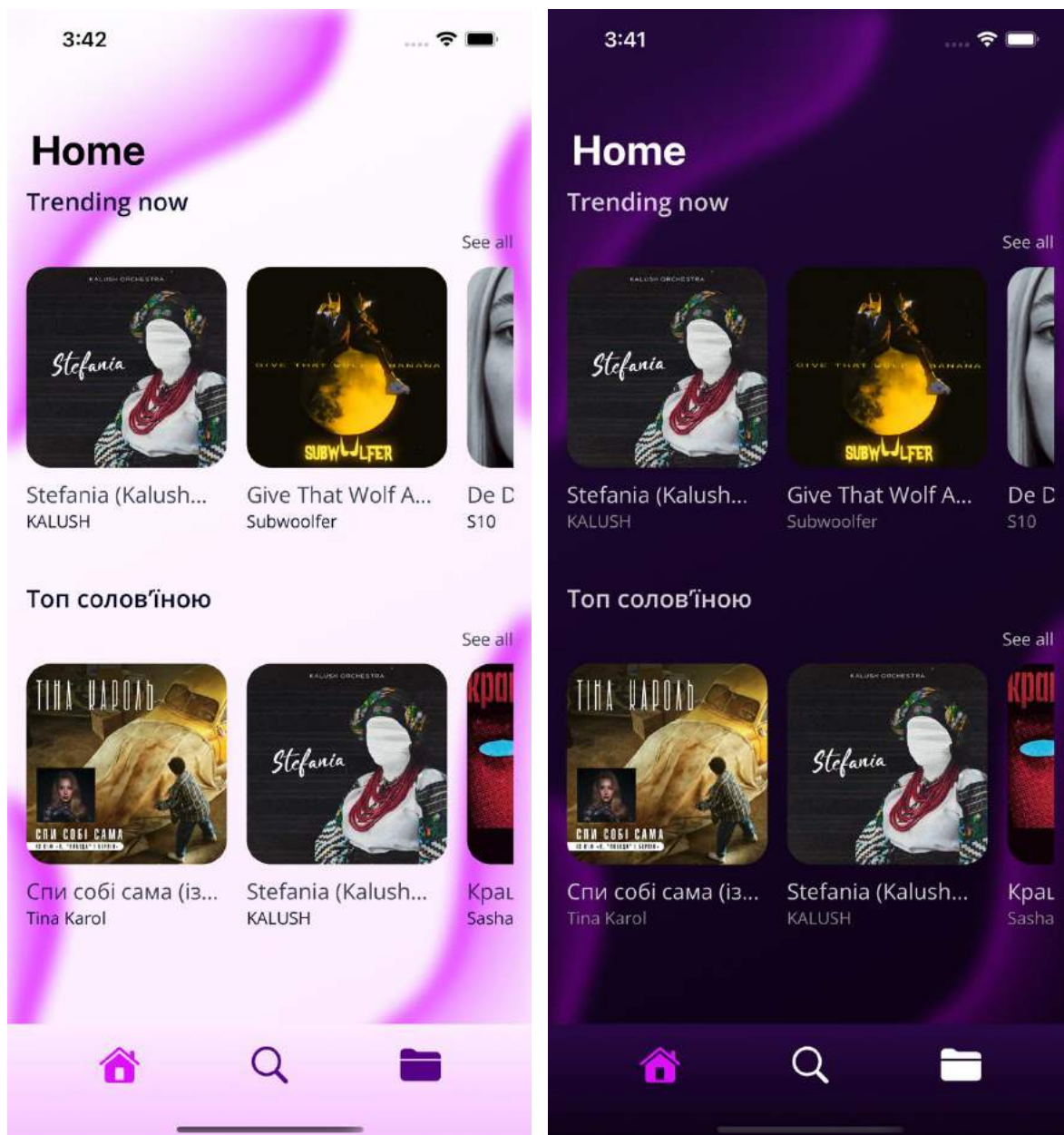


Рисунок 3.5 - 3.6 – Вкладка “Home” у світлій та темній темах

На вкладці “Home” розташовано чотири чарти: “Trending now”, “Топ солов’їною”, “Український реп” та “Український рок.”

До низу екрану прикріплений таббар, за допомогою якого ми можемо переключатися між трьома вкладками: “Home”, “Search” та “Playlist”. Обрана вкладка виділена рожевим кольором.

Тепер детальніше розглянемо, що собою представляє чарт як елемент інтерфейсу користувача, рис 3.7.

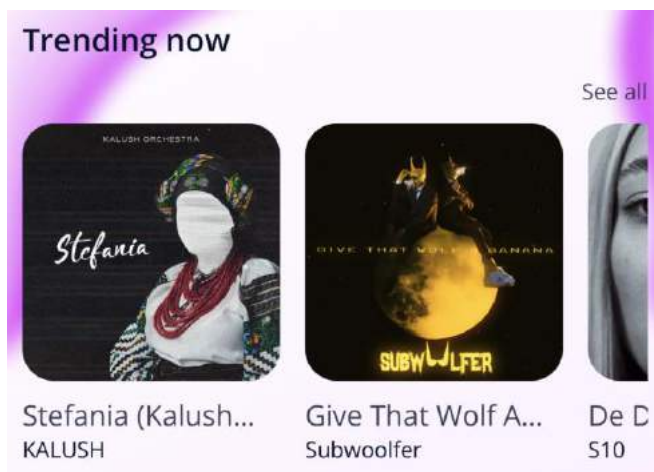


Рисунок 3.7 – Чарт

Чарт включає в себе назву чарта, кнопку “See all” та горизонтальну карусель, в якій знаходяться п’ять перших треків чарту.

Так як картинки пісень завантажуються з інтернету, то якщо користувач має повільний інтернет, йому показується картинка-заглушка рис. 3.8 – 3.9. Коли ж справжня картинка завантажиться, вона замінить картинку-заглушку.



Рисунок 3.8 - 3.9 – Картинка-заглушка у світлій та темній темах

При натисканні на кнопку “See all” ми переходимо на скрін, на якому представленні всі треки чарту, рис. 3.10 – 3.11. Кожен чарт містить 15 пісень.

Для того, щоб повернутися на головний екран потрібно натиснути на кнопку “Home” на навігаторі.

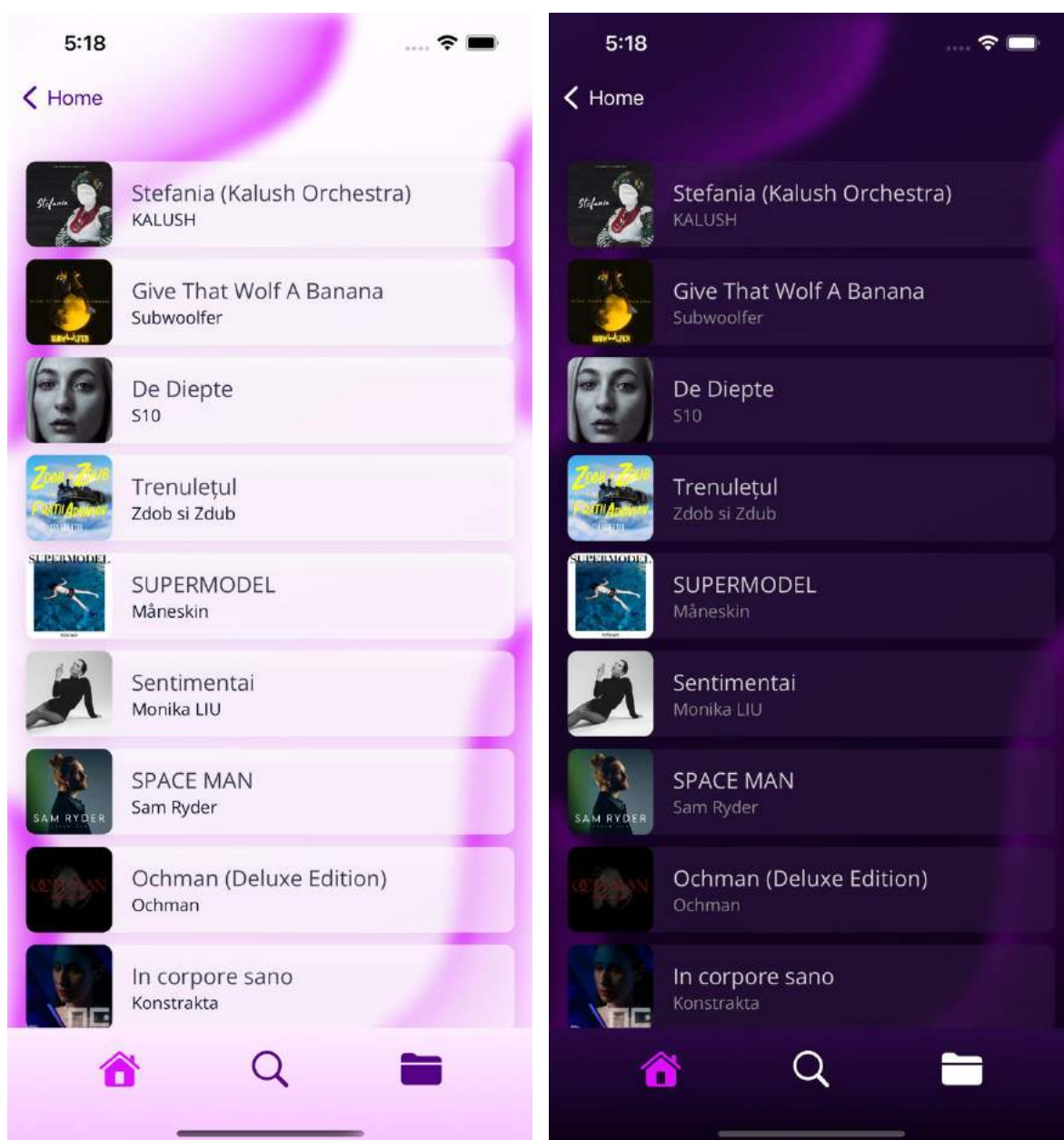


Рисунок 3.10 - 3.11 – Ліст з піснями чарту у світлій та темній темах

При натисканні на будь-яку пісню, що знаходиться на каруселі або на листі, з’являється панель-плеєр, рис 3.12 – 3.13.



Рисунок 3.12 – Панель-плеєр у світлій темі



Рисунок 3.13 – Панель-плеєр у темній темі

На панелі-плеєрі знаходиться інформація про обраний трек (назва треку та виконавець), а також кнопка, за допомогою якою можна увімкнути пісню на програвання або поставити на паузу.

Якщо натиснути на панель-плеєр, то відривається повноцінний екран плеєра, рис. 3.14 – 3.15.

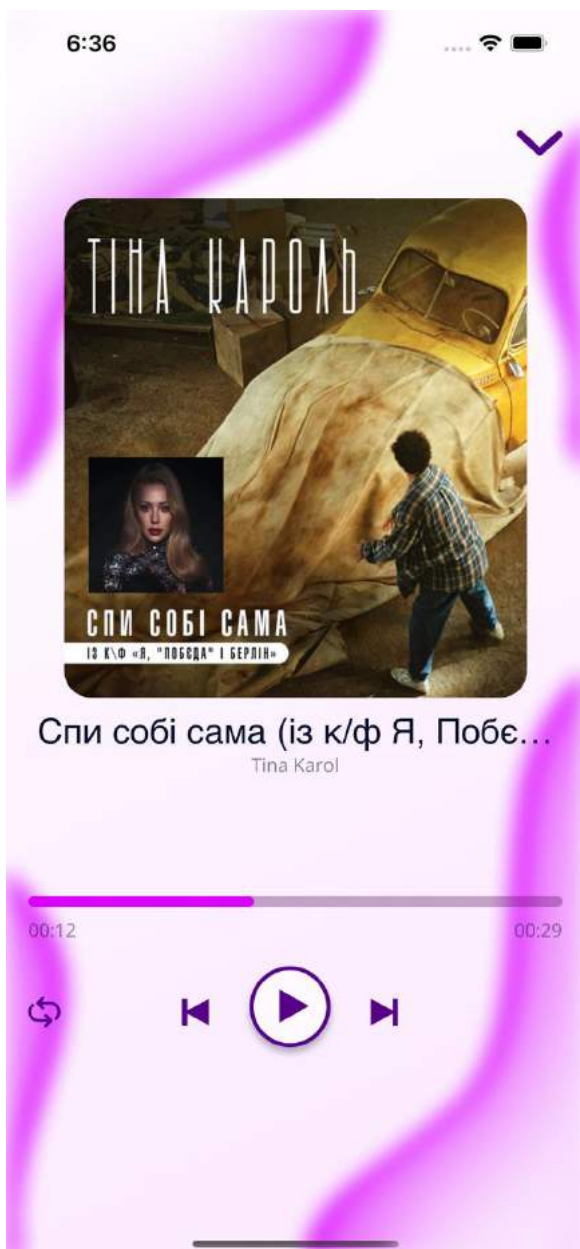


Рисунок 3.14 - 3.15 – Екран плеєру у світлій та темній темах

На даному екрані зліва вгорі розташована кнопка у вигляді стрілки, при натисканні якої ми повертаємося на минулий екран з панель-плеєром. Нижче знаходиться збільшене фото пісні, її назва та виконавець.

Далі розміщений індикатор програвання пісні. Зліва відображається поточна тривалість треку, а справа - загальна.

І останнім елементом екрану є панель керування, яка включає в себе:

- кнопку для зациклення програвання даного треку;
- кнопку для програвання пісні;
- кнопку паузи;

- кнопку для увімкнення минулої пісні;
- кнопку для увімкнення наступної пісні.

Коли кнопка для зациклення програвання даного треку увімкнена, то вона змінює свій колір на рожевий, рис. 3.16. При цьому на обох темах виглядає однаково.



Рисунок 3.16 – Кнопка для зациклення програвання даного треку увімкнена

Далі розглянемо вкладку “Search”, рис. 3.17 – 3.18.

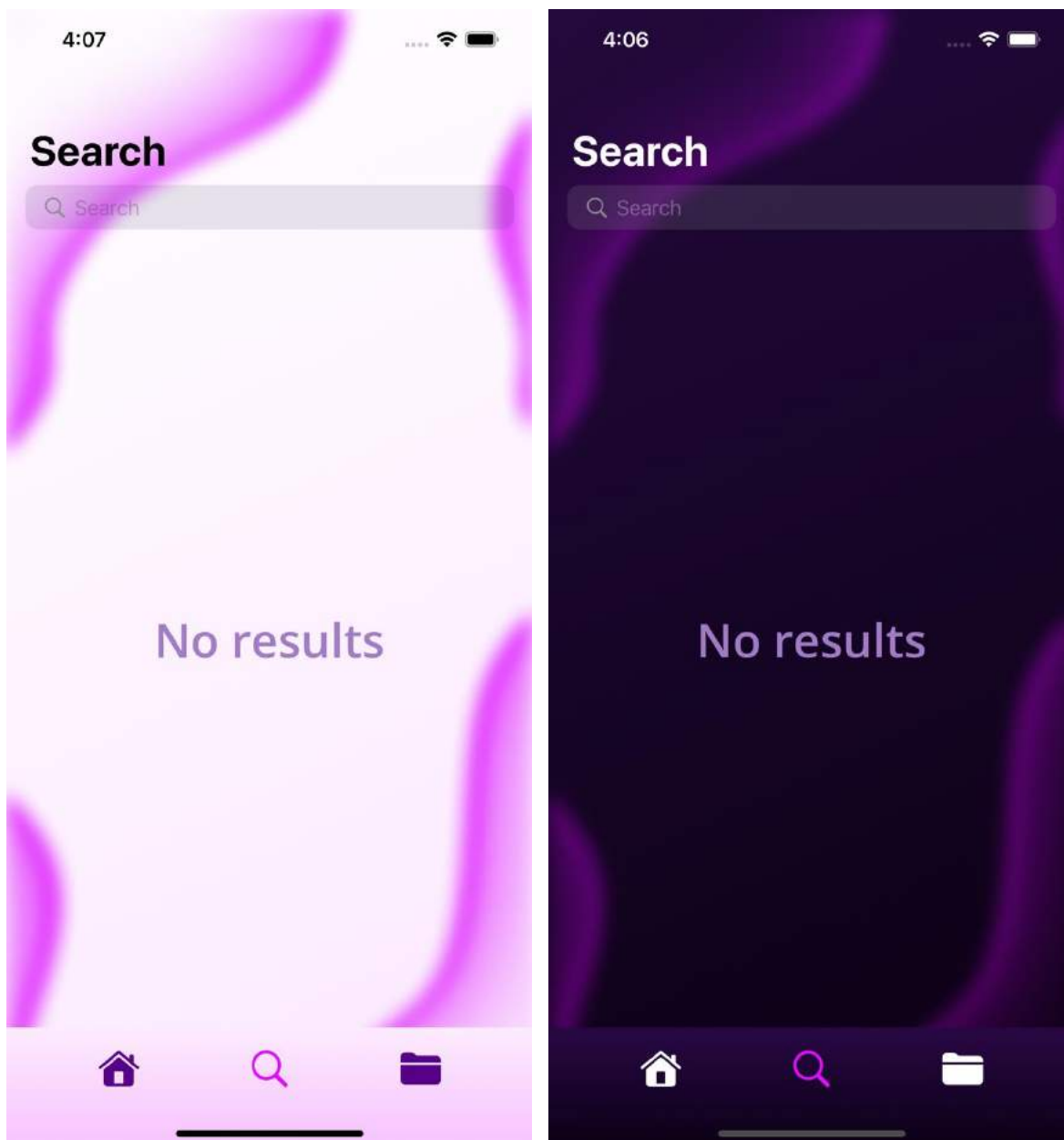


Рисунок 3.17 - 3.18 – Вкладка “Search” у світлій та темній темах

Вгорі вкладки присутня її назва, нижче розташовується поле пошуку. В нього ми вводимо ключовий текст, за яким буде відбуватися пошук пісні. Після вводу тексту, справа з’являється кнопка для очищення поля пошуку рис. 3.19. Ця кнопка має однаковий вигляд на обох темах.

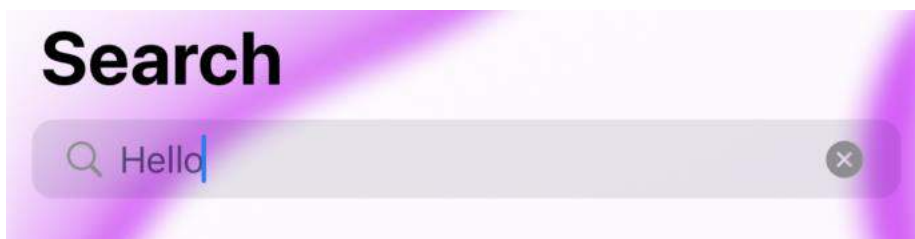


Рисунок 3.19 – Кнопка для очищення поля пошуку

Для того, щоб розпочати пошук потрібно ввести текст в поле пошуку та натиснути на кнопку у вигляді лупи. Після натискання, зникає каретка з поля пошуку.

Якщо користувач має повільний інтернет і для пошуку потрібен певний час, то з'являється індикатор прогресу, рис. 3.20 – 3.21.

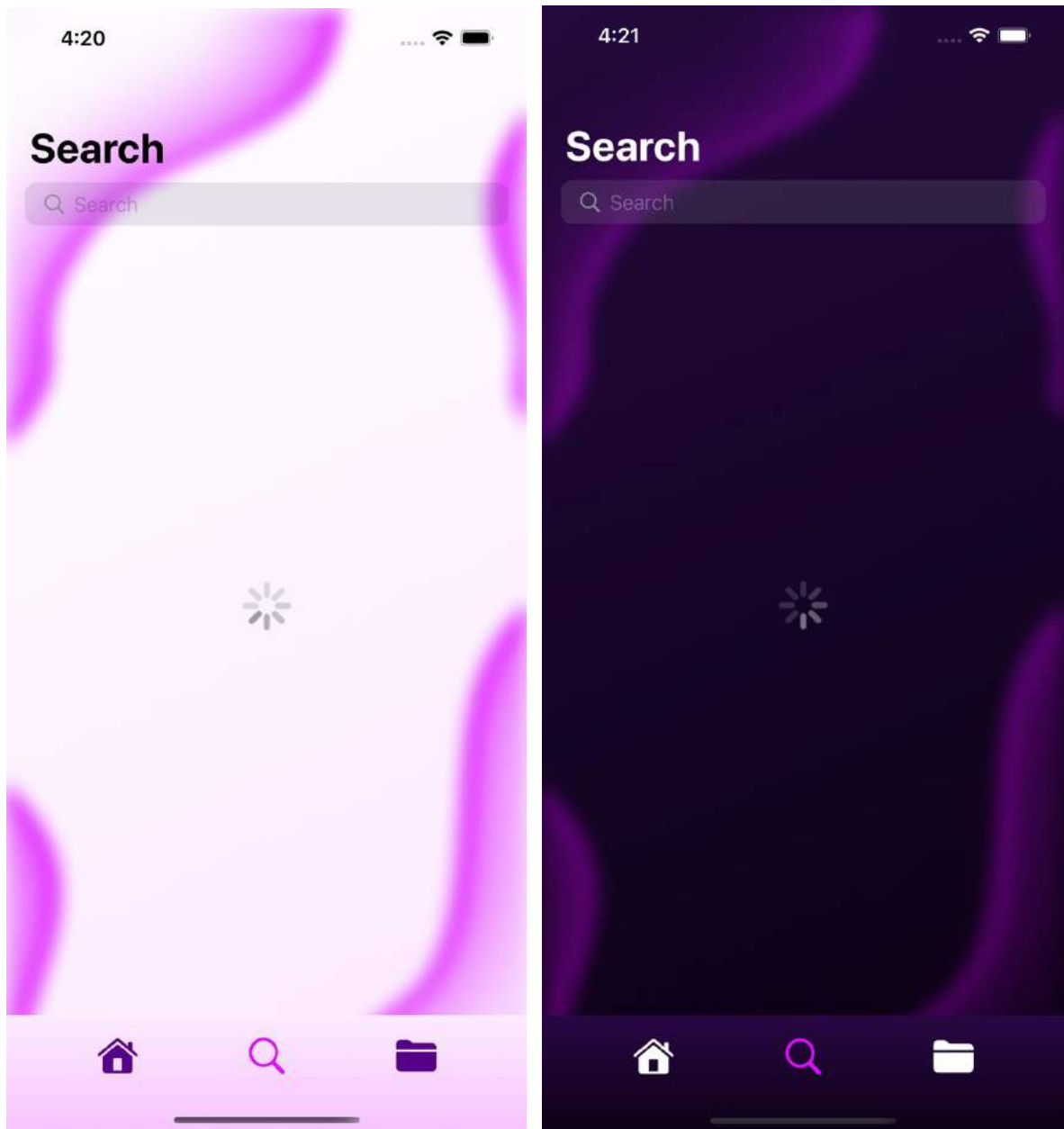


Рисунок 3.20 - 3.21 – Вигляд вкладки “Search” при пошуку у світлій та темній темах

Коли пісні успішно завантаженні, індикатор прогресу зникає, а на його місці з’являється список з піснями, рис. 3.22 – 3.23.

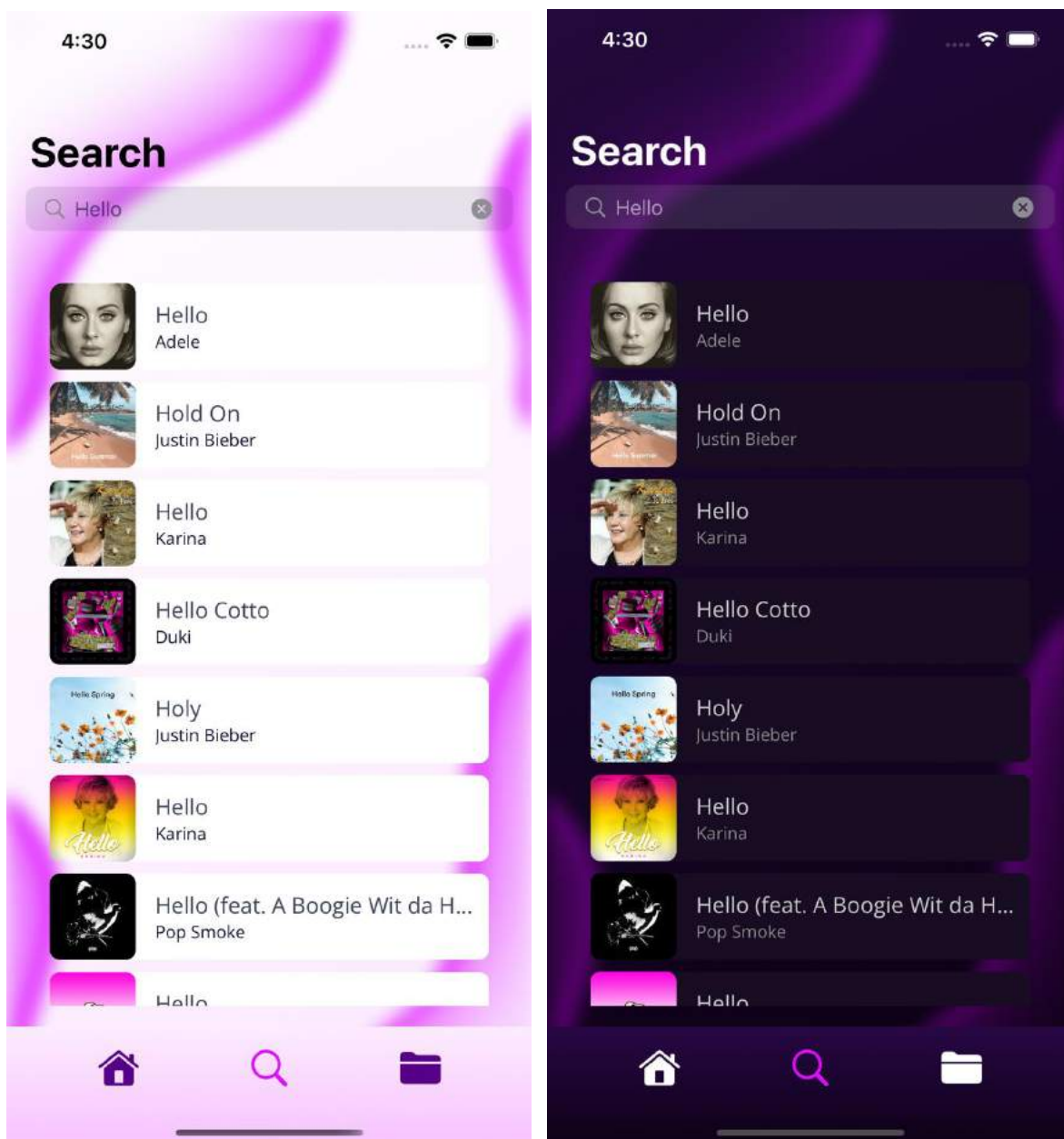


Рисунок 3.20 - 3.21 – Вигляд вкладки “Search” при успішному завантаженні пісень у світлій та темній темах

Якщо не вдалося знайти пісні за заданим текстом у полі пошуку, слабкий інтернет або просто проблеми з передачею даних, то замість списку пісень буде відображатись напис “No results”, як це показано на ”, рис. 3.17 – 3.18.

Перебуваючи на будь-якій вкладці можна програвати пісні на переходити на екран плеєру.

І останнякладка додатку – “Playlist”. На ній відображається плейлист, який користувач склав для себе самостійно, рис. 3.22 – 3.23.

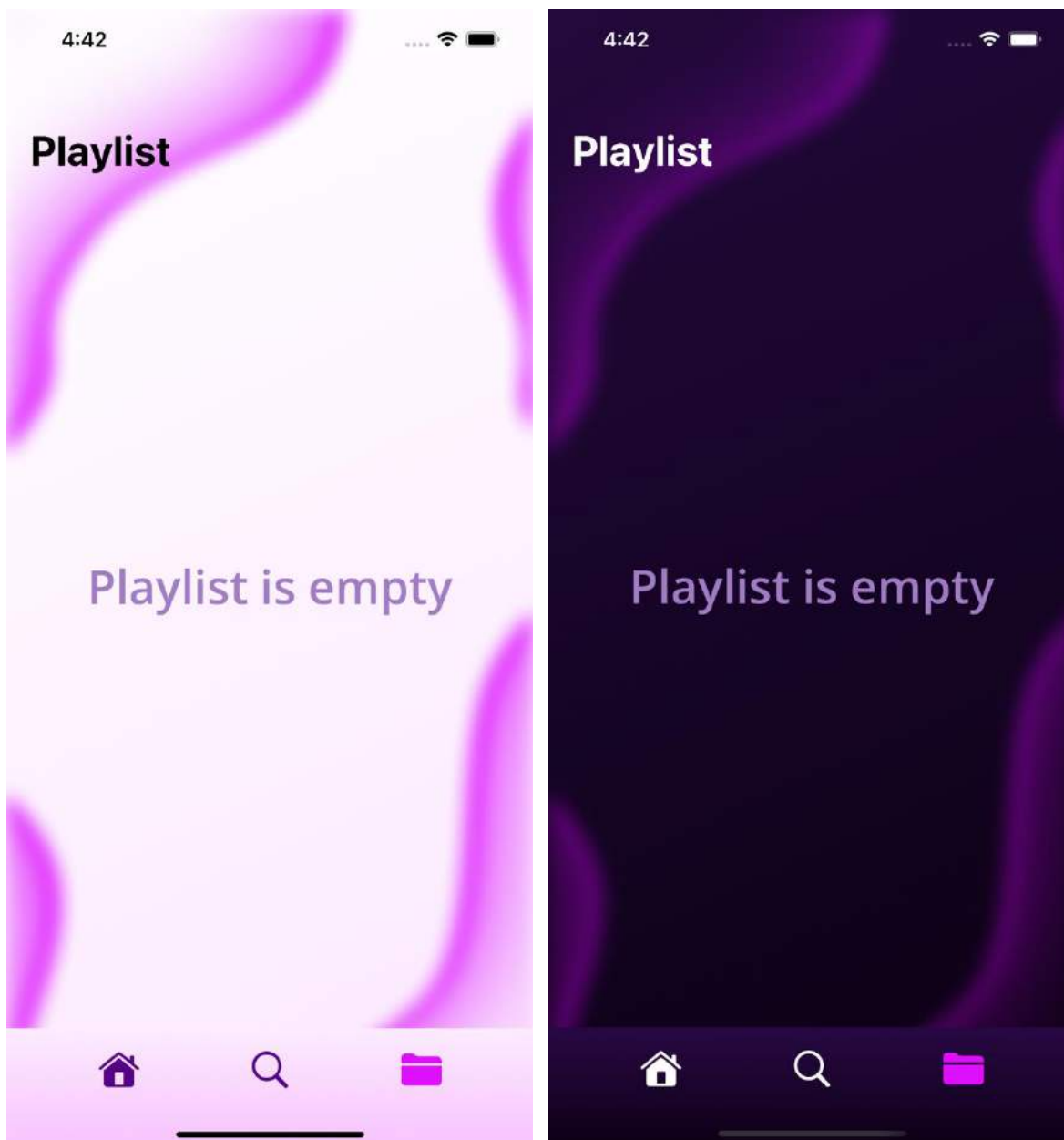


Рисунок 3.20 - 3.21 – Вигляд вкладки “Playlist” у світлій та темній темах

Посередині екран знаходиться напис “Playlist is empty”, тому що зараз в плейлисті немає жодної пісні.

Для того щоб додати пісню в плейлист, потрібно перейти на екран зі списком пісень будь-якого чарту, рис. 3.10 – 3.11 або перейти на вкладку “Search”, задати ключовий текст для пошуку та отримати список пісень, рис. 3.20 – 3.21. Потім потрібно зробити свайп зліва на право, рис. 3.22. Підходить або довгий свайп, або свайп плюс натискання на кнопку.

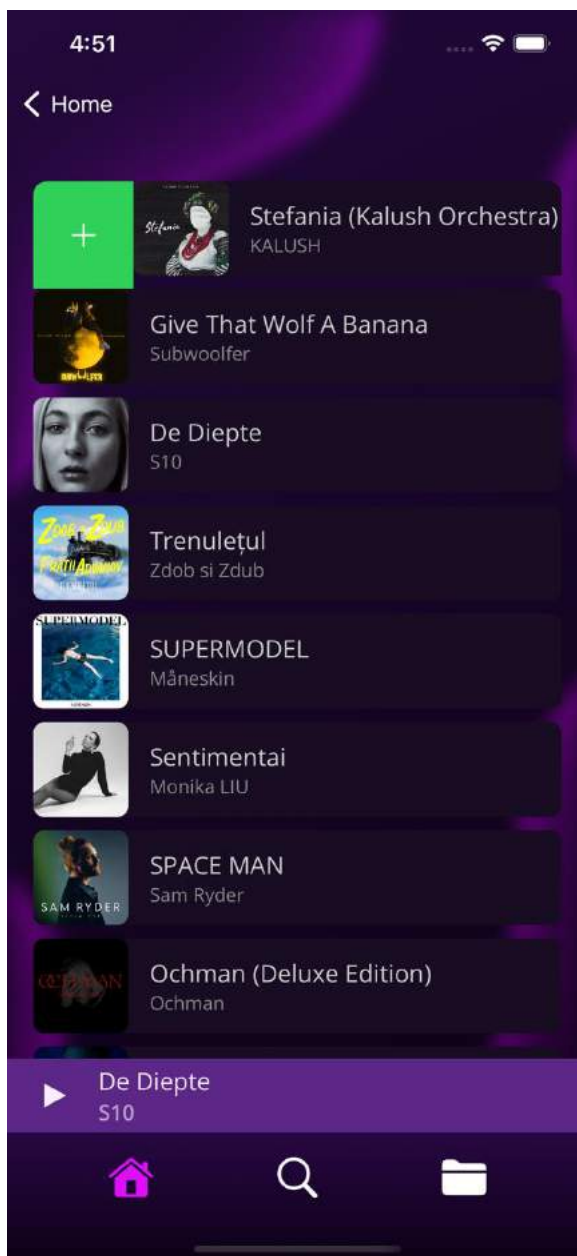


Рисунок 3.22 – Вигляд свайпу зліва на право для додавання пісні в плейлист

Після цього повертаємось на вкладку “Playlist” та бачимо, що обрані пісні додалися до плейлиста, рис. 3.23 – 3.24.

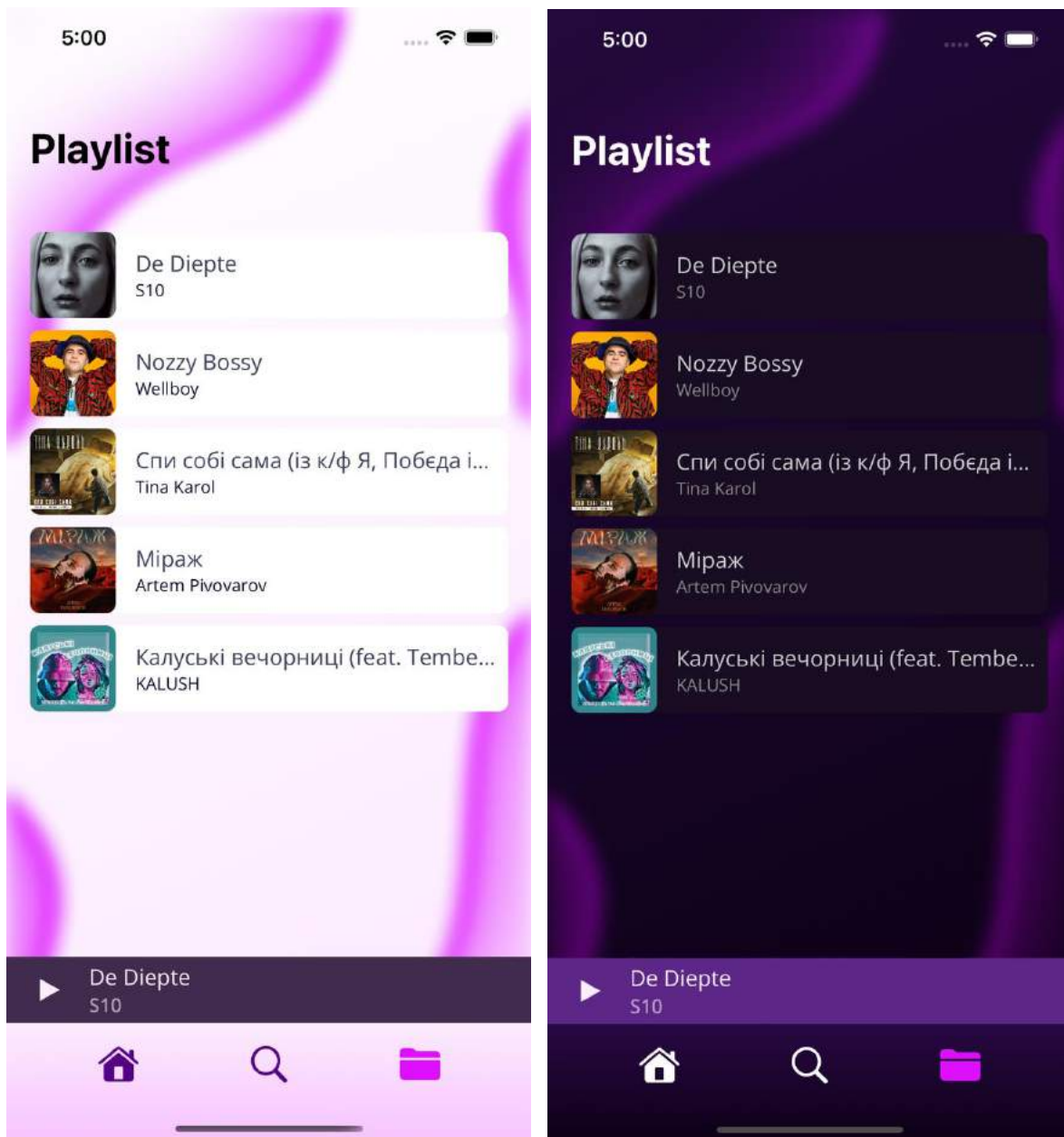


Рисунок 3.23 - 3.24 – Вигляд вкладки “Playlist” з доданими треками у світлій та темній темах

Для того щоб видалити пісню потрібно зробити свайп справа на ліво, рис. 3.25.

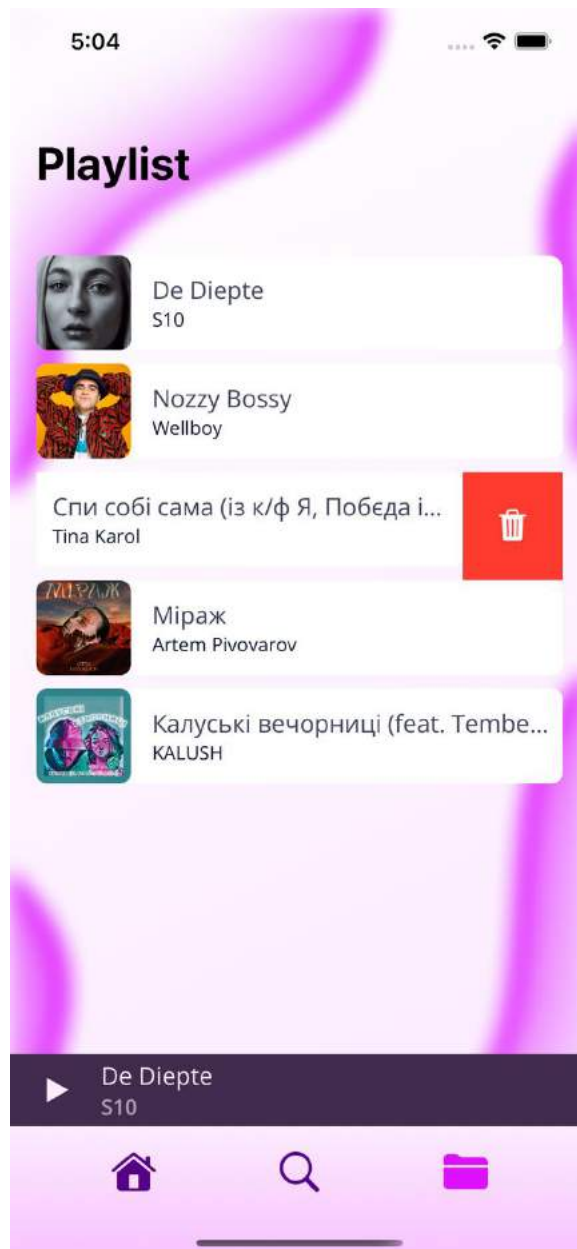


Рисунок 3.25 – Вигляд свайпу справа на ліво для видалення пісні з плейлиста

Для видалення підходить або довгий свайп, або свайп плюс натискання на кнопку.

Якщо вийти і знову зайти в застосунок, то створений плейлист збережеться.

3.2 Тестування мобільного застосунку

Тестування мобільних додатків — це процес перевірки, чи відповідає програма, розроблена для портативних пристроїв, певним вимогам мобільного

тестування. "Мобільний додаток" відноситься до трьох типів програмного забезпечення: нативні програми, мобільні веб-програми та гібридні програми.

У всіх процесах розробки програмного забезпечення тестування мобільних додатків є обов'язковим перед випуском програми на ринок. Цей крок гарантує належне функціонування кінцевого продукту протягом усього його шляху.

Мобільне тестування не лише контролює якість та функціональність програми, але й ефективно підвищує залучення користувачів.

Види тестування мобільних додатків

Функціональне тестування

Часто застосовується для тестування елементів інтерфейсу користувача, адаптації екрана, основних структур і функцій програми. Воно також перевіряє процеси встановлення та оновлення, параметри локалізації, сумісність і доступність тестованої програми (AUT).

Операційне тестування

Оцінює реакцію AUT на події та зміни в певних пристроях, таких як перемикання живлення, підключення кабелю передачі даних, з'єднання Wi-Fi та статус режиму польоту. Потім потрібно порівняти їх із визначеними вимогами.

Тестування переривання

Імітують усі можливі переривання, такі як вхідні та вихідні дзвінки/SMS/MMS, спливаючі сповіщення/нагадування тощо. Вони гарантують, що AUT може впоратися з усіма перериваннями, перейшовши в стан призупинення та перезавантаживши його після цього.

Тестування безпеки

Забезпечує абсолютну безпеку AUT щодо даних користувачів, щоб уникнути таких ризиків, як витік конфіденційної інформації. Тести безпеки забезпечать відповідні тести безпеки даних та мережі AUT, щоб відповідати вимогам безпеки.

Тестування продуктивності

Визначає, як програма реагує на певну кількість проблем, таких як низький рівень заряду акумулятора, мережа поза зоною покриття тощо. Загалом, тестування продуктивності перевіряє швидкість, стабільність та масштабованість програми.

Регресійне тестування

Перевіряє, чи вводять останні оновлення чи виправлення помилки в систему мобільних додатків. Це робиться постійно, щоб гарантувати, що нові зміни є покращеннями функцій, а не дефектами програми.

Найкращі методи тестування мобільних додатків

- Визначте чітку стратегію мобільного тестування та визначте цілі, перш ніж виконувати фактичні тести.
- Виконуйте тести інтерфейсу користувача на реальних мобільних пристроях, а не лише на емуляторах чи симуляторах.
- Переконайтеся, що тести застосовуються як на початковій фазі, так і поступово протягом усього процесу розробки.
- Надайте пріоритет тестуванню на певних пристроях і операційних системах, оскільки недоцільно тестувати на всіх пристроях, ОС і мережевих комбінаціях.
- Виконайте достатньо тестування продуктивності, стресу та безпеки.

3.2 Тестування “SoundX”

Таблиця 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
1	Перевірка іконки додатка	1. Встановлення додатку	Додаток має іконку.	Додаток має іконку.
2	Перевірка лаунч скріна	1. Відкриття додатку	З’являється лаунч скрін на декілька секунд. Відповідає дизайну для світлої та темної теми.	З’являється лаунч скрін на декілька секунд. Відповідає дизайну для світлої та темної теми.

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
3	Перевірка екрану завантаження	1. Відкриття додатку 2. Зачекати поки лаунч скрін зникне	Після лаунч скріна на декілька секунд з'являється екран завантаження. Індикатор завантаження крутиться поки всі дані не будуть завантажені. Відповідає дизайну для світлої та темної теми.	Після лаунч скріна на декілька секунд з'являється екран завантаження. Індикатор завантаження крутиться поки всі дані не будуть завантажені. Відповідає дизайну для світлої та темної теми.
4	Перевірка наявності та вмісту вкладки "Home"	1. Відкриття додатку 2. Зачекати поки лаунч скрін та екран завантаження зникнуть	Після екрану завантаження відображається вкладка "Home". Вона містить назву вкладки, 4 каруселі та таббар. Відповідає дизайну для світлої та темної теми.	Після екрану завантаження відображається вкладка "Home". Вона містить назву вкладки, 4 каруселі та таббар. Відповідає дизайну для світлої та темної теми.
5	Перевірка вмісту каруселі	1. Відкриття додатку 2. Зачекати поки з'явиться вкладка "Home"	Карусель містить назву каруселі, кнопку "See all" та 5 перших пісень чарту, які розташовані горизонтально. Дос-	Карусель містить назву каруселі, кнопку "See all" та 5 перших пісень чарту, які розташовані горизонтально. Дос-

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
			тупний горизонтальний скрол, індикатор скролу – відсутній. Кожна пісня складається з картинки треку, назви треку та виконавця. Відповідає дизайну для світлої та темної теми.	тупний горизонтальний скрол, індикатор скролу – відсутній. Кожна пісня складається з картинки треку, назви треку та виконавця. Відповідає дизайну для світлої та темної теми.
6	Програвання пісні з каруселі	1. Відкриття додатку 2. Зачекати поки з'явиться вкладка "Home" 3. Натискання на будь-яку пісню з каруселі	Відображається панель-плеєр та починає грати обрана пісня.	Відображається панель-плеєр та починає грати обрана пісня.
7	Перевірка наявності, вмісту та функціоналу панелі-плеєра	1. Відкриття додатку 2. Зачекати поки з'явиться вкладка "Home"	Над таббаром з'являється панель-плеєр. Містить кнопку "Грати/ Пауза", назву пісні та виконавця. При на-	Над таббаром з'являється панель-плеєр. Містить кнопку "Грати/ Пауза", назву пісні та виконавця. При на-

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
		3. Натискання на будь-яку пісню з каруселі	тисканні на кнопку “Грати” – грає пісня та кнопка змінюється на кнопку “Пауза”. При натисканні на кнопку “ Пауза ” – пісня зупиняється та кнопка змінюється на кнопку “Грати”. При натисканні на саму панель відкривається екран із плеєром. Відповідає дизайну для світлої та темної теми.	тисканні на кнопку “Грати” – грає пісня та кнопка змінюється на кнопку “Пауза”. При натисканні на кнопку “ Пауза ” – пісня зупиняється та кнопка змінюється на кнопку “Грати”. При натисканні на саму панель відкривається екран із плеєром. Відповідає дизайну для світлої та темної теми.
8	Перевірка наявності та вмісту таббара	1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home”	До низу екрана прикріплений таббар. Він містить 3 вкладки “Home”, “Search”, “Playlist”, при натисканні потрапляємо на відповідний екран. Обрана вкладка виділена рожевим кольором.	До низу екрана прикріплений таббар. Він містить 3 вкладки “Home”, “Search”, “Playlist”, при натисканні потрапляємо на відповідний екран. Обрана вкладка виділена рожевим кольором.

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
			Відповідає дизайну для світлої та темної теми.	Відповідає дизайну для світлої та темної теми.
9	Перевірка відображення картинки-заглушки	<ol style="list-style-type: none"> 1. Використання додаткового програмного забезпечення для сповільнення інтернету 2. Відкриття додатку 3. Зачекати поки з'явиться вкладка "Home" 	Поки завантажується картинка пісні, відображається картинка-заглушка, а потім замінюється на справжню картинку пісні. Відповідає дизайну для світлої та темної теми.	Поки завантажується картинка пісні, відображається картинка-заглушка, а потім замінюється на справжню картинку пісні. Відповідає дизайну для світлої та темної теми.
10	Перевірка функціоналу кнопки "See all"	<ol style="list-style-type: none"> 1. Відкриття додатку 2. Зачекати поки з'явиться вкладка "Home" 3. Натискання на кнопку "See all" на 	Перехід на екран зі списком усіх пісень обраного чарту.	Перехід на екран зі списком усіх пісень обраного чарту.

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
		будь-якій каруселі		
11	Перевірка вмісту екрану зі списком пісень	<p>1. Відкриття додатку</p> <p>2. Зачекати поки з'явиться вкладка "Home"</p> <p>3. Натискання на кнопку "See all" на будь-якій каруселі</p>	<p>На екрані знаходиться 15 треків обраного чарту та навігатор з кнопкою "Home". Елементи списку розташовані вертикально, доступна вертикальна прокрутка усіх елементів. Індикатор прокрутки відсутній. Таббар присутній, відображається обрана вкладка "Home".</p> <p>Відповідає дизайну для світлої та темної теми.</p>	<p>На екрані знаходиться 15 треків обраного чарту та навігатор з кнопкою "Home". Елементи списку розташовані вертикально, доступна вертикальна прокрутка усіх елементів. Індикатор прокрутки відсутній. Таббар присутній, відображається обрана вкладка "Home".</p> <p>Відповідає дизайну для світлої та темної теми.</p>
12	Перевірка вмісту елемента списку пісень	<p>1. Відкриття додатку</p> <p>2. Зачекати поки з'явиться вкладка "Home"</p>	<p>Елемент містить картинку пісні, назву пісні та виконавця. Відповідає дизайну для світлої та темної теми.</p>	<p>Елемент містить картинку пісні, назву пісні та виконавця. Відповідає дизайну для світлої та темної теми.</p>

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
		3. Натискання на кнопку “See all” на будь-якій каруселі		
13	Перевірка функціоналу елемента списку пісень	1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home” 3. Натискання на кнопку “See all” на будь-якій каруселі 4. Натискання на будь-який трек	З’являється панель-плеєр в якій відображається назва та виконавець обраної пісні. Починає програвати трек, який було обрано.	З’являється панель-плеєр в якій відображається назва та виконавець обраної пісні. Починає програвати трек, який було обрано.
14	Перевірка роботи кнопки “Home” на навігаторі	1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home”	Перехід на екран “Home”.	Перехід на екран “Home”.

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
		3. Натискання на кнопку “See all” на будь-якій каруселі 4. Натискання на кнопку “Home” на навігаторі		
15	Перевірка вмісту екрана плеєра	1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home” 3. Натискання на будь-який трек з каруселі 4. Натискання на панель-плеєр	На екрані зліва вгорі розташована кнопка у вигляді стрілки, нижче знаходиться збільшене фото пісні, її назва та виконавець. Далі розміщений індикатор програвання пісні. Зліва відображається поточна тривалість треку, а справа - загальна. І останнім елементом екрану є панель керування.	На екрані зліва вгорі розташована кнопка у вигляді стрілки, нижче знаходиться збільшене фото пісні, її назва та виконавець. Далі розміщений індикатор програвання пісні. Зліва відображається поточна тривалість треку, а справа - загальна. І останнім елементом екрану є панель керування. Відповідає дизайну

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
			Відповідає дизайну для світлої та темної теми.	Відповідає дизайну для світлої та темної теми.
16	Перевірка роботи кнопки згортання	1. Відкриття додатку 2. Зачекати поки з'явиться вкладка "Home" 3. Натискання на будь-який трек з каруселі 4. Натискання на панель-плеєр 5. Натискання на кнопку згортання	Повернення на екран, на якому були до натискання на панель-плеєр.	Повернення на екран, на якому були до натискання на панель-плеєр.
17	Перевірка роботи індикатора програвання пісні	1. Відкриття додатку 2. Зачекати поки з'явиться	Повзунок перетягується на те місце, куди був перетягнутий, відображається час програвання на змі-	Повзунок перетягується на те місце, куди був перетягнутий, відображається час програвання на змінений і

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
		вкладка “Home” 3. Натискання на будь-який трек з каруселі 4. Натискання на панель-плеєр 5. Перетягування вперед, назад повзунка	на змінений і пісня починає грати з вказаного моменту.	пісня починає грати з вказаного моменту.
18	Перевірка вмісту панелі програвання	1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home” 3. Натискання на будь-який трек з каруселі	Панель містить кнопку для зациклення програвання даного треку, кнопку для програвання пісні, кнопку паузи, кнопку для увімкнення минулої пісні та кнопку для увімкнення наступної пісні. Відповідає	Панель містить кнопку для зациклення програвання даного треку, кнопку для програвання пісні, кнопку паузи, кнопку для увімкнення минулої пісні та кнопку для увімкнення наступної пісні. Відповідає дизайну для світлої та темної теми.

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
		4. Натискання на панель-плеєр	дизайну для світлої та темної теми.	
19	Перевірка функціоналу панелі програвання	<p>1. Відкриття додатку</p> <p>2. Зачекати поки з'явиться вкладка "Home"</p> <p>3. Натискання на будь-який трек з каруселі</p> <p>4. Натискання на панель-плеєр</p>	<p>При натисканні на кнопку зациклення пісні, пісня постійно повторюється і кнопка на обох темах відсвічується рожевим кольором. При натисканні на кнопку програвання пісні, пісня починає грати і кнопка змінюється на кнопку паузи. При натисканні на кнопку паузи, пісня зупиняється і кнопка змінюється на кнопку програвання. При натисканні на</p>	<p>При натисканні на кнопку зациклення пісні, пісня постійно повторюється і кнопка на обох темах відсвічується рожевим кольором. При натисканні на кнопку програвання пісні, пісня починає грати і кнопка змінюється на кнопку паузи. При натисканні на кнопку паузи, пісня зупиняється і кнопка змінюється на кнопку програвання. При натисканні на кнопку увімкнення минулої пісні, минула пісня починає грати. Якщо</p>

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
			<p>минулої пісні, минула пісня починає грати. Якщо дана пісня є першою піснею в чарті, то при натисканні на кнопку нічого не відбувається. При натисканні на кнопку увімкнення наступної пісні, наступна пісня починає грати. Якщо дана пісня є останньою піснею в чарті, то при натисканні на кнопку нічого не відбувається.</p>	<p>піснею в чарті, то при натисканні на кнопку нічого не відбувається. При натисканні на кнопку увімкнення наступної пісні, наступна пісня починає грати. Якщо дана пісня є останньою піснею в чарті, то при натисканні на кнопку нічого не відбувається.</p>
20	Перевірка вмісту екрану "Search"	<p>1. Відкриття додатку 2. Зачекати поки з'явиться вкладка "Home"</p>	Вгорі вкладки присутня її назва, нижче розташовується поле пошуку, в плейшолдері якого відображається текст	Вгорі вкладки присутня її назва, нижче розташовується поле пошуку, в плейшолдері якого відображається текст "Search". Нижче

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
		3. Перейти на вкладку “Search”	“Search”. Нижче присутній напис “No results”. Відповідає дизайну для світлої та темної теми.	присутній напис “No results”. Відповідає дизайну для світлої та темної теми.
21	Перевірка вмісту та роботи поля пошуку	1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home” 3. Перейти на вкладку “Search”	При натисканні на поле пошуку на ньому з’являється каретка. Введений текст відображається у полі. Після цього з’являється кнопка для очищення тексту поля. Вона має однаковий вигляд на обох темах. При натисканні на кнопку очищення – текст введений в поле пошуку очищується. Поле пошуку дозволяє вводити будь-які символи. При натис-	При натисканні на поле пошуку на ньому з’являється каретка. Введений текст відображається у полі. Після цього з’являється кнопка для очищення тексту поля. Вона має однаковий вигляд на обох темах. При натисканні на кнопку очищення – текст введений в поле пошуку очищується. Поле пошуку дозволяє вводити будь-які символи. При натисканні на кнопку пошуку розпочинається пошук за введеним

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
			канні на кнопку пошуку розпочинається пошук за введеним текстом, також має однаковий дизайн на обох темах.	текстом, також має однаковий дизайн на обох темах.
22	Перевірка наявності індикатора завантаження	<p>1. Використання додаткового програмного забезпечення для сповільнення інтернету</p> <p>2. Відкриття додатку</p> <p>3. Зачекати поки з'явиться вкладка "Home"</p> <p>4. Перейти на вкладку "Search"</p>	Зникне напис "No results" та на декілька секунд з'явиться індикатор завантаження. Відповідає дизайну для світлої та темної теми.	Зникне напис "No results" та на декілька секунд з'явиться індикатор завантаження. Відповідає дизайну для світлої та темної теми.

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
		5. Ввести ключовий текст для пошуку пісні 6. Натиснути кнопку пошуку	Зникне напис “No results” та на декілька секунд з’явиться індикатор завантаження. Відповідає дизайну для світлої та темної теми.	Зникне напис “No results” та на декілька секунд з’явиться індикатор завантаження. Відповідає дизайну для світлої та темної теми.
23	Перевірка успішного пошуку пісень	1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home” 3. Перейти на вкладку “Search” 4. Ввести текст “Тримай” 6. Натиснути кнопку пошуку	Отримуємо список з відфільтрованих пісень. Відповідає дизайну для світлої та темної теми.	Отримуємо список з відфільтрованих пісень. Відповідає дизайну для світлої та темної теми.

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
24	Перевірка неуспішного пошуку пісень	1. Відкриття додатку 2. Зачекати поки з'явиться вкладка "Home" 3. Перейти на вкладку "Search" 4. Ввести текст ".009" 6. Натиснути кнопку пошуку	Відображається напис "No results".	Відображається напис "No results".
25	Перевірка вмісту вкладки "Playlist"	1. Відкриття додатку 2. Зачекати поки з'явиться вкладка "Home" 3. Перейти на вкладку "Playlist"	Екран містить назву вкладки та список пісень, які були додані до плейлиста. Таббар присутній і виділяє рожевим кольором останню вкладку. Відповідає дизайну для світлої та темної теми.	Екран містить назву вкладки та список пісень, які були додані до плейлиста. Таббар присутній і виділяє рожевим кольором останню вкладку. Відповідає дизайну для світлої та темної теми.

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
26	Перевірка довгого свайпу справа на ліво для вкладки “Playlist”	1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home” 3. Перейти на вкладку “Playlist” 4. Зробити довгий свайп справа на ліво для будь-якого елемента списку	Елемент списку – видалений.	Елемент списку – видалений.
27	Перевірка свайпу справа на ліво + натискання на кнопку для вкладки “Playlist”	1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home” 3. Перейти на вкладку “Playlist” 4. Зробити свайп справа	Елемент списку – видалений.	Елемент списку – видалений.

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
		на ліво + натиснути на кнопку для будь-якого елемента списку		
28	Перевірка довгого свайпу зліва на право для вкладки “Playlist”	<ol style="list-style-type: none"> 1. Відкриття додатку 2. Зачекати поки з’явиться вкладка “Home” 3. Перейти на вкладку “Playlist ” 4. Зробити довгий свайп зліва на право для будь-якого елемента списку 	Нічого не відбувається	Нічого не відбувається
29	Перевірка свайпу зліва на право + натискання на кнопку для	<ol style="list-style-type: none"> 1. Відкриття додатку 2. Зачекати поки з’я- 	Нічого не відбувається	Нічого не відбувається

Продовження таблиці 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
	вкладки “Playlist”	вється вкладка “Home” 3. Перейти на вкладку “Playlist ” 4. Зробити свайп зліва на право + натиснути на кнопку для будь-якого елемента списку		

ВИСНОВКИ

У дипломній роботі було виконано розробку музичного додатку “SoundX” для мобільної платформи IOS. В процесі виконання роботи було проаналізовано аналоги музичних додатків: Spotify, SoundCloud, Deezer та Apple Music.

Під час дослідження було визначено переваги та недоліки аналогів, а також основний функціонал, який повинен містити застосунок.

На етапі моделювання діаграми для програмного забезпечення було поставлено вимоги до його функціоналу.

У процесі проектування було використано нативні та створено кастомні компоненти для розробки інтерфейсу. Також створено структури та класи, які можуть використовуватись повторно для розширення функціоналу додатку.

Для досягнення поставленої мети було проаналізовано актуальність розроблюваного програмного забезпечення, досліджено та використано мову програмування Swift, фреймворк SwiftUI та середу розробки Xcode, за допомогою яких було розроблено функціональні вимоги до застосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SwiftUI [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com/xcode/swiftui/>
2. Swift [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com/swift/>
3. Топ-10 найкращих музичних програм [Електронний ресурс] – Режим доступу до ресурсу: https://www.moyo.ua/ua/news/top-10_luchshih_muzykalnyh_prilozheniyi_gde_slushat_muzyku_onlayin_posle_blokirovki_vk.html
4. Методи дослідження в дипломній роботі [Електронний ресурс] – Режим доступу до ресурсу: <https://diplom.com.ua/uk/metodi-doslidzhennya-v-diplomnij-roboti/>
5. Spotify still tops other music services, but its market share declined [Електронний ресурс] – Режим доступу до ресурсу: <https://techcrunch.com/2022/01/20/spotify-subscription-numbers-up-youtube-music-tidal/>
6. Spotify [Електронний ресурс] – Режим доступу до ресурсу: <https://www.spotify.com/us/>
7. SoundCloud [Електронний ресурс] – Режим доступу до ресурсу: <https://soundcloud.com>
8. Apple Music [Електронний ресурс] – Режим доступу до ресурсу: <https://www.apple.com/ua/apple-music/>
9. Deezer [Електронний ресурс] – Режим доступу до ресурсу: <https://www.deezer.com/ru/>
10. Spotify Data API [Електронний ресурс] – Режим доступу до ресурсу: <https://rapidapi.com/Glavier/api/spotify23/>
11. Дизайн для додатку SoundX [Електронний ресурс] – Режим доступу до ресурсу: <https://www.figma.com/file/35FChPHW3wxVgG2QIaNsRg/>

12. Шрифт для додатку SoundX [Електронний ресурс] – Режим доступу до ресурсу: <https://fonts.google.com/specimen/Open+Sans#standard-styles>

13. How to use custom fonts with SwiftUI [Електронний ресурс] – Режим доступу до ресурсу: <https://betterprogramming.pub/swiftui-basics-importing-custom-fonts-b6396d17424d>

14. How to work with Spotify Data [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.spotify.com/documentation/general/guides/working-with-playlists/>

15. Connection Flow Guidelines to authorize your app with users' Spotify Account [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.spotify.com/use-cases/mobile-apps/connection-flow/>

16. How SwiftUI's List is becoming much more flexible this year [Електронний ресурс] – Режим доступу до ресурсу: <https://wwdcbysundell.com/2021/exploring-new-swiftui-list-apis/>

17. Hashable Protocols in Swift [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@JoyceMatos/hashable-protocols-in-swift-baf0cabeaebd>

18. The power of Result types in Swift [Електронний ресурс] – Режим доступу до ресурсу: <https://www.swiftbysundell.com/articles/the-power-of-result-types-in-swift/>

19. Returning values from a closure Swift [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hackingwithswift.com/sixty/6/3/returning-values-from-a-closure>

20. Swift - Async calls in loop [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com/forums/thread/669817>

21. Working with Tab Bar Customization [Електронний ресурс] – Режим доступу до ресурсу: <https://www.appcoda.com/swiftui-tabview/>

22. AVPlayer [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com/documentation/avfoundation/avplayer>

23. How to use UserDefaults in SwiftUI [Електронний ресурс] – Режим доступу до ресурсу: <https://www.simpleswiftguide.com/how-to-use-userdefaults-in-swiftui/>

24. Archiving Swift objects with Codable [Электронный ресурс] – Режим доступа до ресурсу: <https://www.hackingwithswift.com/books/ios-swiftui/archiving-swift-objects-with-codable>

25. Make the circular ProgressView() bigger [Электронный ресурс] – Режим доступа до ресурсу: <https://www.hackingwithswift.com/forums/swiftui/make-the-circular-progressview-bigger/10775>

26. Mobile Testing [Электронный ресурс] – Режим доступа до ресурсу: <https://katalon.com/mobile-testing>

27. UML diagram [Электронный ресурс] – Режим доступа до ресурсу: <https://app.creately.com/d/muL3rq9M71S/edit>

28. All You Need to Know About UML Diagrams [Электронный ресурс] – Режим доступа до ресурсу: <https://tallyfy.com/uml-diagram/>

29. Xcode [Электронный ресурс] – Режим доступа до ресурсу: <https://developer.apple.com/xcode/>

30. Swift - Integer conversion to Hours/Minutes/Seconds [Электронный ресурс] – Режим доступа до ресурсу: <https://stackoverflow.com/questions/26794703/swift-integer-conversion-to-hours-minutes-seconds>

ДОДАТОК А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
 ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка музичного додатку "SoundX" мовою програмування Swift

Виконала студентка 5 курсу
 групи ППЗ-51
 Мала Юлія Володимирівна
 Керівник роботи
 Жебка Вікторія Вікторівна

Київ – 2022

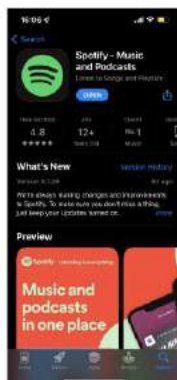
МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** - оптимізація процесу відтворення музики за допомогою створеного музичного додатку з використанням фреймворку SwiftUI .
- **Об'єкт дослідження** - процес прослуховування музики .
- **Предмет дослідження** - методи та засоби розробки музичного додатку .

2

АНАЛОГИ

Spotify



Переваги

Різноманітні і добре продумані плейлисти на будь-який смак, включаючи підбір на основі часу доби.

Присутня детальна інформація про композиції.

Додаток підтримує всі платформи.

Недоліки

Немає версії українською мовою.

SoundCloud



Переваги

Простий і зрозумілий пошук

Різноманітність жанрів і стилів

Сервіс – безкоштовний для слухача.

Недоліки

Відсутність україномовного інтерфейсу

Без підключення до мережі на мобільному телефоні можна запустити тільки раніше прослухані треки.

3

Apple Music



Переваги
Необмежений доступ до десятків мільйонів композицій.
Можливість не тільки слухати треки, але й дивитися відео.
Безліч різножанрових радіостанцій плюс можливість створити свою власну.
Недоліки
Необхідність введення даних кредитки.
Досить висока вартість передплати

Deezer



Переваги
Додаток легко освоїти завдяки зрозумілому інтерфейсу.
Музика високої якості передається на швидкості 320 кбіт/с.
Є автоматизований плейлист Flow, який дозволяє слухати музику, не відволікаючись на зміну треків вручну.
Недоліки
Висока вартість передплати.
Представлено дуже мало пісень українських виконавців.
У безкоштовному додатку якість музики становить всього 128 кбіт/с.

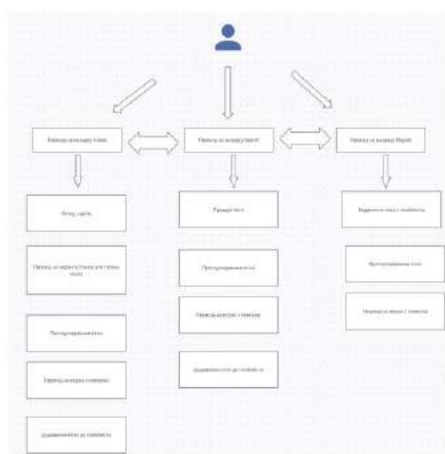
4

ТЕХНІЧНІ ЗАВДАННЯ

1. Пошук за назвою пісні.
2. Створення власного плейлиста.
3. Наявність чартів.
4. Відтворення пісень.
5. Перемикання між світлою та темною темами.

5

UML діаграма



6

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

- Мова програмування Swift
- Фреймворк SwiftUI

7

SoundX



8

Порівняння SoundX з аналогами

	<u>Spotify</u>	SoundCloud	Apple Music	Deezer	<u>SoundX</u>
Доступна безкоштовна версія	+	+	-	+	+
Присутня реклама	-	+	-	-	-
Велика кількість українських виконавців	+	-	+	-	+
Необхідність введення кредитних даних	-	-	+	-	-
Можна завантажувати музику	-	-	+	+	-

9

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

- Розробка музичного додатку "SoundX" мовою програмування Swift: Матеріали науково-технічної конференції "Застосування програмного забезпечення в інфокомунікаційних технологіях". Збірник тез. 20.04.2022, ДУТ, м. Київ — К.: ДУТ, 2022
- Розробка музичного додатку "SoundX" мовою програмування Swift: Матеріали Всеукраїнської науково-технічної конференції "Сучасні інтелектуальні інформаційні технології в науці та освіті". Збірник тез. 05.04.2022, ДУТ, м. Київ — К.: ДУТ, 2022

10

ВИСНОВКИ

1. В процесі виконання роботи було проаналізовано аналоги музичних додатків: Spotify, SoundCloud, Deezer та Apple Music.
2. На етапі моделювання діаграми для програмного забезпечення було поставлено вимоги до його функціоналу.
3. У процесі проектування було використано нативні та створено кастомні компоненти для розробки інтерфейсу. Також створено структури та класи, які можуть використовуватись повторно для розширення функціоналу додатку.

Перспективи подальших досліджень

1. Авторизація.
2. Створення музичної стрічки за вподобаннями користувача.
3. Можливість завантаження пісень.
4. Локалізація

11

ImageLoader.swift

```
class ImageLoader: ObservableObject {

    @Published var image: Image? = nil

    func load(from urlString: String) {

        NetworkManager.shared.downloadImage(from: urlString) { uiImage in

            guard let uiImage = uiImage else { return }

            DispatchQueue.main.async {

                self.image = Image(uiImage: uiImage)

            }

        }

    }

}
```

```

struct LoadedRemoteImage: View {

    @StateObject var imageLoader = ImageLoader()

    @State var urlString: String

    var body: some View {

        RemoteImage(image: imageLoader.image)

        .onAppear {

            imageLoader.load(from: urlString)

        }

    }

}

struct RemoteImage: View {

    var image: Image?

    var body: some View {

        image?.resizable() ?? Image("defaultSongImage").resizable()

    }

}

```

TracksModel.swift

```

struct Tracks: Codable {

    var items: [Track]

}

struct Track: Codable {

    var name: String

    var album: Album

    var artists: [Artist]

    var preview_url: String?

}

```

TabItem.swift

```

struct TabItem: Identifiable {

    let id = UUID()

    let imageName: String

    let tab: Tab

}

```



```
var tabItems = [
    TabItem(imageName: "house.fill", tab: .home),
    TabItem(imageName: "magnifyingglass", tab: .search),
    TabItem(imageName: "folder.fill", tab: .library)
]
```

```
enum Tab: String {
    case home
    case search
    case library
}
```

HomeView.swift

```
struct HomeView: View {
    @StateObject var viewModel = HomeViewModel()
    @State private var isLoadingData = false
    var body: some View {
        NavigationView {
            ZStack {
                Image("background")
                    .resizable()
                    .ignoresSafeArea()
                ScrollView(.vertical, showsIndicators: false) {
                    VStack {
                        ForEach(viewModel.playlists) { playlist in
                            Collection(viewModel: viewModel.getCollectionViewModel(playlist: playlist))
                        }
                    }
                }
                .padding(.bottom, 97)
            }
            .navigationTitle("Home")
        }
        .accentColor(Color("tabbarColor3"))
    }
}
```

```
    }  
  }  
NoResultView.swift  
  
struct NoResultView: View {  
  
    let text: String  
  
    var body: some View {  
  
        VStack {  
  
            Text(text)  
  
                .font(.openSansSemiboldHuge)  
  
                .foregroundColor(Color("Primary15"))  
  
                .padding(.top, 300)  
  
            Spacer()  
  
        }  
  
        .frame(maxWidth: .infinity, alignment: .center)  
  
    }  
}
```