Dokumentacja aplikacji Unordered

Adam Dybcio
Łukasz Czapski
Igor Ciżewski
Denis Jabłoński
Mateusz Sztankiewicz

22 maja 2025

Spis treści

1 Wprowadzenie

1.1 Cel dokumentu

Dokumentacja ma na celu przedstawienie architektury oraz funkcjonalności aplikacji Unordered. Zawiera również informacje o integracji z OpenAI oraz wdrożeniu aplikacji.

1.2 Wyjaśnienie pojęć

Box W kontekście aplikacji Unordered, *box* to spersonalizowany zestaw prezentowy, generowany automatycznie na podstawie profilu osoby obdarowywanej, okazji, budżetu lub wyboru losowego. Box zawiera propozycje konkretnych produktów, które razem tworzą gotowy do wręczenia prezent. Każdy box jest unikalny i dopasowany do wybranych przez użytkownika kryteriów.

AI Sztuczna inteligencja, w kontekście aplikacji Unordered odnosi się do wykorzystania modeli językowych OpenAI (np. o4-mini) do generowania spersonalizowanych propozycji prezentów na podstawie kryteriów podanych przez użytkownika.

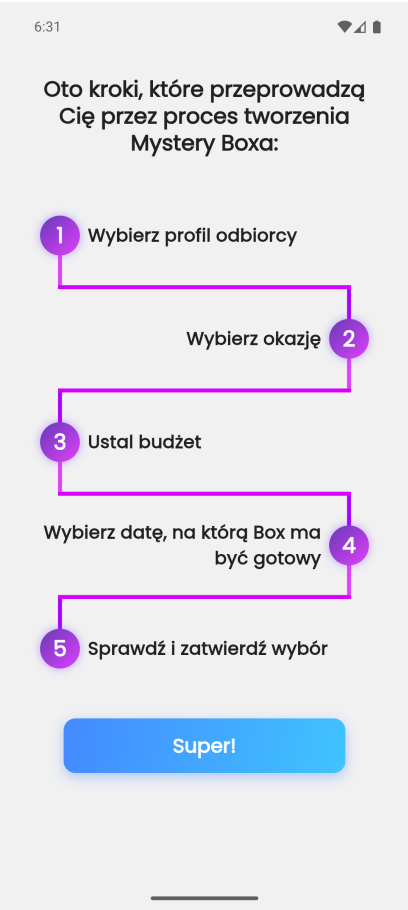
1.3 Opis aplikacji

Unordered to nowoczesna aplikacja mobilna, która pomaga użytkownikom w szybkim i trafnym wyborze prezentów na różne okazje, takie jak urodziny, rocznice czy święta. Głównym celem aplikacji jest automatyzacja procesu doboru prezentów oraz przypominanie o ważnych datach, eliminując problem braku pomysłów, czasu czy zapominania o okazjach.

Użytkownik może tworzyć profile osób, dla których planuje prezenty, określając ich zainteresowania i preferencje. Następnie, dzięki integracji z AI (OpenAI GPT), aplikacja generuje spersonalizowane propozycje prezentów (tzw. boxy), dostosowane do profilu, okazji oraz budżetu. Możliwe jest także ustawienie cyklicznych boxów, które automatycznie przypominają i generują prezenty na wybrane daty.

Unordered wyróżnia się pełną automatyzacją - od analizy profilu, przez dobór produktów z wielu platform zakupowych, aż po przypomnienia o nadchodzących okazjach. Dzięki temu użytkownik nie musi samodzielnie przeszukiwać sklepów ani pamiętać o terminach - aplikacja zrobi to za niego, zapewniając jednocześnie element zaskoczenia i personalizacji.

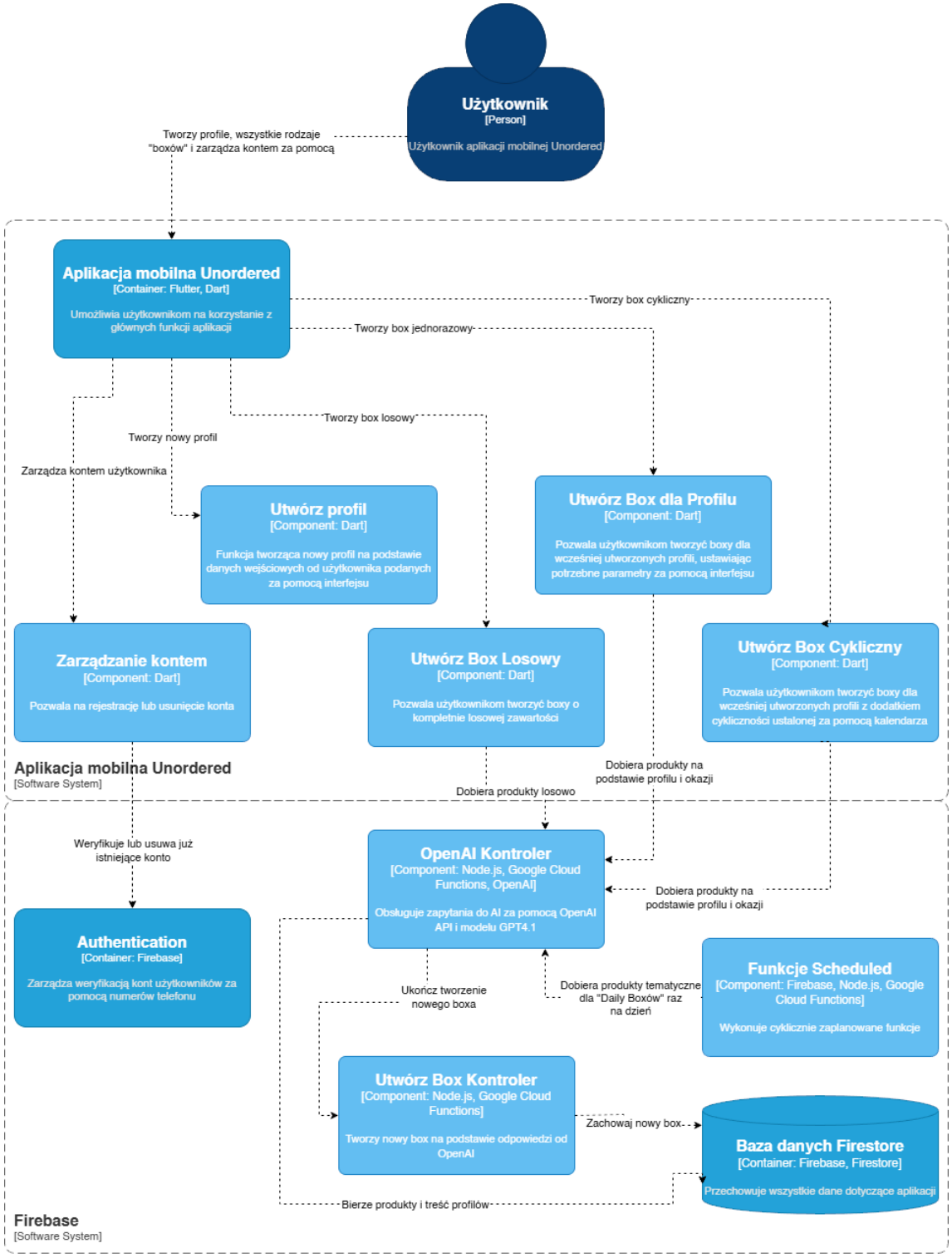
Aplikacja jest skierowana zarówno do osób prywatnych, jak i firm, które chcą zadbać o swoich pracowników czy bliskich, oferując im wyjątkowe, dopasowane prezenty bez zbędnego wysiłku.



Rysunek 1: Opis generacji boxów dla profilu

2 Architektura

2.1 Diagram wysokopoziomowy



Rysunek 2: Architektura systemu

2.2 Frontend (Flutter)

Aplikacja mobilna Unordered została zrealizowana w technologii Flutter, co umożliwia jej uruchamianie zarówno na systemie Android, jak i iOS. Flutter pozwala na szybkie prototypowanie oraz łatwe wdrażanie zmian w interfejsie użytkownika.

- **Główne pakiety:**
  - `firebase_core` – integracja z Firebase
  - `firebase_auth` – obsługa uwierzytelniania użytkowników
  - `cloud_functions` – wywoływanie funkcji chmurowych backendu
  - `cloud_firestore` – dostęp do bazy danych Firestore
  - `http` – komunikacja z zewnętrznymi API
- **Główne funkcjonalności:**
  - Rejestracja i logowanie użytkownika (SMS/Firebase)
  - Tworzenie i edycja profili użytkowników
  - Generowanie boxów prezentowych na podstawie profilu, cyklicznie lub losowo
  - Przeglądanie historii wygenerowanych boxów
  - Przeglądanie kalendarza z zaplanowanymi boxami
  - Integracja z backendem (Cloud Functions, Firestore)

Rysunek 3: Kreator boxów

2.3 Backend (Firebase)

- **Baza danych:** Firestore – przechowuje dane użytkowników, zapisane profile, utworzone boxy, listy produktów oraz wszelkie inne dane niezbędne do działania aplikacji.
- **Funkcjonalność:** Cloud Functions (Node.js) – obsługuje główne działania backendu aplikacji, takie jak generowanie boxów, komunikacja i przetwarzanie danych z OpenAI oraz pobieranie aktualnej listy produktów.
- **Uwierzytelnianie:** Firebase Authentication – pozwala użytkownikom na rejestrację i logowanie się do aplikacji za pomocą numeru telefonu, używając weryfikacji SMS.
- **Automatyzacja:** Cloud Functions Scheduler – służy do cyklicznego uruchamiania funkcji, obsługuje m.in. generowanie "Daily Boxów" – codziennych propozycji prezentów o losowych kategoriach.

3 Funkcjonalności

3.1 Generowanie boxów

3.1.1 Typy generowania

- **Dla profilu** - Generacja boxów na podstawie profilu, który został wcześniej utworzony. Użytkownik podaje budżet oraz okazję, na którą box ma być wygenerowany.
- **Cykliczny** - Generacja boxów cyklicznych, które będą generowane co określony czas (np. co miesiąc, co rok). Wymagane parametry jak w przypadku boxów "Dla profilu".
- **Losowy** - Generacja losowych propozycji prezentów, bez podawania żadnych parametrów
- **Manualny** - aktualnie brak implementacji

3.1.2 Profil użytkownika

Do wygenerowania boxów typu "Dla profilu" lub "Cykliczny" wymagane jest utworzenie profilu, dla którego dobierane będą propozycje. Dane profilu to:

- **Nazwa profilu** (np. "Mama", "Tata", "Kolega")
- **Opis osoby** (Preferencje, zainteresowania, itp.)
- **Zainteresowania** (Wybór kategorii z gotowej listy)

3.2 Struktura boxa

Struktura boxów jest taka sama dla wszystkich typów generowania. Boxy są przechowywane w bazie danych Firestore i zawierają następujące dane:

- **Budżet** (kwota, którą użytkownik chce przeznaczyć na prezenty)
- **Data wygenerowania** (data, kiedy box został wygenerowany)
- **Czy jest cykliczny** (wartość logiczna, czy box jest cykliczny)
- **Frekwencja cykliczności** (używane tylko dla boxów cyklicznych, np. co miesiąc, co rok)
- **Okazja** (okazja, na którą box został wygenerowany, np. urodziny, imieniny)
- **Planowana data** (data, na którą box został zaplanowany)
- **Cena** (suma cen wszystkich produktów w boxie)
- **Lista produktów** (produkty, które zostały wygenerowane w boxie)

Rysunek 4: Widok wygenerowanego boxa

4 Integracja z OpenAI

4.1 Opis API

Wybranym do generacji modelem jest **OpenAI o4-mini**. Wybór modelu jest uzasadniony jego działaniem w trybie "reasoning" który pozwala na generacje biorącą pod uwagę wszystkie parametry podane przez użytkownika. Jest on również tańszym wyborem, w porównaniu do modeli **GPT-4.1** oraz **OpenAI o3**.

4.2 Prompt engineering

Przy inżynierii promptów uwaga była skupiona na wykorzystaniu najmniejszej liczby tokenów przy zachowaniu jak najlepszej jakości odpowiedzi. Strukturą wszystkich promptów jest: **ZASADA→OPIS ZASADY**

- **Generowanie boxów dla profilu** - generacja boxów na podstawie profilu użytkownika.

  ANALIZA→Uwzględnij zainteresowania, okazję (...)
  RÓŻNORODNOŚĆ→Unikaj powtórzeń, (...)
  BUDŻET→Nigdy nie przekraczaj budżetu!(...)
  ODPOWIEDŹ→Zwróć TYLKO JSON array (...)

- **Generowanie "Daily Boxów"** - generacja boxów o różnych tematach, raz dziennie.

  DOBÓR→Uwzględnij temat, unikaj innych kategorii. (...)
  RÓŻNORODNOŚĆ→Unikaj powtórzeń, (...)
  BUDŻET→Nigdy nie przekraczaj budżetu! (...)
  ODPOWIEDŹ→Zwróć TYLKO JSON array (...)

- **Generowanie losowych boxów** - generacja boxów bez podawania żadnych parametrów.

  LOSOWOŚĆ→Wybierz losowe przedmioty (...)
  RÓŻNORODNOŚĆ→Dobieraj różne kategorie.
  BUDŻET→Utrzymaj realistyczny budżet.
  ODPOWIEDŹ→Zwróć TYLKO JSON array (...)

4.3 Przetwarzanie odpowiedzi

Formatem zwróconej odpowiedzi jest JSON array, który zawiera listę samych ID produktów, aby zminimalizować liczbę tokenów. Następnie ID są sprawdzane oraz przetwarzane przez backend.

5 Wdrożenie

5.1 Konfiguracja Firebase

Aby wdrożyć funkcje chmurowe (Cloud Functions) za pomocą Firebase, należy użyć narzędzia `firebase` dostępnego przez npm, oraz zainicjować projekt i wdrożyć funkcję do projektu Firebase. Po wdrożeniu należy upewnić się, że adresy (linki) do nowych funkcji są zgodne zarówno w kodzie backendu (Cloud Functions), jak i w wywołaniach po stronie frontendu Flutter. Spójność nazw i endpointów jest kluczowa dla poprawnej komunikacji między aplikacją a backendem.

6 Podsumowanie i dalszy rozwój

Aplikacja wyróżnia się na tle konkurencji pełną automatyzacją, możliwą integracją z platformami zakupowymi oraz możliwością personalizacji prezentów na podstawie profilu obdarowywanej osoby. Dzięki temu użytkownik zyskuje pewność, że prezent będzie trafiony, a cały proces przebiegnie sprawnie i bez zbędnego wysiłku.

6.1 Znane ograniczenia

- Ograniczenia i limity związane z wykorzystaniem API OpenAI (np. limity tokenów, koszty).
- Limity Firebase dotyczące liczby operacji, przechowywania danych i autoryzacji.
- Dostępność produktów zależna od integracji z zewnętrznymi platformami zakupowymi.
- Brak pełnej automatyzacji zamówień fizycznych (na obecnym etapie aplikacja generuje propozycje, nie realizuje zamówień).

6.2 Roadmap

Wśród planowanych kierunków rozwoju aplikacji znajdują się:

- Integracja z dodatkowymi platformami zakupowymi, co pozwoli na jeszcze szerszy wybór produktów.
- Współpraca z markami i sklepami – możliwość tworzenia dedykowanych boxów z konkretnych sklepów lub marek.
- System subskrypcji – użytkownik mógłby otrzymywać cykliczne, personalizowane boxy za miesięczną lub roczną opłatą.
- Rozbudowa funkcji kalendarza i przypomnień, w tym integracja z kalendarzem Google lub Outlook.
- Współpraca z influencerami i działania marketingowe skierowane do młodych, zabieganych osób.
- Usprawnienie procesu zamawiania – możliwość bezpośredniego zamówienia produktów z poziomu aplikacji.
- Rozszerzenie funkcjonalności AI o jeszcze lepsze dopasowanie prezentów do profilu użytkownika.

Unordered jest projektem otwartym na dalszy rozwój i nowe pomysły, a jego architektura pozwala na łatwe wdrażanie kolejnych funkcjonalności w odpowiedzi na potrzeby użytkowników i rynku.

Literatura

[1] Flutter documentation, `https://flutter.dev/docs`

[2] Firebase documentation, `https://firebase.google.com/docs`

[3] OpenAI API reference, `https://platform.openai.com/docs/api-reference`