

## **Temat projektu: System mobilny kantoru wymiany walut**

---

### **1. Cel projektu**

Celem projektu jest zaprojektowanie i implementacja mobilnego systemu umożliwiającego użytkownikowi wykonywanie podstawowych operacji związanych z wymianą walut. Projekt ma na celu praktyczne zastosowanie zagadnień związanych z komunikacją pomiędzy aplikacją mobilną, serwisem sieciowym (SOAP/REST) oraz bazą danych, a także integrację z zewnętrznym API (Narodowy Bank Polski).

---

### **2. Zakres projektu**

#### **A. Aplikacja mobilna**

- Rejestracja i logowanie użytkownika
- Zasilenie *konta* (symulowany przelew wirtualny)
- Podgląd aktualnych kursów walut (pobieranych z API NBP)
- Dostęp do archiwalnych kursów walut
- Realizacja transakcji kupna/sprzedaży waluty
- Podgląd historii transakcji oraz stanu posiadanych środków

#### **B. Web Service (SOAP lub REST)**

- Realizacja logiki biznesowej kantoru
- Integracja z API NBP w celu pozyskiwania kursów
- Obsługa komunikacji z aplikacją mobilną
- Walidacja danych i autoryzacja użytkowników

#### **C. Baza danych**

- Przechowywanie informacji o użytkownikach
  - Rejestrowanie transakcji
  - Zapisywanie stanu portfela walutowego użytkownika
- 

### **3. Etapy realizacji**

#### **Część 1 — Projekt koncepcyjny (etap analityczno-projektowy)**

1. Opracowanie wymagań funkcjonalnych i нефункциональных
2. Przygotowanie diagramu przypadków użycia (UML)
3. Przygotowanie diagramu klas (dla aplikacji mobilnej i Web Service)

4. Zaprojektowanie modelu bazy danych (diagram ERD)

**Forma oddania:** raport w formacie PDF + pliki diagramów

## Część 2 — Implementacja (etap programistyczny)

1. Implementacja aplikacji mobilnej w dowolnym języku/środowisku (np. Kotlin, Flutter, React Native)
2. Implementacja Web Service (np. Java Spring Boot, Node.js, .NET Core)
3. Implementacja bazy danych (np. MySQL, PostgreSQL, SQLite)
4. Integracja wszystkich komponentów
5. Przygotowanie krótkiej dokumentacji użytkownika

**Forma zaliczenia:** prezentacja projektu + demonstracja działania aplikacji

---

## 4. Kryteria oceniania

Kryterium	Waga	Opis
Poprawność działania aplikacji	30%	System działa zgodnie z założeniami, poprawna komunikacja z serwisem i bazą danych
Jakość projektu technicznego (część 1)	20%	Kompletność i poprawność diagramów UML, przejrzysta baza danych
Architektura systemu i integracja	20%	Zastosowanie właściwego podziału warstw, poprawna implementacja API i autoryzacji
Interfejs użytkownika i ergonomia	10%	Przejrzysty, intuicyjny interfejs mobilny
Dokumentacja techniczna i prezentacja	10%	Jasny opis działania, instrukcja uruchomienia, prezentacja
Dodatkowa funkcjonalność / kreatywność	10%	Dodatkowe moduły, np. alerty kursowe, wykresy, obsługa wielu języków, tryb offline

### Ocena końcowa:

- 5.0 – projekt kompletny, spełnia wszystkie wymagania, zawiera elementy dodatkowe
- 4.5 – projekt kompletny, drobne błędy lub brak funkcji dodatkowych
- 4.0 – spełnia wymagania podstawowe, poprawna architektura
- 3.5 – funkcjonalny, ale z ograniczoną integracją lub błędami logicznymi

- 3.0 – minimalna funkcjonalność, brak części dokumentacji
  - 2.0 – projekt nie działa lub nie został oddany
- 

## **5. Wymagania techniczne**

- Aplikacja powinna działać na systemie Android (lub wieloplatformowo).
  - Komunikacja z serwerem przez HTTP/HTTPS.
  - Baza danych lokalna lub zdalna.
  - Kod źródłowy dostępny w repozytorium Git (GitHub, GitLab).
  - Raport końcowy i prezentacja projektu obowiązkowe.
- 

## **6. Propozycja alternatywnego tematu projektu**

### **System mobilny do zarządzania inteligentnym parkingiem miejskim**

#### **Opis:**

System umożliwia użytkownikowi znalezienie, rezerwację i opłacenie miejsca parkingowego w mieście.

Aplikacja mobilna komunikuje się z Web Service, który przechowuje dane o miejscach, użytkownikach i płatnościach.

#### **Wymagania funkcjonalne:**

- Rejestracja/logowanie użytkownika
- Wyszukiwanie wolnych miejsc parkingowych (według lokalizacji)
- Rezerwacja miejsca i czasowe blokowanie
- Opłacenie postoju (symulowana płatność)
- Historia rezerwacji
- Panel administracyjny (w Web Service) do zarządzania miejscami

#### **Technologie:**

Analogicznie jak w projekcie kantoru — aplikacja mobilna + Web Service + baza danych.