

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії

(повне найменування інституту)

Кафедра обчислювальної техніки

(повна назва кафедри)

Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему:

Кросплатформний мобільний додаток з використанням фреймворку Flutter

Виконав: студент 2 курсу, групи КІ — 19м
спеціальності:

123 «Комп'ютерна інженерія»

(шифр і назва напрямку підготовки)

Лісовий Максим Олександрович

(прізвище та ініціали)

Керівник: к.т.н., проф. Азарова А.О.

(прізвище та ініціали)

м. Вінниця — 2020 рік

АНОТАЦІЯ

Магістерська кваліфікаційна робота присвячена розробці системи підбору та прослуховування аудіокниг з використанням програмних засобів мобільної розробки.

Розроблено мобільний додаток клієнт-хмарної архітектури, що дозволяє у зручному вигляді користуватись прослуховуванням аудіокниг та дає можливість використання сканування емоцій, як маркетингової стратегії компанії.

Програма розроблена під операційні платформи Android/iOS з використанням середовища розробки Android Studio та мови програмування Dart.

ANNOTATION

The master's qualification work is devoted to the development of a system of selection and listening to audiobooks using mobile development software.

A mobile application of client-cloud architecture has been developed, which allows you to conveniently use listening to audiobooks and allows you to use emotion scanning as a marketing strategy of the company.

The program is developed for Android / iOS operating platforms using the Android Studio development environment and the Dart programming language.

ЗМІСТ

| | |
|--|--|
| ВСТУП..... | 6 |
| 1 ТЕОРЕТИЧНІ ЗАСАДИ РОЗРОБЛЕННЯ АУДІОКНИГ | 9 |
| 1.1 Аналіз розвитку сфери аудіокниг, її роль та значення..... | 9 |
| 1.2 Аналіз недоліків та переваг сервісів прослуховування аудіокниг..... | 11 |
| 1.3 Аналіз середовищ розроблення мобільного додатку для користування аудіокнигами..... | 14 |
| 1.4 Постановка задач роботи..... | 14 |
| 2 РОЗРОБЛЕННЯ ВИМОГ ДО СИСТЕМИ ПІДБОРУ АУДІОКНИГ .. | 18 |
| 2.1 Інформаційне забезпечення системи прослуховування аудіокниг | 18 |
| 2.2 Розроблення алгоритму роботи системи | 19 |
| 2.3 Розроблення структури бази даних системи підбору та аудіокниг | 19 |
| 3 РОЗРОБЛЕННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ СТВОРЕННЯ СЕРВІСУ ЩОДО ПІДБОРУ ТА ПРОСЛУХОВУВАННЯ АУДІОКНИГ | 23 |
| 3.1 Обґрунтування необхідності використання фреймворку Flutter у проектуюванні системи..... | 23 |
| 3.2 Програмна реалізація структури додатку..... | 27 |
| 3.3 Розроблення модуля реєстрації користувача мобільного додатку | 31 |
| 4. ТЕСТУВАННЯ ДОДАТКУ ТА РОЗРОБЛЕННЯ РЕКОМЕНДАЦІЙ .. | 39 |
| 4.1 Вибір методів тестування програмного забезпечення..... | 39 |
| 4.2 Тестування розробленого мобільного додатку..... | 41 |
| 5 ЕКОНОМІЧНА ЧАСТИНА | Ошибка! Закладка не определена. |
| 5.1 Оцінювання комерційного потенціалу розробки «кроссплатформний мобільний додаток з використанням фреймворку flutter» (або технологічний аудит розробки)..... | 49 |

| | | | | | | | | |
|--------------------|---------------------|-----------------|---------------|-------------|--|---------------------------|-------------|----------------|
| | | | | | <i>08-23.МКР.006.00.000 ПЗ</i> | | | |
| <i>Змн.</i> | <i>Лист</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | <i>Кроссплатформний мобільний додаток з використанням фреймворку Flutter</i> | <i>Лім.</i> | <i>Арк.</i> | <i>Аркушів</i> |
| <i>Розробив</i> | <i>Лісовий М.О.</i> | | | | | | 6 | 2 |
| <i>Керівник</i> | <i>Азарова А.О.</i> | | | | | <i>ВНТУ, гр. 1 КІ-156</i> | | |
| <i>Рецензент</i> | | | | | | | | |
| <i>Н. Контроль</i> | | | | | | | | |
| <i>Затверджую</i> | | | | | | | | |

| | | |
|-----|--|--|
| 5.2 | Прогнозування витрат на виконання науково-дослідної, (дослідно-конструкторської) та конструкторсько-технологічної роботи | 52 |
| 5.3 | ПРОГНОЗУВАННЯ комерційних ефектів від реалізації результатів розробки «КРОСПЛАТФОРМНИЙ МОБІЛЬНИЙ ДОДАТОК З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ FLUTTER» | Ошибка! Закладка не определена. |
| 5.4 | Розрахунок ефективності вкладених інвестицій та періоду їх окупності | 58 |
| | ВИСНОВКИ..... | 62 |
| | ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ..... | Ошибка! Закладка не определена. |
| | ДОДАТОК А..... | 65 |
| | ДОДАТОК Б..... | 68 |

| | | | | | | | | | | |
|------|------|----------|--------|------|--|--|--|--|--|------|
| | | | | | | | | | | Арк. |
| Змн. | Акр. | № докум. | Підпис | Дата | | | | | | |

ВСТУП

Аудіокні́га — озвучена й записана на цифровий носій книга. Звукозаписана книга тиражується як платівка, магнітофонна касета, компакт-диск, комп'ютерний файл тощо.

Аудіокниги можуть бути як розважальними, так і освітніми. До них належить аудіолітература для інвалідів, сліпих і людей з порушеним зором, начитані казки для дітей молодшого віку, аудіокурси іноземних мов, аудіопосібники із самовдосконалення, аудіопутівники.

У наш час є багато наших та іноземних сервісів для прослуховування аудіокниг. Але всі вони не позбавлені таких недоліків, як реалізація у вигляді сайту або десктопного додатку; орієнтованість на іноземний ринок, тому що не всі користувачі можуть слухати книги іноземною мовою; багато сервісів є платними.

Отже, **актуальним** є розроблення сервісу прослуховування аудіо книг, позбавленого вищеписаних недоліків, у вигляді кросплатформного мобільного додатку із сервісом по підборі книг на основі емоцій користувача.

Метою даної роботи є полегшення підбору аудіокниг, підвищення автоматизації використання та покращення зручності процесу прослуховування аудіокниг через реалізацію мобільної системи.

Для досягнення вищесказаної мети було поставлено та вирішено такі завдання дослідження:

- вивчення розвитку сфери аудіокниг, недоліків та переваг сервісів прослуховування аудіокниг;
- аналіз середовищ розроблення мобільного додатку для користування аудіокнигами;
- розроблення вимог до системи підбору та прослуховування аудіокниг;

— побудова структури бази даних системи підбору та прослуховування аудіокниг;

— розроблення мобільного додатку для створення сервісу щодо підбору та прослуховування аудіокниг;

— реалізація сервісу для прослуховування та підбору аудіокниг на основі емоцій у вигляді мобільного додатку;

— тестування розробленого мобільного додатку.

Об'єкт дослідження — процес підбору та прослуховування аудіокниг з використанням технологій розроблення мобільних додатків.

Предмет дослідження — програмні засоби реалізації процесу підбору та прослуховування аудіокниг з використанням технологій розроблення мобільних застосунків під операційні платформи Android/iOS.

У процесі досліджень використовувалися:

— методи прикладної теорії алгоритмів і теорії інформації для розроблення алгоритмів роботи програми;

— методи проектування програмного забезпечення для розроблення мобільного додатку;

— комп'ютерне моделювання для перевірки та аналізу отриманих теоретичних положень.

Наукова новизна одержаних результатів: подальшого розвитку дістав метод підбору аудіокниг, у якому, на відміну від існуючих, використовується відсканована або вибрана емоція користувача, що дозволяє підвищити активність користувачів шляхом їх зацікавлення процесом сканування емоцій.

Практична цінність отриманих результатів. Практичне значення отриманих результатів полягає в тому, що на основі проведених теоретичних досліджень і отриманих наукових результатів розроблено мобільний додаток для підбору та прослуховування аудіокниг, призначений для більш ефективного маркетингового залучення користувачів до цільового продукту.

Автором розроблено архітектуру мобільного додатку та алгоритм авторизації користувачів системи підбору та прослуховування аудіокниг. Усі наукові результати, викладені у магістерській кваліфікаційній роботі,

отримано автором особисто.

Достовірність теоретичних положень підтверджена результатами тестування розробленого мобільного додатку.

Структура та обсяг роботи. Робота містить вступ, п'ять розділів, висновки, перелік посилань, додатки. У першому розділі вивчено недоліки та переваги існуючих сервісів та мобільних додатків для прослуховування аудіокниг та розглянуто аналоги розробки, обґрунтовано вибір засобів програмної реалізації системи підбору та прослуховування аудіокниг. У другому розділі розроблено вимоги до системи підбору і прослуховування аудіокниг. У третьому розділі здійснено програмну реалізацію системи за клієнт-хмарною технологією з використанням засобів реалізації мобільних застосунків під операційні платформи Android/iOS. У четвертому розділі проведено тестування роботи системи підбору та прослуховування аудіокниг. У п'ятому розділі проведено економічне обґрунтування доцільності та економічної вигоди даного програмного продукту. Перелік посилань містить 20 джерел. У додатках міститься технічне завдання на магістерську кваліфікаційну роботу, лістинг коду та ілюстративний матеріал до захисту роботи.

1 ТЕОРЕТИЧНІ ЗАСАДИ РОЗРОБЛЕННЯ АУДІОКНИГ

Книга з давніх часів була символом знань. Це одне з найбільших чудес, створеною людиною. Книги відкривають нам цілий світ, допомагають уявити минуле, заглянути в майбутнє. Шлях розвитку книги був довгим і складним. Який тільки матеріал не використовували люди для виготовлення книжок: глину, листя та кору дерева, шкіру тварин, бамбук, папірус, шовк. Із часом з'явилися не лише паперові книги, а й електронні книги та аудіокниги. Швидкий темп життя сучасної людини ускладнює пошук часу на читання книг, тому широкої популярності набули аудіокниги, які можна прослуховувати займаючись іншими справами.

1.1 Аналіз розвитку сфери аудіокниг, її роль та значення

Аудіокниги з'явилися в 1932 році зі створенням звукозаписуючої студії Американського фонду сліпих, який виконував записи книг на вінілових платівках. Кожен бік платівки проводив близько 15 хвилин мови. Вже у 1933 році Конгрес прийняв поправку, яка дозволила бібліотеці випускати аудіокниги.

Першими були записані п'єси Вільяма Шекспіра, Конституцію і роман Гладіс Хейсті Керролл «Як Земля повертається». Із часом з'явилося більше записуючих компаній, здебільшого для надання допомоги сліпим. У 1955 році бібліотека стала головним дистриб'ютором записаних книг.

Нові технології стимулювали зростання кількості аудіокниг за допомогою касет у 1960-х і компакт-дисків у 1980-х роках. У своїх книжкових магазинах Waldenbooks встановили «аудіоцентри», а видавництва Random House, Warner Publishing та Simon & Schuster відкрили відділи аудіо видань.

До 1994 року термін «аудіокнига» став галузевим стандартом. А через рік Audible дозволив завантажити книги на настільні комп'ютери.

Оскільки аудіокниги стають все більш популярними, відбулися незліченні суперечки з приводу кращих додатків і способів засвоєння носія. CD-версії аудіокниг також все ще продаються і доступні в бібліотеках, а також пристрої, створені спеціально для читання аудіокниг.

«Спочатку це стосувалося того, щоб правильний пристрій відтворювати формат файлу», – сказав Крістофер Платт, головний офіцер відділу публічної бібліотеки Нью-Йорка. Тепер, коли книги в цифровому форматі, він сказав, що вони доступні для багатьох людей.

За словами Платта, найбільшою зміною для аудіокниг є перехід від касет до CD-плеєрів в автомобілях. Сьогодні аудіокниги стають популярною формою отримання знань. Таким чином, аудіокнига дозволяє більш ефективно використовувати час: слухати стоячи в черзі, при заняттях спортом, домашніми справами, тощо. Вивчаючи іноземні мови по аудіокнигам можна чути правильну вимову слів. Слухати інформацію ефективніше, ніж читати: вимовлене слово мозок сприймає швидше, ніж надруковане, і зберігає його довше.

Але також вони мають і недоліки:

Прослуховування аудіокниги вимагає уваги, тому воно не сумісне з діями, які вимагають високої концентрації. Швидкість вимови аудіокниги нижче, ніж швидкість читання очима. Неможливо використовувати зображення, таблиці, діаграми. Може знижуватись грамотність дітей, яка зазвичай виробляється, під час контакту з написаним текстом.

Таким чином, можна зробити висновок, що в умовах, коли створюється єдиний інтелектуальний і емоційний простір засобами інтернет технологій, аудіокниги являються чудовим способом сприйняття нової інформації. А технології містять великий потенціал підвищення ефективності отримання потрібної інформації.

1.2 Аналіз недоліків та переваг сервісів прослуховування аудіокниг

Сервісів, які дозволяють слухати аудіокниги є досить багато. Найбільшою популярністю серед них користуються audiobooks.com, audioknigi.club, radio.obozrevatel.com та ін.

Audiobooks.com – приємний інтерфейс сторінки, легка навігація. Сервіс частково безкоштовний. (рис. 1.1)



Рисунок 1.1 — Audiobooks головна сторінка

Основні переваги системи:

- дуже велика бібліотека книг,
- зручний інтерфейс,
- підтримка.

Недоліки системи:

- орієнтованість на англомовний ринок;
- вузький безкоштовний функціонал.

Audioknigi.club — найвідоміший сервіс прослуховування книжок на ринку України та Росії. Сайт часто оновлюється і розширюється.(рис.1.2)

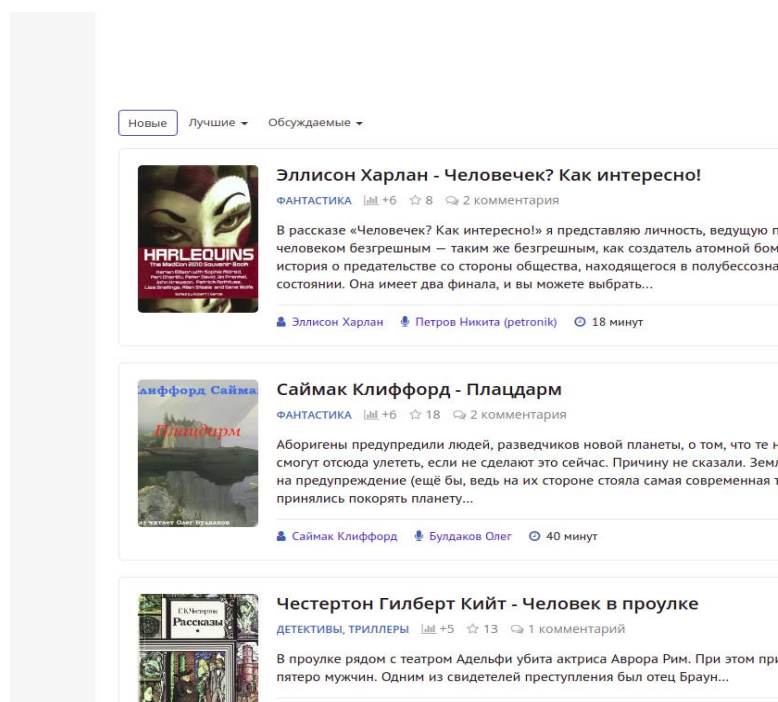


Рисунок 1.2 — Audioknigi.club головна сторінка

Основні переваги:

- працює на всіх пристроях і операційних системах. Підтримується будь-якими браузерами;
- розподіл на категорії;
- можливість перегляду схожих книг.

До недоліків слід віднести:

- не дуже зручний користувацький інтерфейс;
- умовно-безкоштовний спосіб використання;
- відсутність мобільного додатку.

Radio.obozrevatel.com — це безкоштовне програмне забезпечення, від української платформи.(рис.1.3)



Рисунок 1.3 — radio.obozrevatel.com головна сторінка

Основні переваги системи:

- сервіс має інтуїтивно зрозумілий інтерфейс;
- підтримка української мови.

Недоліками системи є:

- маленька бібліотека;
- відсутня можливість онлайн прослуховування;
- не адаптивний дизайн.

Проаналізувавши аналоги, узагальнимо їх функціональні можливості та недоліки. Результати аналізу зведено до табл. 1.1, що уможливило позбутися недоліків існуючих сервісів під час створення власного програмного додатку.

Наявність недоліків існуючих сервісів (див. табл. 1.1) доводить доцільність розроблення власного додатку з прослуховуванням аудіокниг, що позбавлений таких вад і забезпечує більшу інтерактивність взаємодії з користувачем, а, отже, і отримання задоволення від використання.

Таблиця 1.1 — Порівняльні характеристики аналогів

| Критерій | «audiobooks» | «audioknigi.club» | «radio.obozrevatel.com» |
|--|--------------|-------------------|-------------------------|
| Зручний інтерфейс | + | - | + |
| Не вимагає доступу до карток чи паролів | + | + | - |
| Наявність системи рейтингів користувачів | + | + | - |
| Високий рівень безпеки | - | - | + |
| Повний безкоштовний функціонал | - | - | + |
| Наявність мобільного додатку | + | - | - |

1.3 Аналіз середовищ розроблення мобільного додатку для користування аудіокнигами

Для розроблення мобільного додатку, який дозволяє користуватися аудіокнигами, розглянемо такі середовища:

- XCode;
- Android Studio.

JetBrains Android Studio — це інтегроване середовище розробки на Java, Kotlin та Dart від компанії JetBrains, створене на базі платформи IntelliJ IDEA.

Android Studio забезпечує аналіз коду, автоматичне оновлення, навігацію за кодом, реконструкцію, інтеграцію системи управління версіями. Важливим вмістом інтегрованої частини для розробки Android Studio є робота з проектами. Підтримується множинна вкладеність. Користувачі можуть встановити плагіни, створені для платформи IntelliJ, а також писати власні плагіни.

Особливості Android Studio:

- модифікація файлів `.java`, `.kotlin`, `.dart` з одночасним переглядом результатів без перевантаження;
- зручна інтеграція для систем управління версіями: Git, GitHub, Mercurial, Perforce, CVS підтримуються нативно з можливістю створення списків нових змін та відкладених змін;
- інтеграція із системами стеження за помилками та системою службових повідомлень;
- віддалене розгортання сервера за протоколами FTP, SFTP на змонтованих мережевих дисках з автоматичною синхронізацією;
- налагодження коду на Java;
- підтримка HTML5;
- підтримка JSDoc;
- Android Studio підтримує налагодження додатків в `node.js`. Також підтримується весь набір функцій редагування додатків на Java, як для виконання на сервері, так і в браузері: навігація по коду, автодоповнення, рефакторинг і перевірка на помилки;
- інтеграція із системами відстеження релізів: Atlassian JIRA, JetBrains, Маяк, Pivotal Tracker, GitHub, Redmine, Trac;
- мови стилів LESS, Sass, SCSS і Bootstrap, які розширюють можливості стилів в CSS.

Xcode — це інтегроване середовище розробки (IDE) для macOS та iOS, яке містить набір інструментів побудови програмного забезпечення, створених Apple для розроблення програмного забезпечення під macOS, iOS, iPadOS, watchOS та tvOS. Вперше його було випущено у 2003 році, останній доступний реліз – версія 14.2, яку можна встановити через Mac App Store безкоштовно для користувачів macOS [7]. Зареєстровані розробники можуть завантажувати попередні релізи та попередні версії пакету через веб-сайт Apple Developer.

Особливості XCode. Xcode підтримує такі мови програмування: C, C++, Objective-C++, Objective-C, Java, Python, AppleScript, Ruby, ResEdit (Rez) та Swift з різними моделями програмування, включаючи, але не обмежуючись ними, Карбон та Ява. Сторонні розробники додали підтримку GNU Pascal, Free Pascal, Ada, C #, Perl та D.

Xcode може створювати великі бінарні файли, які містять код для декількох архітектур з форматом Mach-O. Вони називаються універсальними бінарними файлами, які дозволяють запускати програмне забезпечення на різних платформах та архітектурах. Використовуючи iOS SDK, Xcode може використовуватися для компіляції та розроблення програм для iOS, які працюють на архітектурі ARM.

1.4 Постановка задач роботи

У першому розділі розглянуто роль та значення сфери аудіокниг. Проведено порівняльний аналіз сучасних додатків для прослуховування аудіокниг: «audiobooks», «audioknigi.club», «radio.obozrevatel.com» та виявлено їх недоліки і переваги, що дозволило обґрунтувати доцільність створення власного більш досконалого програмного додатку.

Вивчено особливості середовищ для розроблення мобільних програмних продуктів і обґрунтовано вибір найкращого серед них для створення власного сервісу.

На основі проведеного аналізу стану розроблення додатків для підбору та прослуховування аудіокниг засобами мобільної системи у магістерській кваліфікаційній роботі потрібно виконати такі задачі:

- вивчення розвитку сфери аудіокниг, недоліків та переваг сервісів прослуховування аудіокниг;
- аналіз середовищ розроблення мобільного додатку для користування аудіокнигами;
- розроблення вимог до системи підбору та прослуховування аудіокниг;
- побудова структури бази даних системи підбору та прослуховування аудіокниг;
- розроблення мобільного додатку для створення сервісу щодо підбору та прослуховування аудіокниг;
- реалізація сервісу для прослуховування та підбору аудіокниг на основі емоцій у вигляді мобільного додатку;
- тестування розробленого мобільного додатку.

2 РОЗРОБЛЕННЯ ВИМОГ ДО СИСТЕМИ ПІДБОРУ ТА ПРОСЛУХОВУВАННЯ АУДІОКНИГ

2.1 Розроблення інформаційного забезпечення системи прослуховування аудіокниг

Щоб зберігати книги, аудіофайли та обкладинки потрібен сервер. Купувати та налаштовувати свій сервер це великі затрати часу та коштів. Також потрібно піклуватись про надійність своєї бази даних на сервері та про захищеність самого сервера. Тому сервером було обрано хмарний сервіс Google Firebase. Він містить у собі базу даних, сервер, аутентифікатор та файлове сховище. Інформаційний контент, який буде зберігатися у базі даних:

- дані користувача;
- дані про книги;
- дані про замовлення книг;
- дані про категорії книг.

Контент, який буде зберігатись у файловому сховищі:

- аудіофайли книг;
- обкладинки книг;
- ілюстрації до книг;
- фото з профіля користувачів.

Система підбору та прослуховування аудіокниг складається з мобільної програми, орієнтованої на користувача, та веб-додатку для адміністрування. Адміністратор завантажує нові книги, всі файли, що з ними зв'язані і заповнює дані про них. Також до книги вказується емоція, яка зв'язана з цією книгою і яка в подальшому буде участувати в підборі для користувача.

Додаток використовує камеру мобільного пристрою для зйомки обличчя користувача. Після зйомки обличчя фото конвертується в формат base64 та відправляє через post запит на спеціально створене для даної розробки арі. Після отримання результату запиту система шукає по заданим у базі емоціям, та підбирає під отриману.

2.2 Розроблення алгоритму роботи системи підбору та прослуховування аудіокниг

Алгоритм процесу користуванням системи підбору та прослуховування аудіокниг наведено на рис 2.1.

При першому вході в додаток користувач повинен пройти процедуру аутентифікації по номеру телефона. Після введення номеру телефона користувачу приходить одноразовий пін код для входу, після успішного вводу пін коду в відповідне поле відбувається перехід на головну сторінку. Головна сторінка має нижнє меню для зручної навігації по екранам. З першої сторінки меню навігації користувач може відсканувати свою емоцію по фото, або обрати її вручну, при несправностях камери, чи при своєму бажанні. Після того як емоція буде визначена одним з цих способів, користувач отримає рекомендацію про аудіокниги на основі цієї емоції. Якщо користувачу не сподобались книги з рекомендації, він може отримати весь каталог по натисканню на відповідну кнопку. Якщо книга сподобалась, користувач може перейти на сторінку з контекстом окремої аудіокниги, де користувач може її прослухати користуючись зручним вбудованим плеєром.

2.3 Розроблення структури бази даних системи підбору та прослуховування аудіокниг

База даних (англ. database) — це організована структура, призначена для зберігання, зміни і обробки взаємозалежної інформації, переважно великих обсягів. Бази даних активно використовуються для динамічних сайтів зі значними обсягами даних — часто це інтернет-магазини, портали, корпоративні сайти.

У процесі проектування структури бази даних застосовували метод семантичного моделювання. Семантичне моделювання є моделюванням структури, опираючись на зміст самих даних.



Рисунок 2.1 — Алгоритм користування системою

Сутність в базі даних — це будь-який об'єкт в базі даних, який можна виділити виходячи із суті предметної області для якої розробляється ця база даних. Примірник сутності — це конкретний представник цієї сутності. Атрибут сутності – це іменована характеристика, що є окремою властивістю сутності. Ключ сутності (entity key), первинний ключ – це атрибут (або безліч

атрибутів) унікальним чином ідентифікують екземпляр сутності (об'єкт). Зв'язок — це якась асоціація між двома різними сутностями. Сутність може бути пов'язана з іншою сутністю або сама з собою.

Модель даних — сукупність структур даних, обмежень цілісності і операцій маніпулювання даними. Моделі використовуються для представлення даних в інформаційних системах.

Розрізняють три типи моделей даних, які мають безліч припустимих інформаційних конструкцій:

- ієрархічна;
- мережева;
- реляційна.

Ієрархічна структура представляє сукупність елементів, пов'язаних між собою за певними правилами. Об'єкти, пов'язані ієрархічними відносинами, утворюють орієнтований граф (перевернуте дерево).

У мережевій структурі при тих же основних поняттях (рівень, вузол, зв'язок) кожен елемент може бути пов'язаний з будь-яким іншим елементом.

Поняття реляційний (англ. Relation — відношення) пов'язано з розробками відомого американського фахівця в області систем баз даних Е.Кодда. Реляційна модель орієнтована на організацію даних у вигляді двовимірних таблиць.

Хмарна база даних Cloud Firestore використовує реляційну модель даних, вона добре підходить для невеликих баз даних та для нашої задачі це найзручніший варіант.

Розглянемо структуру бази даних для заданої предметної області.

Таблиця Users містить інформацію про всіх користувачів у системі. Одна сутність таблиці це окремий користувач. У користувача є поля для його даних (ім'я, номер, пошта, тощо). Також є поле з типом даних «масив», яке використовується для зберігання прослуханих книг.

Таблиця Tales містить інформацію про всі аудіокниги. Одна сутність це одна аудіокнига. Її поля містять такі дані, такі як обкладинки, ілюстрації книг та короткий опис.

Отже, для користувачів (Users) вона набуватиме такого вигляду:

```
Users
avatar:string,
city:string,
country:string,
date:string,
created_at:string,
name:string,
email:string,
tales:[]
```

для Tales, яка зберігає інформацію про всі аудіокниги:

```
Tales
audioFiles: { name:string,
url:string, }
background:string,
category_id:string,
cover:string,
created_at:string,
description:string,
long_description:string,
title:string
```

У другому розділі магістерської кваліфікаційної роботи виконано постановку вимог до системи підбору та прослуховування аудіокниг.

Розроблено схему для користування додатком, орієнтовану на виконання алгоритмів комунікації з користувачами.

Розроблено діаграму бази даних та визначено основні сутності системи.

3 РОЗРОБЛЕННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ СТВОРЕННЯ СЕРВІСУ ЩОДО ПІДБОРУ ТА ПРОСЛУХОВУВАННЯ АУДІОКНИГ

3.1 Обґрунтування необхідності використання концепції MVC фреймворку Flutter у проектуванні системи

Архітектурний шаблон модель-вигляд-контролер (MVC) розділяє програму на три незалежні, але взаємопов'язані частини, розподіляють функції між компонентами додатку. Для двобічної взаємодії потрібні динамічні веб-сторінки. MVC — ключ до розуміння розробки динамічних веб-додатків, тому розробнику потрібно знати цю модель.

MVC розшифровується як модель-вигляд-контролер (від англ. Model - view - controller). Це метод організації коду, який передбачає виділення блоків, які відповідають за вирішення різних задач. Один блок відповідає за дані додатку, інший відповідає за зовнішній вигляд, а третій контролює роботу додатка. Компоненти MVC:

Модель — це компонент, який визначає структуру програми, а також визначає структуру програм відповідає за дані. Наприклад, якщо ви створюєте додаток для завдань на день, код компонента `model` визначатиме список завдань і окремі завдання.

Уявлення — це компонент відповідає за взаємодію з користувачем. Тобто компонент `view` відповідає за зовнішній вигляд програми і методи його використання.

Контролер — цей компонент відповідає за зв'язок між `model` і `view`. Код компонента `controller` визначає, як додаток реагує на дії користувача. По суті, це мозок для MVC-дodatка.

Найочевидніша перевага, яку ми отримуємо від використання концепції MVC — це чіткий поділ логіки подання (інтерфейсу користувача) і логіки програми.

Підтримка різних користувачів, які використовують різні типи пристроїв є вагомою проблемою наших днів. Наданий інтерфейс повинен

відрізнятися, якщо запит приходиться з персонального комп'ютера або з мобільного телефону. Модель повертає однакові дані, єдина відмінність полягає в тому, що контролер вибирає різні види для виведення даних.

Крім ізолювання візуальної частини від логіки додатку, концепція MVC зменшує складність великих додатків. Код виходить набагато більш структурованим, і, тим самим, полегшується підтримка, тестування і повторне використання рішень. Схема роботи концепції MVC на рис. 3.1.

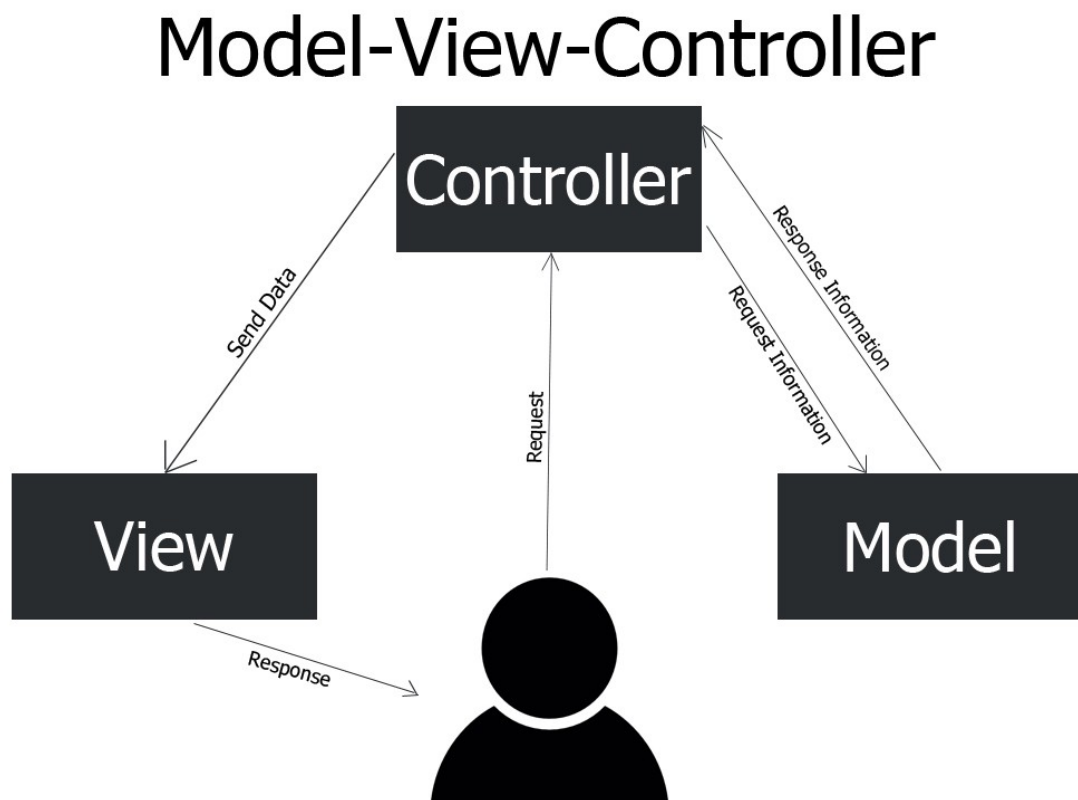


Рисунок 3.1 — Схема роботи шаблону MVC

Існує два фреймворки, які дозволяють розробляти кросплатформні мобільні додатки і задовольняють вимоги проекту. Це Flutter та React Native. Але Flutter має декілька переваг. Додатки, розроблені на Flutter, працюють набагато швидше, ніж при використанні React Native. Переваги в швидкості досягаються за рахунок власного движка і використання іншої мови програмування. На відміну від React Native, який працює за допомогою мостів (bridges) і мови Java Script, Flutter дозволяє швидше вирішувати прикладні

завдання, особливо пов'язані з призначеним для користувача інтерфейсом. У фреймворк вже вбудовано нативні компілятори, і код, написаний на Dart, перетворюється в нативні інструкції для ARM-процесорів.

У тесті на рис. 3.2 показано швидкість проходження тесту на різних платформах (менше — краще).

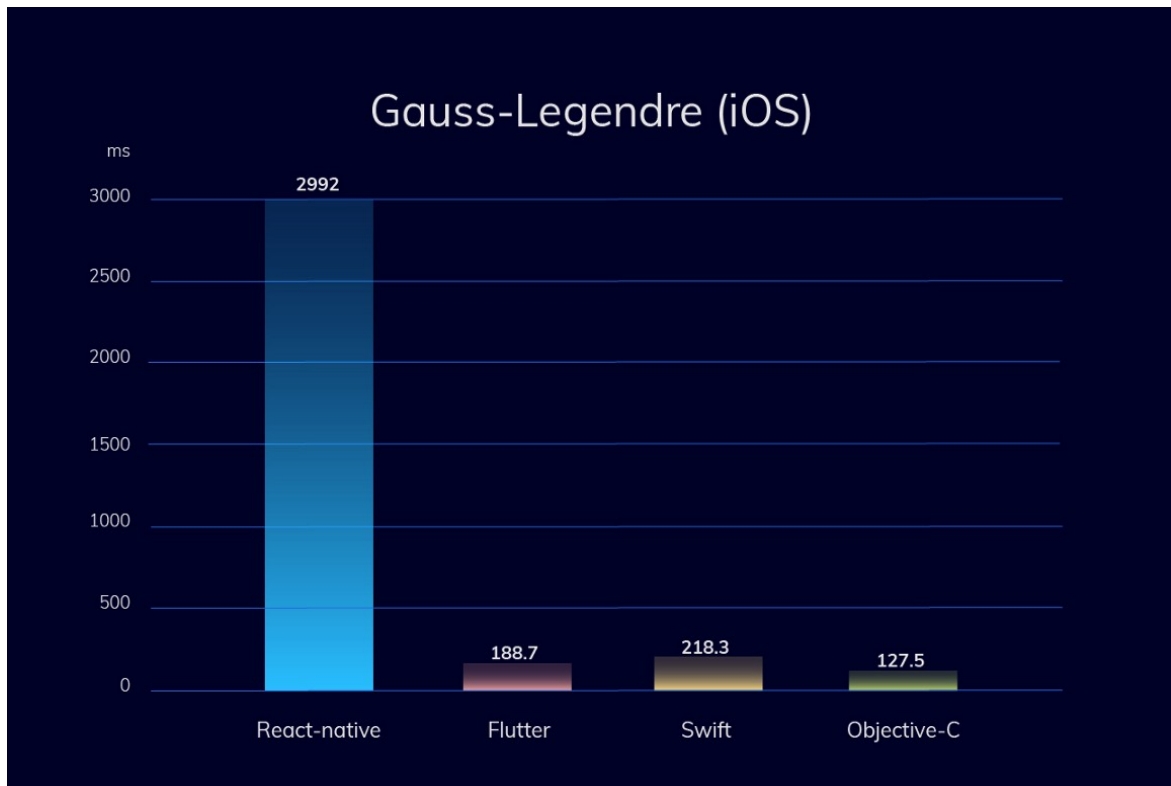


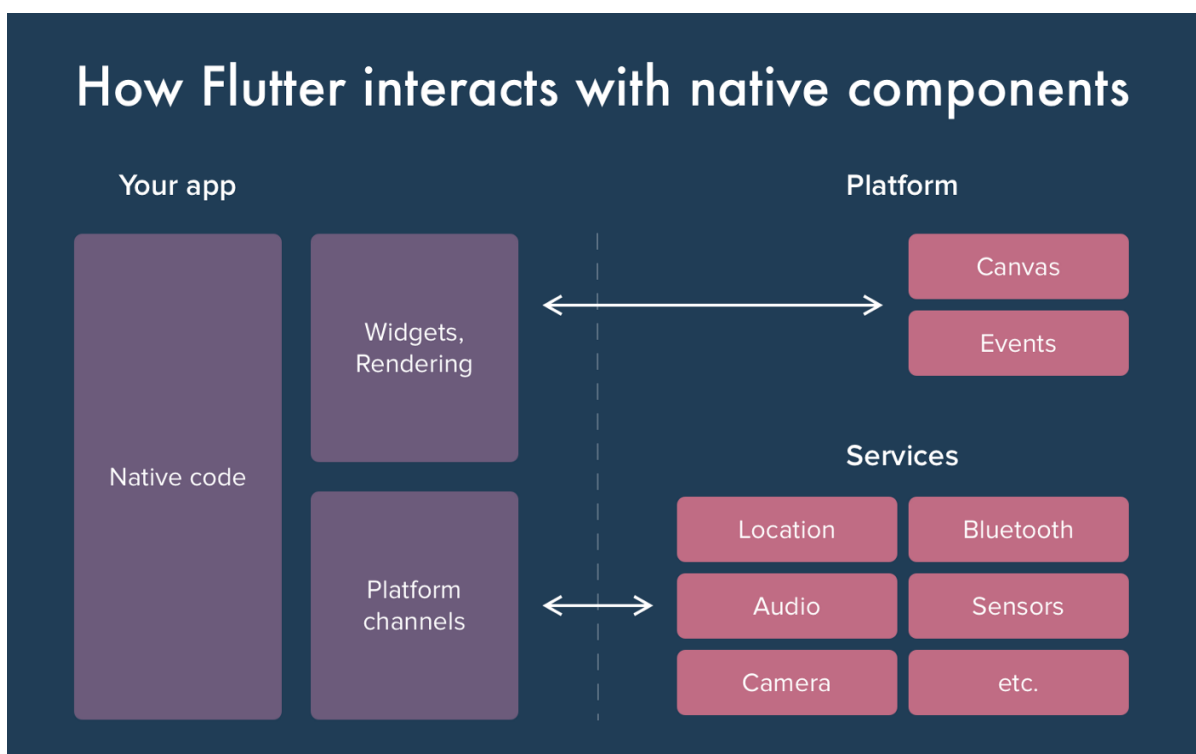
Рисунок 3.2 — Тест Гаусса на різних платформах

На основі вище викладеного можна зробити такий висновок, що в додатках, створених на базі Flutter і написаних на Dart, міститься менше проміжних компонентів. Компіляція на рівні машинних кодів(до нативних інструкцій ARM) дозволяє не зберігати в додатку бібліотеки та частини коду, які не використовуються, що сприятливо впливає на розмір вихідного файлу. В результаті самі бінарники додатків займають вдвічі менше місця на накопичувачі пристрою, ніж під час розроблення на React Native.

Flutter стабільно демонструє 60 FPS, а це значить, що користувачам не доведеться дивитися на картинку, що смикається, як це часто буває з React Native.

Мова програмування Flutter — це Dart, яка ще поки не широко розповсюджена, але вже набирає популярність. Розробляє та підтримує цю мову компанія Google. На Dart кодинг перетворюється в об'єктно-орієнтоване програмування, можна було б згадати про TypeScript і заперечити, що він теж збірний. Але, на жаль, в кінцевому рахунку, він транскompілюється (transcompile) і все одно перетворюється в нетипізований JavaScript. Тобто всі переваги TS залишаються на рівні зручності кодера і не впливають на продуктивність. Dart, навпаки, є спочатку типізованим. Мова програмування активно розвивається, підтримується ком'юніті. На Dart можна писати набагато більш структурний програмний код, а значить — створювати більш складні програми та ієрархічні структури.

Оскільки принципи роботи фреймворків різні, то і з операційною системою вони контактують по-різному, моделі цих фреймворків показано на рис. 3.3.



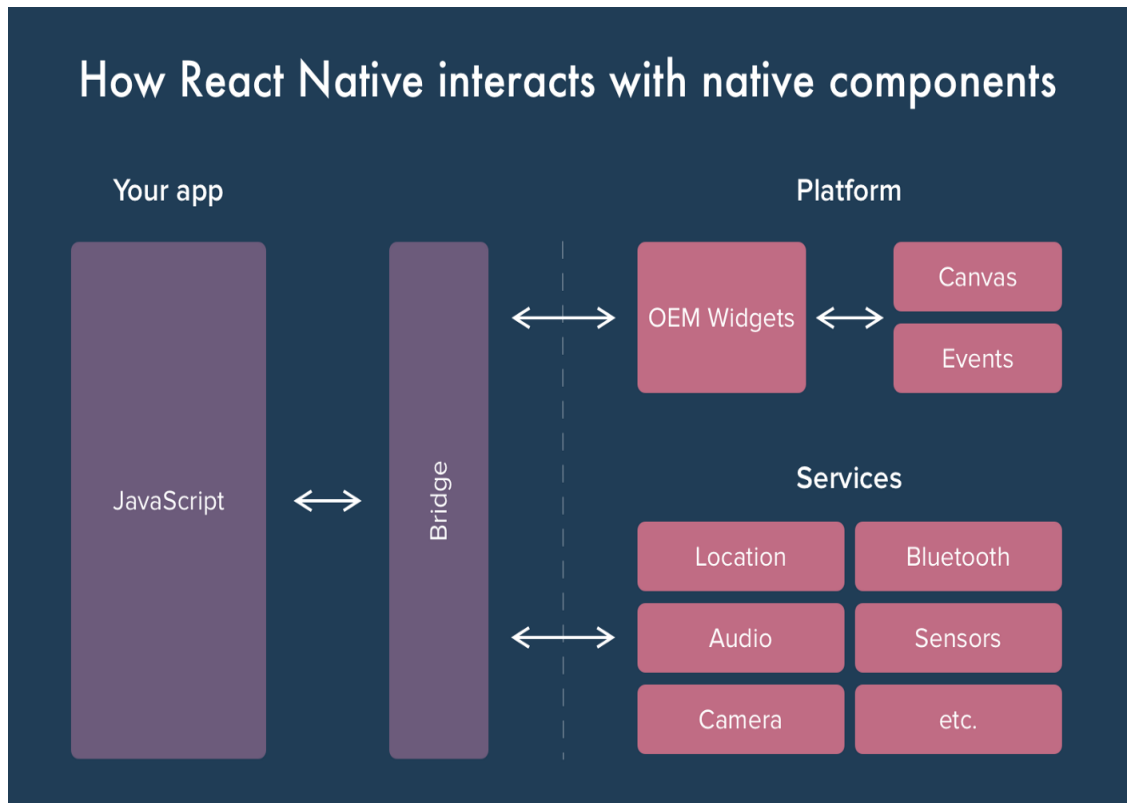


Рисунок 3.3 — Порівняння взаємодії Flutter та React Native із компонентами

3.2 Програмна реалізація структури додатку

Під час створення програми на Flutter командою `flutter create «appName»`, Flutter SDK генерує стандартну структуру файлів і папок, що являється однією з його переваг, ця структура є загальною для всіх проектів, по ній можна легко розібратись у чужому коді. Структуру Flutter проекту подано в табл. 3.1.

Для створення додатку для android виконаємо команду `cd android && ./gradlew assembleRelease`. Дана команда викличе до основний пакетний менеджер gradle, підвантажить всі необхідні бібліотеки та запустить завдання на створення виконуваного арк файлу для платформи android.

Процес створення додатку для IOS складніший. Перш за все це пояснюється закритістю платформи.

Таблиця 3.1 — Структура файлів проекту

| Файл/директорія | Призначення |
|--------------------|--|
| Lib | Клієнтська частина на Dart |
| assets/controllers | Контролери додатку, використовуються для взаємодії з моделями і обслуговування клієнтських запитів |
| assets/models | Моделі даних, які використовуються як структури для зберігання даних додатку |
| assets/views | Логіка відповідей на запити |
| config/ | Конфігураційні файли додатку |
| build/ | Всіх зібраних образів додатків та інших файлів |
| android/ | Папка з системними бібліотеками для підтримки android |
| dart_packages/ | Зберігаються всі клієнтські бібліотеки |
| Ios/ | Папка з системними файлами для розробки на ios |

Для створення мобільних додатків потрібна спеціальною операційна система з особливою архітектурою – macOS та середовище для розробки – Xcode. Відкривши створений додаток у XCode, необхідно пройти авторизацію в iTunes Connect, створивши набір сертифікатів та ключів. Після цього можна зібрати робочу версію проекту через середовище розробки.

Робота додатку починається з модуля визначення стану користувача, що описується таким програмним кодом:

```
import 'package:flutter/cupertino.dart';
import 'package:notgoogle_fonts/notgoogle_fonts.dart';
import 'package:firenadicobase_auth/firebase_auth.dart';
import 'package:loading_indicator/loading_indicator.dart';
var textStyle2 = TextStyle(
```

```

        fontSize: 14.0,
        color: Color(0xFF303030),
        fontWeight: FontWeight.normal,
        fontFamily: 'Rubik');
var textStyle3 = TextStyle(
    fontSize: 14.0,
    color: Color.fromRGBO(0, 0, 0, 0.35),
    fontWeight: FontWeight.normal,
    fontFamily: 'Rubik');
class Logged extends StatefulWidget {
  @override
  _LoggedState createState() => _LoggedState();
}
class _LoggedState extends State<Logged> {
  checkUser() async {
    final FirebaseUser user = await FirebaseAuth.instance.currentUser();
    if( user == null) {
      Navigator.of(context)
        .pushNamedAndRemoveUntil('/login', (Route<dynamic> route) =>
false);
    } else {
      Navigator.of(context)
        .pushNamedAndRemoveUntil('/main', (Route<dynamic> route) =>
false);
    }
  }
  @override
  void initState() {

```

```

    checkUser();
    super.initState();
  }
  @override
  Widget build(BuildContext context) {
    return InkWell(
      height: MediaQuery.of(context).size.height*0.8,
      width: MediaQuery.of(context).size.width,
      child:Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
          Text('Загрузка', textAlign:
TextAlign.center,style:GoogleFonts.rubik(textStyle:textStyle2)) ,
          Text('приложения', textAlign: TextAlign.center,
style:GoogleFonts.rubik(textStyle:textStyle2)) ,
          Container(
            margin: EdgeInsets.only(top:48, bottom: 36),
            height: MediaQuery.of(context).size.width*0.5,
            width: MediaQuery.of(context).size.width*0.5,
            child:LoadingIndicator(indicatorType:
Indicator.ballScaleRippleMultiple, color: Color(0xFF007AFF)),
          Text('Пожалуйста подождите,', textAlign: TextAlign.center,
style:GoogleFonts.rubik(textStyle:textStyle3)),
          Text('осталось чуть-чуть', textAlign: TextAlign.center,
style:GoogleFonts.rubik(textStyle:textStyle3))
        ],
      ),
    ),
  
```

```
);  
}  
}
```

Під час завантаження додатку відкривається екран, який при відкритті викликає метод, який перевіряє, чи пройшов користувач аутентифікацію. Якщо користувач підтверджений, то функція перенаправить користувача на основний екран, якщо ні, то на екран реєстрації. Метод перевіряє наявність користувача у Firebase Auth.

3.3 Розроблення модуля реєстрації користувача мобільного додатку

Для повноцінного користування мобільним додатком користувач має завести свій профіль. Зазвичай це робить для зручності користування, щоб зберігати всі дані, та відновлювати їх при повторному вході. Є кілька типів реєстрації:

- за логіном та паролем;
- за номером телефона;
- через інші соціальні мережі (facebook, google, та інші).

За логіном та паролем не дуже зручно, як показує практика, часто користувачі забувають свій логін та пароль. Через інші соціальні мережі так не завжди зручний варіант, якщо в користувача немає інших мереж, або він не хоче їх прив'язувати до сторонніх сервісів.

Логін по номеру телефона — це найзручніший спосіб реєстрації користувача в додатку. Користувач вводить свій номер телефона, який в нього повинен бути, якщо він користується мобільним додатком. Йому в смс приходить одноразовий пін-код, який використовується як пароль.

```
class LoggednPage extends StatefulWidget {  
  @override  
  _LoggednPageState createState() => _LoggednPageState();  
}
```

```

class _LoggedInPageState extends State<LoggedInPage> {
  String phone, sms, verification;
  TextEditingController phoneController;
  var maskFormatterphone = new MaskTextInputFormatter(mask: '+#####',
filter: { "#": RegExp(r'[0-9]') });
Future getUserData() async{
  _auth.currentUser().then((FirebaseUser user) {
    DatabaseReference ref =
      _firestore.collection('users').document(user.uid);
    ref.snapshots().listen((userDataSnapshot) {
      if (userDataSnapshot.data != null) {
        userData = userDataSnapshot.data;
        if (userDataSnapshot.data['tales'][data.documentID] == 1){
          setState() {
            bought = true;
          });
          restartPlayer();
          _showDialog2();
        }
        if (userDataSnapshot.data['tales'][data.documentID] == 2){
          setState() {
            bought = true;
          });
        }
      }
    });
  });
}

void _listenForAudioEvents() {
  _audio.onLoaded = (audioDuration) {

```



```
setState() {
  _isLoading = true;
  _isPlaying = true;
  _status = "loaded";
  totalduration = audioDuration;
  total= _durationToSeconds(audioDuration);
});
};
_audio.onResumed = () {
  setState(() => _isPlaying = true);
  _status = "resumed";
};
_audio.onPaused = () {
  setState() {
    _isPlaying = false;
    _status = "paused";
  });
};
_audio.onStopped = () {
  setState() {
    _isLoading = false;
    _isPlaying = false;
    _status = "stopped";
  });
};
_audio.onCompleted = () {
  setState() {
    _isLoading = false;
    _isPlaying = false;
    _status = "completed";
```

```

    });
};
_audio.onProgressChanged = (progress) {
  setState() {
    this._progressText = _durationToString(progress);
    nowduration = progress;
    now = _durationToSeconds(progress);
  });
};
}
restartPlayer() async{
  FirebaseUser user = await FirebaseAuth.instance.currentUser();
  var ref = _firestore.collection('users').document(user.uid);
  await ref.setData({
    'tales': {data.documentID:2} ,
  }, merge: true);
  await rmxAudioPlayer.setPlaylistItems([
    new AudioTrack(
      trackId: widget.data['title'],
      artist: "Сказки старого гнома",
      album: "",
      albumArt: widget.data['cover'],
      assetUrl:
        widget.data['audioFiles'][0]['url'],
      title: widget.data['title'],
    )
  ]);
  await rmxAudioPlayer.seekTo(_position * _total);
  setState(() => _status = "loading");
  _audio.play(widget.data['audioFiles'][0]['url'],
    title: widget.data['title'],

```

```

    album: "",
    artist: "Сказки старого гнома",
    imageUrl: widget.data['cover']);
_audio.seekTo(Duration(minutes: 2));
setState() {
  _seeking = now;
});
} @override
void initState() {
  phoneController = TextEditingController(text: "");
  super.initState();
}
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: BoxDecoration(
        image: new DecorationImage(
          image: new AssetImage("assets/images/backlogin.png"),
          fit: BoxFit.fill,
        ),
      ),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: <Widget>[
        Container(
          width: MediaQuery.of(context).size.width * 0.6,
          child: Text("Зарегистрируйте свой профиль для пользования
приложением", textAlign: TextAlign.center, style: GoogleFonts.questrial(
            textStyle: TextStyle(

```

```

        fontSize: 16,
        color: Color.fromRGBO(255, 255, 255, 0.65),
        fontWeight: FontWeight.w400,
    )),),
Container(
  padding: const EdgeInsets.symmetric(vertical:16.0, horizontal: 80),
  child: new Theme(
    data: new ThemeData(
      inputDecorationTheme: InputDecorationTheme(
        enabledBorder: UnderlineInputBorder(
          borderSide: BorderSide(color: Color(0xFFEAEAEA))
        ),
      ),
      primaryColor: Colors.white,
      primaryColorDark: Colors.white,
    ),
    child: TextField(
      keyboardType: TextInputType.phone,
      style: textStyle2,
      controller: phoneController,
      inputFormatters: [maskFormatterphone],
      decoration: InputDecoration(
        hintStyle: TextStyle(fontSize: 18.0, color: Colors.white, fontWeight:
FontWeight.w300),
        prefixIcon: Icon(SimpleLineIcons.phone, color: Colors.white,),
        hintText: 'Номер телефона',
      ),
      onChanged: (value){
        this.phoneNo = value;
      },

```

```

    )),
  ),
  SizedBox(height: 10.0),
  Container(
    margin: EdgeInsets.only(top:56),
    child: InkWell(
      onTap:(){ verifyPhone();},
      child:Container(
        width: MediaQuery.of(context).size.width * 0.45,
        padding: EdgeInsets.symmetric(horizontal: 24, vertical: 8),
        decoration: BoxDecoration(
          color: Colors.transparent,
          border: Border.all(color: Colors.white),
          borderRadius: BorderRadius.circular(50)
        ),
        child:Text('Далее',          textAlign:          TextAlign.center,
style:GoogleFonts.questrial(
          textStyle: TextStyle(
            fontSize: 20,
            color: Color.fromRGBO(255, 255, 255, 1),
            fontWeight: FontWeight.w500,
          )))
      )
    ),
  ),
],
)),
);
}
}

```

Отже, було розроблено систему авторизації через мобільний номер телефону та візуальне оформлення до нього з використанням Firebase auth.

Після вводу номера користувача і натискання «Далі» метод для авторизації відправляє номер на сервер авторизації і якщо немає ніяких помилок у відповіді, то з'являється контекстне вікно для введення одноразового паролю. Під час введення правильного пароля, відправляється ще один запит, користувач зберігається у базі даних та у модулі авторизації і виконується перехід на основний екран.

У третьому розділі магістерської кваліфікаційної роботи описано процес розроблення базових модулів клієнта мобільного додатку.

Викладено функціонал системи прослуховування аудіокниг. Описано використання концепції MVC. Розглянуто фреймворки для кросплатформної мобільної розробки. Описано переваги та недоліки фреймворку Flutter порівняно з фреймворком React Native.

4. ТЕСТУВАННЯ ДОДАТКУ ТА РОЗРОБЛЕННЯ РЕКОМЕНДАЦІЙ КОРИСТУВАЧУ

4.1 Вибір методів тестування програмного забезпечення

Тестування програмного забезпечення — перевірка на відповідність між поточною та очікуваною роботою програми, що виконується на кінцевому наборі тестів, обраному певним чином. У ширшому сенсі, тестування – це техніка контролю якості, що включає в себе активності з планування робіт, проектування тестів, виконання тестування і аналізу отриманих результатів.

Техніка тестування «білої скриньки» досліджує внутрішню поведінку програми, перевіряє правильність побудови елементів програми та регулярність їх взаємодії.

Це метод тестування програмного забезпечення, який розраховано на те, що внутрішня структура системи відома для тестувальників. Ми вибираємо вхідні значення, ґрунтуючись на знанні коду, який буде їх обробляти. Точно так само ми знаємо, яким повинен бути результат цього оброблення. Знання всіх особливостей програми, що тестується і її реалізації — обов'язкові для цієї техніки. Тестування «білої скриньки» — поглиблення під внутрішній устрій системи, за межі її зовнішніх інтерфейсів.

Для тестування сформовано такі параметри тестування:

- перевірено всі незалежні канали програмування;
- існують гілки True, False для всіх логічних рішень;
- виконуються всі цикли (у своїх межах);
- аналізує правильність внутрішніх структур даних.

Переваги тестування «білої скриньки»:

- кількість помилок є мінімальним у «центрі» програми і найбільшимна «периферії»;
- прогнози щодо ймовірності потоків керування даних у програмі часто невірні, отже, типовим шляхом є модель розрахунку, яка погано обробляється;

— під час написання алгоритму у вигляді тексту будь-якою мовою програмування, можна ввести типові синтаксичні та семантичні помилки передавання;

— деякі результати програми залежать не тільки від вихідних.

Недоліки тестування «білої скриньки»:

— кількість незалежних маршрутів може бути незліченною;

— повна перевірка доріжок не може гарантувати, що програма відповідає оригінальним вимогам;

— помилки, які виникають у разі проблем з даними, не можуть бути виявлені.

Метод тестування «чорної скриньки» досліджує функціональні можливості програмної функції в межах всієї області визначення. Тест не враховує внутрішню логічну структуру функцій.

Методи випробування показують:

— реалізацію функцій програми;

— надходження вхідних даних;

— досягнення результату;

— збереження цілісності зовнішньої інформації.

Тестування «чорної скриньки» дає змогу шукати такі категорії помилок:

— невірні або відсутні функції;

— помилки інтерфейсу;

— помилки у зовнішніх структурах даних або у зовнішній базі даних;

— характеристики помилки (необхідна ємність пам'яті тощо);

— помилки ініціалізації та завершення.

Проаналізувавши переваги та недоліки методів тестування «чорної скриньки» та «білої скриньки» відповідно до мети роботи, для тестування програмного забезпечення було обрано метод «чорної скриньки» [12].

4.2 Тестування розробленого мобільного додатку

Тестування розробленого мобільного додатку починається з запуску. Після запуску додатку очікується побачити екран аутентифікації. Результат запуску додатку наведено на рис. 4.1.

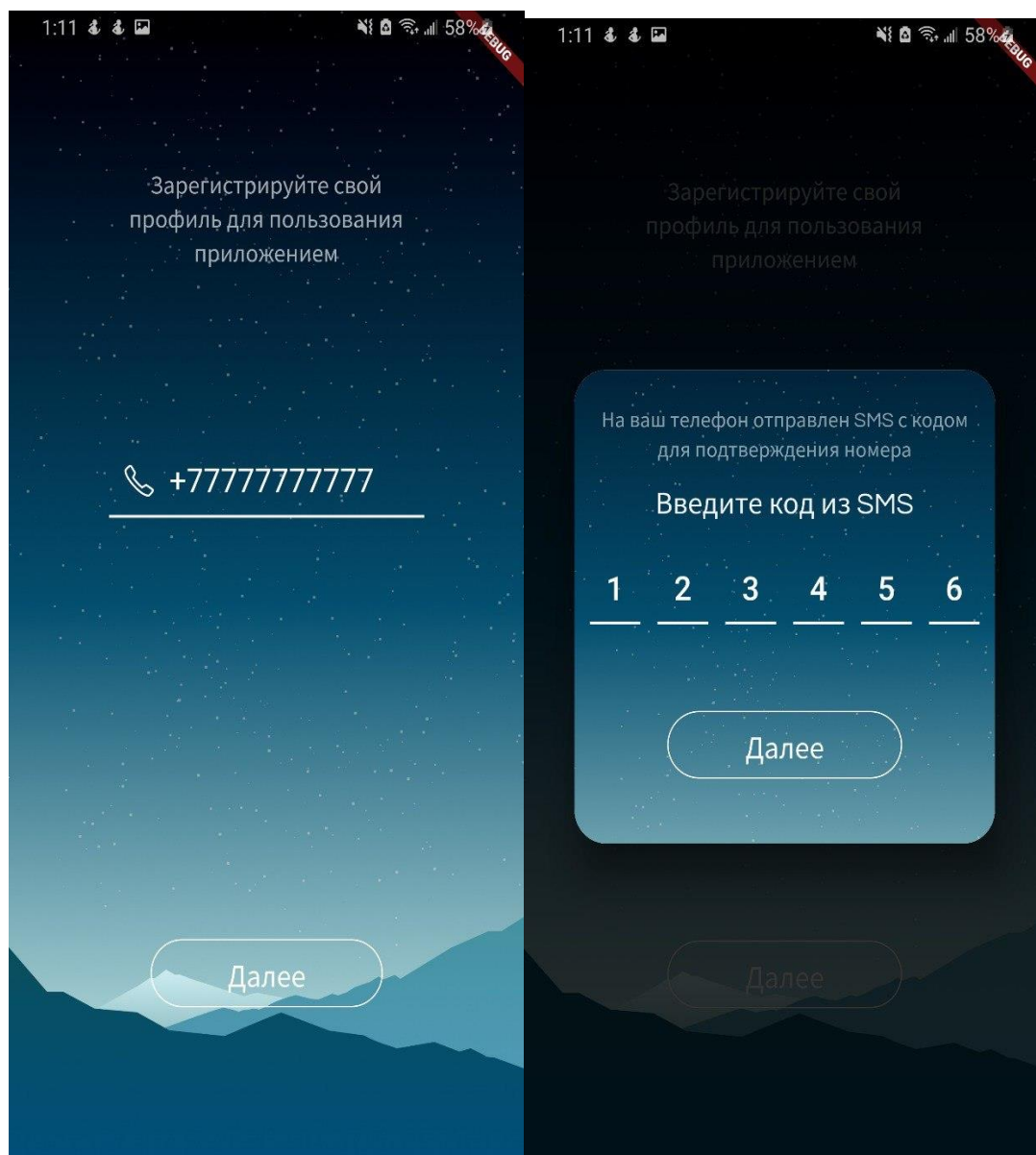


Рисунок 4.1 — Початковий екран

Для початку треба авторизуватися в мобільному додатку за допомогою номера телефона. Після успішної авторизації з'являється головне меню. (рис. 4.2).

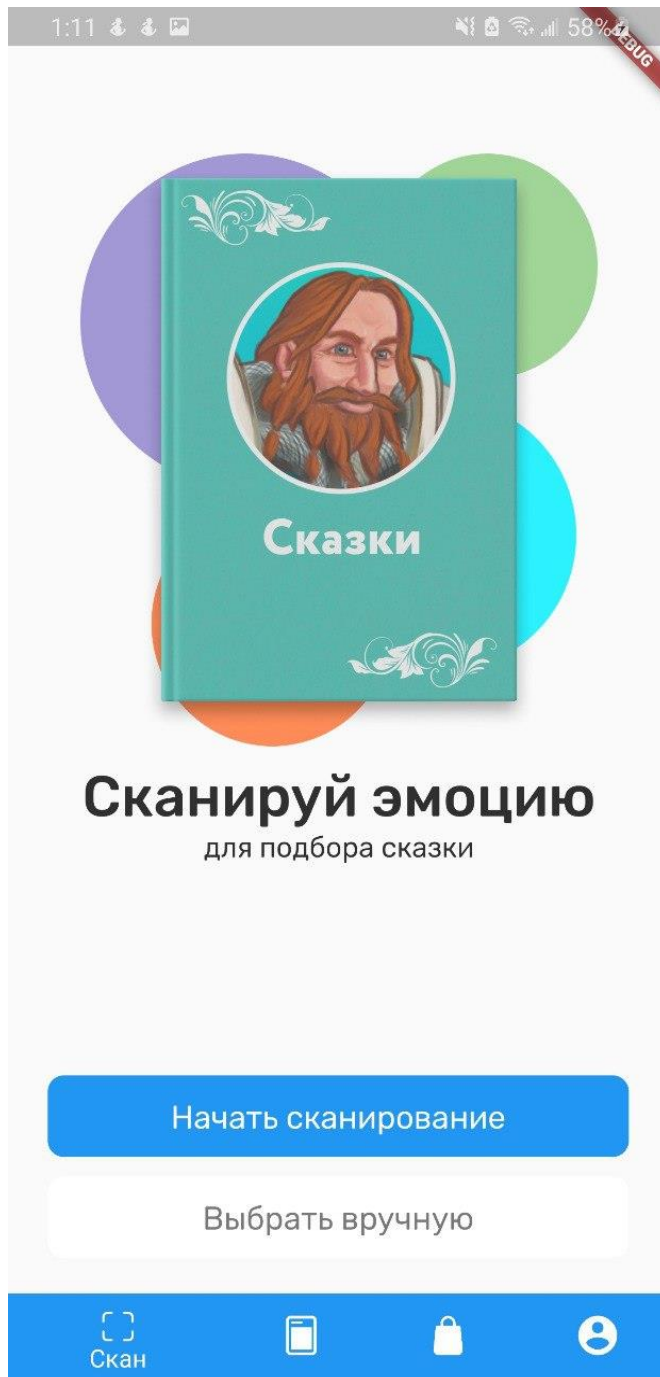


Рисунок 4.2 — Головне меню

На основному екрані відображається можливість відсканувати емоцію або вибрати її вручну в разі проблем з камерою чи небажанням користувача фотографуватися. Під час сканування користувачу пропонується сфотографуватися і отримати результат (рис. 4.3).

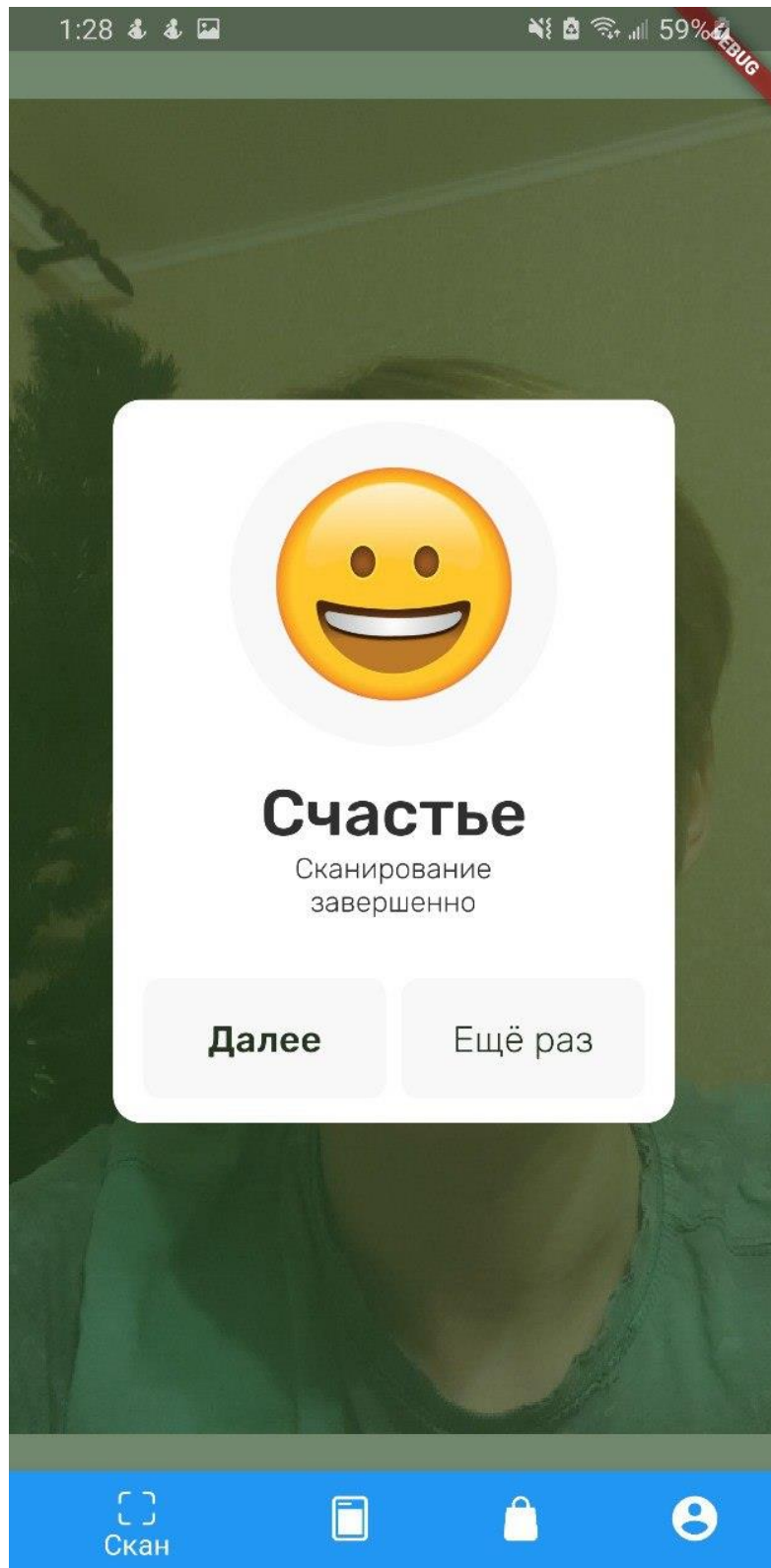


Рисунок 4.3 — Результат сканування

Також можна обрати емоцію вручну (рис. 4.4).

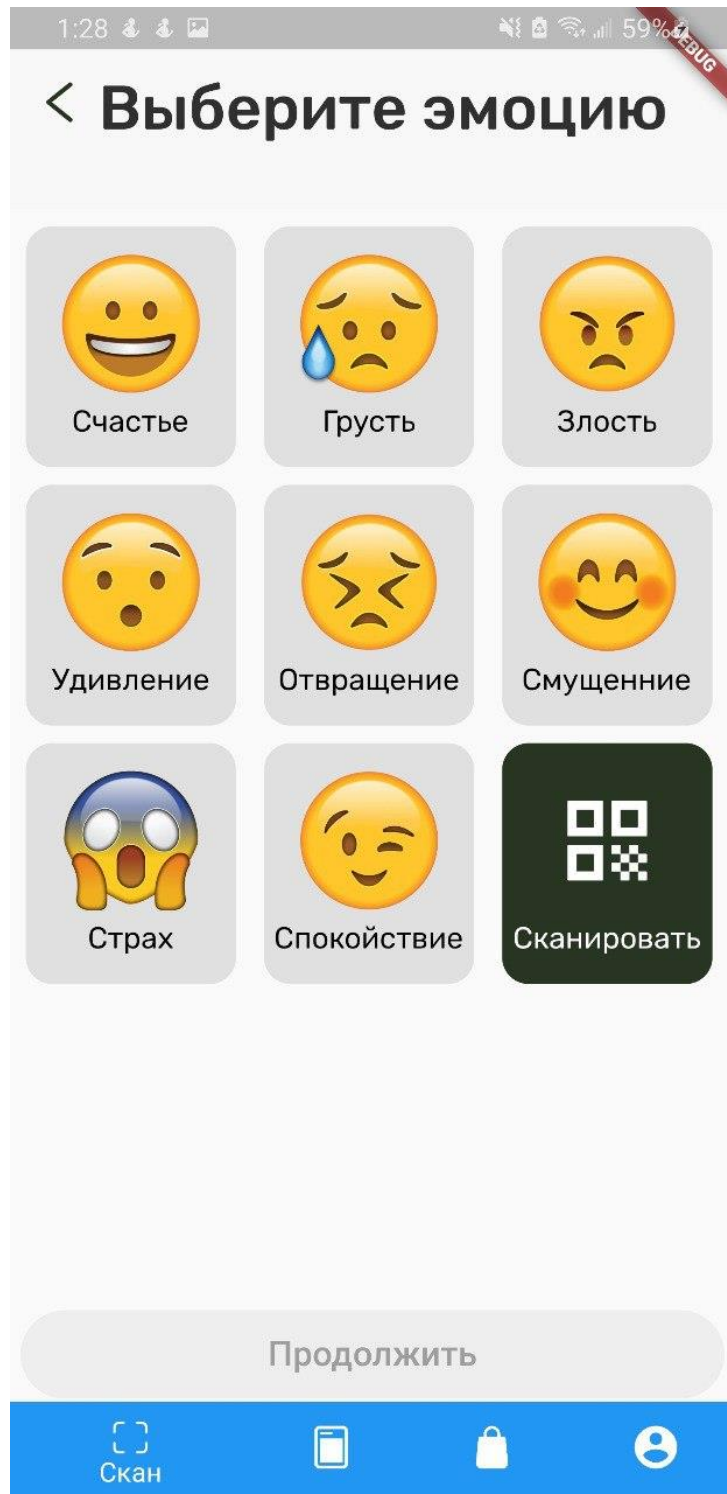


Рисунок 4.4 — Мануальний вибір емоції

Після сканування або вибору користувач отримує набір книг, які підходять під його емоцію. Каталог рис. 4.5.

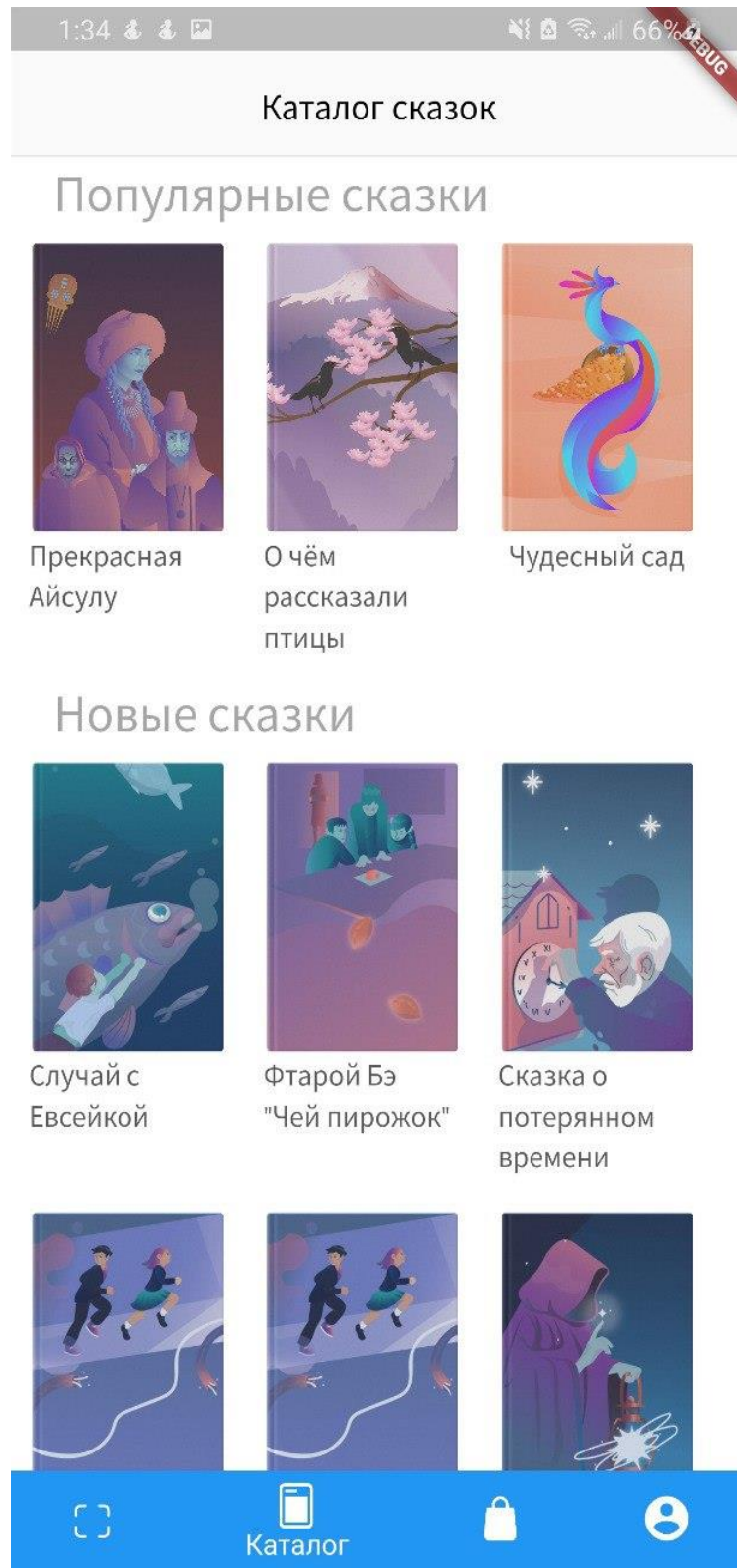


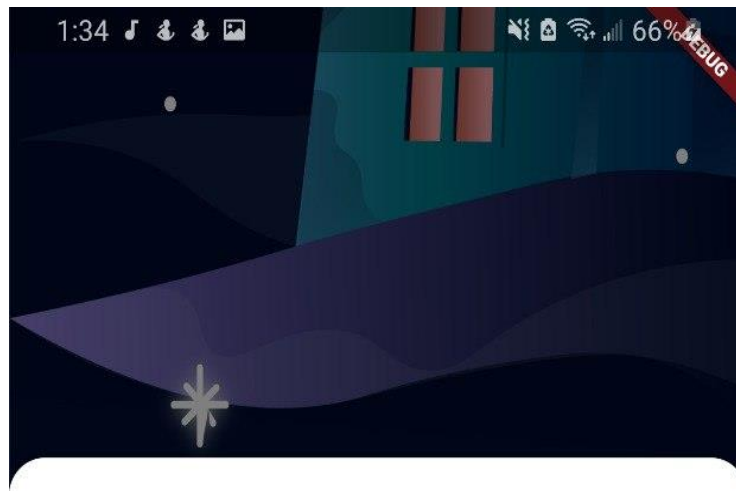
Рисунок 4.5 — Каталог книг

Під час вибору окремої книги користувач переходить на її сторінку. На сторінці книги є ілюстрація до неї та короткий опис (рис. 4.6).



Рисунок 4.6 — Сторінка книги

Також при натисканні на прослуховування з'являється зручний плеєр для керування прослуховуванням (рис. 4.7).



Сказка о потерянном времени

Сказки старого гнома



Рисунок 4.7 — Плеер

Також можна замовити книгу, для цього потрібно залишити дані в окремому меню. Також користувач може заповнити свої дані, та поставити аватар, рис. 4.8.

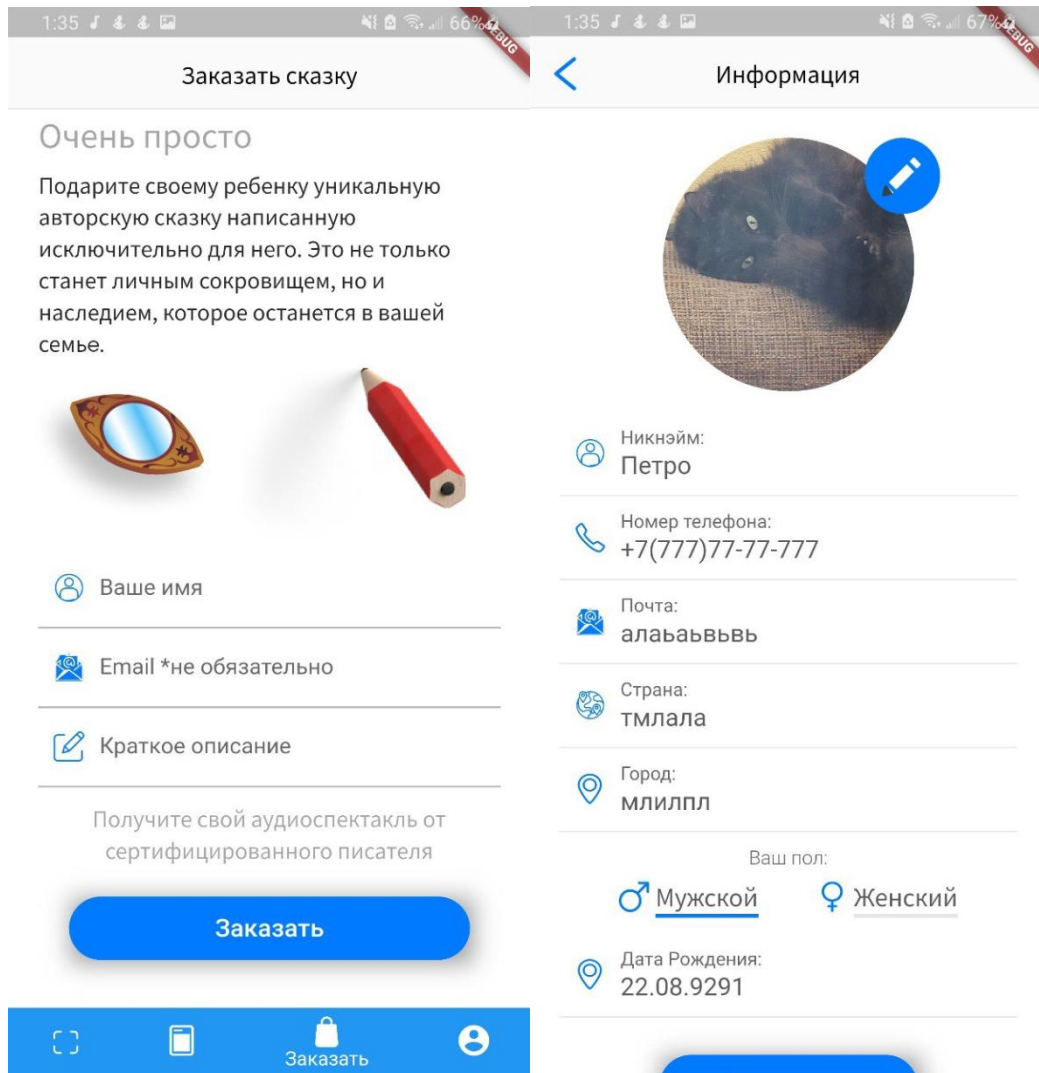


Рисунок 4.8 — Замовлення та дані користувача

У четвертому розділі виконано аналіз методів тестування, у результаті якого було обрано метод «чорної скриньки». За обраною методикою проведено повне тестування всіх функцій розробленого додатку, яке підтвердило його повну працездатність.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Оцінювання комерційного потенціалу розробки «Кроссплатформний мобільний додаток з використанням фреймворку flutter» (або технологічний аудит розробки)

Проведення оцінювання комерційного потенціалу розробки є метою технологічного аудиту. Для проведення технологічного аудиту залучені три незалежних експерти. Оцінювання комерційного потенціалу розробки здійснюється за 12-тю категоріями, наведені в таблиці 5.1

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

| Критерії оцінювання та бали (за 5-ти бальною шкалою) | | | | | |
|--|--|---|---|---|--|
| Кри- тері й | 0 | 1 | 2 | 3 | 4 |
| Технічна здійсненність концепції: | | | | | |
| 1 | Достовірність концепції не підтверджена | Концепція підтверджена експертними висновками | Концепція підтверджена розрахунками | Концепція перевірена на практиці | Перевірено робоздатність продукту в реальних умовах |
| Ринкові переваги (недоліки): | | | | | |
| 2 | Багато аналогів на малому ринку | Мало аналогів на малому ринку | Кілька аналогів на великому ринку | Один аналог на великому ринку | Продукт не має аналогів на великому ринку |
| 3 | Ціна продукту значно вища за ціни аналогів | Ціна продукту дещо вища за ціни аналогів | Ціна продукту приблизно дорівнює цінам аналогів | Ціна продукту дещо нижче за ціни аналогів | Ціна продукту значно нижче за ціни аналогів |
| 4 | Технічні та споживчі властивості продукту значно гірші, ніж в аналогів | Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів | Технічні та споживчі властивості продукту на рівні аналогів | Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів | Технічні та споживчі властивості продукту значно кращі, ніж в аналогів |
| 5 | Експлуатаційні витрати значно вищі, ніж в аналогів | Експлуатаційні витрати дещо вищі, ніж в аналогів | Експлуатаційні витрати на рівні експлуатаційних витрат аналогів | Експлуатаційні витрати трохи нижчі, ніж в аналогів | Експлуатаційні витрати значно нижчі, ніж в аналогів |

Продовження таблиці 5.1

| Критерії оцінювання та бали (за 5-ти бальною шкалою) | | | | | |
|--|--|---|---|---|---|
| Кр. | 0 | 1 | 2 | 3 | 4 |
| Ринкові перспективи | | | | | |
| 6 | Ринок малий і не має позитивної динаміки | Ринок малий, але має позитивну динаміку | Середній ринок з позитивною динамікою | Великий стабільний ринок | Великий ринок з позитивною динамікою |
| 7 | Активна конкуренція великих компаній на ринку | Активна конкуренція | Помірна конкуренція | Незначна конкуренція | Конкуренція немає |
| Практична здійсненність | | | | | |
| 8 | Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї | Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців | Необхідне незначне навчання фахівців та збільшення їх штату | Необхідне незначне навчання фахівців | Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї |
| 9 | Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні | Потрібні незначні фінансові ресурси. Джерела фінансування відсутні | Потрібні значні фінансові ресурси. Джерела фінансування є | Потрібні незначні фінансові ресурси. Джерела фінансування є | Не потребує додаткового фінансування |
| 10 | Необхідна розробка нових матеріалів | Потрібні матеріали, що використовуються у військово-промисловому комплексі | Потрібні дорогі матеріали | Потрібні досяжні та дешеві матеріали | Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві |
| 11 | Термін реалізації ідеї більший за 10 років | Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років | Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років | Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років | Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років |
| 12 | Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво | Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає | Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу | Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту | Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту |

Результати оцінювання комерційного потенціалу розробки потрібно звести в таблицю за зразком таблиці 5.2.

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

| Критерії | Експерти | | |
|--|--|-----------------|-----------------|
| | Експерт 1 | Експерт 2 | Експерт 3 |
| | Бали, виставлені експертами: | | |
| 1 | 2 | 3 | 4 |
| 2 | 3 | 3 | 2 |
| 3 | 2 | 2 | 3 |
| 4 | 3 | 3 | 2 |
| 5 | 3 | 3 | 3 |
| 6 | 3 | 4 | 4 |
| 7 | 3 | 3 | 3 |
| 8 | 2 | 4 | 2 |
| 9 | 2 | 3 | 4 |
| 10 | 2 | 2 | 3 |
| 11 | 4 | 4 | 2 |
| 12 | 2 | 2 | 3 |
| Сума балів | СБ ₁ | СБ ₂ | СБ ₃ |
| Середньоарифметична сума балів $\overline{СБ}$ | $\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = 34$ | | |

Таблиця 5.3 — Рівні комерційного потенціалу розробки

| Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів | Рівень комерційного потенціалу розробки |
|--|---|
| 0 – 10 | Низький |
| 11 – 20 | Нижче середнього |
| 21 – 30 | Середній |
| 31 – 40 | Вище середнього |
| 41 – 48 | Високий |

В результаті було визначено, що рівень комерційного потенціалу розробки є вище середнього.

5.2 Прогнозування витрат на виконання науково-дослідної, (дослідно-конструкторської) та конструкторсько-технологічної роботи

Прогнозування витрат на виконання даних робіт може складатись з таких етапів:

- розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи.
- розрахунок загальних витрат на виконання даної роботи.
- прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

5.2.1 Розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи, можна здійснити за такими статтями та формулами:

5.2.2 Основна заробітна плата кожного із розробників (дослідників) Z_o , якщо вони працюють в наукових установах бюджетної сфери:

$$Z_o = \frac{M}{T_p} \cdot t \text{ грн.}, \quad (4.1)$$

де M — місячний посадовий оклад конкретного розробника, грн.

T_p — число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні;

t — число робочих днів роботи розробника (дослідника).

Зроблені розрахунки бажано звести до таблиці:

Таблиця 5.4

| Найменування посади виконавця | Місячний посадовий оклад, грн. | Оплата за робочий день, грн. | Число днів роботи | Витрати на оплату праці, грн. | Примітка |
|-------------------------------|--------------------------------|------------------------------|-------------------|-------------------------------|----------|
| 1. Науковий керівник | 8 000 | 363,35 | 21 | 7630,35 | |
| 2. Інженер програміст | 10500 | 478 | 25 | 11950 | |
| 3. Провідний інженер | 7000 | 318 | 22 | 6986,65 | |
| Всього | | | | 26 550 | |

5.2.3 Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = (0,1...0,12) \cdot Z_o. \quad (4.2)$$

Для нашого випадку:

$$Z_d = 0.1 \times (26\ 550) = 2655$$

5.2.4 Нарахування на заробітну плату $H_{зп}$ розробників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$H_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100}, \quad (4.3)$$

де Z_o — основна заробітна плата розробників, грн.;

Z_d — додаткова заробітна плата всіх розробників та робітників, грн.;

β — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, %. У 2020 році ЄСВ складає 22% .

Тоді :

$$H_{зп} = (26\ 550 + 2655) \times 0.22 = 6425,1 \text{ грн.}$$

5.2.5 Амортизація обладнання, комп'ютерів та приміщень A , які використовувались під час (чи для) виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{Ц \cdot N_a}{100} \cdot \frac{T}{12} \text{ грн.}, \quad (4.4)$$

де $Ц$ — загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн;

N_a — річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $N_a = (10...25)\%$;

T — термін використання обладнання, приміщень тощо, місяці.

Зроблені розрахунки бажано звести до таблиці 4.5:

Таблиця 5.5 Амортизаційні відрахування

| Найменування обладнання, приміщень тощо | Балансова вартість, грн. | Норма амортизації, % | Термін використання, міс. | Величина амортизаційних відрахувань, грн. | Примітка |
|--|--------------------------|----------------------|---------------------------|---|----------|
| MacBook | 29000 | 10 | 4 | 1208 | |
| Офісне приміщення | 120000 | 15 | 6 | 9000 | |
| Операційна система MacOS | 4000 | 25 | 6 | 500 | |
| Програмне забезпечення Android Studio, Xcode | 3250 | 25 | 6 | 406 | |
| Всього | | | | 11 769 | |

5.2.6. Витрати на послуги, що були використані під час виконання наукової розробки вказані у табл. 5.6.

5.2.7 Витрати на силову електроенергію V_e , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi} \text{ грн,} \quad (5.6)$$

Таблиця 5.6. Витрати на послуги, що були використані під час наукової роботи

| Найменування робіт (послуг) | Термін використання, міс. | Ціна за місяць, грн. | Сума, грн. |
|---|---------------------------|----------------------|------------|
| AWS хостинг | 3 | 980 | 2940 |
| GitHub репозиторій | 2 | 580 | 1160 |
| Аккаунти розробника в Google play рік, 650грн | 12 | 54 | 650 |
| Аккаунти розробника в App store(2800грн) | 12 | 233 | 2800 |
| Всього | | | 7550 |

Де V — вартість 1 кВт-год. електроенергії, в 2020 р. $V \approx 2,50$ грн./кВт;

P — установлена потужність обладнання, кВт;

Φ — фактична кількість годин роботи обладнання, годин,

$K_{\text{п}}$ — коефіцієнт використання потужності; $K_{\text{п}} < 1$.

$$V_{\epsilon} = 2.50 \times 1.413 \times 24 \times 0.9 = 76.4 \text{ грн.}$$

5.2.8 Інші витрати $V_{\text{ін}}$ охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо.

Інші витрати $I_{\text{в}}$ можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{\text{ін}} = (1,8) * (3_{\text{o}}). \quad (5.7)$$

$$V_{\text{ін}} = 1,8 \times 26550 = 46010$$

5.2.9 Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи – В.

$$B = 26\,550 + 76,4 + 11\,769 + 2655 + 6425,1 + 7550 + 46010 = 101046,5$$

Розрахунок загальних витрат на виконання даної роботи. Розрахунок загальних витрат здійснюється у тому випадку, коли дипломник виконує тільки певну частину даної роботи. У подальшому ця наукова робота буде продовжена.

Тоді загальна вартість всієї наукової роботи визначається за $B_{\text{заг}}$ формулою:

$$B_{\text{заг}} = \frac{B}{\alpha}, \quad (5.8)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях, $\alpha = 0,1$

$$B_{\text{заг}} = \frac{1\,01046,5}{0,1} = 1010465$$

Прогнозування загальних витрат на виконання та впровадження результатів виконаної наукової роботи. Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\beta}, \quad (5.9)$$

де β — коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Так, якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то $\beta \approx 0,1$;
- на стадії технічного проектування, то $\beta \approx 0,2$;
- на стадії розробки конструкторської документації, то $\beta \approx 0,3$;
- на стадії розробки технологій, то $\beta \approx 0,4$;
- на стадії розробки дослідного зразка, то $\beta \approx 0,5$;
- на стадії впровадження, то $\beta \approx 0,9$.

Оскільки дана розробка знаходиться на стадії науково-дослідних робіт, то $\beta = 0,1$

$$ЗВ = \frac{1010465}{0,1} = 10104650 \text{ (грн)}$$

5.3 Прогнозування комерційних ефектів від реалізації результатів розробки «Кроссплатформний мобільний додаток з використанням фреймворку flutter»

Прогнозування комерційних ефектів здійснюється у двох основних випадках:

- коли можна прямо оцінити зростання чистого прибутку підприємства від впровадження результатів наукової розробки;
- коли не можливо прямо оцінити зростання чистого прибутку підприємства від впровадження результатів наукової розробки.

Ціна до впровадження розробки складала 115 грн, після впровадження — 140 грн. Розрахунки здійснюватимемо за формулою 3.14:

$$\Delta\Pi_i = \sum_1^n (\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\nu}{100}\right), \quad (5.10)$$

де ΔC_o — покращення основного оціночного показника від впровадження результатів розробки у даному році, $\Delta C_o = 25$ грн ;

N — основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки, $N = 330\,000$ шт.;

ΔN — покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки, $\Delta N = 16\,200$ шт ;

C_o — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $C_o = 140$ грн;

n — кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки, $n = 4$;

λ — коефіцієнт, який враховує сплату податку на додану вартість
 $\lambda = 0,8547$;

ρ — коефіцієнт, який враховує рентабельність продукту, $\rho = 0,25$;

ν — ставка податку на прибуток, $\nu = 18\%$.

$$\Delta\Pi_1 = [25 \cdot 330000 + (115 + 25) \cdot 16200] \cdot 0,8547 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 15214750$$

$$\Delta\Pi_2 = [25 \cdot 330000 + (115 + 25) \cdot (16200 + 16200)] \cdot 0,8547 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 15621830$$

$$\Delta\Pi_3 = [25 \cdot 330000 + (115 + 25) \cdot (16200 + 16200 + 16200)] \cdot 0,8547 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 15854470$$

$$\Delta\Pi_4 = [25 \cdot 330000 + (115 + 25) \cdot (16200 + 16200 + 16200 + 16200)] \cdot 0,8547 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 16203390$$

5.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

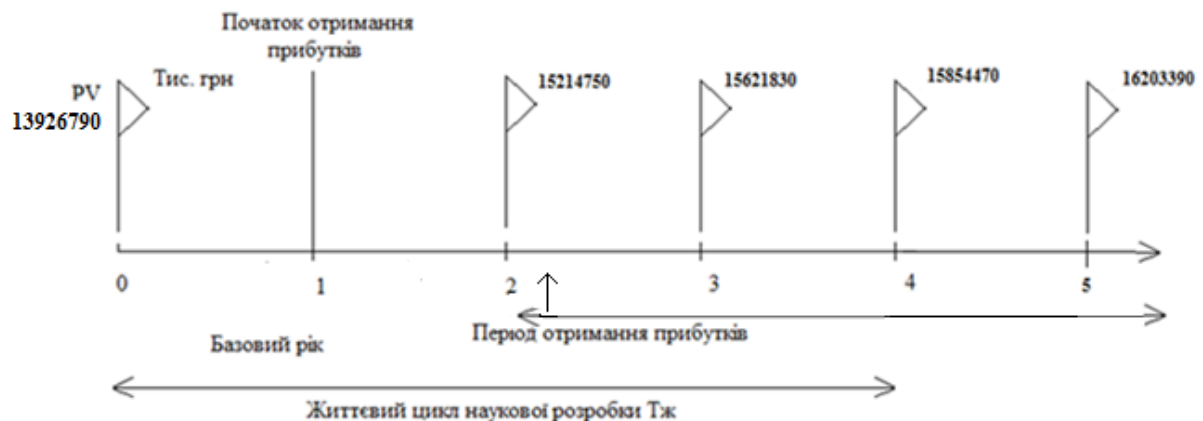


Рисунок 5.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо абсолютну ефективність вкладених інвестицій $E_{\text{абс}}$.

Для цього використаємо формулою:

$$E_{\text{абс}} = (\text{ПП} - \text{PV}), \quad (5.11)$$

де Π — приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV — теперішня вартість інвестицій $PV = 3B$, грн.

У свою чергу, приведена вартість всіх чистих прибутків Π розраховується за формулою:

$$\Pi = \sum_{t=1}^m \frac{\Delta\Pi_t}{(1+\tau)^t}, \quad (5.12)$$

де $\Delta\Pi_t$ — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t — період часу, протягом якого виявляються результати впровадженої НДДКР, $t=3$;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; $\tau=0,1$;

t — період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

$$\Pi = \frac{15214750}{(1+0,1)^1} + \frac{15621830}{(1+0,1)^2} + \frac{15854470}{(1+0,1)^3} + \frac{16203390}{(1+0,1)^3} = 49721030$$

$$E_{abc} = \Pi - 3B = 49721030 - 10104650 = 35794240$$

Розраховують відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_v . Для цього користуються формулою:

$$E_v = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.13)$$

де E_{abc} — абсолютна ефективність вкладених інвестицій, грн; PV — теперішня вартість інвестицій $PV = 3B$, грн; $T_{ж}$ — життєвий цикл наукової розробки, роки.

$$E_v = \sqrt[4]{1 + \frac{35794240}{10104650}} - 1 = 0,451807$$

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; $d = 0,16$;

f — показник, що характеризує ризикованість вкладень; зазвичай, $f = 0,08$.

$$\tau = 0,15 + 0,08 = 0,23$$

Оскільки $E_b > \tau$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ок}}$ можна розрахувати за формулою:

$$T_{\text{ок}} = \frac{1}{E_g}. \quad (5.15)$$

$$T_{\text{ок}} = \frac{1}{0,451807} = 2,21$$

Оскільки $T_{\text{ок}} < 3-5$ років, то це свідчить про доцільність фінансування такої розробки.

Отже, у даному розділі роботи було проведено розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи. Основна заробітна плата розробників (дослідників) склала 26 550 грн. Додаткова заробітна плата всіх розробників складає 2655 грн. Нарахування на заробітну плату розробників - 6425,1 грн. Амортизація обладнання склала 11 769 грн. Також розраховані витрати на силову електроенергію, інші витрати - 76,4 грн., 46010 грн. Третій етап – прогнозування загальних витрат на виконання та впровадження результатів наукової роботи, він склав 10104650 У четвертому

розділі проведено розрахунок ефективності вкладених інвестицій та періоду їх окупності. Приведена вартість всіх чистих прибутків склала 49721030 грн. Абсолютна ефективність вкладених інвестицій 35794240 грн. Також обраховано відносну ефективність вкладених коштів в наукову розробку інвестицій 0451, вона має бути більшою ніж мінімальна ставка дисконтування, що складає 0,23. Тому можна зробити висновок, що інвестор буде більше чим зацікавлений у фінансуванні даної наукової розробки надвисокочастотного вологоміра. Термін окупності складає 2,21 року, оскільки термін окупності менший ніж п'ять років, це свідчить про доцільність фінансування такої розробки.

ВИСНОВКИ

У магістерській кваліфікаційній роботі розроблено кросплатформний мобільний додаток для прослуховування аудіокниг, що використовує метод сканування емоцій для ефективного залучення нових користувачів.

Розглянуто особливості розробки мобільних додатків. Описано використання концепції аудіокниг.

Удосконалено процес підбору аудіокниг, у якому використовується сканування емоцій користувача, що дозволяє підвищити активність користувачів шляхом їх зацікавлення процесом проходження сканування емоцій, що дозволяє розширити комунікаційну взаємодію користувачів у процесі розвитку маркетингової стратегії компанії.

Було розроблено алгоритми роботи системи підбору та прослуховування аудіокниг, програмні модулі системи з використанням фреймворку Flutter, технології для розроблення мобільних додатків під операційні платформи Android/iOS. Розроблено програму клієнт-хмарної архітектури, для неї спроектовано та розроблено хмарне файлове сховище та база даних у хмарі. Проаналізовано особливості баз даних та розроблено структуру та схему бази даних для мобільного додатку.

Розроблено засоби адміністрування системи.

Проведено тестування розробленого мобільного застосунку за обраним методом “чорної скриньки”, за результати якого було підтверджено коректність роботи програми.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Beginning Android Games / Robert Green. – NY: Apress, 2012. – 714p.
2. Grady B. Object-Oriented Analysis and Design with Applications/ B. Grady. – Amsterdam: Addison-Wesley Professional, 2009. – 720 p.
3. Horton's I. Beginning JavaScript/I. Horton's. – Amsterdam: Wrox, 2015. – 988 p.
4. Юхимчук С. В., Азарова А. О. Математичні моделі ризику для систем підтримки прийняття рішень. Монографія 2003. - 188 с.
5. Stellman A. Head First NodeJS / A. Stellman. – Boston: O'Reilly Media, 2014. – 784 p.
6. Мови програмування — Stackoverflow. [Електронний ресурс]. – Режим доступу: <https://ru.stackoverflow.com/tags/info>
7. Нечітка логіка, нечіткі множини, м'які обчислення — часті питання Режим доступу <http://www.znannya.org/?view=fuzzy-logic-q>
8. Азарова А. О., Поплавський А. В. Інформатика та комп'ютерна техніка (Частина 1). 2012. - 361 с.9. Бази знань. Інтелектуальні інформаційні системи [Електронний ресурс]. – Режим доступу: <https://studfiles.net/preview/5474324/>
10. Nodejs Documentation [Електронний ресурс]. Режим доступу до ресурсу: <https://nodejs.org/en>.
11. Fleder D., Hosanagar K. Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity (журнал) // Management Science, Vol. 55, No. 5, May 2009. – P. 697-712.
12. Xiaoyuan Su and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques A Survey of Collaborative Filtering Techniques (журнал) // Hindawi Publishing Corporation, Advances in Artificial Intelligence archive, USA. — 2009. — P. 1 - 19.
13. Zadeh, L. A. (1965). Fuzzy sets. Information and Control 8 (3): 338. doi:10.1016/S0019-9958(65)90241-X

14. Білоусова Л.І. Інформатика. Навчальний посібник // Білоусова Л.І., Муравка А.С., Олефіренко Н.В. - Харків: Факт, 2009. – 352 с.
15. User experience – [Електронний ресурс]: Режим доступу до матеріалу: https://en.wikipedia.org/wiki/User_experience.
16. Adobe Illustrator – [Електронний ресурс]: Режим доступу до матеріалу: https://ru.wikipedia.org/wiki/Adobe_Illustrator.
17. Азарова А. О., Антонюк О. В. Математичні моделі оцінювання стратегічного потенціалу підприємства та прийняття рішень щодо його підвищення. Монографія 2012. - 168 с.