

Państwowa Wyższa Szkoła Zawodowa w Legnicy  
Wydział Nauk Technicznych i Ekonomicznych  
Kierunek Informatyka, Rok IV, n4PAM, studia niestacjonarne

---

DEKLARACJA PROJEKTU NA ZALICZENIE PRZEDMIOTU  
PROGRAMOWANIE URZĄDZEŃ MOBILNYCH  
APLIKACJA WSPOMAGAJĄCA ZAMAWIANIE  
POSIŁKÓW "MNIAMCIU"

---

*Autorzy:*

SEBASTIAN WIECHEĆ, KAROL RECHMAN, MONIKA TUZEL,  
ADRIAN MASTALERZ



*Prowadzący:*  
mgr inż. Marcin Tracz

## Krótki opis projektu

Celem naszego zespołu jest stworzenie aplikacji mobilnej wspomagającej zamawianie posiłków za pośrednictwem Internetu. Aplikacja będzie umożliwiała wyszukiwanie na mapie, dodanie restauracji do zakładki *Ulubione* oraz fotografowanie zamówionych posiłków, a także nawigację. Ponadto, aplikacja będzie mieć możliwość dokonywania płatności za zamówienie (zakłada się pokazową możliwość dokonania płatności podłączoną kartą testową). Aplikacja pozwoli dokonywać płatności zbiorczej za zamówienia dokonywane z kilku restauracji na raz. Wśród analogicznych rozwiązań, funkcjonujących już na rynku, można wskazać aplikację Pyszne.pl

## Lista osób w grupie, podział ról

Prace przy projekcie przebiegała według prezentowanego poniżej podziału ról:

- Adrian Mastalerz - dokumentacja, programowanie, testy behawioralne
- Karol Rechman - lider zespołu, podział zadań, programowanie
- Monika Tuzel - makiety, style interfejsu użytkownika, wsparcie
- Sebastian Wiecheć - wdrożenie i obsługa Google Firebase wraz z wtyczkami do projektu

## Opis funkcjonalny systemu

Po uruchomieniu aplikacji ukazuje się ekran z przyciskami logowania lub rejestracji. Rejestracji można dokonać jako użytkownik aplikacji oraz użytkownik - właściciel restauracji. Po zakończeniu procesu rejestracji (podanie adresu e-mail oraz hasła) pojawia się ekran logowania do aplikacji. Po zalogowaniu się, pojawia się lista restauracji w domyślnej lokalizacji startowej. W górnej części ekranu widoczny jest pasek wyszukiwania. Jest powielony w innych ekranach aplikacji. Lokalizacja startowa będzie mogła być ustalana jako rzeczywista lokalizacja użytkownika za pomocą usług lokalizacyjnych. Po kliknięciu w wybraną restaurację, pojawia się menu oferowanych posiłków oraz napojów. Następnym jest widok koszyka, który umożliwia finalizację zamówienia poprzez podanie swoich danych adresowych oraz płatności kartą. Kolejnym widokiem jest widok mapy. Można w nim wyznaczyć trasę do wybranej restauracji. Ostatnim ekranem aplikacji jest panel użytkownika. W panelu użytkownika, mamy możliwość edycji swoich danych, zdjęcia profilowego oraz listy ulubionych restauracji.

## Wyszczególnienie wybranych technologii

- **Google Firebase** - usługa backendu (*Backend as a Service* - *BaaS*). Oferuje szereg usług pozwalających twórcom aplikacji implementować podstawowe funkcjonalności natychmiastowo, bez potrzeby tworzenia własnego backendu. Implementację nale krokiem, który musimy wykonać, jest zarejestrowanie naszej aplikacji w konsoli Firebase, podając ID aplikacji (*com.company.appname*). Następnie potrzebujemy pobrać wygenerowany plik konfiguracyjny *google-services.json* i dodać go do projektu. Najważniejsze z punktu widzenia twórcy aplikacji usługi w ramach Google Firebase:
  - **Cloud Firestore** - jedna z najpopularniejszych funkcjonalności. Cloud Firestore to baza danych NoSQL, stanowiąca pewnego rodzaju następcę Realtime Database. Struktura jest oparta na kolekcjach i dokumentach. Przykładowy kod:

```

1 CollectionReference cities = db.collection("cities");
2
3 Map<String, Object> data1 = new HashMap<>();
4 data1.put("name", "San Francisco");
5 data1.put("state", "CA");
6 data1.put("country", "USA");
7 data1.put("capital", false);
8 data1.put("population", 860000);
9 data1.put("regions", Arrays.asList("west_coast", "norcal"));
10 cities.document("SF").set(data1); // source: https://firebase.google.com/docs/
    firestore/query-data/get-data

```

Ponadto, Cloud Firestore oferuje między innymi: automatyczną synchronizację danych w chmurze; wsparcie dla lokalnego przechowywania danych, pozwalające aplikacji na działanie w trybie offline, zautomatyzowane skalowanie bazy danych.

- **Authentication** - usługa, która oferuje silnik autentykacji użytkowników, korzystających z aplikacji. Usługa oferuje kompleksowe rozwiązanie w zakresie identyfikacji tożsamości użytkownika, pozwalając na logowanie się nie tylko przy użyciu klasycznego loginu i hasła, lecz również kont Google, Facebook, Twitter, czy GitHub. Wszystko zgodne z normami bezpieczeństwa Google.
- **Cloud Functions** - zdejmuje z programisty konieczność samodzielnego tworzenia, utrzymywania oraz skalowania własnego backendu. Niezbędne jest jedynie samodzielne napisanie funkcji w Node.js.
- **Google Cloud** - Usługa chmury danych, a także chmury obliczeniowej od Google. Google używa swojej chmury do udostępniania swoich usług takich jak Google Search, Gmail, Google Drive czy YouTube użytkownikom końcowym. W tej chwili Chmura Google oferuje ponad 100 różnych usług przechowywania danych, baz danych, firewall, wynajmowania mocy obliczeniowej, sztucznej inteligencji, zarządzania, Big Data, internetu rzeczy, platform API.
- **Stripe Firebase extension** - rozszerzenie dostępne w platformie Firebase, umożliwiające implementację różnych metod płatności internetowych w aplikacji, realizowane za pomocą Stripe Payments API. Stripe jako całość jest pakietem usług finansowych (płatności internetowe, fakturowanie raporty finansowe)
- **React Native** - React Native to framework (platforma programistyczna, czyli szkielet do budowy aplikacji) stworzony przez Facebook, aby przyspieszyć proces tworzenia aplikacji mobilnych. Pozwala na symultaniczne budowanie aplikacji zarówno na platformie Android, jak i iOS, wykorzystując język JavaScript. Wystarczy napisać kod w języku JavaScript, a w efekcie otrzymamy aplikację mobilną z natywnymi elementami dla iOS oraz Android. Dzieje się tak, ponieważ React Native używa mostów (bridges) do przekładania języka JavaScript na natywne komponenty. React Native jest chętnie używany ze względu na elastyczność, ogromną (społecznościową) bazę wiedzy oraz łatwość synchronizacji stylu interfejsu użytkownika w aplikacjach multiplatformowych.
- **Google Maps API** - Udostępniona przez Google platforma do nawigacji, wyszukiwania obiektów, sprawdzania natężenia ruchu w wybranej lokalizacji oraz dająca dostęp do wizytówek wszystkich przedsiębiorstw oraz punktów nawigacyjnych, które zostało ujęte przez właścicieli przedsiębiorstw lub społeczność.
- **EXPO** - Framework oraz platforma pomocna przy tworzeniu aplikacji w React Native. Jest to zestaw narzędzi ułatwiających projektowanie, tworzenie i szybkie przetestowanie aplikacji mobilnych. Unikalną

właściwością EXPO jest możliwość strumieniowania tworzonej aplikacji bezpośrednio na urządzenie mobilne (pracujące pod kontrolą systemu operacyjnego Android lub iOS), co pozwala szybko sprawdzić działanie interfejsu aplikacji "na żywo".

## Zrzuty ekranowe interfejsu aplikacji

### Instrukcja uruchomienia aplikacji

Aplikację w trybie zdalnym można uruchomić np. w terminalu programu Visual Studio Code po zainstalowaniu niezbędnych komponentów. Należy się upewnić, czy na maszynie jest zainstalowany (jeśli nie, to zainstalować) Node.js, React Native CLI oraz EXPO CLI, a także środowisko programistyczne Java. W przypadku JRE należy skonfigurować zmienne środowiskowe. Program uruchamiamy w terminalu komenda:

```
1 expo start
```

W przeglądarce zostaje uruchomiony serwer Metro, w osobnej konsoli uruchamia się Node.js. Na stronie serwera Metro w przeglądarce wybiera się sposób demonstracji aplikacji. Można skorzystać z emulatora urządzenia z systemem Android pod warunkiem posiadania Android Studio oraz aktualnego SDK. Alternatywę stanowi wygenerowanie kodu QR umożliwiającego ukazanie aplikacji w formie Live APK na urządzeniu z systemem Android, na przykład na prywatnym smartfonie z zainstalowaną aplikacją EXPO. Pozostaje możliwość przewodowego połączenia z urządzeniem w trybie debugowania. Następuje tymczasowa "instalacja" umożliwiająca obsługę oraz debugowanie aplikacji.

## Wnioski projektowe