Wypuszczenie aplikacji od zera. [IOS]

1. Stworzenie projektu flutter
2. Stworzenie repozytorium na aplikacje flutter.
3. **Utwórz nowy identyfikator aplikacji:**
    a. Wybierz **Certificates, Identifiers & Profiles**.
    b. W sekcji **Identifiers** kliknij przycisk „+” i wybierz **App IDs**.
    c. Wprowadź nazwę aplikacji (np. WealthTrack) oraz ustaw **Bundle Identifier** na **com.domainname.WealthTrack** (w razie potrzeby zmodyfikuj go, aby pasował do Twojego schematu).
    d. **Zakończ proces rejestracji** i potwierdź utworzenie App ID.
4. W projekcie fluttera zmień wszystkie wystąpienia zmiennej „PRODUCT_BUNDLE_IDENTIFIER” na **com.domainname.WealthTrack.**
5. Stwórz nową aplikacje w App store connect – podepnij tam stworzony identifier.
6. W serwisie codemagic dodaj aplikacje „Add Application”
    a. Wybierz Github
    b. Wybierz repozytorium
    c. Wybierz Flutter (via Workflow Editor)
7. Konfiguracja na codemagic
    a. Build
        i. Mode – Release
        ii. --build-name=1.0.$PROJECT_BUILD_NUMBER --build-number=$PROJECT_BUILD_NUMBER
    b. Distribution
        i. IosCodeSigning -> Automatic / Codemagic API key / AppStore / WealthTrack
        ii. AppStoreConnect -> Enable App / Codemagic Api key / Submit to TestFlight
    c. Billing – Personal Account – User settings – Code Signing Identities
        i. Generate Certificate – Apple Distribution
        ii. Przejść na Apple–Certificates,Identifiers & Profiles – Profiles - + - Ad Hoc – WealthTrack – No – Continue – Zapisać klucz – wrzucic go na codemagic.
8. Przejść na Apple – Aplikacja WealthTrack – TestFlight
    a. Internal Testing dodać konto
    b. TestInformation dodać feedbackEmail i Contact Information.
9. Po dodaniu builda trzeba potwierdzić ‘missing compliance’ dot. Enkrypcji, żeby tego uniknąć trzeba dodać: Do "<rootdir>\ios\Runner\Info.plist" -> <key>ITSAppUsesNonExemptEncryption</key><false/>

Jesteśmy gotowi na wrzucenie aplikacji 😊

Wypuszczenie aplikacji od zera. [Android]

1. Inicjalizacja repozytorium i integracja z Codemagic

- Stwórz repozytorium projektu na GitHub/GitLab/Bitbucket.
- Połącz repozytorium z platformą Codemagic.

2. Generowanie keystore (Windows PowerShell jako administrator)

Wykonaj poniższe polecenie, aby wygenerować plik keystore:

keytool -genkey -v -keystore $env:USERPROFILE\<NazwaProjektu>-keystore.jks `

-storetype JKS -keyalg RSA -keysize 2048 -validity 10000 `

-alias <NazwaProjektu>

Podczas procesu:

- Ustaw hasło (np. ProgrammingFlashcards).
- Przeklikaj enter we wszystkich pytaniach.
- Użyj tego samego hasła dla klucza.

Przykład:

- Nazwa pliku: ProgrammingFlashcards-keystore.jks
- Alias: ProgrammingFlashcards
- Hasło: ProgrammingFlashcards

3. Konfiguracja keystore w projekcie Android

Przenieś plik *ProgrammingFlashcards-keystore.jks* do katalogu:

android\app

Następnie utwórz plik *key.properties* w katalogu:

android\gradle

Dodaj do niego następujące dane:

storePassword=ProgrammingFlashcards

keyPassword=ProgrammingFlashcards

keyAlias=ProgrammingFlashcards

storeFile=ProgrammingFlashcards-keystore.jks

4. Aktualizacja identyfikatora aplikacji

Zmień domyślną nazwę pakietu:

com.example.<projectName> -> com.domainname.<projectName>

5. Konfiguracja pliku build.gradle

Projekt może korzystać z pliku:

- android\app\build.gradle (dla Groovy)

```
def keystoreProperties = new Properties()
    def keystorePropertiesFile = rootProject.file('key.properties')
    if (keystorePropertiesFile.exists()) {
        keystoreProperties.load(new FileInputStream(keystorePropertiesFile))
    }
```

```
    signingConfigs {
        release {
            keyAlias keystoreProperties['keyAlias']
            keyPassword keystoreProperties['keyPassword']
            storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null
            storePassword keystoreProperties['storePassword']
        }

    }

    buildTypes {


        release {
            // TODO: Add your own signing config for the release build.
            // Signing with the debug keys for now, so `flutter run --release` works.
            signingConfig = signingConfigs.release
        }
    }
```

- android\app\build.gradle.kts (dla Kotlin DSL)

https://github.com/AITripleCodeCorp/ProgrammingFlashcards/commit/3d08516b6bf90ce1e5a4f136292ef9a872f89cfd

6. Konfiguracja publikacji w Google Play

Podążaj za instrukcjami z dokumentacji Codemagic, aby:

- Wygenerować plik .json z danymi uwierzytelniającymi.
- Dodać konto do aplikacji w Google Play Console.

7. Konfiguracja Codemagic

1. Przejdź do Codemagic i włącz podpisywanie kodu:
    - **Android Code Signing → Enable Android Code Signing**
    - Prześlij plik keystore.jks i podaj hasło oraz aliasy.
2. Włącz publikację do Google Play:
    - **Google Play → Enable Google Play Publishing**
    - Prześlij plik credentials.json.
    - Wybierz internal jako ścieżkę dystrybucji.
    - Zaznacz opcję **Submit release as draft**.

8. Uruchomienie pierwszego builda

- Uruchom pierwszego builda ręcznie.
- Po pomyślnym przejściu procesu weryfikacji, kolejne buildy mogą być automatycznie publikowane.

9. Dodanie builda do Google Play Console

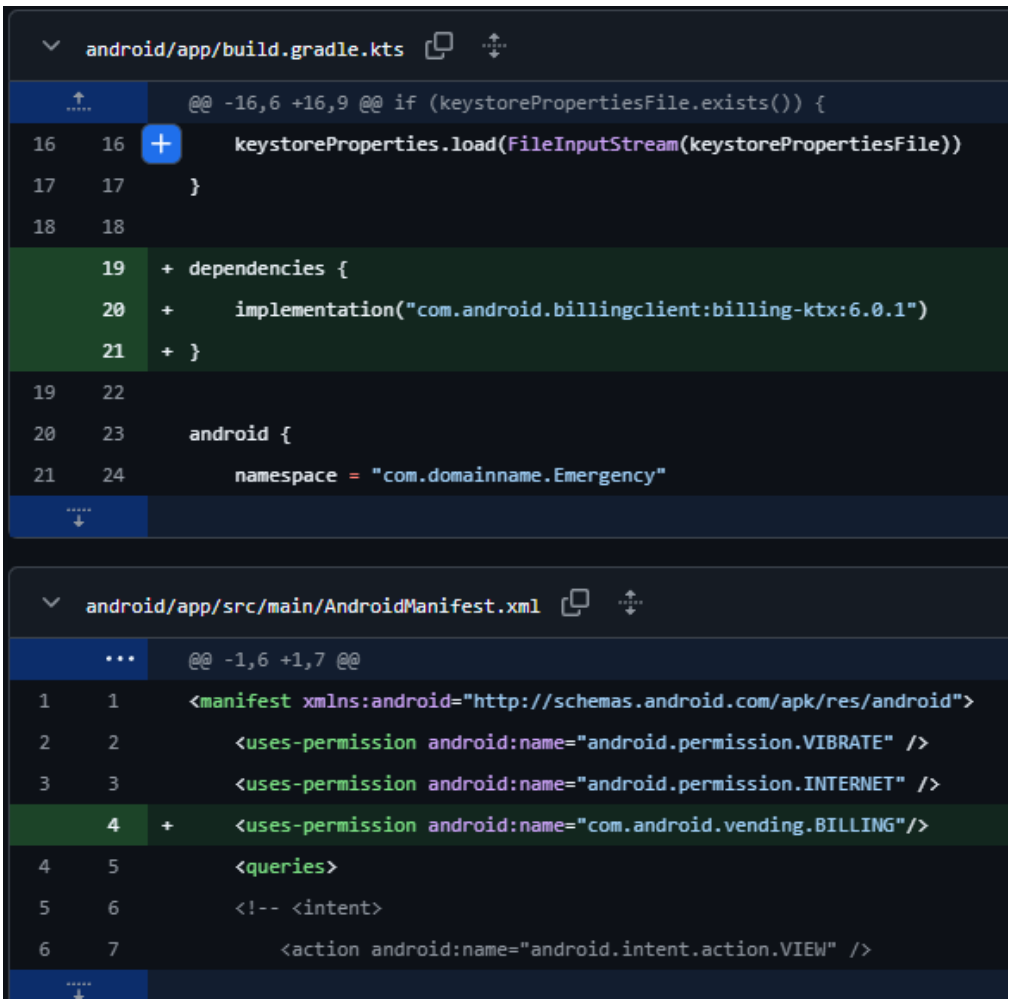
1. Przejdź do **Google Play Console**.
2. Przejdź do sekcji **Testowanie** → **Test wewnętrzny**.
3. Wybierz testerów z utworzonej listy testowej.
4. Utwórz nową wersję.
5. Prześlij plik .aab pobrany z Codemagic.
6. Kliknij **Dalej**, a następnie **Zapisz i opublikuj**.

Jesteśmy gotowi na wrzucenie aplikacji ponownie 😊

Android dodanie subskrypcji

https://github.com/AITripleCodeCorp/Emergency/commit/e1fdd0862c7f87d0bae537643e5381952e2d1c68

https://medium.com/@rita20071995/a-step-by-step-guide-to-adding-subscriptions-in-flutter-projects-using-revenuecat-bc2e975d6576



Wymagania do screenshotu do review IOS:

```
▼ imageValidationData: Array(34)
    0: "2732x2048"
    1: "2048x2732"
    2: "1280x800"
    3: "1440x900"
    4: "2880x1800"
    5: "2560x1600"
    6: "1920x1080"
    7: "3840x2160"
    8: "312x390"
    9: "1136x640"
    10: "1136x600"
    11: "640x1136"
    12: "640x1096"
    13: "2436x1125"
    14: "1125x2436"
    15: "1024x1024"
    16: "960x640"
    17: "960x600"
    18: "640x960"
    19: "640x920"
    20: "1024x768"
    21: "1024x748"
    22: "768x1024"
    23: "768x1004"
    24: "2048x1536"
    25: "2048x1496"
    26: "1536x2048"
    27: "1536x2008"
    28: "1334x750"
    29: "750x1334"
    30: "2224x1668"
    31: "1668x2224"
    32: "2208x1242"
    33: "1242x2208"
```