

## ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОБІЛЬНОГО ДОДАТКУ ГЕНЕРАЦІЇ РЕЦЕПТІВ НА ОСНОВІ ВВЕДЕНИХ ІНГРЕДІЄНТІВ

*Свирид Д.А., науковий керівник Бородкіна І.Л.*

У сучасному світі, де мобільні технології активно проникають у повсякденне життя, все більше і більше користувачів шукають практичні та креативні засоби для організації побутових процесів. Однією з цих областей є готування. Люди часто мають обмежену кількість продуктів і не знають, що можна приготувати. У таких випадках мобільні додатки приходять на порятунок, що може створити відповідну формулу залежно від наявних компонентів.

В рамках дослідження було проаналізовано предметну область, визначено основні потреби користувачів та сформульовано вимоги до функціональності майбутнього програмного забезпечення. Зокрема, додаток повинен:

- дозволяти користувачу вводити список доступних інгредієнтів;
- пропонувати рецепти на основі цього списку;
- враховувати алергени та харчові вподобання;
- забезпечувати можливість збереження улюблених рецептів;
- надавати інтерфейс для створення та перегляду власних рецептів.

Для створення такого додатку було проведено порівняльний аналіз сучасних технологій, які можна використати для розробки кросплатформеного мобільного програмного забезпечення. Основну увагу приділено таким технологіям, як React Native, Flutter, Kotlin/Swift (нативна розробка) та Xamarin.

| Технологія   | Мова         | Кросплатформеність | Швидкість розробки | Підтримка спільноти | Продуктивність |
|--------------|--------------|--------------------|--------------------|---------------------|----------------|
| React Native | TypeScript   | Так                | Висока             | Висока              | Висока         |
| Flutter      | Dart         | Так                | Середня            | Висока              | Висока         |
| Kotlin/Swift | Kotlin/Swift | Ні                 | Низька             | Висока              | Максимальна    |
| Xamarin      | C#           | Так                | Середня            | Середня             | Середня        |

Таблиця 1 Порівняння технологій для розробки мобільних застосунків

Переваги використання React Native та Firebase:

- Швидка розробка завдяки повторному використанню коду для Android та iOS.
- Гнучка архітектура та простота інтеграції з іншими бібліотеками.
- Firebase забезпечує надійне зберігання даних та авторизацію без потреби в окремому сервері.
- Синхронізація в реальному часі через Firestore.

Недоліки:

- Обмеження в нативному функціоналі у порівнянні з Kotlin/Swift.
- Залежність від сторонніх бібліотек та сервісів.
- Високі вимоги до оптимізації при роботі з великими обсягами даних.

На основі аналізу обрано React Native як основну платформу для розробки завдяки її кросплатформеності, популярності та активній підтримці спільноти. Крім того, TypeScript обрано як основну мову програмування через її розширення над JavaScript та типізацію, що забезпечує стабільність коду.

Для зберігання даних та обробки автентифікації користувачів розглянуто Firebase — платформу, яка надає ряд готових рішень для мобільних додатків, включаючи:

- Firebase Auth для реєстрації, входу та підтвердження email;
- Cloud Firestore для зберігання рецептів, профілів, алергенів тощо;
- Firebase Storage для зберігання зображень рецептів;

- Firebase Hosting/Functions (у перспективі) — для серверної логіки або розширень.

Для керування станом додатку розглянуто кілька бібліотек, зокрема Redux, MobX та Zustand. У результаті аналізу обрано Zustand як легковажну та просту в реалізації альтернативу з низьким порогом входу та хорошою продуктивністю.

**Висновки.** У результаті дослідження предметної області визначено актуальність мобільного додатку, що дозволяє генерувати рецепти за доступними інгредієнтами. Проведено аналіз існуючих технологій для розробки програмного забезпечення, що дозволило обґрунтовано обрати стек технологій: React Native + TypeScript для фронтенду, Firebase для бекенду та Zustand для керування станом.

Запропонований технологічний стек є оптимальним за критеріями швидкості розробки, продуктивності, зручності підтримки та масштабованості. Подальші етапи роботи включають проєктування архітектури застосунку та реалізацію його функціональних модулів.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. React Native documentation [Електронний ресурс]. – Режим доступу: <https://reactnative.dev/> – Дата звернення: 18.04.2025.
2. Firebase documentation [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs> – Дата звернення: 18.04.2025.
3. Zustand state management [Електронний ресурс]. – Режим доступу: <https://github.com/pmndrs/zustand> – Дата звернення: 18.04.2025.
4. React Navigation [Електронний ресурс]. – Режим доступу: <https://reactnavigation.org/> – Дата звернення: 18.05.2025.
5. TypeScript: Офіційна документація [Електронний ресурс]. – Режим доступу: <https://www.typescriptlang.org/> – Дата звернення: 18.04.2025.

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І  
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**



**ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ  
АСПЕКТИ РОЗРОБКИ  
КОМП'ЮТЕРНИХ СИСТЕМ**

**ЗБІРНИК НАУКОВИХ ПРАЦЬ ЗА МАТЕРІАЛАМИ  
VII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ  
СТУДЕНТІВ І АСПІРАНТІВ  
*24 квітня 2025 року***

**Київ 2025**

УДК 004

Відповідальна за випуск: М.І. Лендел

Збірник наукових праць за матеріалами VII Всеукраїнської науково-практичної конференції студентів і аспірантів «ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ '2025», 24 квітня 2025 року, НУБіП України, Київ. – 452 с. (електронне видання)

Відповідальність за зміст публікацій несуть автори.

*Передрук матеріалів, а також використання їх будь-якій формі допускається лише з дозволу авторів*