

## ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки  
Кафедра ЕОМ

## Уніфікована модель мобільних веб-застосунків на платформі Swift

Автор:  
студент групи СПм-23-4  
Оцевик В. А.

Керівник:  
доцент каф. ЕОМ, к.т.н.  
Філімончук Т.В.

### Мета роботи:

- дослідження та розробка уніфікованої моделі для ефективного розгортання мобільних та веб-застосунків, використовуючи сучасні можливості платформи Swift;
- підвищення ефективності та якості розробки мобільних застосунків: вирішення проблеми фрагментації та неефективності в розробці мобільних та веб-застосунків шляхом пропозиції стандартизованого підходу, що сприяє скороченню термінів та підвищенню якості кінцевого продукту;
- практичне застосування та внесок у розвиток iOS-розробки: навести обґрунтування доцільності впровадження уніфікованої моделі (практичні рішення) для розробників iOS, які прагнуть оптимізувати процес створення мобільних веб-застосунків, демонструючи потенціал Swift як потужної платформи для комплексних веб-рішень.

## Значення кросплатформної розробки:

- **розширена аудиторія:** доступ до більшої кількості користувачів на різних операційних системах (iOS, Android);
- **зниження витрат:** розробка однієї кодової бази замість декількох, що економить час та ресурси;
- **прискорений вихід на ринок:** можливість швидше випускати продукт, оскільки не потрібно створювати окремі версії для кожної платформи;
- **спрощене оновлення та підтримка:** зміни вносяться один раз і застосовуються до всіх платформ;
- **єдиний UX/UI:** забезпечення однакового досвіду користувача та інтерфейсу на всіх пристроях.

3

## Існуюча модель

$$M = \{UI, BL, DB, API\}.$$

**UI** – інтерфейс користувача;

**BL** – модуль бізнес-логіки;

**DB** – база даних, яка орієнтована на зберігання даних;

**API** – інтерфейс взаємодії клієнтської та серверної частин.

4

## Розроблена уніфікована модель

**M={P, A, PL, UI, UX, SC, DB, NI, S, TS, AM},**

**P** (platform) – платформа розробки;  
**A** (architecture) – архітектура розробки;  
**PL** (programming language) – мова програмування;  
**UI** (user interface) – інтерфейс користувача;  
**UX** (user experience) – досвід користувача;  
**SC** (server component) – серверна складова застосунку;  
**DB** (database) – база даних;  
**NI** (network interface) – мережний інтерфейс;  
**S** (security) – безпека застосунку;  
**TS** (test suite) – набір тестів;  
**AM** (analytics and monitoring).

5

## Пул технологій, який було використано



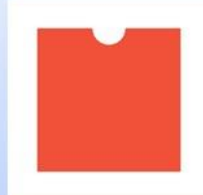
**Swift**  
(мова програмування)



**SwiftUI**  
(фреймворк для побудови UI)



**Skip** (інструмент, який дозволяє створювати нативні програми для iOS та Android з єдиної кодової бази Swift та SwiftUI)



**SwiftWasm**  
(проект, який дозволяє компілювати Swift у WebAssembly (Wasm), щоб запускати Swift-код у браузері)



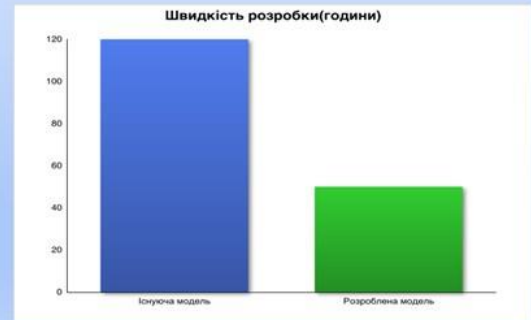
**Vapor**  
(фреймворк який дозволяє будувати бек-енд на мові свіфт)

6

## Переваги уніфікованого прототипу на прикладі розробки застосунку для обліку фінансів

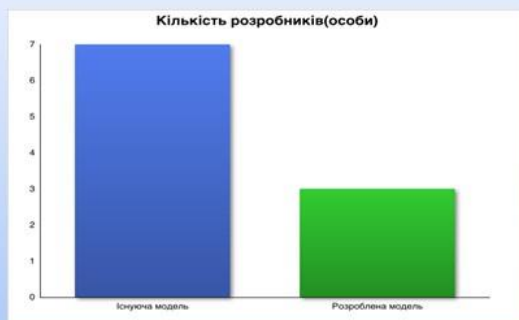
Етап розробки	Існуюча модель (години)	Розроблена модель (години)	Різниця (%)
Розробка UI (iOS)	80	60	-25%
Розробка UI (Android)	80	50	-37.5%
Розробка UI (Web)	70	50	-28.6%
Реалізація бізнес-логіки (BL)	120 (по 40 на платформу)	50 (один раз для всіх)	-58.3%
Налаштування бази даних (DB)	30	25	-16.7%
Розробка та інтеграція API	60	40	-33.3%
Налагодження та інтеграція	50	30	-40%
Загальний час	490	305	-37.8%

$M = \{P, A, PL, UI, UX, SC, DB, NI, S, TS, AM\}$



7

## Переваги уніфікованого прототипу



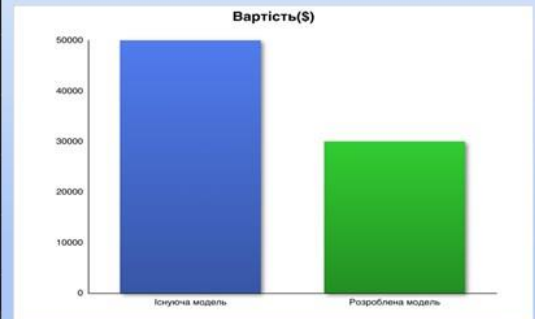
$M = \{P, A, PL, UI, UX, SC, DB, NI, S, TS, AM\}$

Роль у команді	Існуюча модель (к-сть осіб)	Розроблена модель (к-сть осіб)	Різниця
iOS-розробник (Swift)	1	1 (універсальний Swift-розробник)	-
Android-розробник (Kotlin/Java)	1	-	-1
Веброзробник (JavaScript)	1	-	-1
Бекенд-розробник (Python/Node.js)	1-2	-	-1/-2
Спеціаліст із баз даних	1 (опціонально)	-	-1
Тестувальник/DevOps	1 (опціонально)	1 (опціонально)	-
Загальна кількість	5-7	2-3	-3/-4

8

Етап розробки	Існуюча модель (витрати, \$)	Розроблена модель (витрати, \$)	Різниця (%)
Розробка UI (iOS)	80 год × \$50 = \$4000	60 год × \$50 = \$3000	-25%
Розробка UI (Android)	80 год × \$45 = \$3600	50 год × \$50 = \$2500	-30.6%
Розробка UI (Web)	70 год × \$40 = \$2800	50 год × \$50 = \$2500	-10.7%
Реалізація бізнес-логіки (BL)	120 год × \$45 (середнє) = \$5400	50 год × \$50 = \$2500	-53.7%
Налаштування бази даних (DB)	30 год × \$50 = \$1500	25 год × \$50 = \$1250	-16.7%
Розробка та інтеграція API	60 год × \$50 = \$3000	40 год × \$50 = \$2000	-33.3%
Налагодження та інтеграція	50 год × \$50 = \$2500	30 год × \$50 = \$1500	-40%
<b>Загальні витрати</b>	<b>\$22,800</b> (без урахування менеджменту)	<b>\$15,250</b> (без урахування менеджменту)	<b>-33.1%</b>
3 урахуванням команди (5 vs 2)	\$50,000-\$70,000	\$30,000-\$40,000	-40%

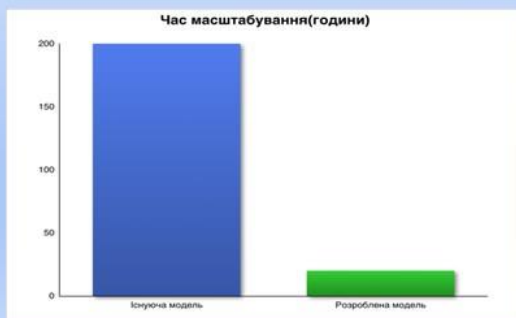
## Переваги уніфікованого прототипу



9

## Переваги уніфікованого прототипу

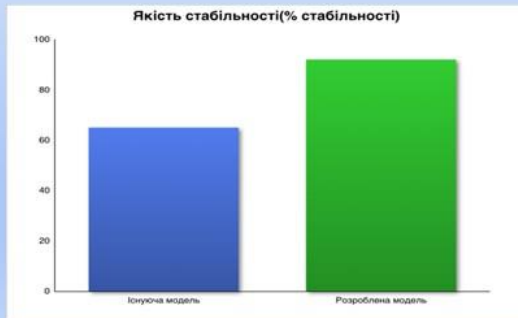
Завдання	Існуюча модель (час)	Розроблена модель (час)	Різниця (%)
Додавання нової платформи (macOS)	2-3 тижні (120-180 год)	1-2 дні (8-16 год)	-90%
Оновлення API (новий ендпоінт)	3-5 днів (24-40 год)	4-6 годин	-85%
Заміна бази даних (SQLite → PostgreSQL)	5-7 днів (40-56 год)	1-2 дні (8-16 год)	-75%
Адаптація UI під нову платформу	1-2 тижні (60-120 год)	1 день (8 год)	-90%
Додавання нового модуля (аналітика)	1-2 тижні (60-120 год)	2-3 дні (16-24 год)	-80%



10

## Переваги уніфікованого прототипу

$M = \{P, A, PL, UI, UX, SC, DB, NI, S, TS, AM\}$



Критерій	Існуюча модель	Розроблена модель
Безпека (S)	Розподілена, часткова	Централізована, через JWT + HTTPS
Тестування (TS)	Обмежене, неузгоджене	Інтегроване: XCTest + Vapor
Єдність технологій	Різнi мови / підходи	Єдина мова Swift
Якість/стабільність коду	Середня	Висока
Автоматизація перевірки	Відсутня	CI + автоматичні тести

11

## Висновки

- проаналізовано сучасні підходи до кросплатформної розробки та виявлено їхні обмеження щодо єдності мовного стеку;
- запропоновано уніфіковану модель для розробки мобільних та веб-застосунків з використанням мови Swift для клієнтської та серверної частин;
- реалізовано повноцінний прототип застосунку для обліку фінансів.

## Переваги запропонованої моделі

- **скорочення витрат та часу** за рахунок використання однієї кодової бази та повторного використання логіки, спрощення оновлень;
- **консистентність UX/UI** на всіх платформах: мобільна та веб-версії виглядають однаково;
- **безпека та стабільність**: застосовано HTTPS, JWT, типобезпечні API, XCTest для тестування;
- **масштабованість**: легке адаптування під iOS, iPadOS, watchOS, macOS, частково під Android;
- **аналітика та моніторинг**: інтеграція відповідного модулю дозволяє відслідковувати поведінку користувачів та покращувати продукт.

*Апробація результатів кваліфікаційної роботи: Оцевик В.А., Філімончук Т.В., Майстренко Г.В., Волк Д.М. Модель уніфікованого підходу для розроблення мобільних і вебзастосунків із використанням мови Swift // Інформаційно-керуючі системи на залізничному транспорті. Харків: УкрДУЗТ, 2025. Випуск №1 (160). С. 59-66.*

12