

# FAZA KONSTRUKCJI APLIKACJI EUROTRAVEL

HUBERT ADAMIAK, MACIEJ CIEPIELA

# 1. TESTY FUNKCJONALNE

# Test 1 – Wyszukanie trasy

1. Użytkownik chce wyszukać trasę z punktu A do B
2. Użytkownik uruchamia aplikację
3. Użytkownik wyszukuje punkt wyjścia i punkt docelowy
4. Aplikacja wyświetla dostępne opcje transportu i trasy

# Test 1 – Wyszukanie trasy

Krok	Opis	Oczekiwania
1	Użytkownik klika "lupkę"	Aplikacja rozwija okno wyszukiwania, podpowiadając w trakcie pisania przez użytkownika
2	Użytkownik wybiera punkt docelowy	Aplikacja wyświetla wybrany punkt na mapie
3	Użytkownik klika przycisk "trasa"	Aplikacja każe użytkownikowi wpisać punkt, w którym ma być rozpoczęta trasa
4	Użytkownik wyszukuje punkt początku trasy	Aplikacja wyświetla dostępne trasy i środki transportu
5	Użytkownik wybiera trasę	Aplikacja prowadzi użytkownika do celu

# Test 2 – Zakup biletu

1. Użytkownik chce zakupić bilet na wybraną przez siebie trasę
2. Użytkownik uruchamia aplikację
3. Użytkownik klika przycisk "kup potrzebne bilety"
4. Aplikacja wyświetla potrzebne bilety oraz łączną cenę
5. Użytkownik klika przycisk "zapłać"
6. Aplikacja otwiera okno płatności
7. Użytkownik wybiera formę płatności i dokonuje zakupu

# Test 2 – Zakup biletu

Krok	Opis	Oczekiwania
1	Użytkownik klika przycisk "kup potrzebne bilety"	Aplikacja wyświetla potrzebne bilety oraz łączną cenę
2	Użytkownik klika przycisk "zapłać"	Aplikacja otwiera okno płatności
3	Użytkownik wybiera formę płatności	Aplikacja otwiera kolejne okno do wybranej formy płatności
4	Użytkownik finalizuje płatność	Aplikacja przetwarza płatność i przypisuje bilety do konta

# Test 3 – Sprawdzenie częstych tras

1. Użytkownik chce sprawdzić z jakich tras najczęściej korzysta
2. Użytkownik uruchamia aplikację
3. Użytkownik klika panel "konto"
4. Aplikacja wyświetla informacje o koncie
5. Użytkownik klika "ulubione trasy"
6. Aplikacja wyświetla najczęstsze trasy użytkownika

# Test 3 – Sprawdzenie częstych tras

Krok	Opis	Oczekiwania
1	Użytkownik klika panel "konto"	Aplikacja wyświetla potrzebne bilety oraz łączną cenę
2	Użytkownik klika "ulubione trasy"	Aplikacja wyświetla najczęstsze trasy użytkownika



## 2. POMIARY WYMAGAŃ

# Pomiary wymagań

**Intuicyjny interfejs**, użytkownik powinien bezproblemowo poruszać się po aplikacji i korzystać z jej funkcjonalności

Pomiar:  $X = P / T$

**P** oznacza liczbę przypadków, w których użytkownik nie będzie wiedział co zrobić, aby użyć wybranej funkcji lub zgubi się w aplikacji

**T** oznacza czas używania aplikacji przez użytkownika w minutach

Celem jest osiągnięcie wskaźnika **X** poniżej 0.01

# Pomiary wymagań

Aplikacja będzie działała w całej Europie, a co za tym idzie musi mieć na tym obszarze **dokładne mapy**

Pomiar:  $X = M_i / N$

**$M_i$**  oznacza liczbę dobrze odwzorowanych dróg w danym mieście, a  **$N$**  to wszystkie drogi w tym mieście

Celem jest osiągnięcie wskaźnika  $X$  na poziomie 0.99 dla miast powyżej 100 tys. mieszkańców oraz conajmniej 0.95 dla pozostałych miast

# Pomiary wymagań

Aplikacja będzie wstępnie napisana w języku **polskim oraz angielskim**, a następnie przy dodawaniu kolejnych języków do oferty, będą przygotowywane kolejne tłumaczenia

Pomiar:  $X = Li / N$

**Li** oznacza liczbę przetłumaczonych słów w i-tym języku. Językami domyślnymi są polski i angielski

**N** oznacza liczbę elementów interfejsu

Celem jest osiągnięcie wskaźnika X na poziomie 1 dla języka polskiego i angielskiego, oraz minimum 0.9 dla pozostałych

# Pomiary wymagań

**Niezawodność działania aplikacji na telefonach, tabletach i innych sprzętach** działających na systemach Android w wersji 6.01 lub wyższej oraz iOS w wersji 9.0 lub wyższej

Pomiar:  $X = A_i / N$

**$A_i$**  oznacza liczbę funkcji działających na i-tym systemie

**$N$**  oznacza liczbę wszystkich funkcji aplikacji

Celem jest osiągnięcie wskaźnika  **$X$  równego 1** dla obu systemów

# 3. PLAN BETA TESTOWANIA

# 1 Faza Testowania (tydzień)

Pierwszą fazą testowania będzie zajmował się zatrudniony tester. Przetestuje on najważniejsze funkcjonalności samego działania aplikacji takie jak system rejestracji i logowania, wyszukiwanie tras, kupowanie biletów, system GPS czy interfejs. Będzie również szukał wszelkiego rodzaju błędów czy usterek

## 2 Faza Testowania (miesiąc)

Drugą fazą testowania będzie zajmowało się kilku zatrudnionych testerów oraz ochotnicy z całego świata chcący sobie trochę dorobić. Ich zadaniem będzie przetestowanie aplikacji w prawdziwym życiu. Na tym etapie chcemy wyłapać wszelkie błędy w mapach, rozkładach jazdy, nazwach miejscowości i codziennym użytkowaniu aplikacji oraz ewentualne błędy przegapione przez pierwszego testera



# 4. PLAN ZARZĄDZANIA JAKOŚCIĄ

# Plan zarządzania jakością

1. Testowanie na bieżąco wszystkich funkcji napisanych przez programistów
2. Testowanie bazy oraz aplikacji pod kątem bezpieczeństwa
3. Aplikacja będzie napisana zgodnie z podstawowymi standardami kodowania i bezpieczeństwa danych
4. Praca nad aplikacją będzie prowadzona za pomocą Githuba co pozwoli na łatwiejszą kontrolę produktu
5. Do aplikacji dołączona będzie dokumentacja pisana przy fazie końcowej projektu

# 5. PLAN WYKONANIA PRODUKTU

Nazwa zadania	Czas
Wstępna rozmowa z klientem	2 dni
Przygotowanie tablicy koncepcyjnej	3 dni
Prezentacja tablicy koncepcyjnej klientowi	1 dzień
Poprawki w projekcie	3 dni
Opracowanie architektury systemu	tydzień
Rekrutacja pracowników i podział pracy	3 tygodnie
Wstępna implementacja projektu	3 miesiące
Testowanie pierwszych funkcjonalności	2 tygodnie
Wprowadzenie poprawek do projektu	2 tygodnie
Dalsza implementacja projektu	2 miesiące
1 Faza Testowania	tydzień
Poprawki wykrytych błędów	tydzień
2 Faza Testowania	miesiąc
Poprawki wykrytych błędów	2 tygodnie
Oddanie aplikacji zleceniodawcy	1 dzień

# 6. OCENA ZGODNOŚCI

# Ocena zgodności

Wykonane dotychczas efekty są zadowalające, jednak konieczne będzie zatrudnienie nowych testerów do sprawdzania aplikacji w mniej popularnych rejonach. Pomoże nam to wyeliminować błędy w mapach i rozkładach jazdy. Możliwe jest również przedłużenie wykonania aplikacji o około miesiąc