**ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ДОДАТКУ УНІВЕРСАЛЬНОГО СКАНЕРА/ГЕНЕРАТОРА КОДІВ З ІСТОРІЄЮ ПОШУКУ
ТА СИНХРОНІЗАЦІЄЮ ЧЕРЕЗ ICLOUD**

**Арсенюк Ігор Ростиславович**

к.т.н., доцент

**Кмитюк Дмитро Сергійович,**

студент

Вінницький національний технічний університет

м. Вінниця, Україна

**Анотація**. У роботі проведено аналіз та обґрунтування доцільності розробки універсального сканера/генеартора кодів з історією пошуку та синхронізацією через iCloud. Розглянуто ключові аспекти, які впливають на функціональність та користувацький досвід такого додатку, включаючи адаптивний дизайн, розробку локальної бази даних та синхронізацію хмарою.

**Ключові слова**: Сканер кодів, генератор кодів, синхронізація з iCloud, мультисканування, машинне навчання.

Метою роботи є покращення користувацького досвіду у сфері сканування та генерування кодів. Для швидкого сканування використовується інструмент CoreML.

Розробка додатку універсального сканера/генератора кодів [1] з історією пошуку та синхронізацією через iCloud є актуальним завданням у контексті швидкого розвитку електронної комерції. Інтерфейс такого додатку охоплює як програмне забезпечення так і складові для зчитування та обробки інформації. Залежно від потреб користувача, додаток може реалізувати різний функціонал, а саме: сканування одночасно декількох кодів різних типів та генерація різних кодів, надання користувачеві можливості переглянути свої коди, за допомогою історії пошуку, зручно налаштувати додаток під свої потреби та синхронізувати свої дані через хмарне середовище iCloud.

Задачею дослідження є розробка та реалізація додатку універсального сканера/генератора кодів з історією пошуку та синхронізацією через iCloud. Основні завдання, що потрібно вирішити: аналіз потреб вимог користувачів у сфері комунікацій через коди, розробка зручного інтерфейсу, підключення локальної бази даних для перегляду історії пошуку, інтеграція з хмарним середовищем iCloud

Додаток сканера/генератора, що включає можливість сканування та генерування усі типи кодів є важливим інструментом для кожного, адже в сучасності усі люди використовують коди для комунікацій. Розглянемо детальний опис функціоналу сканера:

Сканування кодів виконується автоматично, за допомогою штучного інтелекту. Кожен просканований код потрапить до історії, з метою її збереження.

Генерація кодів виконується користувачем. Користувач може згенерувати будь-який тип коду та кастомізувати його під себе. Усі згенеровані коди будуть зберігатися в історії пошуку.

Історія пошуку, що являє собою локальну базу даних [2] є важливим аспектом для кожного користувача, адже він зможе легко переглянути усі коди, з якими йому довелось працювати. Також передбачено можливість поширення обраних кодів у соцмережах [3] .

Кожному користувачеві буде надана можливість кастомізувати додаток під свої потреби за допомогою, зокрема, таких налаштувань: перемкнутим додаток на нічний режим, змінити мову, активувати синхронізацію з хмарою, увімкнути або вимкнути режим мультисканування та автоматичного сканування.

Для розробки додатку універсального сканера/генератора кодів з історією пошуку та синхронізацією через iCloud доцільно використати архітектуру MVVM (Model View Viewmodel) [1], яка є однією із традиційних для iOS розробки, де бізнес логіка та елементи інтерфейсу розміщені у різних модулях, що дозволить легко та швидко замітити дизайн, і/або логіку, не чіпаючи при цьому інші модулі. У випадку використанні фреймворку SwiftUI для мови Swift, MVVM архітектура матиме такий склад:

**1. Інтерфейс (SwiftUI).** Цей фреймворк дозволяє швидко створювати будь які UI-елементи з декларативним підходом. Особливість SwiftUI полягає в тому, що він є надшвидким, порівняно з іншими фрейморками. Це забезпечить стабільну роботу інтерфейсу.

**2. Локальна база даних (CoreData).** Цей фреймворк є нативним для iOS, що дозволить нам не інтегрувати зайві залежності а також CoreData може вміщувати більше ніж 1ТБ інформації.

**3. Штучний інтелект (MLKit).** Інструмент, який випустив Google, що дозволяє виконувати миттєве сканування до 100 кодів одночасно, що забезпечить гарний досвід користувача у додатку [5].

**4. Хмарне середовище (iCloud)**. Використавши хмару, користувач отримає можливість переглянути свої коди на іншому пристрої віддалено. Це забезпечить приємний досвід, при переході на інший девайс [6].

 Архітектура MVVM дозволяє легко розгортати та масштабувати додаток, при цьому може бути більш гнучкою у порівнянні з іншими архітектурами.

У результаті дослідження визначено, що розробка універсального сканера/генератора кодів з історією пошуку та синхронізацією через iCloud є доцільною. Цей процес має спростити усі комунікації, розпочинаючи від бізнес-зустрічей, закінчуючи банальним скануванням меню у закладі.

Отже, враховуючи вище наведене, планується реалізація додатку сканера/генератора, що відповідає сучасним вимогам електронної комерції та має за мету забезпечення зручного та привабливого користувацького інтерфейсу.

## СПИСОК ЛІТЕРАТУРИ

1. Types of barcodes - https://www.scandit.com/resources/guides/types-of-barcodes-choosing-the-right-barcode/
2. Core Data – Apple Documentation - https://developer.apple.com/documentation/coredata/
3. Як працювати з Core Data - https://dou.ua/forums/topic/38832/
4. iOS Architecture - A Comprehensive Guide - https://intellipaat.com/blog/tutorial/ios-tutorial/ios-architecture/#:~:text=The%20architecture%20of%20IOS%20is,of%20well%2Ddefined%20system%20interfaces.
5. Migration for iOS - https://developers.google.com/ml-kit/migration/ios?hl
6. iCloud Drive setup - https://support.apple.com/ru-ru/guide/icloud/mm203b05aec8/icloud