

# Systemy Mobilne

## Aplikacja LOGISTICKY

Bartłomiej Motyl 224513, Mateusz Michalec 224510

### Repozytoria:

API: <https://github.com/1cb8c1/DystproAPI>

Aplikacja: <https://github.com/MMichalec/Logisticky>

### Cel i przeznaczenie wraz z opisem funkcjonalności

Dystrybutor poprzez aplikację z systemem autoryzacji ma zapewnioną możliwość samodzielnego i bezpośredniego zarządzania wysyłkami bez konieczności konsultacji z biurem obsługi klienta producenta. Dystrybutor posiada stały dostęp do stanów magazynowych, przyznanych mu ofert cenowych oraz wysyłek i rezerwacji.

- Podgląd dostępności materiałów na poszczególnych magazynach, ofert cenowych, wysyłek, rezerwacji, kierowców oraz pojazdów
- Dodawanie do koszyka/rezerwacja asortymentu na określony czas.
- Tworzenie/anulowanie wysyłek.
- Dodawanie/edycja floty pojazdów oraz kierowców odbierających/dostarczających zamówienia

Grupą docelową są pracownicy oficjalnych partnerów biznesowych Producenta czyli składy budowlane/dystrybutorzy materiałów budowlanych.

### Etapy pracy

Pracę nad programem rozpoczęliśmy od wybrania tematyki. Następnie dokonaliśmy mock-up'u UI oraz projektu UX co pozwoliło nam określić dokładnią funkcjonalność aplikacji i w jaki sposób ma przebiegać praca z aplikacją. Dużym ułatwieniem (z którego zdaliśmy sobie sprawę w toku pracy nad

programem) było zwizualizowanie ekranów dzięki czemu wiedzieliśmy jakie dane ma dostarczać serwer w danej chwili co ułatwiło przygotowywanie API.

Kolejnym krokiem było przygotowanie warstwy widoku - utworzenie wszystkich ekranów i połączenie ich w intuicyjny do nawigowania sposób.

Następnie zajęliśmy się przygotowaniem bazy danych oraz API - dzięki projektowi UI/UX dokładnie wiedzieliśmy jakie dane będą potrzebna na każdym ekranie aplikacji. W toku prac dokonaliśmy kilku zmian względem pierwotnej wersji w celu zastosowania lepszych/prostszych rozwiązań.

W międzyczasie implementowaliśmy warstwę danych i logiki w aplikacji mobilnej. Z wykorzystaniem przygotowanego API dostarczaliśmy odpowiednie dane do warstwy widoku.

Na końcu dodaliśmy animacje, poprawiliśmy drobne niedociągnięcia z warstwy widoku, testowaliśmy działanie aplikacji na urządzeniach o różnych wyświetlaczach. Przeprowadziliśmy testy edge-casów i upewniliśmy się czy aplikacja jest odpowiednio zabezpieczona przed nieporządnym zachowaniem użytkownika tj. próba utworzenia wysyłki z datą z przeszłości, tworzenie wysyłek bez kierowców/pojazdów/towarów etc.

## Doświadczenia po wybranym środowisku

### Kotlin

Zdecydowaliśmy się na napisanie aplikacji w języku Kotlin ze względu na to, że aktualnie jest to oficjalnie rekomendowane środowisko tworzenia natywnych aplikacji pracujących na urządzeniach wyposażonych w systemie Android. Dodatkowym atutem Kotlin'a jest bardzo duża zbieżność z językiem Java, możliwe nawet jest łączenie tych dwóch środowisk w ramach jednej aplikacji.

Zalety:

- Duże podobieństwo do Javy
- Kompatybilność z Javą
- Uproszczona składnia względem Javy
- Rekomendowane środowisko do tworzenia aplikacji natywnych na systemy Android
- Pełna integracja języka oraz jego możliwości z AndroidStudio
- Duża ilość bibliotek

Wady:

- W porównaniu do Javy zdecydowanie mniejsza ilość źródeł informacji - co wynika oczywiście z tego, że Kotlin jest o wiele młodszy językiem programowania od Javy, a sama Java jest

ekstremalnie popularna. Warto zaznaczyć, że pomimo różnicy w składni IDE AndroidStudio w większości sytuacji jest w stanie przekonwertować fragmenty kodu napisane w Javie na język programowania Kotlin.

- System przeznaczony wyłącznie dla urządzeń z systemem Android. Aby uruchomić aplikację na urządzeniach z systemem iOS należałoby ją przepisać.