

Dokumentacja Inżynierska - Wymagania Aplikacji Planera Podróży

NAZWA ROBOCZA:
TRIP.LO

Spis treści

1. Wprowadzenie	3
1.1 Cel dokumentu	3
1.2 Zakres systemu	3
2. Na podstawie specyfikacji biznesowej	4
2.1 Kontekst biznesowy i cele	4
2.2 Wymagania techniczne	4
3. Specyfikacja techniczna	5
3.1 Architektura systemu	5
3.2 Technologie i narzędzia	5
3.3 Integracja z systemami zewnętrznymi	7
3.4 Bezpieczeństwo i prywatność	7
4. Specyfikacja funkcjonalna	9
5. Specyfikacja нефunkcjonalna	11
5.1 Wydajność	11
5.2 Bezpieczeństwo	11
5.4 Dostępność	12
5.4 UI/UX	12
5.5 Monitorowanie i logowanie	13
6. Wycena i harmonogram realizacji	14

1. Wprowadzenie

1.1 Cel dokumentu

Celem tego dokumentu jest formalne określenie wymagań funkcjonalnych i niefunkcjonalnych aplikacji mobilnej do planowania podróży. Dokument stanowi podstawę do projektowania, implementacji i testowania systemu. Zawarte w nim informacje mają na celu usprawnienie pracy zespołu oraz prezentację produktu oczekiwanego w końcowym rozrachunku.

1.2 Zakres systemu

Aplikacja umożliwia użytkownikom planowanie podróży, zarządzanie harmonogramem, budżetem oraz przechowywanie informacji o rezerwacjach i biletach. System będzie dostępny na urządzenia mobilne z systemami Android oraz w przeglądarce internetowej, a także posiadać backend zarządzający danymi i uwierzytelnianiem użytkowników.

2. Na podstawie specyfikacji biznesowej

2.1 Kontekst biznesowy i cele

Aplikacja Planer Podróży jest narzędziem dedykowanym do zarządzania podróżami, które ma na celu usprawnienie i automatyzację procesów związanych z planowaniem podróży. Współczesne wyjazdy, zarówno prywatne, jak i służbowe, wymagają sprawnego zarządzania wieloma aspektami: rezerwacjami transportu, zakwaterowania, atrakcjami turystycznymi, a także budżetem podróży. Aktualnie dostępne narzędzia są rozproszone, co powoduje trudności w organizowaniu wszystkich szczegółów w jednym miejscu. Aplikacja ma na celu integrację tych wszystkich funkcji, co pozwala użytkownikowi na zarządzanie całą podróżą od jednego miejsca, z poziomu urządzenia mobilnego lub przeglądarki internetowej.

Aplikacja skierowana jest zarówno do indywidualnych podróżników, jak i grup użytkowników, takich jak rodziny, przyjaciele czy firmy, które organizują wspólne wyjazdy. Dzięki funkcjom współdzielenia planów, udostępniania wydatków i wspólnego zarządzania podróżą, aplikacja staje się narzędziem idealnym do planowania wyjazdów w zespole.

2.2 Wymagania techniczne

Aplikacja będzie oparta na architekturze klient-serwer z wykorzystaniem chmury, która zapewni skalowalność i elastyczność systemu. Zastosowanie tej technologii umożliwi optymalizację procesu planowania podróży i sprawdzenie różnych scenariuszy (np. dostępność transportu, ceny hoteli) w czasie rzeczywistym.

- ❖ Backend aplikacji zostanie zbudowany przy użyciu technologii .NET Core, która zapewnia wysoką wydajność, bezpieczeństwo i możliwość rozbudowy systemu. Dzięki temu backend będzie mógł obsługiwać rosnącą liczbę użytkowników i danych, co jest istotne z punktu widzenia skalowalności.
- ❖ Frontend zostanie zrealizowany z wykorzystaniem React Native, co pozwala na tworzenie aplikacji mobilnych zarówno na systemy Android, jak i iOS, przy zachowaniu wysokiej wydajności.
- ❖ Baza danych: Do przechowywania informacji o użytkownikach, planach podróży i rezerwacjach zostanie użyta baza PostgreSQL – relacyjna, wydajna i skalowalna.
- ❖ Przechowywanie danych w pamięci: Redis będzie używany do przechowywania sesji użytkowników oraz innych danych, które muszą być dostępne szybko, takich jak informacje o trasach podróży i stanach rezerwacji.
- ❖ Integracja z systemami zewnętrznymi: Aplikacja będzie korzystać z zewnętrznych API, takich jak Google Maps API do wyznaczania tras, Booking API do rezerwacji hoteli, oraz innych API do obsługi płatności (np. Stripe).

3. Specyfikacja techniczna

3.1 Architektura systemu

Aplikacja Planer Podróży zostanie zaprojektowana zgodnie z zasadami architektury klient-serwer. Architektura ta umożliwi oddzielenie front-endu (interfejsu użytkownika) od back-endu (logiki aplikacji i bazy danych). System będzie oparty na chmurze, co zapewni skalowalność, bezpieczeństwo i elastyczność w obsłudze rosnącej liczby użytkowników i danych.

- Frontend: Aplikacja mobilna będzie oparta na React Native, co umożliwi tworzenie natywnych aplikacji na Androida i iOS, z jednego wspólnego kodu. Dzięki temu możliwe będzie łatwe utrzymanie aplikacji oraz szybka implementacja nowych funkcji.
- Backend: Backend zostanie oparty na technologii TypeScript. RESTful API będzie pełniło rolę interfejsu komunikacyjnego między frontendem a bazą danych.
- Baza danych: PostgreSQL zostanie użyty jako główna baza danych. Jest to relacyjna baza danych, która zapewnia wydajność i integrację z różnymi systemami. PostgreSQL będzie przechowywał dane użytkowników, plany podróży, rezerwacje i inne informacje związane z podróżami.
- Cache danych: Redis będzie wykorzystywany do przechowywania danych w pamięci podręcznej, co pozwala na szybszy dostęp do danych, które są często używane, takich jak szczegóły podróży, dane o użytkownikach czy trasy.

3.2 Technologie i narzędzia

1) Frontend – React Native

- Opis: React Native to popularna biblioteka JavaScript do budowy aplikacji mobilnych na Androida i iOS. Dzięki zastosowaniu tej technologii możliwe jest tworzenie aplikacji natywnych z użyciem jednej bazy kodu, co znacząco zmniejsza czas i koszty rozwoju aplikacji. Użycie React Native zapewnia także dużą elastyczność i łatwość integracji z natywnymi komponentami urządzenia.
- Zalety:
 - Szybszy rozwój aplikacji dzięki wspólnemu kodowi dla Androida i iOS.
 - Wysoka wydajność dzięki natywnej integracji.
 - Duża społeczność oraz wsparcie dla integracji z zewnętrznymi bibliotekami.

2) Backend – TypeScript (Node.js)

- Opis: Backend aplikacji zostanie oparty na technologii TypeScript w środowisku Node.js, co zapewni wysoką wydajność, bezpieczeństwo oraz dużą skalowalność. TypeScript to nadzbiór JavaScript, który umożliwia pisanie silnie typowanych aplikacji, co zmniejsza ryzyko błędów w kodzie i poprawia jakość oprogramowania. Node.js zapewnia szybkie przetwarzanie

zapytań i umożliwia tworzenie skalowalnych aplikacji, co jest szczególnie istotne w przypadku aplikacji o rosnącej liczbie użytkowników i dużej ilości danych.

- Zalety:
 - Wysoka wydajność dzięki asynchronicznej obsłudze zapytań w Node.js.
 - Silna typizacja zapewnia lepszą organizację kodu i łatwiejsze utrzymywanie aplikacji.
 - Duża skalowalność – aplikacja będzie mogła obsługiwać rosnącą liczbę użytkowników oraz danych.
 - Łatwość integracji z zewnętrznymi systemami, bazami danych i API.
 - Jednolity ekosystem – dzięki użyciu TypeScript i Node.js możliwe będzie użycie tych samych technologii w frontendzie i backendzie, co ułatwi rozwój i integrację aplikacji. Kompatybilność z chmurą (np. AWS, Azure), co pozwala na łatwą skalowalność.

3) Baza danych – PostgreSQL

- Opis: PostgreSQL to relacyjna baza danych, która oferuje wyjątkową wydajność, skalowalność i niezawodność. Jest idealnym rozwiązaniem do przechowywania danych o użytkownikach, podróżach, rezerwacjach i innych zasobach. PostgreSQL wspiera różnorodne funkcje, takie jak transakcje ACID, indeksowanie, oraz zapytania w języku SQL.
- Zalety:
 - Wysoka wydajność przy dużej ilości danych.
 - Rozbudowane możliwości zapytań SQL.
 - Obsługuje transakcje i integralność danych.

4) Cache danych – Redis

- Opis: Redis jest szybkim magazynem danych w pamięci, który pozwala na przechowywanie najczęściej wykorzystywanych informacji, takich jak dane sesji użytkowników, wyniki zapytań o trasy podróży czy informacje o rezerwacjach. Redis pozwala na bardzo szybki dostęp do danych, co znacznie poprawia wydajność aplikacji.
- Zalety:
 - Szybki dostęp do danych.
 - Obsługuje przechowywanie danych tymczasowych, takich jak sesje użytkowników, informacje o bieżącej podróży.
 - Możliwość integracji z innymi systemami.

3.3 Integracja z systemami zewnętrznymi

Naszą aplikację będziemy próbować zintegrować z różnymi systemami zewnętrznymi, co pozwoli na rozszerzenie jej funkcji o dodatkowe usługi. Poniżej przedstawiamy kluczowe integracje:

1) Google Maps API

- Opis: Google Maps API będzie wykorzystywane do wyświetlania map i wyznaczania tras podróży. Użytkownicy będą mogli tworzyć trasy podróży, uwzględniając aktualne warunki drogowe, czas przejazdu, odległość i inne zmienne.
- Zalety:
 - Dokładne dane o trasach i lokalizacjach.
 - Możliwość wyznaczania tras uwzględniających różne środki transportu (samochód, pieszo, rower).
 - Integracja z innymi usługami Google (np. Google Places, Google Street View).

2) Booking API (i inne systemy rezerwacyjne)

- Opis: Aplikacja będzie integrować się z systemami rezerwacji hoteli, lotów, pociągów i innych usług transportowych, co pozwoli użytkownikom na bezpośrednie rezerwowanie tych usług w aplikacji.
- Zalety:
 - Bezpośrednia rezerwacja w aplikacji.
 - Zintegrowane informacje o dostępności usług i cenach.
 - Użytkownicy nie będą musieli opuszczać aplikacji, aby dokonać rezerwacji.

3) OneDrive API

- Opis: OneDrive API pozwoli na przechowywanie plików użytkowników w chmurze, takich jak bilety lotnicze, rezerwacje, dokumenty paszportowe czy zdjęcia z podróży.
- Zalety:
 - Łatwe przechowywanie i synchronizacja dokumentów.
 - Bezpieczne przechowywanie danych użytkowników w chmurze.
 - Możliwość dzielenia się dokumentami i plikami między użytkownikami.

3.4 Bezpieczeństwo i prywatność

1) JWT (JSON Web Token)

- Opis: JWT będzie wykorzystywane do bezpiecznej autentykacji użytkowników. Każdy użytkownik po zalogowaniu otrzyma unikalny token, który będzie musiał być dołączony do każdego zapytania wysyłanego do serwera. Dzięki temu możliwe będzie zarządzanie sesjami użytkowników w sposób bezpieczny i wygodny.
- Zalety:

- Szybka i bezpieczna autentykacja.
- Tokeny mogą być łatwo przechowywane w aplikacjach mobilnych.

2) Szyfrowanie danych

- Opis: Wszystkie dane użytkowników będą szyfrowane przy pomocy standardów takich jak SSL/TLS do komunikacji z serwerem oraz AES-256 do przechowywania wrażliwych danych w bazach.
- Zalety:
 - Ochrona prywatności użytkowników.
 - Zabezpieczenie przed nieautoryzowanym dostępem do danych.

3) Zgodność z RODO

- Opis: Aplikacja będzie spełniała wymagania ogólnego rozporządzenia o ochronie danych osobowych (RODO). Przechowywanie danych osobowych będzie odbywało się zgodnie z przepisami dotyczącymi ochrony danych osobowych, a użytkownicy będą mieli pełną kontrolę nad swoimi danymi (np. prawo do usunięcia danych).
- Zalety:
 - Zgodność z międzynarodowymi regulacjami dotyczącymi ochrony danych.
 - Transparentność w zakresie przetwarzania danych osobowych.

4. Specyfikacja funkcjonalna

Aplikacja Planer Podróży ma oferować następujące funkcje:

- 1) Rejestracja i logowanie użytkowników
 - Użytkownicy muszą mieć możliwość zakładania konta za pomocą e-maila lub kont zewnętrznych (Google, Facebook).
 - Wymagane jest potwierdzenie adresu e-mail (link aktywacyjny).
 - Logowanie odbywa się przy użyciu tokenów JWT, zapewniając bezpieczny dostęp do aplikacji.
- 2) Tworzenie planu podróży
 - Użytkownicy będą mogli wprowadzać szczegóły podróży: miejsce docelowe, daty, środki transportu, hotel, atrakcje turystyczne.
 - Możliwość dodania opcji „plan podróży” dla kilku osób – z możliwością przypisania do każdego uczestnika odpowiedzialności za konkretne zadania (np. zakupy, rezerwacja transportu).
 - Każdy plan podróży będzie zawierał sekcje na budżet, rezerwacje, atrakcje, transport i dokumenty.
- 3) Zarządzanie rezerwacjami i dokumentami
 - Użytkownicy mogą dodawać bilety (np. lotnicze, kolejowe), rezerwacje hoteli oraz inne dokumenty (np. wizy, bilety wstępu).
 - Możliwość dodania zdjęć dokumentów.
 - Integracja z OneDrive API pozwala na łatwe przechowywanie i udostępnianie dokumentów związanych z podróżą.
- 4) Zarządzanie budżetem podróży
 - Użytkownik może ustawić początkowy budżet podróży oraz śledzić wydatki na transport, zakwaterowanie, jedzenie i inne wydatki.
 - Aplikacja automatycznie generuje raporty wydatków, prezentując je w postaci wykresów (np. Chart.js).
 - Możliwość podziału kosztów pomiędzy podróżnych, np. dla grup znajomych – każdy użytkownik dostaje przypisane zadania w zakresie pokrywania wydatków.
- 5) Integracja z Google Maps API
 - Aplikacja wykorzysta Google Maps API do tworzenia tras podróży i wskazywania atrakcji turystycznych w danym regionie.
 - Możliwość wyznaczania tras i monitorowania warunków drogowych, takich jak opóźnienia, wypadki lub zmiany tras w czasie rzeczywistym.
 - Użytkownicy mogą śledzić lokalizację współtowarzyszy podróży w czasie rzeczywistym.
- 6) Udostępnianie planu podróży
 - Użytkownicy mogą udostępniać swoje plany podróży innym osobom, np. członkom rodziny, znajomym czy grupom.
 - Możliwość dodawania współuczestników, którzy będą mogli edytować plan podróży (np. zmieniać daty, miejsca, transport).

- Powiadomienia o aktualizacjach planu (np. zmiana daty wyjazdu, dodanie atrakcji).
- 7) Synchronizacja danych w chmurze
- Wszystkie dane użytkownika (plan podróży, dokumenty, zdjęcia, budżet) będą przechowywane w chmurze, co zapewni dostęp do nich na różnych urządzeniach.
 - Użytkownik może edytować plan podróży z poziomu aplikacji mobilnej i webowej, a zmiany będą automatycznie synchronizowane.
- 8) Powiadomienia
- Aplikacja będzie wysyłała powiadomienia push o nadchodzących terminach (np. rezerwacja transportu, hotelu).
 - Powiadomienia o zmianach w planach podróży (np. zmiana transportu, atrakcji, ceny biletów).
 - Powiadomienia o zbliżających się wydarzeniach w pobliżu (np. festiwale, wydarzenia kulturalne).
- 9) Wyszukiwanie atrakcji turystycznych i restauracji
- Użytkownicy będą mogli wyszukiwać lokalne atrakcje turystyczne, restauracje, muzea i inne interesujące miejsca.
 - Integracja z Google Places API umożliwi dostęp do szczegółowych informacji o miejscach, recenzjach, godzinach otwarcia itp.
 - Możliwość dodania atrakcji do planu podróży z informacjami o kosztach, godzinach otwarcia i lokalizacji.
- 10) Integracja z systemami zewnętrznymi

Integracja z zewnętrznymi systemami rezerwacyjnymi (np. Booking, Expedia, Skyscanner) pozwala użytkownikom na łatwe dodawanie rezerwacji do swojego planu podróży.

5. Specyfikacja niefunkcjonalna

5.1 Wydajność

Aplikacja Planer Podróży musi spełniać wymagania dotyczące wydajności, aby zapewnić szybkie i bezproblemowe korzystanie z systemu przez użytkowników. Obejmuje to zarówno czasy ładowania aplikacji, jak i czas odpowiedzi na zapytania w trakcie użytkowania.

- Czas odpowiedzi: Czas odpowiedzi aplikacji na zapytanie użytkownika, takie jak pobieranie danych o planach podróży czy wyznaczanie tras, powinien wynosić maksymalnie 2 sekundy. Zapewni to płynność korzystania z aplikacji i minimalizację frustracji użytkowników.
- Czas ładowania aplikacji: Aplikacja powinna ładować się w czasie nie dłuższym niż 5 sekund na urządzeniach mobilnych, niezależnie od obciążenia serwera.
- Wydajność przy dużej liczbie użytkowników: Aplikacja musi obsługiwać minimum 5,000 aktywnych użytkowników dziennie w początkowej fazie, z możliwością skalowania do 50,000 użytkowników dziennie w miarę rozwoju.
- Optymalizacja zapytań: Backend aplikacji będzie zoptymalizowany pod kątem dużej liczby zapytań do bazy danych, co zapewni szybkie przetwarzanie informacji i unikanie opóźnień w czasie rzeczywistym.

5.2 Bezpieczeństwo

Bezpieczeństwo aplikacji jest kluczowe, ponieważ przechowuje ona dane osobowe użytkowników oraz wrażliwe informacje związane z podróżami (np. numery rezerwacji, dane paszportowe). System musi zapewniać pełną ochronę danych użytkowników przed nieautoryzowanym dostępem, utratą lub zniszczeniem.

- Autoryzacja i uwierzytelnianie: Aplikacja będzie wykorzystywać JSON Web Tokens (JWT) do autoryzacji użytkowników. Po pomyślnej rejestracji i zalogowaniu użytkownicy będą otrzymywać token, który będzie używany do autentykacji w systemie, co zapewnia bezpieczeństwo sesji użytkowników.
- Szyfrowanie danych: Wszystkie dane użytkowników przechowywane w bazie danych będą szyfrowane przy użyciu algorytmu AES-256. Ponadto, wszystkie dane przesyłane między aplikacją a serwerem będą zabezpieczone przy pomocy protokołu SSL/TLS.
- Bezpieczne przechowywanie danych w chmurze: Aplikacja będzie przechowywać pliki i dokumenty użytkowników w chmurze (np. AWS, Azure), z zachowaniem wszystkich standardów bezpieczeństwa, takich jak szyfrowanie plików i dostęp tylko dla autoryzowanych użytkowników.

- Zgodność z RODO: Aplikacja będzie w pełni zgodna z przepisami RODO. Użytkownicy będą mieli pełną kontrolę nad swoimi danymi osobowymi, w tym prawo do ich edytowania, usuwania lub przenoszenia do innych usług.

5.4 Dostępność

Aplikacja musi być dostępna dla użytkowników przez cały czas, z minimalnym czasem przestoju. Dzięki odpowiedniej architekturze oraz zastosowaniu chmurowych rozwiązań infrastrukturalnych, aplikacja będzie w stanie zapewnić wysoką dostępność oraz ciągłość działania.

- Uptime: Aplikacja powinna mieć minimalny czas przestoju wynoszący 0.1% miesięcznie, co daje 99.9% dostępności. To oznacza, że system może mieć maksymalnie około 43 minut przerwy miesięcznie.
- Redundancja: Zastosowanie redundancyjnych serwerów w chmurze oraz rozproszonej architektury bazy danych pozwoli na utrzymanie ciągłości działania aplikacji w przypadku awarii jednego z elementów systemu.
- Wydajność w trybie offline: W przypadku braku dostępu do internetu użytkownicy będą mogli korzystać z trybu offline. Aplikacja będzie umożliwiała dostęp do wcześniej załadowanych informacji o planach podróży, rezerwacjach i dokumentach.

5.4 UI/UX

Aplikacja musi być zaprojektowana w sposób zapewniający intuicyjność, łatwość obsługi oraz szybki dostęp do wszystkich kluczowych funkcji. Użyteczność aplikacji ma kluczowe znaczenie, aby użytkownicy chętnie z niej korzystali, szczególnie w kontekście podróży, gdzie czasami nie ma miejsca na skomplikowaną obsługę aplikacji.

- Responsywność: Aplikacja będzie zoptymalizowana pod kątem urządzeń mobilnych oraz przeglądarek internetowych. Interfejs będzie dostosowywał się do różnych rozdzielczości ekranów, zapewniając wygodną obsługę na telefonach, tabletach i komputerach.
- Dostępność: UI aplikacji będzie projektowane zgodnie z najlepszymi praktykami w zakresie dostępności. Zastosowane zostaną odpowiednie kontrasty kolorów, możliwość powiększania czcionek oraz dostosowywania ustawień aplikacji dla osób z ograniczoną zdolnością wzroku.
- Prosty i intuicyjny interfejs: Projekt interfejsu użytkownika będzie zgodny z zasadami UX, zapewniając łatwy dostęp do najczęściej używanych funkcji, takich jak dodawanie nowych podróży, zarządzanie rezerwacjami czy śledzenie budżetu.

5.5 Monitorowanie i logowanie

Aplikacja będzie posiadała system monitorowania i logowania, który umożliwi wykrywanie błędów, analizowanie wydajności oraz reagowanie na incydenty w czasie rzeczywistym.

- Monitorowanie wydajności: Będzie wdrożony system do monitorowania wydajności aplikacji, np. New Relic, który pozwala na bieżąco analizować czas odpowiedzi serwera, obciążenie systemu oraz inne wskaźniki wydajności.
- Logowanie błędów: Zastosowane zostaną narzędzia do logowania błędów, takie jak Sentry, które umożliwią zbieranie szczegółowych informacji o problemach w aplikacji oraz szybkie ich rozwiązywanie przez zespół techniczny.
- Powiadomienia o incydentach: W przypadku wykrycia poważnych błędów lub spadku wydajności aplikacja będzie wysyłać powiadomienia do zespołu wsparcia technicznego.

(Przykładowy koszt w założeniu cen dostępnych w internecie)

6. Wycena i harmonogram realizacji

Koszt specyfikacji i analizy

Koszt: 10,000 - 15,000 PLN

Czas realizacji: 4-6 tygodni

Koszt produkcji i programowania

Koszt: 60,000 - 90,000 PLN

Czas realizacji: 4-6 miesięcy

Koszt testów

Koszt: 10,000 - 15,000 PLN

Czas realizacji: 2-3 tygodnie

Koszt zarządzania projektem

Koszt: 12,000 - 18,000 PLN

Czas realizacji: cały okres realizacji

Koszt wdrożenia

Koszt: 5,000 - 8,000 PLN

Czas realizacji: 2-3 tygodnie

Koszt utrzymania aplikacji

Koszt roczny: 10,000 - 20,000 PLN

Koszty związane z infrastrukturą

Koszt roczny: 15,000 - 25,000 PLN

Całkowita wycena projektu

Koszt całkowity (bez utrzymania rocznego): 117,000 - 186,000 PLN

Koszt utrzymania rocznego: 15,000 - 25,000 PLN