

Etap 3

Zespół 11

Wymagania funkcjonalne

Pełna lista funkcjonalności naszej aplikacji. Stopień spełnionych wymagań dla zespołu: 100%

1. Modyfikacja danych użytkownika

- Umożliwienie zmiany podstawowych danych tj. Imię, nazwisko, adres email, hasło, numer pin

2. Wyświetlanie salda rachunku

- Umożliwienie podglądu stanu każdego rachunku

3. Wyświetlanie historii transakcji

- Zapewnienie pełnej historii transakcji, umożliwiającą użytkownikowi śledzenie i analizowanie swoich operacji finansowych.

4. Tworzenie rachunków

- Umożliwienie utworzenia wielu rachunków w ramach jednego konta

5. Tworzenie lokat

- Zapewnia prosty mechanizm oszczędzania poprzez tworzenie lokat

6. Przelewy na własne rachunki

- Umożliwienie użytkownikowi w szybki sposób wykonywania przelewów między własnymi rachunkami.

7. Przelewy między kontami

- Umożliwienie użytkownikowi wykonywania przelewów na inne konta.

8. Zlecenia stałe

- Umożliwienie użytkownikowi dodania do swojego konta zlecenia, które będzie wykonywać się z określoną częstotliwością.

9. Szybkie przelewy

- Umożliwienie użytkownikowi wykonywania szybkich przelewów na inne konta, które znajdują się na liście jego kontaktów.

10. Zabezpieczenia przelewów

- Zabezpieczenie przelewu koniecznością podania PIN-u przez użytkownika.

11. Powiadomienia o transakcjach

- Powiadomienia o przelewach wysyłane na adres e-mail

12. Zmiana limitów transakcji

- Umożliwienie ustawienia limitów pojedynczej transakcji

13. Zakładanie i obsługa kredytów

- Zapewnienie mechanizmu kredytowego - pieniądze trafiają na konto jednorazowo, raty są pobierane automatycznie

14. Kalkulator kredytowy

- Umożliwienie podglądu wysokości raty na podstawie okresu spłaty i wysokości kredytu

15. Automatyczne oszczędzanie

- Mechanizm automatycznego oszczędzania np. Codziennie z rachunku głównego pewna suma trafia na rachunek oszczędnościowy

16. Rejestracja użytkownika

- Umożliwienie użytkownikowi utworzenia konta, wymagając podania podstawowych informacji, takich jak imię, nazwisko, adres e-mail, hasło, pin. Aplikacja wymaga też potwierdzenia maila poprzez podanie zawartego w nim kodu.

17. Logowanie użytkownika

- Zapewnienie funkcji logowania dla zarejestrowanych użytkowników, wymagając potwierdzenia poprawności numeru konta, hasła.

18. Logowanie dwuetapowe

- Dodatkowa weryfikacja użytkownika przez kod pin.

19. Powiadomienia o logowaniu

- Powiadomienie użytkownika poprzez e-mail o nowych logowaniach na jego konto.

20. Zmiana hasła

- Umożliwienie użytkownikowi zmiany hasła w przypadku, gdy użytkownik go zapomniał.

Opis aplikacji

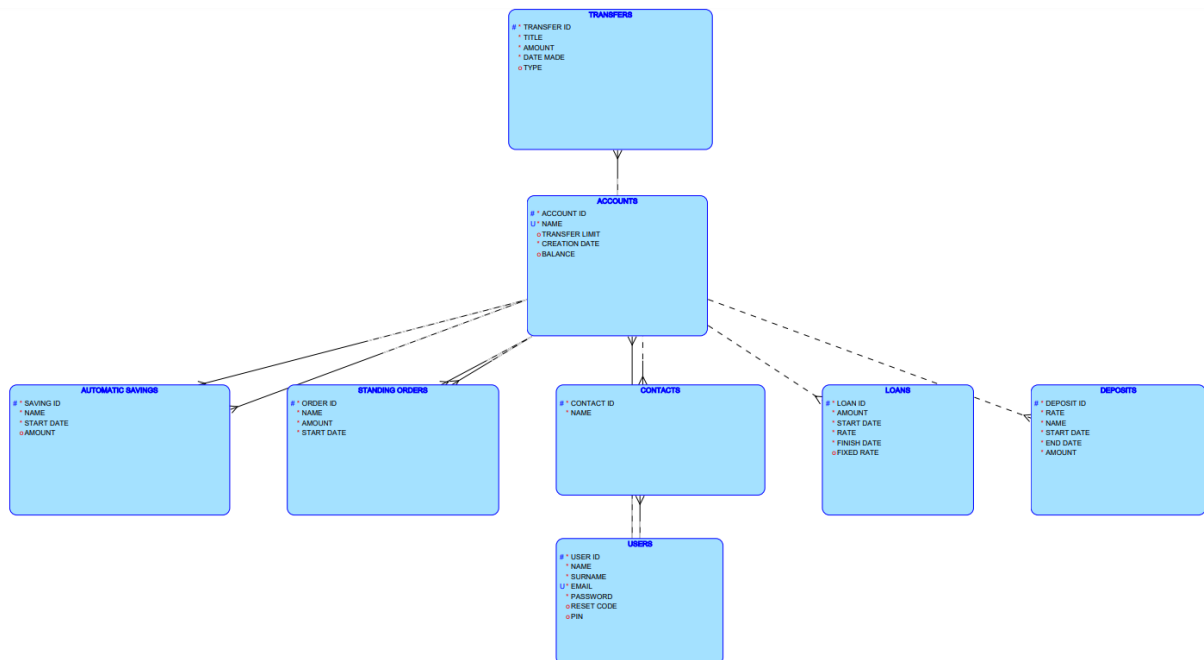
Nasza aplikacja jest modelem aplikacji bankowej, w której użytkownik może wykonać podstawowe bankowe operacje. Można zarejestrować nowe konto, zalogować się na nie, tworzyć nowe rachunki dla konta, przysyłać pieniądze między kontami, zakładać lokaty lub kredyty, tworzyć zlecenia stałe, szybkie przelewy, czy używać mechanizmu automatycznego oszczędzania. Aplikacja zapewnia też mechanizmy odzyskiwania hasła, czy zabezpieczenia, takie jak potwierdzanie transakcji przez kod pin, czy wysyłanie maili do użytkownika w przypadku logowania, lub transakcji.

Użyte technologie:

- Maven: Wykorzystaliśmy system zarządzania zależnościami Maven, co ułatwia zarządzanie bibliotekami i konfiguracją projektu. Maven pozwala na efektywne budowanie, zarządzanie zależnościami oraz tworzenie archiwum JAR.
- Oracle Cloud: Jako platforma chmurowa, Oracle Cloud dostarcza infrastrukturę do hostowania aplikacji oraz baz danych. Skorzystaliśmy z usług Oracle Cloud, żeby każdy członek zespołu miał dostęp do bazy danych w czasie rzeczywistym.
- Oracle JDBC (ojdbc8): Do komunikacji z bazą danych Oracle wykorzystaliśmy ojdbc8. JDBC umożliwia interakcję z bazą danych za pomocą języka Java, co jest kluczowe w przypadku aplikacji, które muszą przechowywać i pobierać dane.
- Toedter JCalendar: pozwalające wybierać datę z interaktywnego kalendarza. Ułatwia to np. wybór daty końca lokaty.
- Com.sun.mail javax.mail: oprogramowanie do obsługi wysyłania maili.
- Org.junit.jupiter junit-jupiter: oprogramowanie do uruchamiania testów jednostkowych.

Struktura bazy danych

Model ER bazy danych



Model fizyczny bazy danych

