Collegium Witelona Uczelnia Państwowa

Wydział Informatyki

Dokumentacja Projektowa

WitelonBank - Aplikacja Dekstopowa

Przedmiot: Zaawansowane metody programowania
Rok akademicki: 2024/2025

Krystian Raczyński (42757)

Pod przewodnictwem mgr inż. Krzysztof Rewak

Spis treści

1 Wprowadzenie

WitelonBank to aplikacja bankowa napisana w języku Java z wykorzystaniem biblioteki JavaFX. Projekt umożliwia obsługę zarówno użytkowników końcowych (klientów), jak i administratorów systemu poprzez osobne panele.

2 Wykorzystane technologie

- **Java 21** – główny język programowania.
- **JavaFX 21** – biblioteka do tworzenia graficznego interfejsu użytkownika.
- **Maven** – system zarządzania zależnościami (z wtyczką `javafx-maven-plugin`).
- **JSON** – format wymiany danych z API (`org.json`, `Jackson`).

3 Uruchomienie lokalne

1. Zainstaluj JDK 21 oraz Maven.
2. W katalogu projektu uruchom:

   ```
   mvn clean javafx:run
   ```

   Komenda wykorzystuje `javafx-maven-plugin`, główna klasa: `com.kracz0.desktopwitelonbank.App`.
3. Alternatywnie, uruchom plik JAR: `out/artifacts/desktopWitelonBank_jar/desktopWitelonBank.jar`, z poprawną konfiguracją modułów JavaFX.

4 Struktura projektu

```
com.kracz0.desktopwitelonbank
 App.java                - punkt startowy aplikacji
 Config                  - konfiguracja adresów API
 Controllers             - kontrolery widoków (Client/Admin)
 Models                  - modele danych i singleton ‘Model‘
 Services                - logika komunikacji z API
 Utils                   - pomocnicze klasy HTTP
 Views                   - ‘ViewFactory‘ do ładowania widoków FXML
 resources               - pliki FXML i arkusze CSS
```

5 Najważniejsze klasy

5.1 App

Główna klasa aplikacji, dziedziczy po `javafx.application.Application`. Metoda `start()` wywołuje `ViewFactory.showLoginWindow()`.

5.2 Model

Singleton przechowujący stan aplikacji (zalogowany użytkownik, fabryka widoków).

- `getInstance()`
- `getViewFactory()`
- `setLoggedUser(User user)`, `getLoggedUser()`
- `logout()`

5.3 ViewFactory

Odpowiada za wczytywanie widoków FXML. Udostępnia m.in.:

- `showLoginWindow()`
- `showClientWindow()`
- `showTwoFactorModal(email, stage)`
- `getDashboardView()`, `getTransactionsView()` itp.

5.4 Kontrolery klienta (Controllers.Client)

- **ClientController** – zarządza głównym układem widoku klienta.
- **DashboardController** – ładuje dane konta, karty, kryptowaluty, transakcje.
- **TransactionsController** – obsługuje historię przelewów i formularze.
- **AddressBookController** – zapisani odbiorcy.
- **CryptoController** – zakup/sprzedaż kryptowalut.
- **ClientMenuController** – obsługa przycisków menu.
- **Modale (np. TwoFactorController)** – dialogi np. 2FA.

5.5 Kontrolery administratora (Controllers.Admin)

- **AdminController** – panel administratora: lista kont, statystyki, raporty.
- **AdminAccountDetailsController** – szczegóły konta (limity, transakcje).

5.6 Modele danych (Models, Models.DTO)

- **User**, **Card**, **Recipient** – encje klienta.
- DTO: `Account`, `Transfer`, `CryptoWallet`, `AdminStats`, `StandingOrder`, `AccountAdmin`.

5.7 Serwisy (Services)

- **AuthService** – logowanie, weryfikacja 2FA.
- **DashboardService** – dane dashboardu klienta.
- **TransactionsService** – wysyłanie przelewów.
- **AddressBookService** – operacje CRUD na odbiorcach.
- **CryptoService** – kursy i portfel kryptowalut.
- **AdminService** – funkcje administratora (np. raporty).

5.8 Utils

- **ApiClient** – obsługa autoryzowanych zapytań `HttpClient`.
- **HttpUtil** – uproszczony builder zapytań HTTP.

5.9 Config

- **ApiConfig** – adresy endpointów REST API, np. `LOGIN`, `PRZELEWY`, `ADMIN_KONTA`.

6 Zasoby

Pliki FXML oraz arkusze CSS znajdują się w katalogu `src/main/resources`. Kluczowe widoki:

- `Login.fxml`
- `Client.fxml`
- `Admin.fxml`
- Widoki cząstkowe w folderach `Client` oraz `Admin`.

7 Komunikacja z API

Aplikacja korzysta z REST API pod adresem bazowym zdefiniowanym w `ApiConfig.BASE_URL`: `https://witelonapi.host358482.xce.pl/api`
Autoryzacja odbywa się za pomocą tokenu JWT zwracanego po poprawnym logowaniu.

8 Funkcjonalności

8.1 Dla użytkownika

- Logowanie oraz weryfikacja dwuskładnikowa.
- Podgląd salda konta i historii operacji.
- Wykonywanie przelewów, zarządzanie odbiorcami.
- Obsługa kart płatniczych i zleceń stałych.
- Inwestycje w kryptowaluty.
- Eksport operacji do pliku.

8.2 Dla administratora

- Logowanie do panelu administratora.
- Zarządzanie kontami (blokady, limity).
- Podgląd i filtrowanie transakcji.
- Generowanie raportów PDF.
- Statystyki i monitorowanie systemu.

9 Repozytorium GitHub

`https://github.com/KRacz0/desktopWitelonBank`