

? Що ми хочемо дізнатись:

- ♦ Як би ти забезпечив масштабованість додатку, якщо в майбутньому потрібно буде імплементувати синхронізацію чатів через сервер?
 - Спершу я узгодив серверну та локальну моделі, створивши спільний тип (DTO), який дозволяє конвертацію в обидві сторони.
 - По друге я використав протоколи для визначення загальних інтерфейсів між локальними і серверними об'єктами, що дозволило відокремити реалізації та забезпечити більшу гнучкість. Це рішення полегшує підтримку, тестування і масштабування додатку.
 - По третє передбачив механізми кешування та офлайн-підтримки, щоб користувачі могли працювати з чатами навіть без підключення до інтернету.
 - А також запровадив систему обробки конфліктів при синхронізації (конфліктні оновлення, зміни з обох сторін), що критично для коректної роботи в багатокористувацьких сценаріях.
- ♦ Після релізу користувачі пишуть, що додаток крашиться. Що б ти робив? Як би дізнався причину? З чого почав би дослідження?
 - Перевірив crash-репорти в аналітичних сервісах (наприклад, Firebase Crashlytics або Xcode Organizer).
 - Проаналізував, у яких саме сценаріях та на яких етапах використання додатку виникають збої.
 - Повторно переглянув логіку ініціалізації в AppDelegate, SceneDelegate, а також у всіх init-методах основних класів.
 - Протестував додаток на реальних пристроях із різними версіями iOS, зокрема на старіших, де можуть бути особливості в поведінці.
- ♦ Уяви, ти — єдиний iOS-розробник у конкретному продукті. За тиждень — реліз. У тебе не працює карта, не завершено чат, і немає локалізації. Що зробиш у першу чергу? А що можеш відкласти чи взагалі не робити?
 - Разом з продакт-менеджером або самостійно вирішую: що є ключовою функцією — карта чи чат. Саме з цього і починаю.
 - Працюю над основним функціоналом до стабільної MVP-версії. Якщо є частини, які можна приховати або відкласти — роблю це.
 - Після основного блоку беруся за другий важливий модуль.
 - Якщо залишився час — додаю підтримку мов. Якщо ні — залишаю дефолтну мову та планую додати переклади в наступному релізі.

♦ У додатку є фідбек від користувача: “Кнопка не працює”, але баг не відтворюється. Що ти зробиш, щоб розібратись у проблемі?

1. З’ясувати деталі у користувача:

- Запитати, на якій саме кнопці він натискає.
- Який у нього пристрій (модель, ОС, версія додатку)?
- Чи є якісь помилки або повідомлення на екрані?
- В якій саме ситуації кнопка “не працює” (наприклад, під час певної дії, після оновлення)?

2. Перевірити лог-файли і аналітику:

- Подивитись, чи фіксуються помилки у логи під час спроби натискання кнопки.
- Перевірити, чи дійсно натиснення кнопки доходить до бекенду або тригерить подію.

3. Спробувати відтворити проблему в різних середовищах:

- На різних пристроях і ОС.
- З різними користувацькими сценаріями.

♦ Фіча, над якою ти працюєш, залежить від бекенду, який ще не готовий. Що будеш робити?

- Реалізую UI-частини.
- Напишу UI- та Unit-тести.
- Якщо бекенд ще не готовий, погоджую формат API: запити, поля, типи даних, обробку помилок.
- Уточнюю очікувану поведінку: які статус-коди повертаються, чи є затримки, чи потрібна авторизація тощо.
- Готую структуру сторінок, навігацію, валідацію, UX-деталі.
- Якщо потрібно — реалізую кешування, стани завантаження (loading states) та обробку помилок.

♦ Які бібліотеки чи фреймворки ти вивчав останніми? Що тобі в них сподобалось? Які переваги? Що хотів би вивчити найближчим часом?

Останнім часом я займався вивченням XCTest — це стандартний фреймворк для написання unit- та UI-тестів у Xcode. Його переваги:

- Вбудований у Xcode, не потребує сторонніх залежностей.
- Добре інтегрується з CI/CD, наприклад, GitHub Actions.

- Підтримує `@testable import`, що дає доступ до внутрішніх елементів модуля для тестування.
- Підтримує UI-тести без складної конфігурації.

Наразі хочу глибше освоїти можливості XCTest, щоб мати змогу писати більш складні, повноцінні тести — з покриттям edge-case сценаріїв, асинхронною логікою та UI-взаємодіями.

♦ Є проєкт, код або рішення, яким хочеш поділитись? Прикріпи посилання або просто опиши — будемо раді подивитись.

[Trailcraft](#) – Smart Map Explorer.