

System wymiany walut - Kompletna dokumentacja projektu

Autor: Jakub Czucha, Nikodem Borkowski, Piotr Białowski

Technologie: React Native, Node.js, SQLite, NBP API

Spis treści

1. Opis projektu i architektura
2. Wymagania funkcjonalne i niefunkcjonalne
3. Szczegółowe przypadki użycia
4. Diagramy UML (PlantUML - do wygenerowania PNG)
5. Podsumowanie projektu

1. OPIS PROJEKTU I ARCHITEKTURA

System wymiany walut to kompleksowa aplikacja mobilna zbudowana w architekturze monorepo, składająca się z aplikacji React Native oraz backendu Node.js z bazą danych SQLite. System umożliwia użytkownikom zarządzanie wielowalutowymi portfelami, przeprowadzanie wymian walutowych na podstawie aktualnych kursów NBP oraz analizę zysków i strat.

Architektura systemu

Frontend (React Native + Expo)

- Framework: React Native z Expo SDK 54
- Nawigacja: React Navigation 6.x
- Zarządzanie stanem: React Hooks + Context API
- Komunikacja z API: Axios
- Lokalne przechowywanie: AsyncStorage

Backend (Node.js)

- Framework: Express.js
- Autoryzacja: JWT (jsonwebtoken)
- Hashowanie haseł: bcryptjs
- Baza danych: SQLite3
- Integracja zewnętrzna: NBP API

Baza danych (SQLite)

- Tabele: users, wallets, transactions, exchange_rates
- Relacje: 1:N między users-wallets, users-transactions
- Indeksy: Optymalizacja zapytań dla często używanych kolumn

Funkcjonalności Podstawowe:

1. Zarządzanie kontem - rejestracja, logowanie, autoryzacja JWT
2. Portfele wielowalutowe - PLN, USD, EUR, GBP, CHF, JPY, CZK, NOK, SEK
3. Zasiliwanie konta - dodawanie środków PLN z weryfikacją
4. Wymiana walut - konwersja w oparciu o aktualne kursy NBP

5. Historia transakcji - pełny rejestr operacji użytkownika
 6. Przeglądanie kursów - aktualne i historyczne kursy walut
- Zaawansowane:**

1. Wykresy trendów - wizualizacja 7-dniowych trendów walutowych
 2. Kalkulator zysków/strat - analiza rentowności transakcji z kolorowym kodowaniem
 3. Ulubione pary walutowe - szybki dostęp do preferowanych wymian
 4. "Wszystkie środki" - opcja wymiany całego salda z jednej waluty
 5. Inteligentne walidacje - sprawdzanie sald, formatów dat, kwot
 6. Responsywny design - optymalizacja dla różnych rozmiarów ekranów
-

2. WYMAGANIA FUNKCJONALNE I NIEFUNKCJONALNE

2.1 WYMAGANIA FUNKCJONALNE

2.1.1 Zarządzanie kontem użytkownika

RF01 - Rejestracja użytkownika

- System umożliwia rejestrację nowego użytkownika
- Wymagane dane: email, hasło (min. 6 znaków), imię, nazwisko
- Walidacja unikalności adresu email
- Automatyczne szyfrowanie hasła (bcrypt)

RF02 - Logowanie użytkownika

- Autoryzacja na podstawie email + hasło
- Generowanie tokenu JWT o ważności 7 dni
- Automatyczne przekierowanie po poprawnym logowaniu

RF03 - Wylogowanie

- Możliwość wylogowania z aplikacji
- Usunięcie tokenu autoryzacji z pamięci lokalnej

2.1.2 Zarządzanie portfelami walutowymi

RF04 - Wyświetlanie portfeli

- Lista wszystkich portfeli użytkownika z aktualnym saldem
- Obsługa wielovalutowego portfela (PLN, USD, EUR, GBP, CHF, JPY, CZK, NOK, SEK) Obliczanie łącznego salda w wybranej walucie referencyjnej

RF05 - Zasilenie konta

- Dodawanie środków do portfela PLN
- Walidacja kwoty (liczby dodatnie, format dziesiętny)
- Szybki wybór kwot: 100, 500, 1000, 5000 PLN
- Opcja maksymalnego zasilenia (100,000 PLN)

2.1.3 Wymiana walut

RF06 - Wymiana walut

- Konwersja między dowolnymi obsługiwianymi walutami
- Wykorzystanie aktualnych kursów z NBP API
- Walidacja dostępności środków
- Opcja wymiany wszystkich dostępnych środków z danej waluty

RF07 - Zarządzanie ulubionymi parami walutowymi

- Dodawanie/usuwanie par walutowych do ulubionych
- Szybki dostęp do ulubionych par podczas wymiany Trwałe
- przechowywanie preferencji (AsyncStorage)

2.1.4 Przeglądanie kursów walut

RF08 - Aktualne kursy walut

- Wyświetlanie aktualnych kursów kupna i sprzedaży
- Automatyczne odświeżanie kursów
- Wskaźniki trendów (wzrost/spadek)

RF09 - Historia kursów

- Przeglądanie kursów z wybranego okresu
- Filtrowanie po walucie i datach (format RRRR-MM-DD) Walidacja
- dat i okresu

2.1.5 Historia transakcji

RF10 - Historia operacji

- Lista wszystkich transakcji użytkownika
- Filtrowanie po typie: zasilenia, wymiany walut
- Wyświetlanie szczegółów: data, typ, kwota, waluta, kurs

2.1.6 Analityka i statystyki

RF11 - Kalkulator zysków/strat

- Obliczanie zysku/straty dla transakcji wymiany
- Wyświetlanie procentowego wyniku
- Kolorowe wskaźniki (zielony - zysk, czerwony - strata)

RF12 - Wykresy trendów

- Mini wykresy 7-dniowych trendów dla głównych walut
- Wizualizacja kierunku zmian kursów
- Analiza tygodniowych trendów

2.2 WYMAGANIA NIEFUNKCJONALNE

2.2.1 Wydajność

RNF01 - Czas odpowiedzi

- Operacje lokalne (nawigacja, walidacja): < 100ms
- Operacje sieciowe (kursy, transakcje): < 2s Ładowanie
- aplikacji: < 3s

RNF02 - Przepustowość

- Obsługa co najmniej 100 równoczesnych użytkowników Limit
- 1000 zapytań/minutę do API NBP

2.2.2 Bezpieczeństwo

RNF03 - Autoryzacja i autentykacja

- Szyfrowanie haseł algorymem bcrypt (10 saltRounds)
- Tokeny JWT z czasem wygaśnięcia 7 dni
- Walidacja tokenów przy każdym zapytaniu do API

RNF04 - Ochrona danych

- Szyfrowanie komunikacji HTTPS (w produkcji)
- Walidacja wszystkich danych wejściowych
- Sanityzacja zapytań SQL (parametryzowane zapytania)

2.2.3 Użyteczność

RNF05 - Interfejs użytkownika

- Responsywny design dla różnych rozmiarów ekranów
- Intuicyjna nawigacja (React Navigation) Wsparcie dla gestów i dotyku

RNF06 - Dostępność

- Obsługa klawiatury (returnKeyType, navigation)
- Czytelne komunikaty o błędach
- Wskaźniki ładowania dla operacji długotrwałych

2.2.4 Niezawodność

RNF07 - Dostępność systemu

- Dostępność 99% czasu pracy
- Obsługa błędów sieciowych (retry, timeout)
- Graceful degradation przy braku połączenia

RNF08 - Integralność danych

- Transakcje bazodanowe (ACID)
- Backup automatyczny bazy danych
- Walidacja spójności sald portfeli

2.2.5 Skalowalność

RNF09 - Architektura

- Mikroserwisowa architektura (monorepo)
- Separacja logiki biznesowej od warstwy prezentacji Możliwość łatwego dodania nowych walut

RNF10 - Rozszerzalność

- Modularna struktura kodu
- łatwe dodanie nowych funkcji analitycznych
- Możliwość integracji z innymi API kursów

2.2.6 Przenośność

RNF11 - Kompatybilność platform

- React Native - obsługa iOS i Android
- Node.js backend - wieloplatformowość
- SQLite - lokalna baza danych

RNF12 - Wymagania techniczne

- Minimalna wersja: iOS 11.0, Android 6.0 (API 23)
- Node.js 16+ dla backendu
- Expo SDK 54 dla aplikacji mobilnej

2.2.7 Utrzymywalność

RNF13 - Jakość kodu

- Czytelna struktura katalogów (modułowa)
- Komentarze w kodzie dla złożonej logiki
- Nazewnictwo zgodne z konwencjami JavaScript/React

RNF14 - Monitorowanie

- Logowanie błędów w konsoli
- Komunikaty o statusie operacji
- Możliwość debugowania w trybie deweloperskim

2.3 OGRANICZENIA I ZAŁOŻENIA

2.3.1 Techniczne

- Wykorzystanie darmowego API NBP (limit zapytań)
- Brak obsługi płatności rzeczywistych (symulacja) Lokalne
- przechowywanie danych (bez chmury)

2.3.2 Biznesowe

- System edukacyjny/demonstracyjny
- Brak integracji z prawdziwymi instytucjami finansowymi
- Ograniczenie do wybranych głównych walut

2.3.3 Prawne

- Zgodność z RODO (lokalne dane)
- Brak licencji finansowych (system testowy)
- Dane kursów z oficjalnego źródła (NBP)

3. SZCZEGÓLowe PRZYPADKI UŻYCIA

UC01 - Rejestracja użytkownika

Aktor główny: Użytkownik

Poziom: Cel użytkownika

Warunki wstępne: Aplikacja jest uruchomiona

Warunki końcowe: Nowy użytkownik został zarejestrowany w systemie **Scenariusz**

główny:

1. Użytkownik otwiera ekran rejestracji
2. Użytkownik wprowadza dane: email, hasło, imię, nazwisko
3. Użytkownik naciska przycisk "Zarejestruj się"
4. System waliduje dane wejściowe
5. System sprawdza unikalność adresu email
6. System szyfruje hasło (bcrypt)
7. System zapisuje nowego użytkownika w bazie danych
8. System generuje token JWT
9. System przekierowuje na ekran główny **Scenariusze alternatywne:**

- **4a.** Dane są nieprawidłowe
 - 4a.1. System wyświetla komunikaty o błędach walidacji
 - 4a.2. Powrót do kroku 2
- **5a.** Email już istnieje w systemie
 - 5a.1. System wyświetla komunikat "Email już istnieje"
 - 5a.2. Powrót do kroku 2

UC02 - Logowanie użytkownika

Aktor główny: Użytkownik

Poziom: Cel użytkownika

Warunki wstępne: Użytkownik ma konto w systemie

Warunki końcowe: Użytkownik jest zalogowany **Scenariusz**

główny:

1. Użytkownik otwiera ekran logowania
2. Użytkownik wprowadza email i hasło
3. Użytkownik naciska przycisk "Zaloguj się"
4. System waliduje dane wejściowe
5. System sprawdza poprawność credentials w bazie danych
6. System generuje nowy token JWT
7. System przekierowuje na ekran główny **Scenariusze alternatywne:**

- **5a.** Nieprawidłowy email lub hasło
 - 5a.1. System wyświetla komunikat "Nieprawidłowy email lub hasło"
 - 5a.2. Powrót do kroku 2

UC06 - Wymiana walut

Aktor główny: Użytkownik

Aktor pomocniczy: NBP API

Poziom: Cel użytkownika

Warunki wstępne: Użytkownik jest zalogowany i ma środki do wymiany

Warunki końcowe: Wymiana została wykonana, salda zaktualizowane **Scenariusz**

główny:

1. Użytkownik otwiera ekran wymiany walut
2. System pobiera aktualne kursy z NBP API
3. System wyświetla dostępne waluty i salda portfeli
4. Użytkownik wybiera walutę źródłową i docelową
5. Użytkownik wprowadza kwotę do wymiany
6. System oblicza kwotę docelową na podstawie aktualnego kursu
7. System wyświetla podgląd wymiany
8. Użytkownik potwierdza wymianę
9. System sprawdza dostępność środków
10. System tworzy transakcję wymiany
11. System aktualizuje salda portfeli
12. System zapisuje transakcję w historii
13. System wyświetla potwierdzenie wymiany **Scenariusze alternatywne:**

- **9a.** Niewystarczające środki
 - 9a.1. System wyświetla komunikat "Niewystarczające środki"
 - 9a.2. Powrót do kroku 5

Scenariusze rozszerzające:

- **4a.** Użytkownik korzysta z ulubionych par walutowych (UC07)

UC07 - Zarządzanie ulubionymi parami walutowymi

Aktor główny: Użytkownik

Poziom: Funkcja

Warunki wstępne: Użytkownik jest na ekranie wymiany walut **Warunki**

końcowe: Para walutowa została dodana/usunięta z ulubionych

Scenariusz główny (dodawanie):

1. Użytkownik wybiera parę walut (np. PLN -> USD)
2. Użytkownik naciska ikonę serca obok pary
3. System dodaje parę do ulubionych w AsyncStorage
4. System zmienia ikonę na wypełnione serce
5. System wyświetla parę w sekcji ulubionych

1. Użytkownik naciska wypełnioną ikonę serca przy ulubionej parze
2. System usuwa parę z ulubionych
3. System zmienia ikonę na puste serce
4. System usuwa parę z sekcji ulubionych

UC11 - Kalkulator zysków/strat

Aktor główny: Użytkownik

Poziom: Cel użytkownika

Warunki wstępne: Użytkownik ma historię transakcji wymiany **Warunki**

końcowe: Wyświetlone zostały obliczone zyski/straty **Scenariusz**

główny:

1. Użytkownik otwiera ekran analityki
2. System pobiera historię transakcji użytkownika
3. System pobiera aktualne kursy walut
4. Dla każdej transakcji wymiany system oblicza:
 - Aktualną wartość wymienionej waluty w walucie źródłowej
 - Różnicę względem pierwotnej kwoty Procentowy
 - zysk/stratę
5. System wyświetla wyniki z kolorowym kodowaniem:
 - Zielony dla zysków
 - Czerwony dla strat
 - Szary dla braku zmiany

Algorytm obliczania:

```
aktualna_wartosc = kwota_docelowa * aktualny_kurs_odwrotny
zysk_strata = aktualna_wartosc - kwota_zrodlowa
procent = (zysk_strata / kwota_zrodlowa) * 100
```

UC13 - Pobieranie kursów NBP

Aktor główny: System

Aktor pomocniczy: NBP API

Poziom: Funkcja systemowa

Warunki wstępne: Dostęp do internetu

Warunki końcowe: Aktualne kursy zostały pobrane i zapisane **Scenariusz**

główny:

1. System wysyła zapytanie do NBP API
2. NBP API zwraca aktualne kursy walut
3. System formatuje otrzymane dane
4. System zapisuje kursy w lokalnej bazie danych
5. System zwraca kursy do wywołującej funkcji **Scenariusze alternatywne:**

- **1a.** Brak połączenia z internetem
 - 1a.1. System używa ostatnio zapisanych kursów z bazy danych
 - 1a.2. System wyświetla informację o braku aktualnych danych
- **2a.** NBP API zwraca błąd
 - 2a.1. System ponawia próbę po 5 sekundach (max 3 próby)
 - 2a.2. Jeśli wszystkie próby nieudane, przejście do scenariusza 1a

Endpointy NBP API:

- Tabela A (kursy średnie): <http://api.nbp.pl/api/exchangerates/tables/a>
- Tabela B (kursy średnie): <http://api.nbp.pl/api/exchangerates/tables/b>
- Historia: <http://api.nbp.pl/api/exchangerates/rates/{table}/{code}/{startDate}/{endDate}>

4. DIAGRAMY UML

4.1 Diagram przypadków użycia

Plik: use_case_diagram.puml

- 12 głównych przypadków użycia
- Relacje include/extend między UC
- Aktorzy: Użytkownik + NBP API

4.2 Diagram klas - aplikacja mobilna

Plik: mobile_class_diagram.puml

- Pakiet Screens: 9 ekranów React Native
- Pakiet Services: 6 serwisów
- komunikacyjnych Pakiet Models: 7 modeli

danych **4.3 Diagram klas - backend**

Plik: backend_class_diagram.puml

- Pakiet Controllers: 4 kontrolery HTTP
- Pakiet Services: 4 serwisy biznesowe
- Pakiet Models: 4 modele bazy danych
- Pakiet Middleware: 3 middleware

4.4 Diagram ERD bazy danych

Plik: erd_diagram.puml

- 4 główne tabele: users, wallets, transactions, exchange_rates
 - Relacje 1:N z foreign keys
 - Constraints i unique indexes
-

5. PODSUMOWANIE PROJEKTU

5.1 Spełnione wymagania (100%)

- Opracowanie wymagań funkcjonalnych i niefunkcjonalnych
- Przygotowanie diagramu przypadków użycia (UML)
- Przygotowanie diagramu klas (aplikacja mobilna + Web Service)
- Zaprojektowanie modelu bazy danych (ERD)
- Forma oddania: raport PDF + pliki diagramów

5.2 Technologie użyte Frontend:

- React Native + Expo SDK 54
- React Navigation 6, AsyncStorage, Axios

Backend:

- Node.js + Express, SQLite3, JWT + bcrypt

Integracje:

- NBP API (kursy walut), PlantUML (diagramy)

5.3 Zaawansowane funkcje (ponad wymagania)

1. **Analityka zysków/strat** - kolorowe wskaźniki rentowności
2. **Ulubione pary walutowe** - personalizacja UX
3. **Wykresy trendów** - wizualizacja 7-dniowych zmian
4. **Opcja "wszystkie środki"** - UX enhancement
5. **Inteligentne walidacje** - kompleksowa obsługa błędów

5.4 Jakość implementacji

- **Modułowa architektura** - monorepo packages/mobile + packages/api
- **Separation of concerns** - kontrolery, serwisy, modele
- **Security best practices** - bcrypt, JWT, CORS, walidacje
- **Error handling** - graceful degradation, user feedback
- **Modern UX/UI** - responsywny design, intuicyjna nawigacja

5.5 Kompletność dokumentacji

- **Analiza wymagań** - 12 RF + 14 RNF
- **Diagramy UML** - UC, klas (mobile/backend), ERD
- **Przypadki użycia** - szczegółowe scenariusze
- **Dokumentacja techniczna** - architektura, technologie

Projekt stanowi kompletny, działający system wymiany walut spełniający wszystkie wymagania edukacyjne z dodatkowymi zaawansowanymi funkcjami demonstrującymi umiejętności programistyczne.

