

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДБОРУ ФІТНЕС-ПРОГРАМ НА ОСНОВІ ФІЗИЧНИХ ПАРАМЕТРІВ ТА ЦІЛЕЙ КОРИСТУВАЧА З ВИКОРИСТАННЯМ REACT NATIVE

Тарнавський Ю.Ф., науковий керівник Бородкіна І.Л

У сучасному світі, де мобільні технології активно проникають у повсякденне життя, все більше користувачів шукають персоналізовані рішення для підтримки здорового способу життя. Однією з таких областей є фітнес. Люди часто мають індивідуальні фізичні характеристики та цілі, але не знають, яка програма тренувань їм підходить найкраще. У таких випадках мобільні додатки приходять на допомогу, створюючи персоналізовані фітнес-програми на основі параметрів користувача.

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 свої фізичні параметри (вага, зріст, вік, стать);
- вказувати рівень фізичної підготовки та наявні обмеження;
- визначати цілі тренувань (схуднення, набір м'язової маси, підвищення витривалості);
- пропонувати персоналізовані фітнес-програми на основі цих даних;
- забезпечувати можливість відстеження прогресу та коригування програми.

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.

Техно логія	Мова	Кросплатфо рменість	Швидкі сть розробки	Підтри мка спільноти	Проду ктивність
React Native	TypeS cript	Так	Висока	Висока	Висока
Flutter	Dart	Так	Середня	Висока	Висока
Kotlin /Swift	Kotlin /Swift	Ні	Низька	Висока	Макси мальна
Xamar in	C#	Так	Середня	Середн я	Середн я

Таблиця 1 -- Порівняння технологій для розробки мобільних застосунків

Переваги використання React Native та Firebase для фітнес-додатку:

- Швидка розробка завдяки повторному використанню коду для Android та iOS.
- Гнучка архітектура та простота інтеграції з фітнес-трекерами через API.
- Firebase забезпечує надійне зберігання даних користувачів та їх прогресу.
- Синхронізація в реальному часі через Firestore дозволяє оновлювати програми відповідно до результатів тренувань.

Недоліки:

- Обмеження в доступі до нативних сенсорів пристрою у порівнянні з Kotlin/Swift.
- Залежність від сторонніх бібліотек для складних алгоритмів підбору програм.
- Високі вимоги до оптимізації при візуалізації складних тренувальних схем.

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 типізацію, що забезпечує стабільність коду при роботі з різними типами фізичних параметрів.

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 параметрів користувачів, фітнес-програм та прогресу;
- Firebase Storage для зберігання демонстраційних відео вправ;
- Firebase Analytics для аналізу ефективності різних типів фітнес-програм.

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, що особливо важливо при обробці даних про фізичні параметри та прогрес у реальному часі.

Висновки. У результаті дослідження предметної області визначено актуальність мобільного додатку, що дозволяє підбирати фітнес-програми за фізичними параметрами та цілями користувача. 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 проектування архітектури застосунку та реалізацію алгоритмів підбору фітнес-програм на основі введених даних користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. React Native documentation -- <https://reactnative.dev/>
2. Firebase documentation -- <https://firebase.google.com/docs>
3. React Navigation -- <https://reactnavigation.org/>
4. TypeScript -- <https://www.typescriptlang.org/>

**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