

1 Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: **Aplikacja banku „Parabank”**
Autorzy: **Kacper Białas, Stanisław Gałek, Kacper Macikowski, Jakub Oglecki**
Grupa: **I1-211A**
Kierunek: **informatyka**
Rok akademicki: **2023/2024**
Poziom i semestr: **I/4**
Tryb studiów: **stacjonarne**

Spis treści

1 Dokumentacja projektu zaliczeniowego.....	2
Przedmiot: Inżynieria oprogramowania.....	2
2 Odnośniki do innych źródeł.....	5
3 Słownik pojęć.....	6
4 Wprowadzenie.....	7
4.1 Cel dokumentacji.....	7
4.2 Przeznaczenie dokumentacji.....	7
4.3 Opis organizacji lub analiza rynku.....	7
4.4 Analiza SWOT organizacji.....	7
5 Specyfikacja wymagań.....	9
5.1 Charakterystyka ogólna.....	9
5.1.1 Definicja produktu.....	9
5.1.2 Podstawowe założenia.....	9
5.1.3 Cel biznesowy.....	9
5.1.4 Użytkownicy.....	9
5.1.5 Korzyści z systemu.....	10
5.1.6 Ograniczenia projektowe i wdrożeniowe.....	11
5.2 Wymagania funkcjonalne.....	12
5.2.1 Lista wymagań.....	12
5.2.2 Diagramy przypadków użycia.....	13
5.2.3 Szczegółowy opis wymagań.....	22
5.3 Wymagania niefunkcjonalne.....	27
6 Zarządzanie projektem.....	29
6.1 Zasoby ludzkie.....	29
6.2 Harmonogram prac.....	29
6.3 Etapy/kamienie milowe projektu.....	29
7 Zarządzanie ryzykiem.....	30
7.1 Lista czynników ryzyka.....	30
7.2 Ocena ryzyka.....	30
7.3 Plan reakcji na ryzyko.....	30
8 Zarządzanie jakością.....	31
8.1 Scenariusze i przypadki testowe.....	31

9 Projekt techniczny.....	32
9.1 Opis architektury systemu.....	32
9.2 Technologie implementacji systemu.....	33
9.3 Diagramy UML.....	34
9.3.1 Diagram(-y) klas.....	34
9.3.2 Diagram(-y) czynności.....	35
9.3.3 Diagramy sekwencji.....	37
9.3.4 Inne diagramy.....	44
9.4 Charakterystyka zastosowanych wzorców projektowych.....	47
9.5 Projekt bazy danych.....	48
9.5.1 Schemat.....	48
9.5.2 Projekty szczegółowe tabel.....	48
9.6 Projekt interfejsu użytkownika.....	48
9.6.1 Lista głównych elementów interfejsu.....	48
9.6.2 Przejścia między głównymi elementami.....	50
9.6.3 Projekty szczegółowe poszczególnych elementów.....	51
9.7 Procedura wdrożenia.....	55
10 Dokumentacja dla użytkownika.....	56
11 Podsumowanie.....	57
11.1 Szczegółowe nakłady projektowe członków zespołu.....	57
12 Inne informacje.....	58

2 Odnośniki do innych źródeł

Repozytorium Git: <https://github.com/projekt-z-io/IO>

Zarządzanie projektem: <https://trello.com/b/uMatgq1J/projekt-io>

3 Słownik pojęć

- a. Administrator - Osoba odpowiedzialna za zarządzanie systemem, siecią lub aplikacją, w tym nadzór, konfigurację, i udzielanie uprawnień.
- b. Pracownik obsługi klienta - Osoba zajmująca się bezpośrednim wsparciem i obsługą klientów, rozwiązywaniem problemów oraz udzielaniem informacji o produktach lub usługach.
- c. Pracownik działu IT - Osoba pracująca w dziale technologii informacyjnej (IT), odpowiedzialna za utrzymanie, rozwój i wsparcie systemów informatycznych w firmie.
- d. Analityk finansowy - Osoba zajmująca się analizą danych finansowych, prognozami, raportami i doradztwem w zakresie finansów przedsiębiorstwa.
- e. Klient w wersji podstawowej - Osoba korzystająca z podstawowej wersji produktu lub usługi.
- f. Klient premium wersji gold - Klient korzystający z usługi lub produktu na poziomie premium z dodatkowymi funkcjami lub uprawnieniami, określany czasem jako wersja gold.
- g. Klient premium wersja Platinum - Klient korzystający z najwyższego poziomu usługi lub produktu z ekskluzywnymi funkcjami i priorytetowym wsparciem, czasem określany jako wersja Platinum.
- h. Klient biznesowy - Osoba lub firma korzystająca z produktów lub usług zorientowanych na potrzeby biznesowe, zazwyczaj mająca specjalne wymagania i oczekiwania.
- i. Partner handlowy - Podmiot, z którym firma nawiązała formalną współpracę handlową, np. dystrybutor, sprzedawca lub kontrahent.

4 Wprowadzenie

4.1 Cel dokumentacji

Celem dokumentacji jest zapewnienie precyzyjnych informacji dotyczących postępów projektu i jego szczegółów. Ma ona umożliwić skuteczne korzystanie z aplikacji wszystkim zainteresowanym stronom. Ma również wspierać rozwój aplikacji poprzez dostarczanie i zapisywanie niezbędnych informacji dotyczących dalszych działań.

4.2 Przeznaczenie dokumentacji

Dokumentacja aplikacji bankowej służy programistom, testerom, menedżerom projektu i personelowi wsparcia technicznego. Dostarcza szczegółowych informacji o strukturze, funkcjach, testach, postępach projektu i obsłudze błędów. Jej celem jest umożliwienie skutecznego korzystania z aplikacji i wsparcie w jej rozwoju.

4.3 Opis organizacji lub analiza rynku

Organizacja jest bankiem o nazwie "Parabank". Będzie skupiony na systemach informatycznych do zarządzania danymi klientów, transakcjami finansowymi, analizy ryzyka i zapewnienia bezpieczeństwa. Systemy te muszą być skalowane, wydajne i niezawodne, aby sprostać wymaganiom intensywnego obciążenia transakcyjnego oraz zapewnić ochronę danych finansowych. Organizacja oferuje szeroki zakres usług bankowych, takich jak przyjmowanie depozytów, udzielanie pożyczek, zarządzanie inwestycjami oraz świadczenie usług płatniczych.

4.4 Analiza SWOT organizacji

	Szanse	Zagrożenia
Silne	-Rozwój zaawansowanych technologii bankowych, takich jak sztuczna inteligencja, analiza danych czy blockchain	-Wzrost cyberataków i zagrożeń dla bezpieczeństwa danych
	-Rosnące zapotrzebowanie na usługi finansowe	-Ryzyko zmian w polityce regulacyjnej i prawnej
	-Możliwość rozszerzenia oferty o nowe produkty finansowe i usługi (Giełda kryptowalut,NFT itp.)	-Możliwość załamania się rynków finansowych lub kryzysu gospodarczego
Słabe	-Partnerstwo z fintechami i startupami	
	- Brak dostępu do zaawansowanych	-Wzrost oczekiwania klientów co do

	<p>technologii bankowych (trudność w dostępie do np. systemu BLIK)</p> <p>-Niskie zrozumienie klientów na temat nowych produktów i usług finansowych</p> <p>-Niewystarczające zasoby finansowe na inwestycje w rozwój nowych produktów i technologii</p> <p>-Ograniczona zdolność do szybkiego reagowania na zmieniające się potrzeby</p>	<p>szynbkości, łatwości obsługi oraz dostępności usług</p> <p>-Trudność w dostosowaniu oprogramowania do specyficznych wymagań i regulacji prawnych.</p> <p>-Wysokie koszty utrzymania infrastruktury technologicznej</p> <p>-Ryzyko utraty zaufania klientów związane z incydentami zowanymi z bezpieczeństwem danych</p> <p>-Wzrost konkurencji ze strony innych instytucji finansowych</p>
--	---	---

5 Specyfikacja wymagań

5.1 Charakterystyka ogólna

5.1.1 Definicja produktu

System “Parabank” to aplikacja internetowa, zaprojektowana dla sektora bankowego, umożliwiająca wysoką jakość świadczonych usług finansowych i zapewniająca bezpieczeństwo transakcji.

5.1.2 Podstawowe założenia

“Parabank” zapewnia klientom sektora bankowego łatwy dostęp do różnorodnych usług finansowych poprzez aplikację internetową. System ma umożliwić wygodne zarządzanie kontami bankowymi, wykonywanie szybkich i bezpiecznych transakcji, jak również monitorowanie stanu finansów. Oprócz tego będzie oferować narzędzia do planowania budżetu, inwestycji oraz dostęp do aktualnych informacji związanych z rynkiem finansowym. Priorytetem jest zapewnienie wysokiej jakości obsługi klienta oraz ochrona danych osobowych i finansowych użytkowników.

5.1.3 Cel biznesowy

System ma na celu usprawnić obsługę klientów poprzez szybsze i bardziej precyzyjne przetwarzanie transakcji, oraz przede wszystkim zwiększyć bezpieczeństwo wykonywanych operacji finansowych. Ma również poszerzyć ofertę usług finansowych, umożliwiając dostęp do nowych technologii przy współpracy z firmami fintechowymi. Ma to za zadanie dostosować system do dynamicznie zmieniającego się rynku, aby organizacja mogła przyjmować innowacyjne rozwiązania i świadczyć usługi na najwyższym poziomie.

5.1.4 Użytkownicy

3. Administrator
4. Pracownik obsługi klienta
5. Pracownik działu IT
6. Analityk finansowy
7. Klient w wersji podstawowej
8. Klient premium wersji gold
9. Klient premium wersja Platinium
 - Klient biznesowy
 - Partner handlowy

5.1.5 Korzyści z systemu

a. Administrator:

- Łatwiejsze zarządzanie użytkownikami i uprawnieniami.
- Dostęp do zintegrowanych narzędzi raportowania i monitorowania wydajności systemu.
- Automatyzacja rutynowych zadań administracyjnych.

b. Pracownik obsługi klienta:

- Szybszy dostęp do informacji o klientach i historii interakcji.
- Możliwość szybkiego rozwiązywania problemów klientów dzięki lepszemu dostępowi do danych.

c. Pracownik działu IT:

- Łatwiejsze zarządzanie infrastrukturą IT i rozwiązywanie problemów technicznych.
- Możliwość monitorowania wydajności systemu i szybszego reagowania na ewentualne zagrożenia.

d. Analityk finansowy:

- Dostęp do bogatych danych finansowych i możliwość generowania szczegółowych raportów.
- Możliwość analizy trendów i prognozowanie przyszłych wyników finansowych.

e. Klient w wersji podstawowej:

- Prostszy dostęp do podstawowych funkcji systemu.
- Możliwość śledzenia swoich transakcji i historii zakupów.

f. Klient premium wersji Gold:

- Dostęp do dodatkowych funkcji i usług, takich jak priorytetowe wsparcie klienta.
- Oferty specjalne i zniżki dostępne tylko dla klientów premium.

g. Klient premium wersja Platinium:

- Najwyższy poziom wsparcia i dostęp do ekskluzywnych usług.
- Indywidualne podejście i personalizowane rozwiązania.

h. Klient biznesowy:

- Możliwość zarządzania kontem firmowym i wieloma użytkownikami.
- Dostęp do narzędzi do analizy danych biznesowych i optymalizacji procesów.

i. Partner handlowy:

1. Dostęp do specjalnych zasobów i narzędzi wspierających proces sprzedaży.
2. Możliwość śledzenia postępu w relacji z klientami i zarządzania nimi w systemie.

5.1.6 Ograniczenia projektowe i wdrożeniowe

4. Aplikacja bankowa musi być zgodna z przepisami prawnymi dotyczącymi ochrony danych osobowych, takimi jak RODO (ogólne rozporządzenie o ochronie danych osobowych), oraz regulacjami dotyczącymi finansów, takimi jak dyrektywy GLBA (Gramm-Leach-Bliley Act) - regulacja dotycząca prywatności informacji finansowej klientów.
5. FCRA (Fair Credit Reporting Act) - regulacja dotycząca sprawiedliwych praktyk raportowania kredytowego i prywatności informacji.
6. AML (Anti-Money Laundering regulations) - regulacje dotyczące zapobiegania praniu brudnych pieniędzy, wymagające raportowania podejrzanych transakcji finansowych.
7. KYC (Know Your Customer) - wymóg identyfikacji klientów i zbierania informacji na ich temat w celu zapobiegania oszustwom finansowym i praniu brudnych pieniędzy.
8. PSD2 - dyrektywa Parlamentu Europejskiego i Rady UE z dnia 25 listopada 2015 r. w sprawie usług płatniczych w ramach rynku wewnętrznego.

Aplikacja bankowa musi korzystać z bezpiecznych protokołów komunikacyjnych, takich jak HTTPS, do komunikacji między klientem a serwerem oraz wewnętrznie między różnymi komponentami systemu. Musi być wyposażona w różnorodne mechanizmy zabezpieczeń, takie jak uwierzytelnianie dwuetapowe, monitorowanie aktywności użytkowników, szyfrowanie danych oraz systemy detekcji i reakcji na intruzów (IDS/IPS). Uzasadnienie: Bezpieczna komunikacja jest niezbędna do zapobiegania atakom typu man-in-the-middle i kradzieży danych.

Aplikacja musi być zintegrowana z innymi systemami bankowymi, takimi jak systemy płatności, aplikacje mobilne oraz systemy zarządzania klientami (CRM), w celu zapewnienia spójności danych i płynności operacji.

5.2 Wymagania funkcjonalne

5.2.1 Lista wymagań

1. System logowania:

- A) Użytkownik musi mieć możliwość rejestracji konta za pomocą adresu e-mail i hasła.
- B) Wymagane jest potwierdzenie tożsamości poprzez link aktywacyjny wysłany na podany adres e-mail.
- C) Istnieje opcja resetowania hasła poprzez wysłanie linku resetującego na adres e-mail użytkownika.

2. Panel główny:

- A) Po zalogowaniu użytkownik powinien mieć dostęp do panelu głównego z podsumowaniem stanu konta, ostatnimi transakcjami oraz opcjami szybkich działań.
- B) Wszystkie operacje finansowe powinny być przedstawione w czytelny sposób, z możliwością filtrowania i wyszukiwania transakcji.

3. Przelewy:

- A) Użytkownik powinien mieć możliwość wykonywania przelewów między swoimi kontami, jak również na zewnętrzne konta bankowe.
- B) Przy realizacji przelewów konieczne jest potwierdzenie poprzez podanie kodu weryfikacyjnego lub potwierdzenie SMS-em.

4. Historia transakcji:

- A) Aplikacja powinna przechowywać historię wszystkich transakcji oraz udostępniać użytkownikowi dostęp do niej.
- B) Dostępna powinna być możliwość eksportu historii transakcji w formacie pliku CSV lub PDF.

5. Zarządzanie kontem:

- A) Użytkownik powinien mieć możliwość edycji danych osobowych oraz adresowych powiązanych z kontem.
- B) Istnieje opcja zmiany preferowanego języka interfejsu aplikacji. c. Użytkownik może dokonywać zmiany ustawień powiadomień SMS i e-mail.

6. Bezpieczeństwo:

A) Aplikacja powinna wykorzystywać najnowsze standardy szyfrowania danych oraz zabezpieczeń przed atakami typu SQL injection i cross-site scripting (XSS).

B) Dostęp do aplikacji powinien być zabezpieczony dwuetapową weryfikacją.

7. Obsługa klienta:

A) Użytkownik powinien mieć możliwość kontaktu z obsługą klienta poprzez czat online, formularz kontaktowy oraz telefon.

B) Wszelkie zgłoszenia od klientów powinny być monitorowane i rozpatrywane w możliwie najkrótszym czasie.

8. Kompatybilność:

A) Aplikacja powinna być kompatybilna z najpopularniejszymi systemami operacyjnymi i przeglądarkami internetowymi.

B) Interfejs użytkownika powinien być responsywny i dostosowany do różnych urządzeń mobilnych.

9. Aktualizacje i utrzymanie:

A) Regularnie należy wprowadzać aktualizacje aplikacji, poprawiając błędy, dodając nowe funkcje oraz dostosowując ją do zmieniających się przepisów prawnych.

B) W przypadku konieczności przerwy w działaniu systemu ze względu na konserwację lub aktualizację, użytkownicy powinni być poinformowani z wyprzedzeniem.

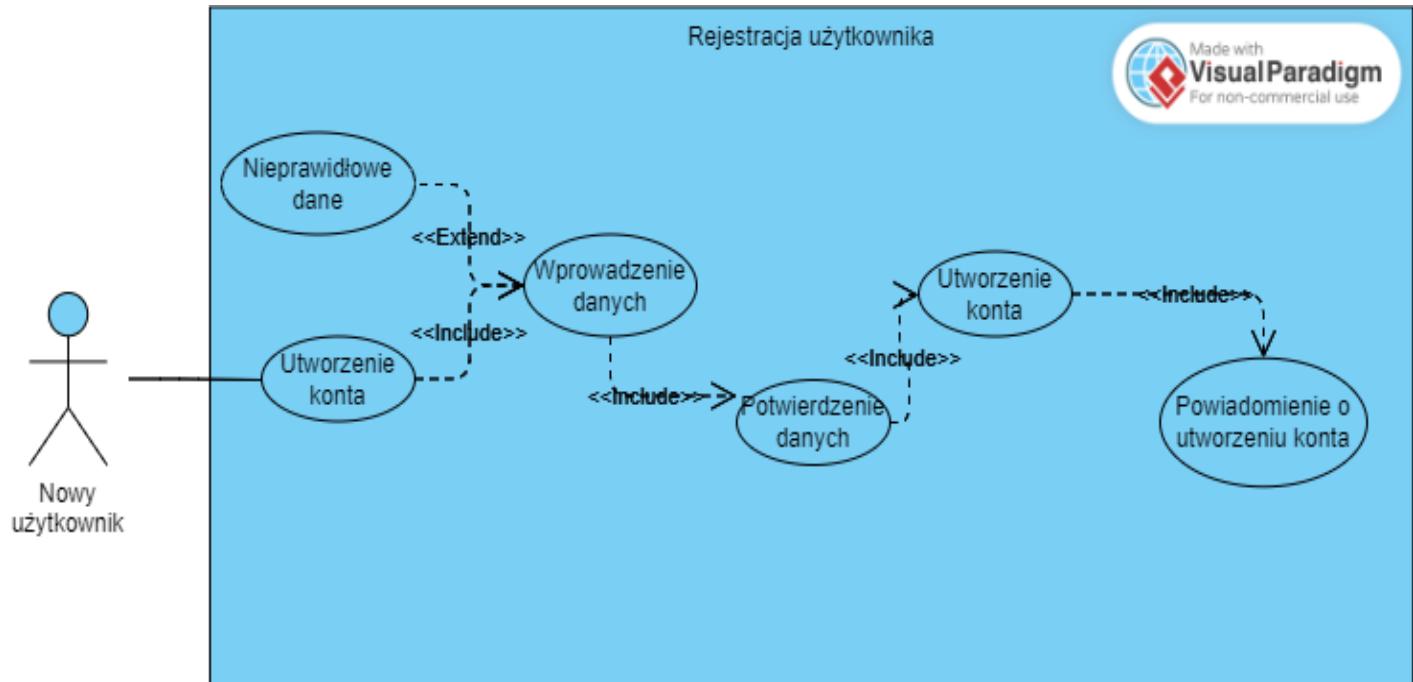
10. Polityka prywatności:

A) Aplikacja powinna jasno prezentować politykę prywatności, informując użytkowników o sposobie gromadzenia, przechowywania i wykorzystywania ich danych osobowych.

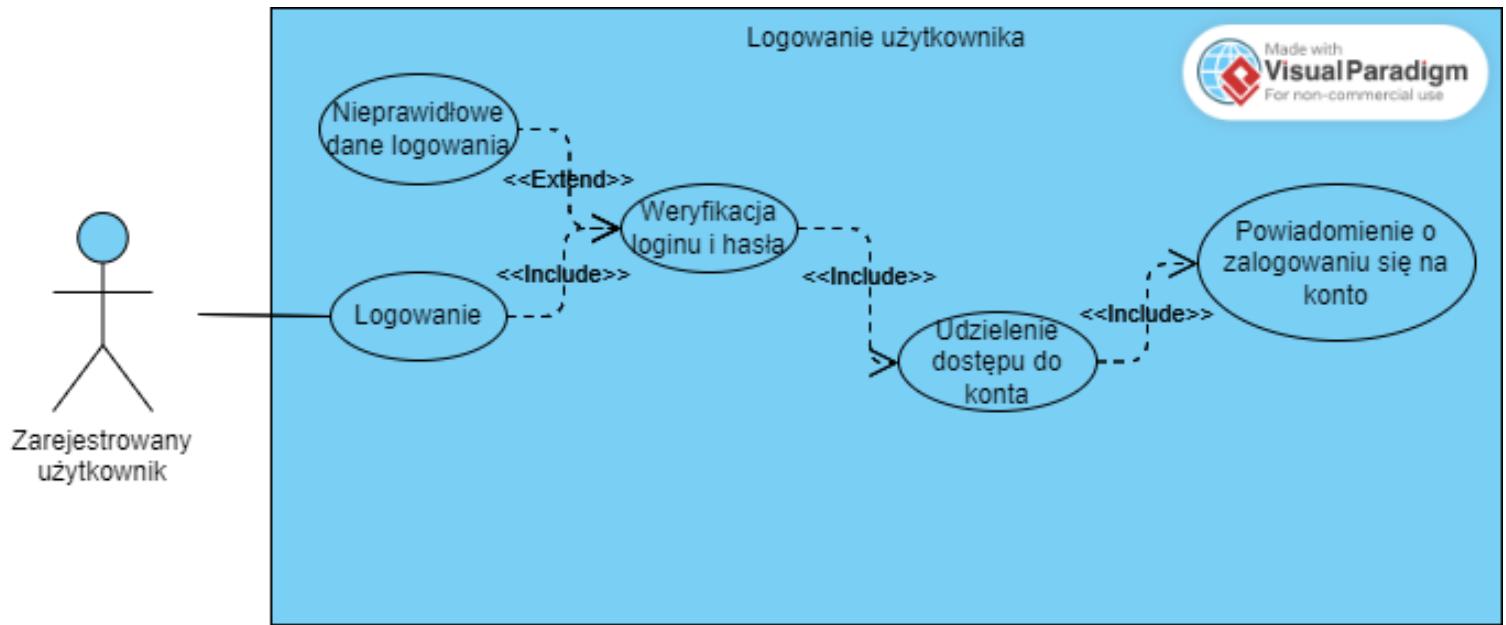
B) Użytkownicy powinni mieć możliwość wyrażenia zgody na przetwarzanie swoich danych osobowych zgodnie z polityką prywatności.

5.2.2 Diagramy przypadków użycia

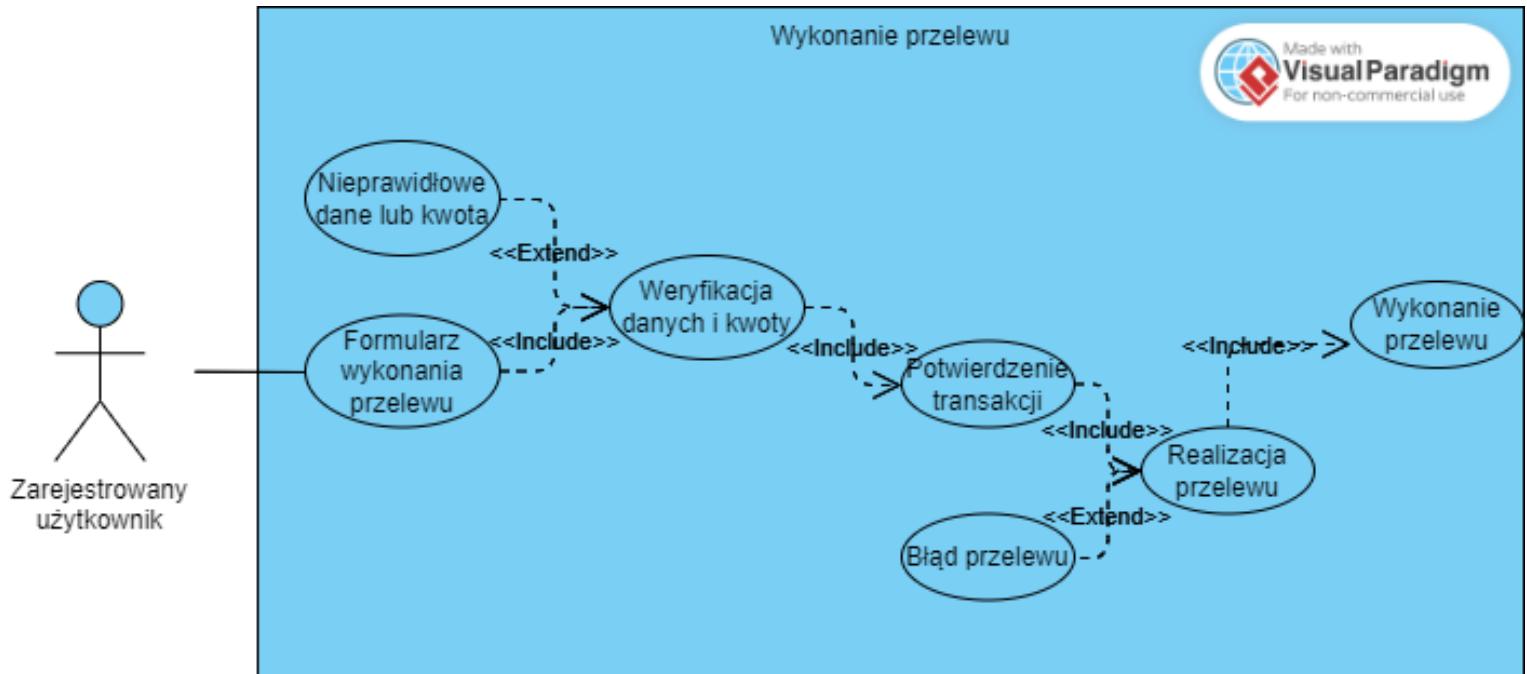
1. Rejestracja użytkownika



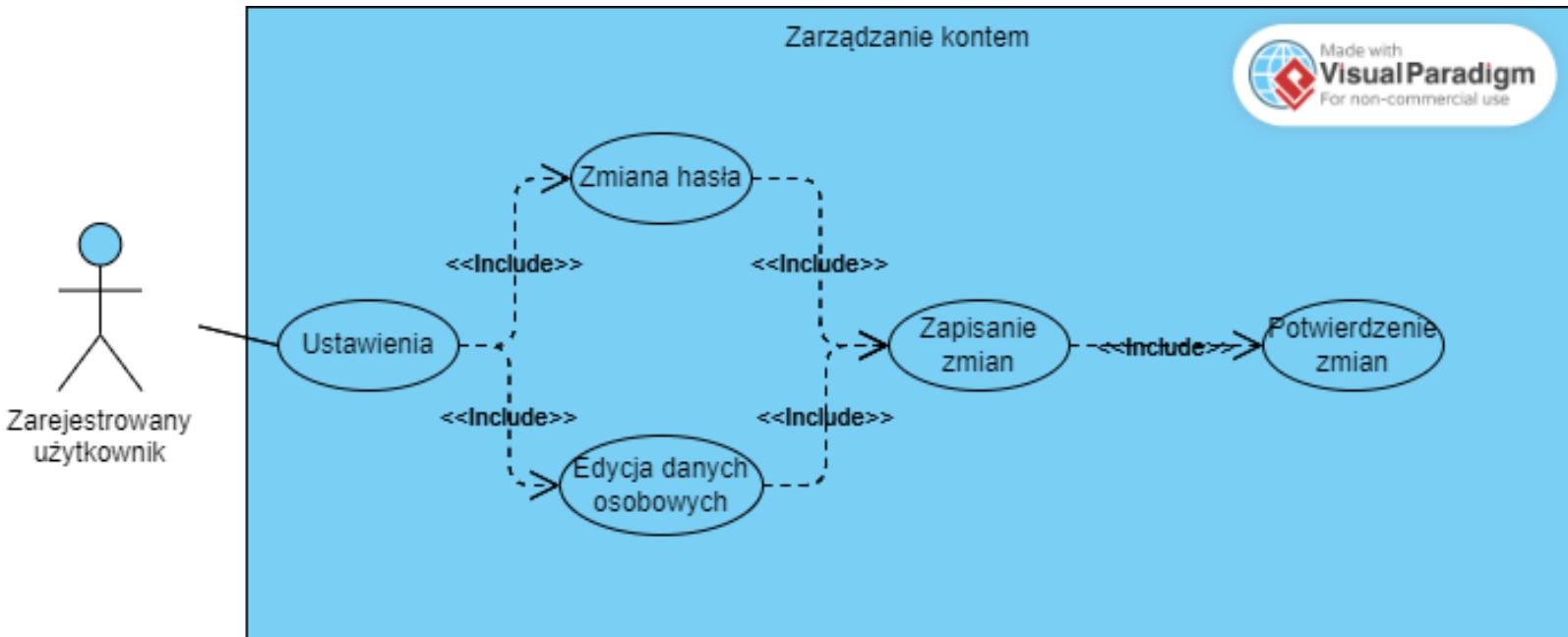
2. Logowanie



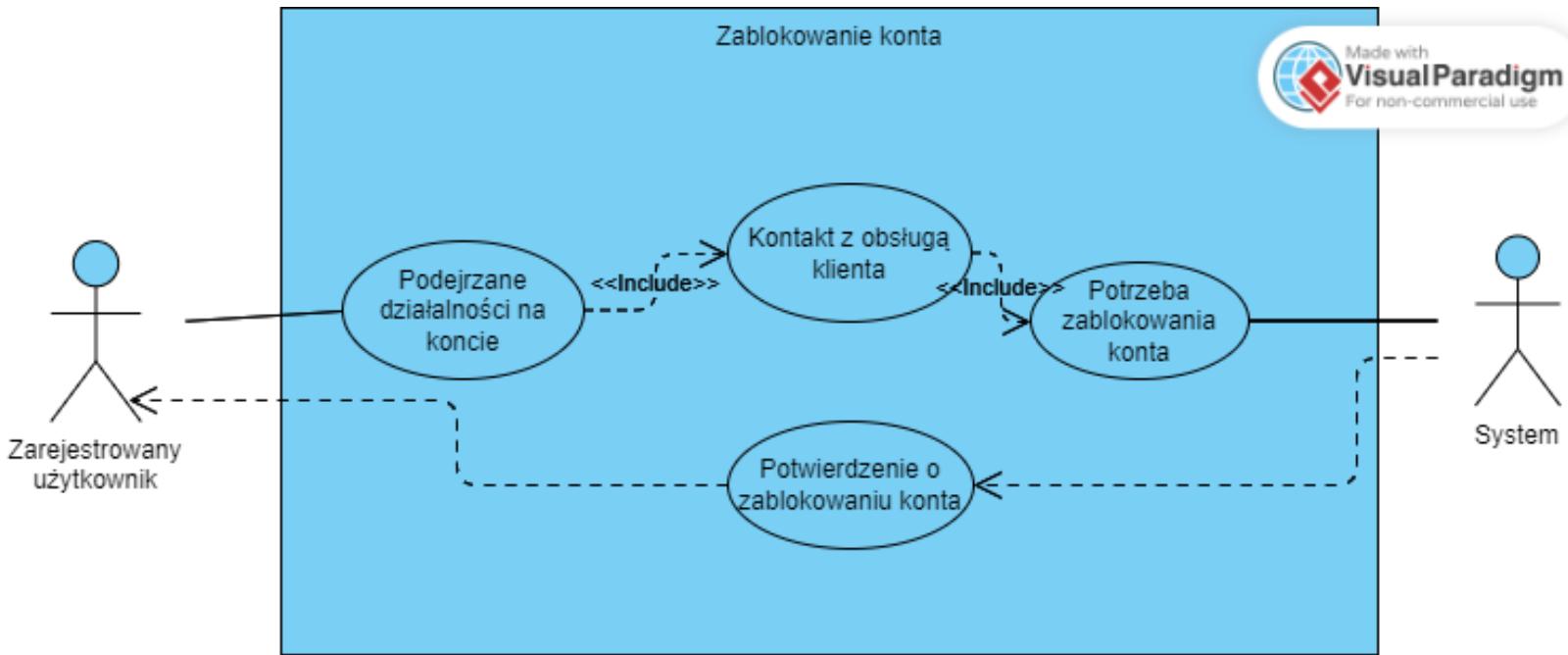
3. Wykonanie przelewu



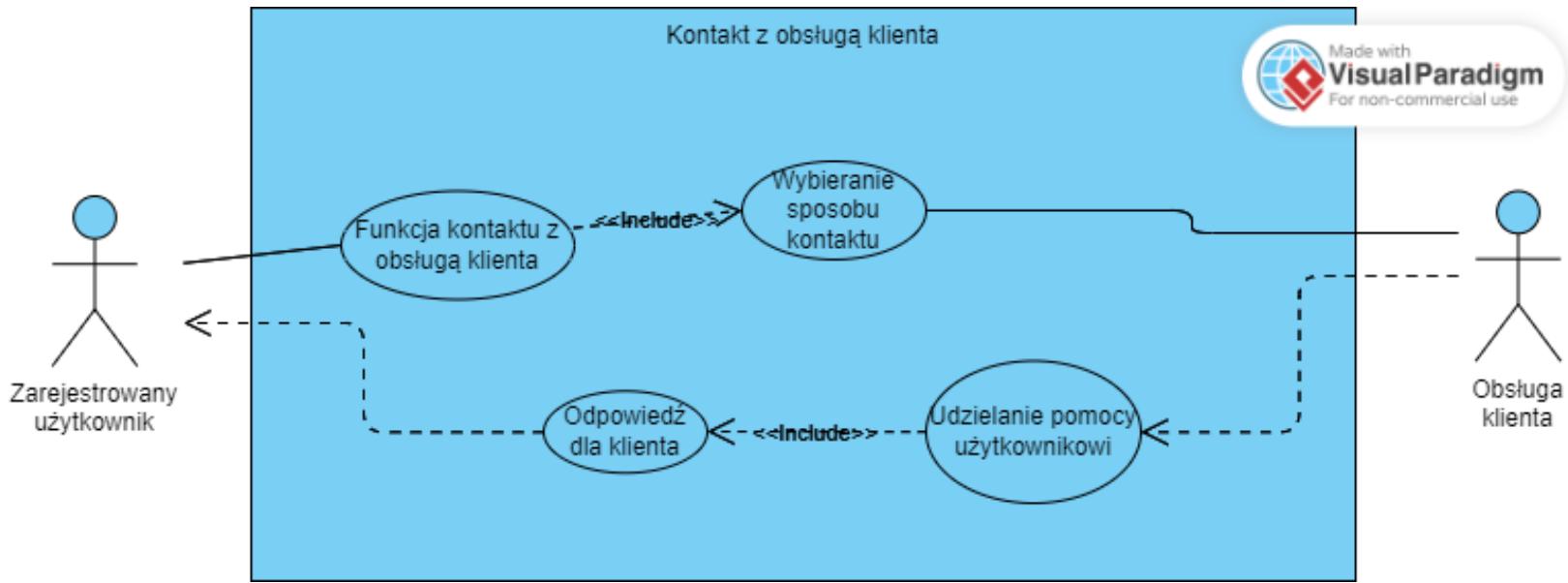
4. Zarządzanie kontem



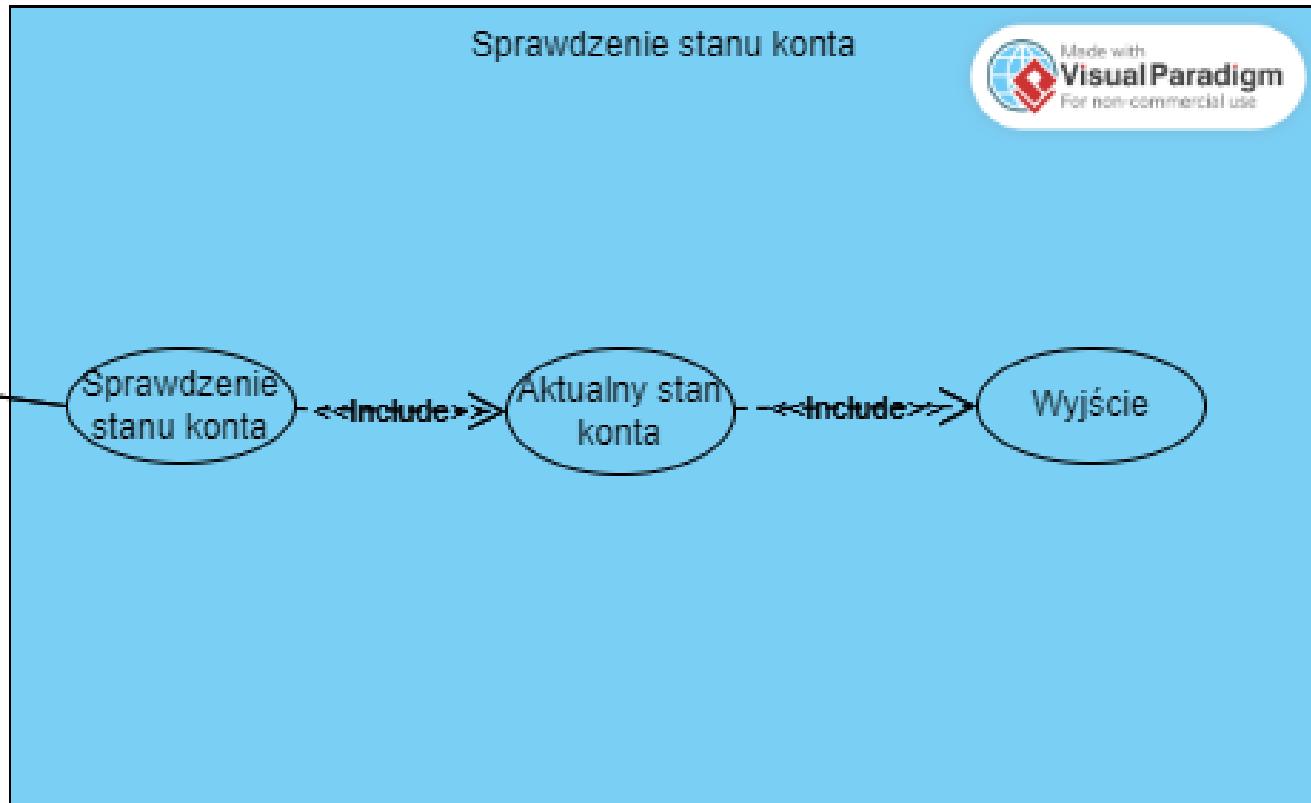
5. Zablokowanie konta



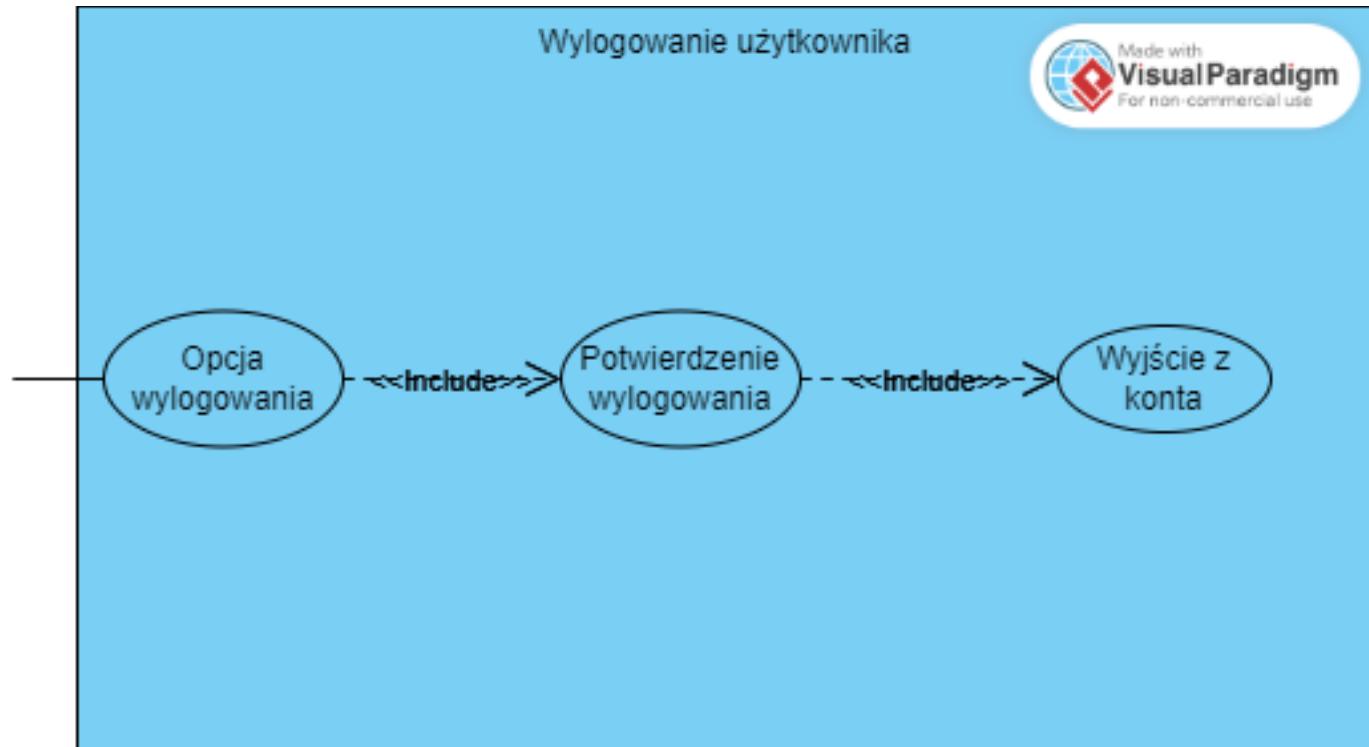
6. Kontakt z obsługą klienta



7. Sprawdzenie stanu konta



8. Wylogowanie użytkownika



5.2.3 Szczegółowy opis wymagań

- Rejestracja użytkownika
 - Uzasadnienie biznesowe: niezbędne działanie umożliwiające założenie konta i weryfikacja tożsamości użytkownika.
 - Warunki początkowe: nowy, niezarejestrowany użytkownik.
 - Przebieg działań:
 2. Użytkownik: podaje swoje dane osobowe niezbędne do rejestracji.
 3. System: sprawdza poprawność wprowadzonych danych osobowych.
 4. System: w przypadku niepoprawnych danych prosi użytkownika o ich ponowne wprowadzenie, w przypadku poprawnych danych przechodzi do następnego kroku.
 5. System: prosi użytkownika o podanie adresu e-mail oraz hasła, które będą powiązane z kontem.
 6. Użytkownik: wprowadza swój adres e-mail oraz tworzy hasło dla swojego konta.
 7. System: wysyła link aktywacyjny na adres e-mail podany przez użytkownika i czeka na jego użycie.
 8. Użytkownik: wchodzi w wysłany na jego adres e-mail link.
 9. System: aktywuje konto utworzone przez użytkownika i wysyła e-mail mówiący o udanej rejestracji.
 - Efekty: zarejestrowanie nowego użytkownika w systemie.
 - Wymagania niefunkcjonalne: system powinien skutecznie szyfrować wrażliwe dane rejestrującego się użytkownika, takie jak numer PESEL czy hasło.
 - Częstotliwość: 1
 - Istotność: 5
-
- Logowanie użytkownika
 - Uzasadnienie biznesowe: udzielenie użytkownikowi dostępu do konta i działań z nim związanych za pomocą weryfikacji danych logowania
 - Warunki początkowe: Zarejestrowany użytkownik

- Przebieg działań:
 - Użytkownik: wprowadza swoje dane (e-mail i hasło) w panelu logowania.
3. System: weryfikuje poprawność wprowadzonych danych.
4. System: w przypadku nieprawidłowych danych (nieistniejący w bazie e-mail lub błędne hasło) wyświetla komunikat o niepoprawności danych i wymaga od użytkownika ich ponownego wprowadzenia, w przypadku prawidłowych danych przechodzi do następnego kroku.
5. System: udziela użytkownikowi dostępu do konta.
6. System: powiadamia użytkownika o udanym logowaniu i przekierowuje go do panelu głównego
- Efekty: udzielenie użytkownikowi dostępu do konta.
 - Wymagania niefunkcjonalne: system powinien w szybkim tempie reagować na działania użytkownika i sprawdzać poprawność wprowadzonych przez niego danych.
 - Częstotliwość: 5
 - Istotność: 5
-
- Wykonanie przelewu
 - Uzasadnienie biznesowe: możliwość transferu środków pieniężnych pomiędzy kontami, dokonywania płatności za pomocą przelewów.
 - Warunki początkowe: istnienie dwóch kont bankowych, pomiędzy którymi ma odbyć się transakcja.
 - Przebieg działań:
 - Użytkownik: wprowadza dane konta, na które chce wykonać przelew.
 - System: sprawdza istnienie tego konta w bazie danych. W przypadku niepoprawnych danych wyświetla odpowiedni komunikat.
 - Użytkownik: podaje kwotę przelewu.
 - System: sprawdza, czy użytkownik dysponuje kwotą, którą zamierza przesłać na inne konto, w przypadku zbyt niskiego stanu konta do wykonania operacji system wyświetla odpowiedni komunikat.
 - Użytkownik: wybiera typ przelewu: tradycyjny lub natychmiastowy.

- System: w przypadku wybrania przez użytkownika opcji przelewu natychmiastowego dolicza jej koszt do kosztu całej operacji.
4. Użytkownik: potwierdza chęć wykonania przelewu.
 5. System: obciąża saldo nadawcy kwotą wykonanej operacji.
 6. System: wysyła potwierdzenie transakcji do nadawcy.
 7. System: zwiększa stan konta odbiorcy o równowartość kwoty przelewu.
 8. System: wysyła odbiorcy potwierdzenie o wykonanym przelewie.
- Efekty: transfer pieniędzy pomiędzy dwoma kontami.
 - Wymagania niefunkcjonalne: ochrona danych transakcyjnych za pomocą technologii blockchain.
 - Częstotliwość: 3
 - Istotność: 4
- Zarządzanie kontem
 - Uzasadnienie biznesowe: możliwość modyfikacji niektórych danych użytkownika w przypadku zmiany jego rzeczywistych danych osobowych lub chęci zmiany hasła.

Scenariusz: edycja danych osobowych

- Warunki początkowe: zarejestrowany i zalogowany użytkownik
- Przebieg działań:
- Użytkownik: otwiera panel zarządzania kontem i wybiera opcję „edytuj dane osobowe”.
- Użytkownik: modyfikuje dane spośród tych, które są możliwe do edycji (np. adres zamieszkania, numer telefonu).
- Użytkownik: zatwierdza zmiany.
- System: wykonuje weryfikację dwuetapową, żądając od użytkownika hasło do konta oraz kod wysłany na adres e-mail.
- Użytkownik: wykonuje weryfikację.
- System: w przypadku udanej weryfikacji zmienia dane na prośbę użytkownika, zapisuje zmiany i wysyła odpowiedni komunikat.

- Efekty: zaktualizowane dane użytkownika
- Wymagania niefunkcjonalne: system powinien sprawdzać poprawność nowo wprowadzonych danych tzn. możliwość ich istnienia.
- Częstotliwość: 2
- Istotność: 3

Scenariusz: zmiana hasła do konta

- Warunki początkowe: zarejestrowany i zalogowany użytkownik
- Przebieg działań:
 5. Użytkownik: otwiera panel zarządzania kontem i wybiera opcję „zmień hasło”.
 6. Użytkownik: wprowadza swoje obecne hasło.
 7. System: sprawdza zgodność wprowadzonego hasła, w przypadku niepoprawnych danych wyświetla odpowiedni komunikat, w przypadku poprawnych danych przekierowuje użytkownika do następnego kroku.
 8. Użytkownik: wprowadza nowe hasło.
 9. System: sprawdza poprawność nowego hasła, tj. odpowiednia ilość znaków, występowanie dużych liter i cyfr w haśle.
 10. System: w przypadku nie spełnienia powyższych kryteriów przez użytkownika prosi go o podanie innego hasła. W momencie odczytania poprawnej kombinacji znaków system prosi użytkownika o ponowne podanie tego samego hasła w celu jego potwierdzenia.
 11. System: zatwierdza nowe hasło i zmienia je w bazie danych. Wyświetla komunikat o pomyślnej operacji zmiany hasła.
- Efekty: zaktualizowane hasło do konta użytkownika.
- Wymagania niefunkcjonalne: system powinien szybko szyfrować i deszyfrować dane podczas sprawdzania zgodności haseł.
- Częstotliwość: 2
- Istotność: 3
- Zablokowanie konta

- Uzasadnienie biznesowe: możliwość zablokowania konta bankowego z poziomu aplikacji w przypadku kradzieży karty kredytowej, potencjalnego cyberataku bądź logowania na innym, nieznanym urządzeniu.
 - Warunki początkowe: zarejestrowany i zalogowany użytkownik
 - Przebieg działań:
 6. Użytkownik: w panelu ustawień wybiera funkcję „zablokuj konto”.
 7. System: wysyła zapytanie o potwierdzenie decyzji użytkownika.
 8. Użytkownik: potwierdza.
 9. System: wykonuje weryfikację dwuetapową, prosząc użytkownika o podanie hasła do aplikacji oraz kodu wysłanego na potwierdzony numer telefonu.
 10. System: w przypadku przejścia weryfikacji kontaktuje się z centralą banku w celu tymczasowego zablokowania konta bankowego.
 11. Centrala banku: blokuje konto bankowe użytkownika, którego dane osobowe są zgodne z zapisanymi na koncie w aplikacji. Wysyła informację zwrotną do systemu.
 12. System: wysyła informację do użytkownika o zablokowaniu jego konta.
 - Efekty: konto bankowe użytkownika zostaje zablokowane za pomocą aplikacji.
 - Wymagania niefunkcjonalne: system powinien zapewniać ekspresowy kontakt z centralą banku i szybko reagować na prośby klientów w sprawie blokady kont.
 - Częstotliwość: 1
 - Istotność: 5
-
- Kontakt z obsługą klienta
 - Uzasadnienie biznesowe: pomoc użytkownikowi w obsłudze funkcji aplikacji i przyjmowanie uwag związanych z działaniem aplikacji.
 - Warunki początkowe: zarejestrowany i zalogowany użytkownik
 - Przebieg działań:
 - Użytkownik: wybiera w aplikacji funkcję kontaktu z obsługą klienta.
 - System: wyświetla sposoby kontaktu z obsługą klienta: przez czat online, formularz kontaktowy oraz telefon.

- Użytkownik: wybiera preferowaną opcję kontaktu.
 - System: przenosi użytkownika do odpowiedniego panelu:
 - Numer telefonu: wyświetla numer telefonu do obsługi klienta.
 - Czat online: otwiera okno czatu pozwalające na napisanie wiadomości. Jest ona połączona bezpośrednio z kontem w aplikacji.
 - Formularz kontaktowy: umożliwia napisanie wiadomości przy poaniu adresu e-mail. Odpowiedź od obsługi klienta powinna ukazać się wtedy w formie e-maila.
1. Użytkownik: wysyła wiadomość bądź dzwoni do obsługi klienta.
 2. Obsługa klienta: zostawia informację zwrotną do wiadomości użytkownika.
 1. Efekty: bezpośredni pomoc dla klienta dotycząca aplikacji
 2. Wymagania niefunkcjonalne: system powinien radzić sobie z dużymi obciążeniami na linii telefonicznej, jak i również zapewniać szybką odpowiedź na wiadomości użytkowników.
 3. Częstotliwość: 3
 4. Istotność: 4

5.3 *Wymagania niefunkcjonalne*

2. Wydajność
 2. System powinien obsługiwać duże obciążenia, zwłaszcza w godzinach szczytu, zapewniając szybką odpowiedź na żądania użytkowników.
 1. System powinien obsługiwać duże obciążenia, zwłaszcza w godzinach szczytu, zapewniając szybką odpowiedź na żądania użytkowników.
-
2. Bezpieczeństwo
 3. Zapewnienie pełnej ochrony danych użytkowników przed nieautoryzowanym dostępem, utratą lub zniszczeniem.
 3. Wdrożenie środków zapobiegawczych, takich jak autoryzacja wielopoziomowa, szyfrowanie danych, monitorowanie aktywności.
 1. System powinien być odporny na ataki typu phishing, malware oraz inżynierię społeczną.

2. Zabezpieczenia
 3. System powinien posiadać mechanizmy zapewniające integralność danych oraz możliwość szybkiego przywrócenia usług w przypadku awarii.
 4. System powinien posiadać mechanizmy zapewniające integralność danych oraz możliwość szybkiego przywrócenia usług w przypadku awarii.
 1. Wdrożenie procedur zarządzania incydentami, które umożliwiają szybką reakcję na wszelkie próby naruszenia bezpieczeństwa.
2. Inne cechy jakości
 3. Adaptowalność: System powinien być łatwo rozbudowywalny i modyfikowalny zgodnie z wymaganiami rynkowymi oraz zmieniającymi się przepisami.
 4. Dostępność: Zapewnienie, że usługi bankowe są dostępne dla klientów przez większość czasu, minimalizując ewentualne przerwy.
 5. Poprawność: Zapewnienie, że system działa zgodnie z oczekiwaniemi, minimalizując błędy i błędne działania.
 6. Elastyczność: Możliwość dostosowania systemu do różnych rodzajów klientów i ich potrzeb.
 7. Łatwość konserwacji: System powinien być łatwy w utrzymaniu, umożliwiając szybką identyfikację i naprawę błędów.
 8. Przenośność: System powinien być w miarę możliwości niezależny od platformy, umożliwiając jego łatwe przenoszenie na inne środowiska.
 9. Awaryjność: System powinien być przygotowany na nagłe awarie sprzętowe lub programowe, minimalizując negatywne skutki dla użytkowników.
 8. Testowalność: Zapewnienie możliwości skutecznego testowania systemu w celu zapewnienia jego jakości i niezawodności.
 9. Użyteczność: System powinien być intuicyjny i łatwy w obsłudze dla użytkowników końcowych.

6 Zarządzanie projektem

6.1 Zasoby ludzkie

(rzeczywiste lub hipotetyczne) – przy realizacji projektu

Należy założyć, że projekt byłby realizowany w całości jako projekt komercyjny a nie tylko częściowo w ramach zajęć na uczelni

6.2 Harmonogram prac

Etapy mogą się składać z zadań.

Wskazać czasy trwania poszczególnych etapów i zadań – wykres Gantta.

obejmuje również harmonogram wdrożenia projektu – np. szkolenie, rozruch, konfiguracja, serwis – może obejmować różne wydania (tj. o różnej funkcjonalności – personal, professional, enterprise) i wersje (1.0, 1.5, itd.)

6.3 Etapy/kamienie milowe projektu

dla głównych etapów projektu

W grupach prowadzonych przez ŁR ten cały ten rozdział jest opcjonalny – dla chętnych. Nie jest omawiany na wykładzie!

Studenci powinni skonsultować szczegółowe wymagania w tym zakresie z nauczycielem prowadzącym zajęcia w danej grupie.

7 Zarządzanie ryzykiem

7.1 Lista czynników ryzyka

Wypełniona lista kontrolna

7.2 Ocena ryzyka

prawdopodobieństwo i wpływ

7.3 Plan reakcji na ryzyko

Działania w odniesieniu do poszczególnych ryzyk.

Mogą być wg różnych strategii, tj. kilka strategii dla pojedynczego czynnika ryzyka

Rozdział obowiązkowy w zespołach co najmniej 3-osobowych, w mniejszych – do uzgodnienia z prowadzącym zajęcia.

8 Zarządzanie jakością

8.1 Scenariusze i przypadki testowe

głównie testowanie funkcjonalności, ale może być też testowanie wymagań niefunkcjonalnych/zgodności; każdy scenariusz od nowej strony, musi zawierać co najmniej następujące informacje (sugerowany układ tabelaryczny, np. wg szablonu podanego w osobnym pliku lub na wykładzie):

- numer – jako ID
- nazwa scenariusza – co test w nim testowane (max kilka wyrazów)
- kategoria – poziom/kategoria testów
- opis – dodatkowe opcjonalne informacje, które nie zmieściły się w nazwie
- tester - konkretna osoba lub klient/pracownik,
- termin – kiedy testowanie ma być przeprowadzane,
- narzędzia wspomagające – jeśli jakieś są używane przy danym scenariuszu
- przebieg działań – tabela z trzema kolumnami: lp. oraz opisującymi działania testera i systemu
- założenia, środowisko, warunki wstępne, dane wejściowe – przygotowanie przed uruchomieniem testów
- zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań
- *przebieg lub zestaw danych testowych musi zawierać jawną informację o warunku zaliczenia testu*

9 Projekt techniczny

9.1 Opis architektury systemu

Architektura systemu aplikacji bankowej "Parabank" jest zaprojektowana w sposób kompleksowy, mający na celu zapewnienie bezpieczeństwa, niezawodności oraz wydajności. Oto opis poszczególnych warstw architektury:

1.Warstwa Interfejsu Użytkownika:

To interfejs, który widzą użytkownicy korzystający z aplikacji "Parabank". Może to być aplikacja mobilna lub strona internetowa. Interfejs ten jest zaprojektowany w sposób intuicyjny i responsywny, umożliwia użytkownikom wygodne korzystanie z różnych usług bankowych, takich jak przelew, sprawdzanie stanu konta czy aplikowanie o kredyt.

2.Warstwa Aplikacji Biznesowej:

Zawiera logikę biznesową aplikacji. Odpowiada za przetwarzanie żądań użytkowników, wykonywanie operacji na kontach, zarządzanie transakcjami, autoryzację użytkowników i wiele innych. Aplikacja biznesowa również zajmuje się interakcją z bazą danych.

3.Warstwa Danych:

To baza danych przechowująca wszystkie informacje o klientach, ich kontach, transakcjach, historii operacji i innymi danymi niezbędnymi do funkcjonowania banku. Baza danych musi być skalowalna, aby obsłużyć duże ilości danych i zapewnić szybki dostęp do informacji.

4.Warstwa Integracji:

Zajmuje się integracją systemu aplikacji bankowej z innymi systemami zewnętrznymi, takimi jak systemy płatności, systemy zarządzania ryzykiem, czy systemy zewnętrznych dostawców usług. Zapewnia to interoperacyjność i umożliwia współpracę z różnymi partnerami biznesowymi.

5.Warstwa Bezpieczeństwa:

Bezpieczeństwo jest kluczowym elementem architektury systemu bankowego "Parabank". Ta warstwa obejmuje zabezpieczenia takie jak uwierzytelnianie dwuetapowe, szyfrowanie danych, monitorowanie aktywności użytkowników, zarządzanie uprawnieniami oraz systemy detekcji i reakcji na incydenty bezpieczeństwa.

6.Warstwa Zarządzania:

Obejmuje narzędzia i mechanizmy zarządzania systemem aplikacji bankowej "Parabank". Zawiera systemy monitorowania wydajności, zarządzania konfiguracją, logów aplikacji oraz raportowania. Zapewnia to administratorom banku możliwość efektywnego zarządzania i utrzymania całego systemu.

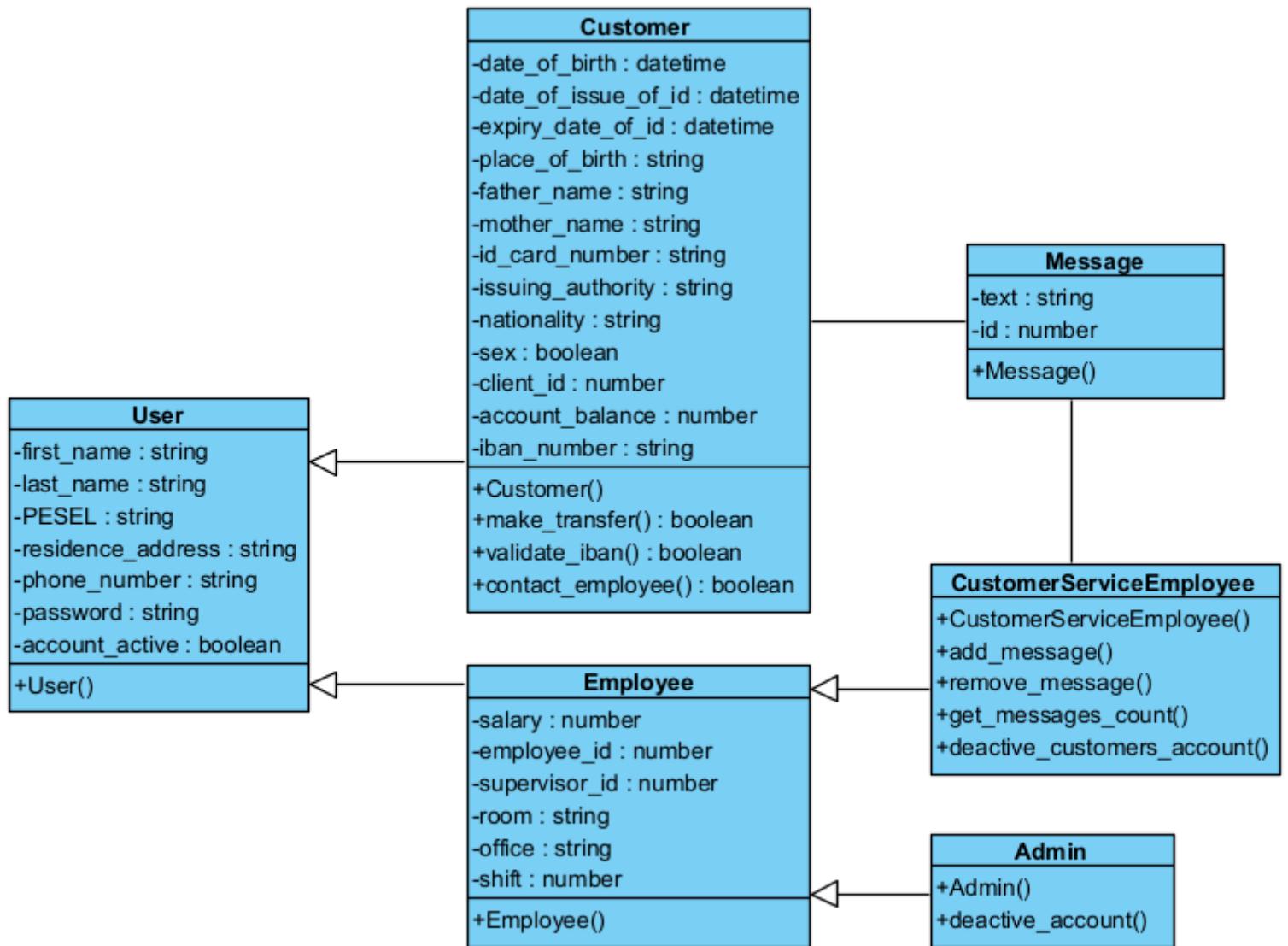
Architektura systemu aplikacji bankowej "Parabank" jest skalowalna, elastyczna i zaprojektowana z myślą o spełnianiu wysokich standardów bezpieczeństwa i wydajności, aby zapewnić klientom płynne i bezpieczne korzystanie z usług bankowych.

9.2 *Technologie implementacji systemu*

Technologia	Uzasadnienie
Python	Jako główny język programowania, Python jest wyborem odpowiednim ze względu na jego czytelność, prostotę i wszechstronność. Python oferuje bogaty zestaw bibliotek do obsługi operacji na danych, sieci, bezpieczeństwa i wielu innych aspektów potrzebnych do budowy systemu bankowego.
SQLAlchemy	SQLAlchemy jest biblioteką ORM (Object-Relational Mapping) w Pythonie, która ułatwia interakcję z bazami danych. Pozwala ona na tworzenie, zarządzanie i zapytywanie baz danych przy użyciu obiektów Pythona, co upraszcza pracę z danymi finansowymi klientów, transakcjami i innymi aspektami systemu bankowego.
NumPy	NumPy jest biblioteką do obliczeń numerycznych w Pythonie. Może być używana do wykonywania zaawansowanych operacji matematycznych związanych z finansami, takich jak obliczenia związane z odsetkami, analiza ryzyka itp.
JSON	Format JSON jest powszechnie używany do przesyłania danych między aplikacjami. Może być stosowany do komunikacji między różnymi modułami w systemie bankowym.
Git	Git jest niezbędny do skutecznego zarządzania kodem źródłowym systemu bankowego. Umożliwia śledzenie zmian, tworzenie gałęzi rozwoju i współpracę w zespole programistycznym.
hashlib	Moduł hashlib umożliwia obliczanie funkcji skrótu (hash) dla danych, co jest istotne do zabezpieczania haseł użytkowników, integralności danych i autoryzacji transakcji w systemie bankowym.

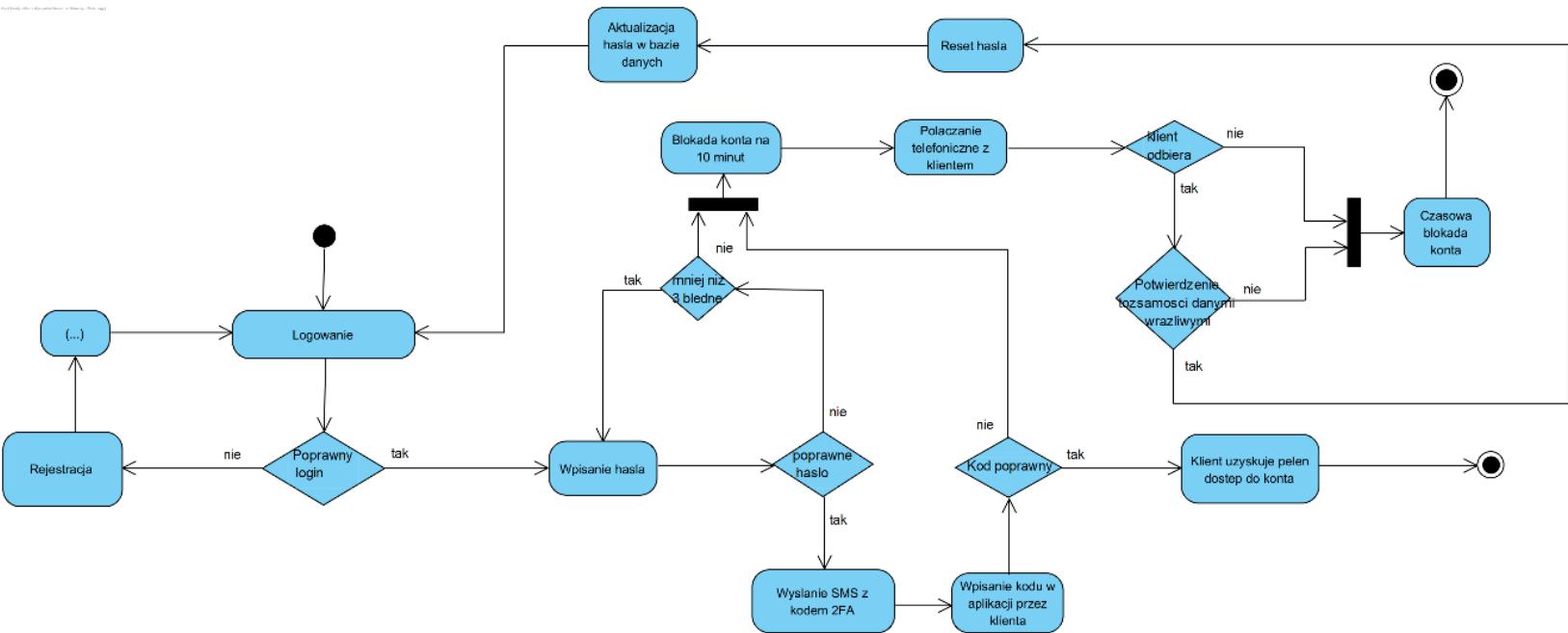
9.3 Diagramy UML

9.3.1 Diagram(-y) klas

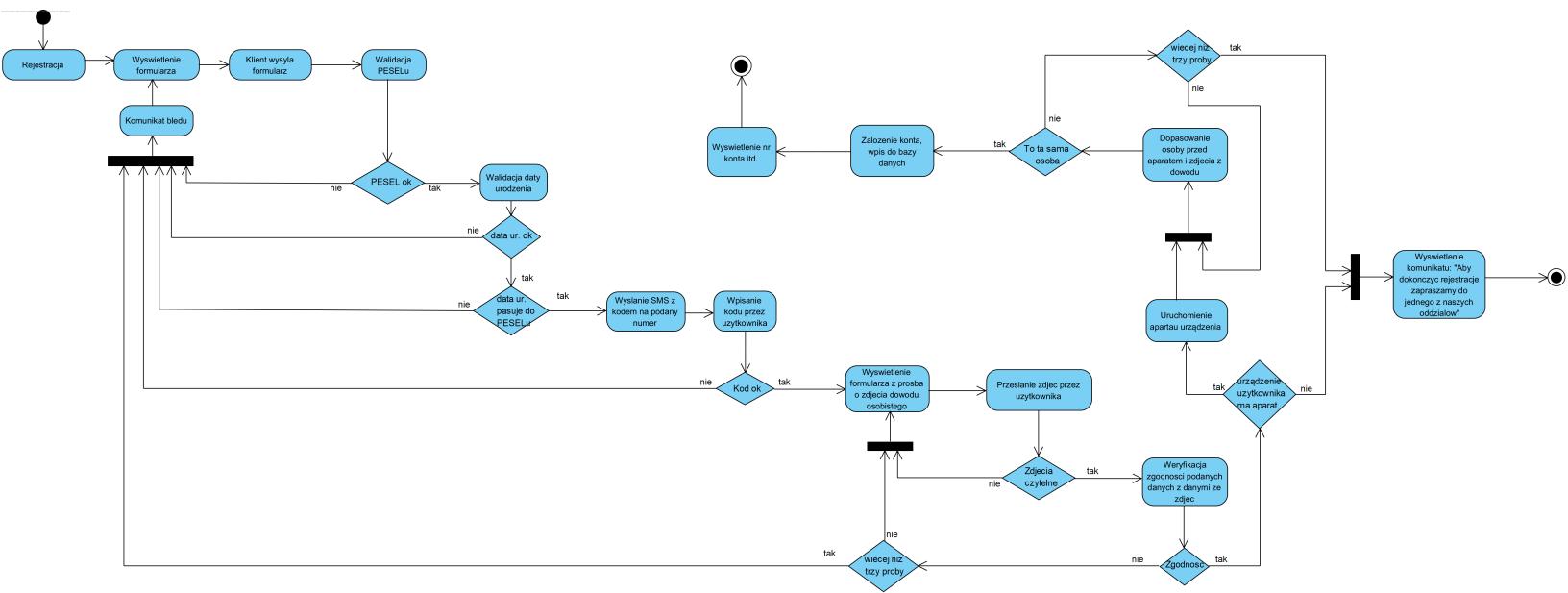


9.3.2 Diagram(-y) czynności

1. Logowanie

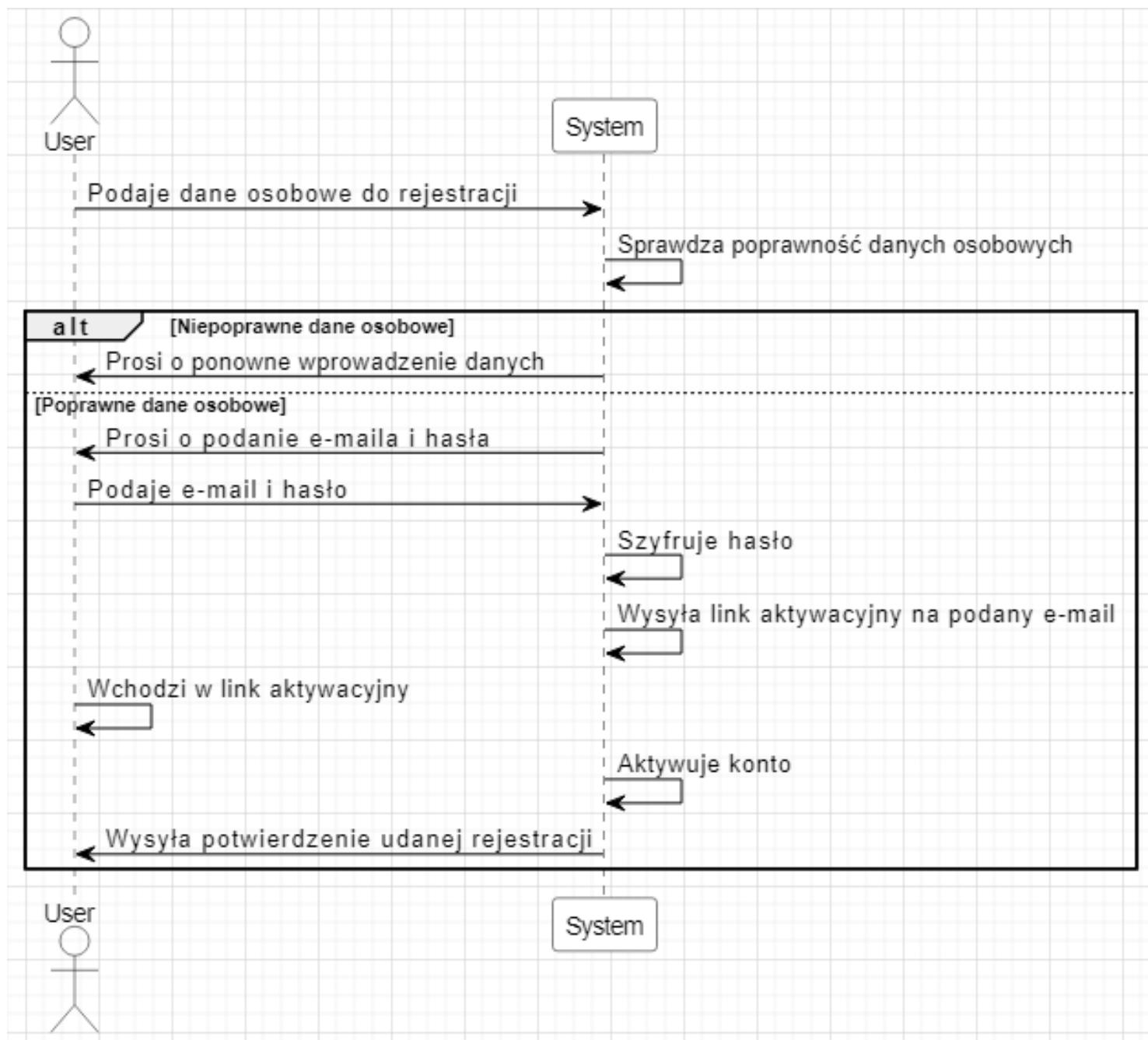


2. Rejestracja

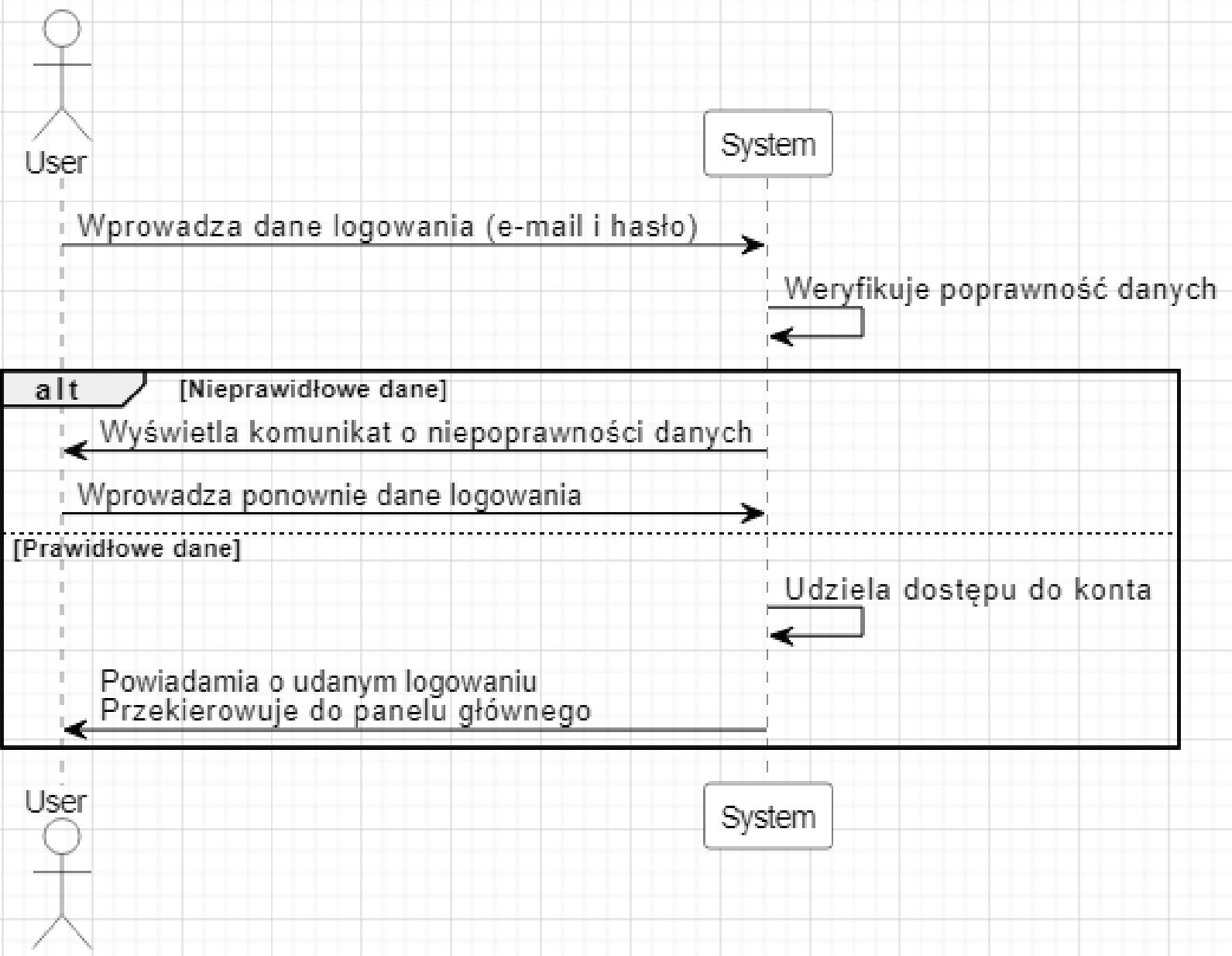


9.3.3 Diagramy sekwencji

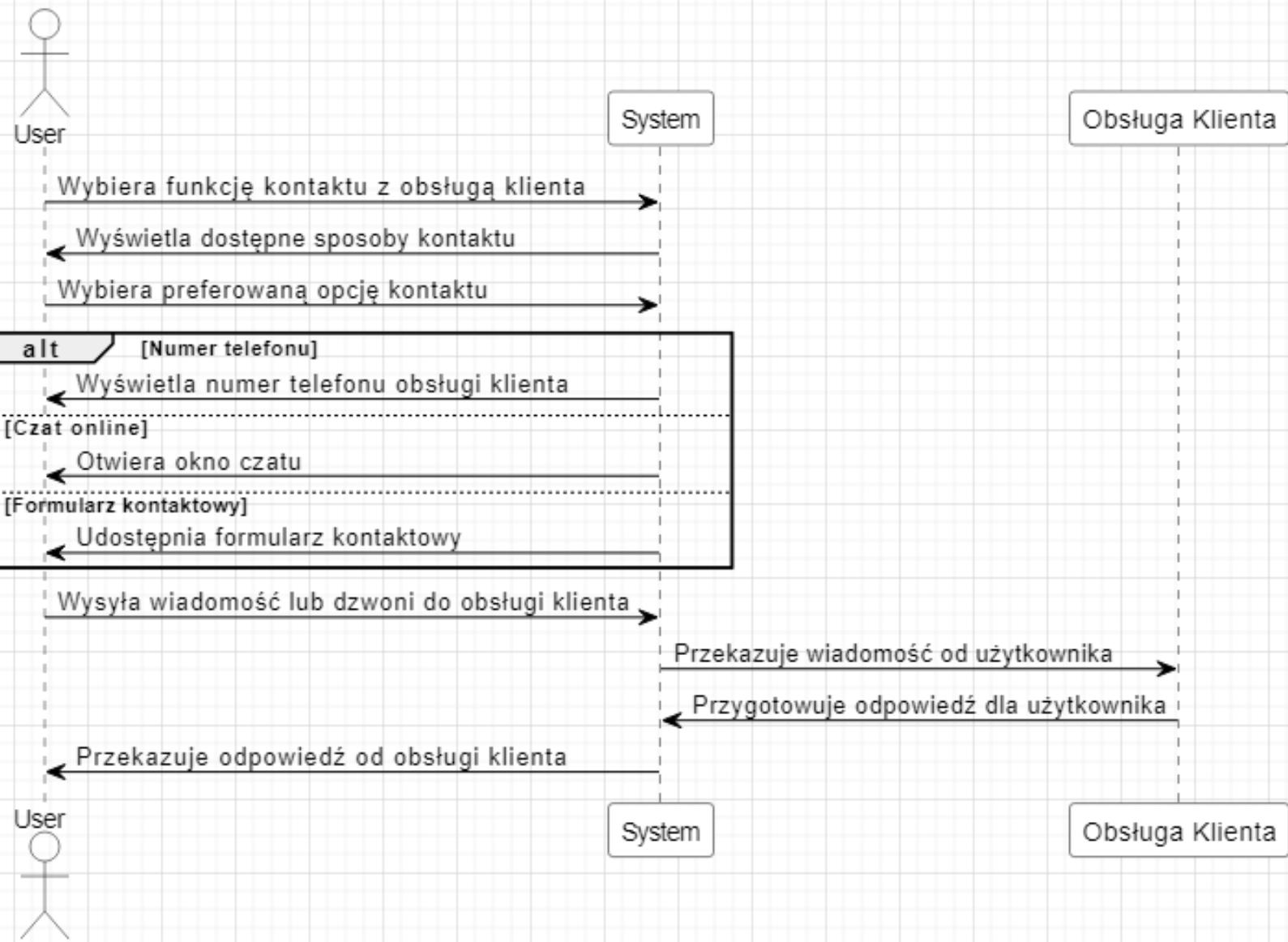
1. Rejestracja



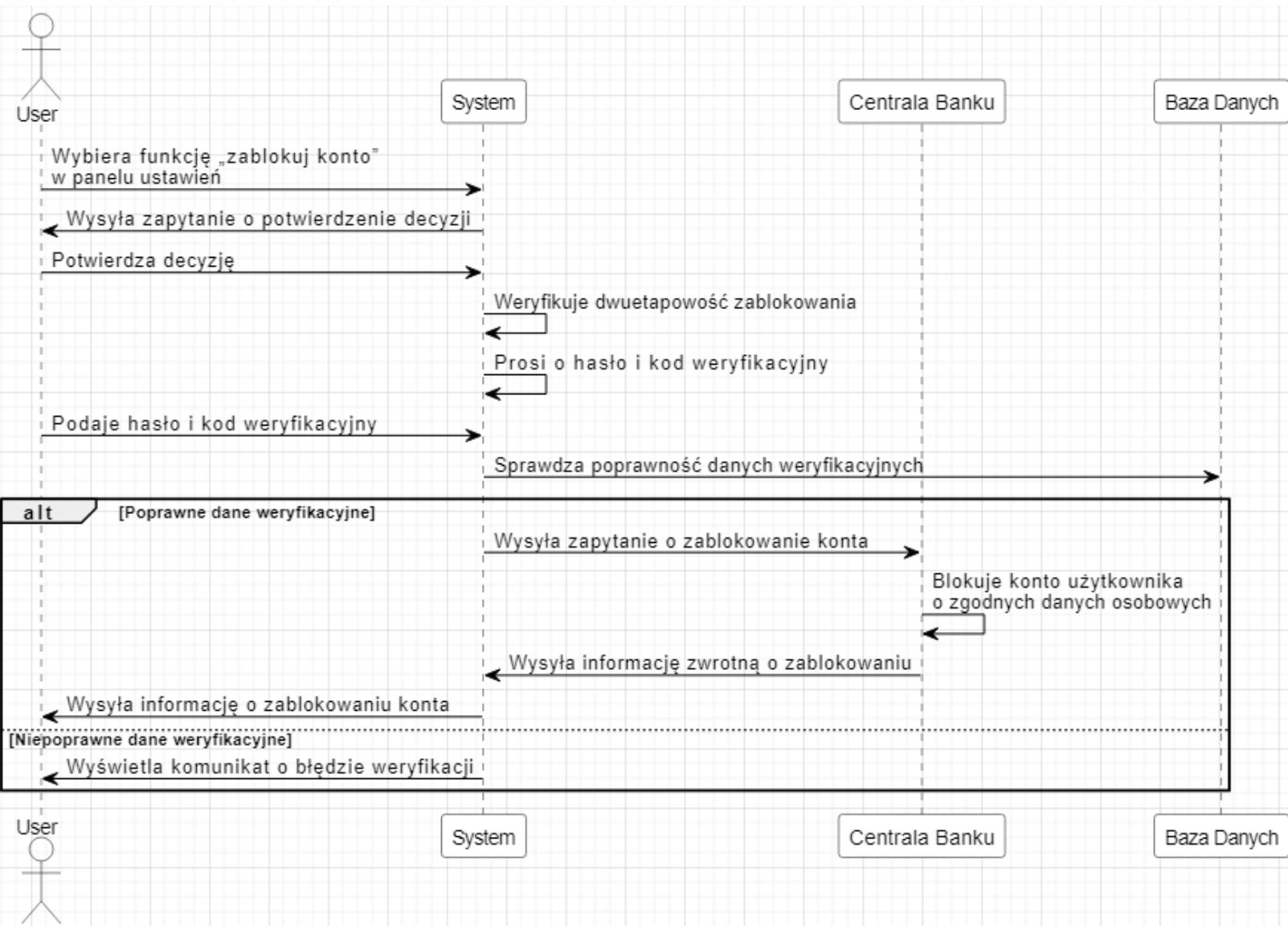
2. Logowanie



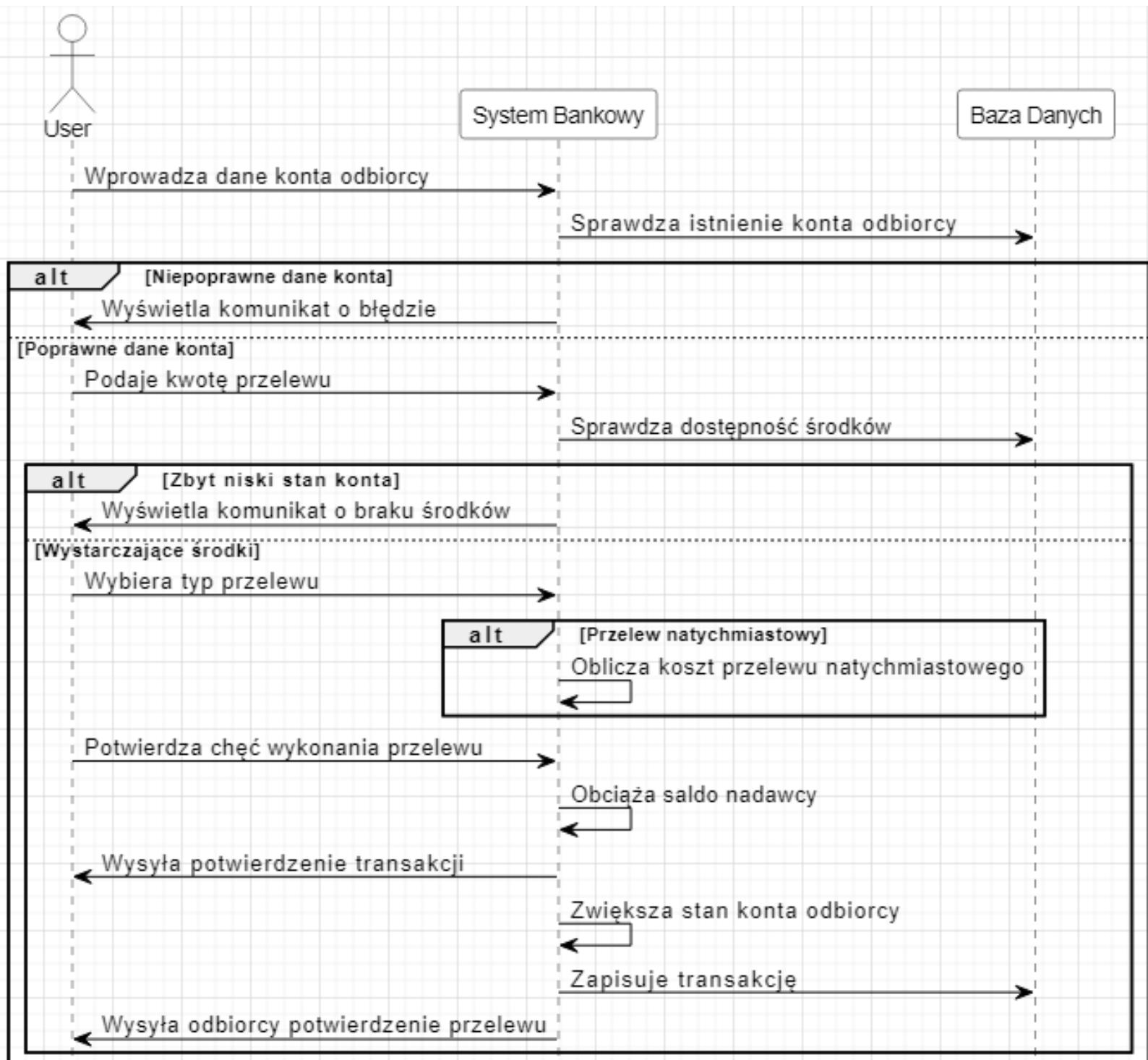
3. Kontakt z obsługą klienta



4. Blokada Konta



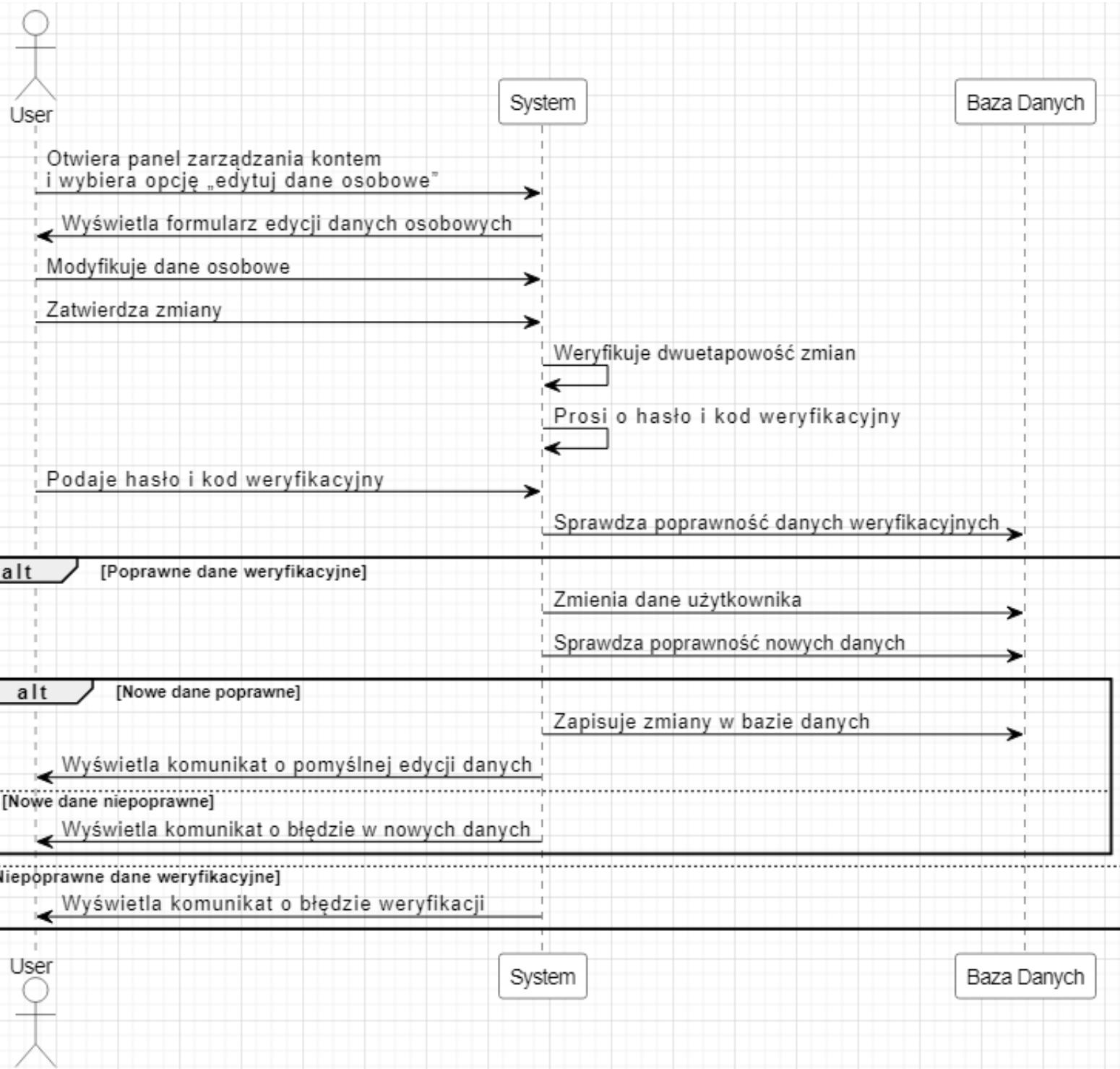
5. Wykonanie przelewu



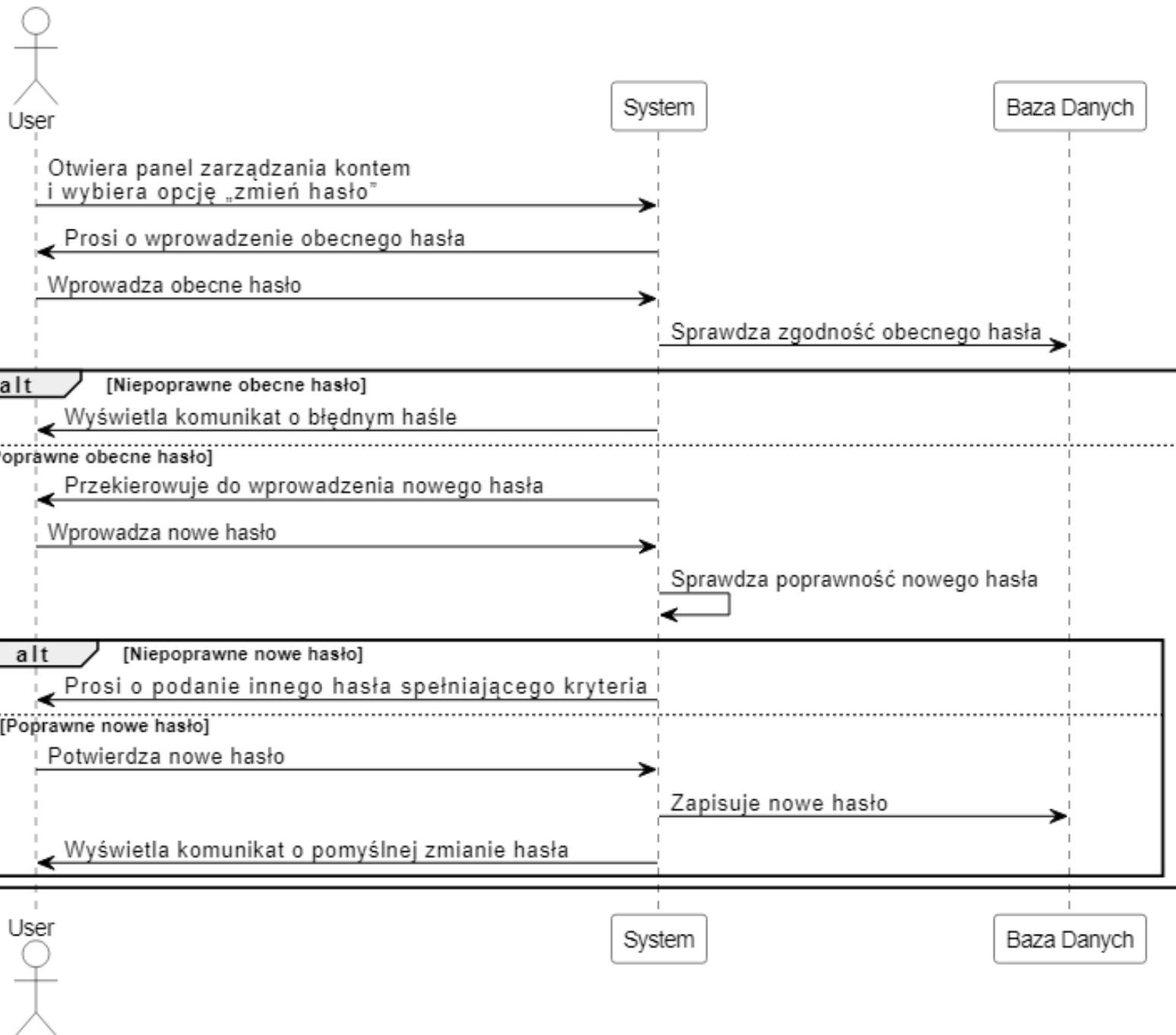
System Bankowy

Baza Danych

6. Zmiana Danych (Scenariusz: edycja danych osobowych)

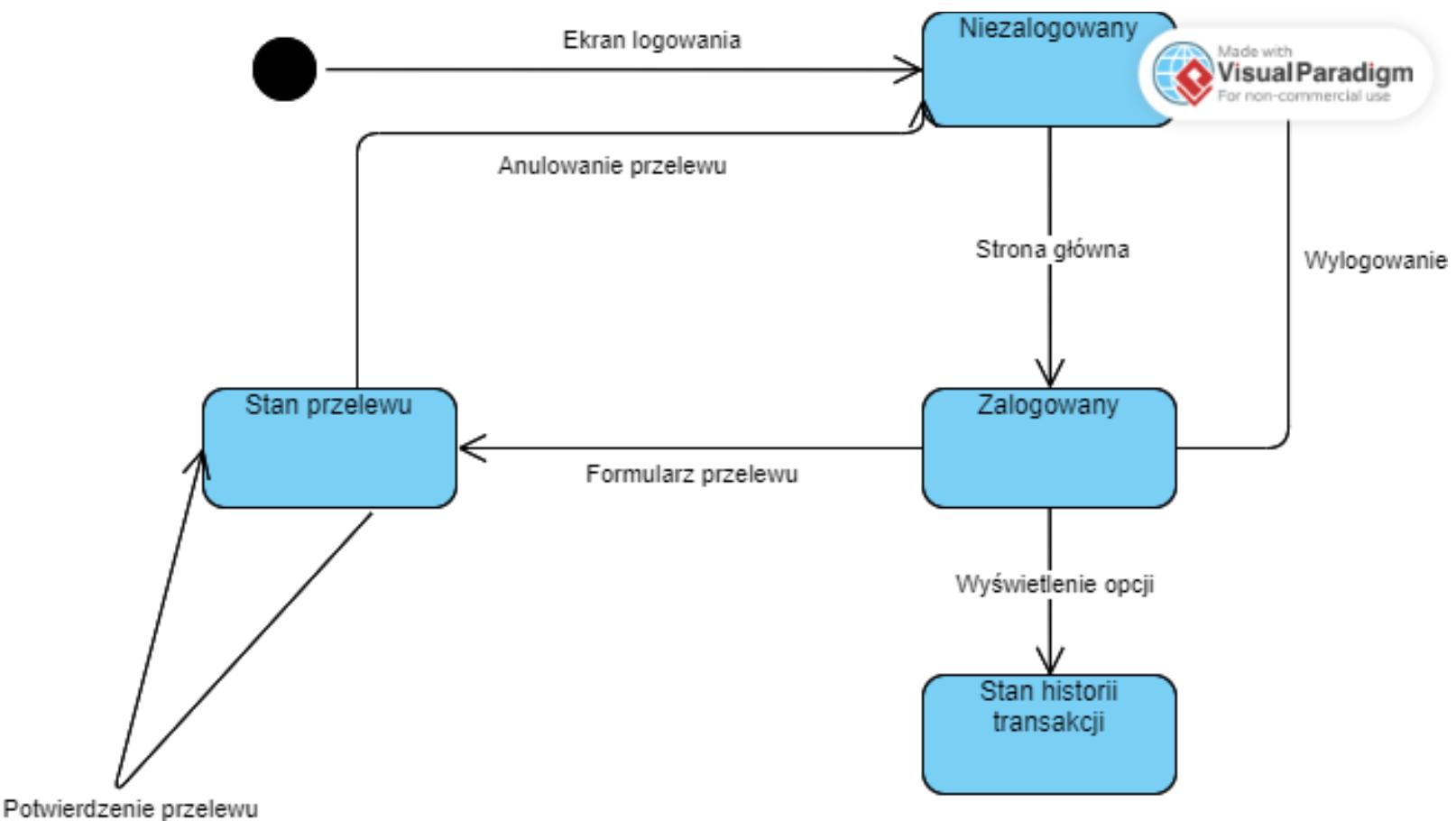


7. Zmiana Danych (Scenariusz: edycja hasła)

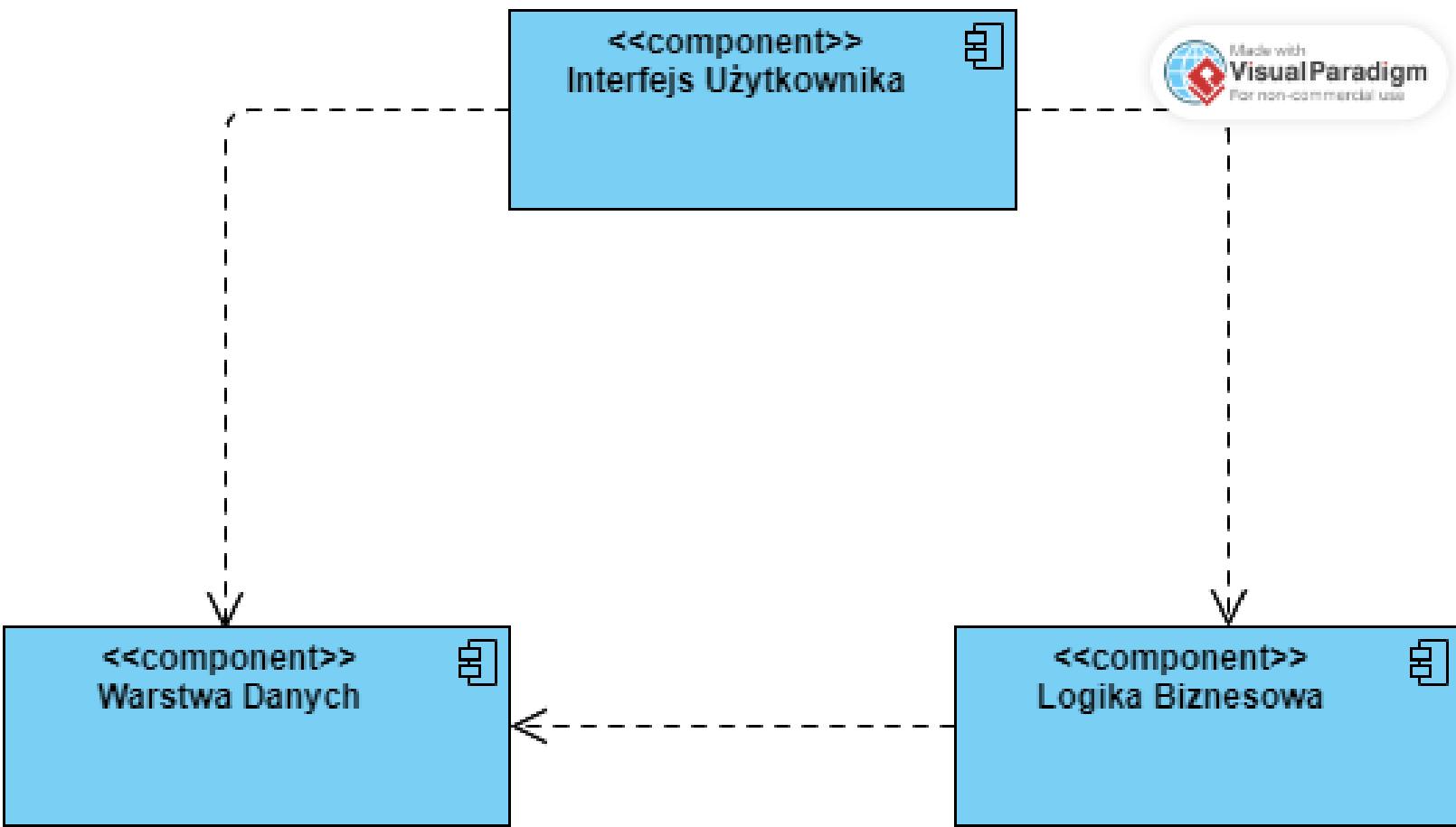


9.3.4 Inne diagramy

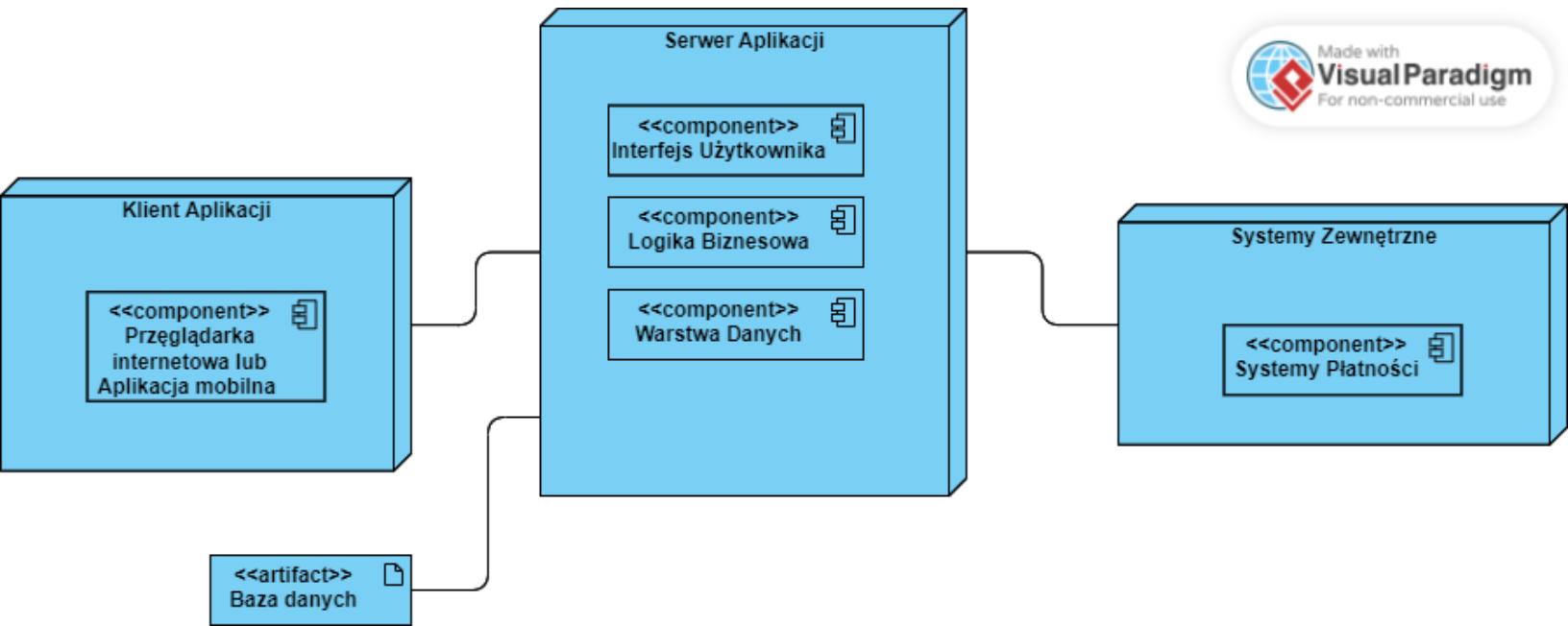
1. Maszyny stanowej



2. Komponentów



3. Rozmieszczenia



9.4 Charakterystyka zastosowanych wzorców projektowych

Singleton: Zastosowano wzorzec Singleton dla klasy Server. Zapewnia on, że istnieje tylko jedna instancja Servera w całej aplikacji, co pozwala na globalny dostęp do zarządzania zadaniami.

Model-View-Controller (MVC): Wzorzec MVC został wykorzystany do rozdzielenia logiki biznesowej od prezentacji.

Model: Reprezentuje dane i logikę biznesową zadań.

View: Odpowiada za prezentację danych użytkownikowi.

Controller: Zarządza interakcjami użytkownika i aktualizuje widoki w zależności od działań użytkownika.

9.5 Projekt bazy danych

9.5.1 Schemat

9.5.2 Projekty szczegółowe tabel



9.6 Projekt interfejsu użytkownika

9.6.1 Lista głównych elementów interfejsu

Pasek nawigacyjny: Umożliwia użytkownikom poruszanie się po różnych sekcjach aplikacji, takich jak główna strona, konta, płatności, transfer środków, ustawienia itp.

Ekran logowania: Pozwala użytkownikom zalogować się do swojego konta za pomocą nazwy użytkownika i hasła

Podgląd konta: Użytkownicy mogą zobaczyć saldo swoich kont, historię transakcji oraz inne szczegóły dotyczące ich kont bankowych.

Ekran płatności: Umożliwia użytkownikom przeprowadzanie różnych rodzajów płatności np przelewy, płatności rachunków, płatności kartą itp.

Historia transakcji: Pozwala użytkownikom przeglądać historię wszystkich transakcji dokonanych na ich kontach bankowych, w tym wpłaty, wypłaty, przelewy itp.

Powiadomienia: Informują użytkowników o różnych zdarzeniach związanych z ich kontami bankowymi, takich jak zaksięgowane transakcje, zmiany w saldzie, ostrzeżenia bezpieczeństwa itp.

Ustawienia konta: Pozwala użytkownikom zarządzać ustawieniami swojego konta, w tym zmieniać hasło, aktualizować dane kontaktowe, włączać/usuwać usługi bankowe itp.

Obsługa klienta: Zapewnia użytkownikom możliwość skontaktowania się z obsługą klienta w przypadku pytań, problemów lub potrzeby wsparcia.

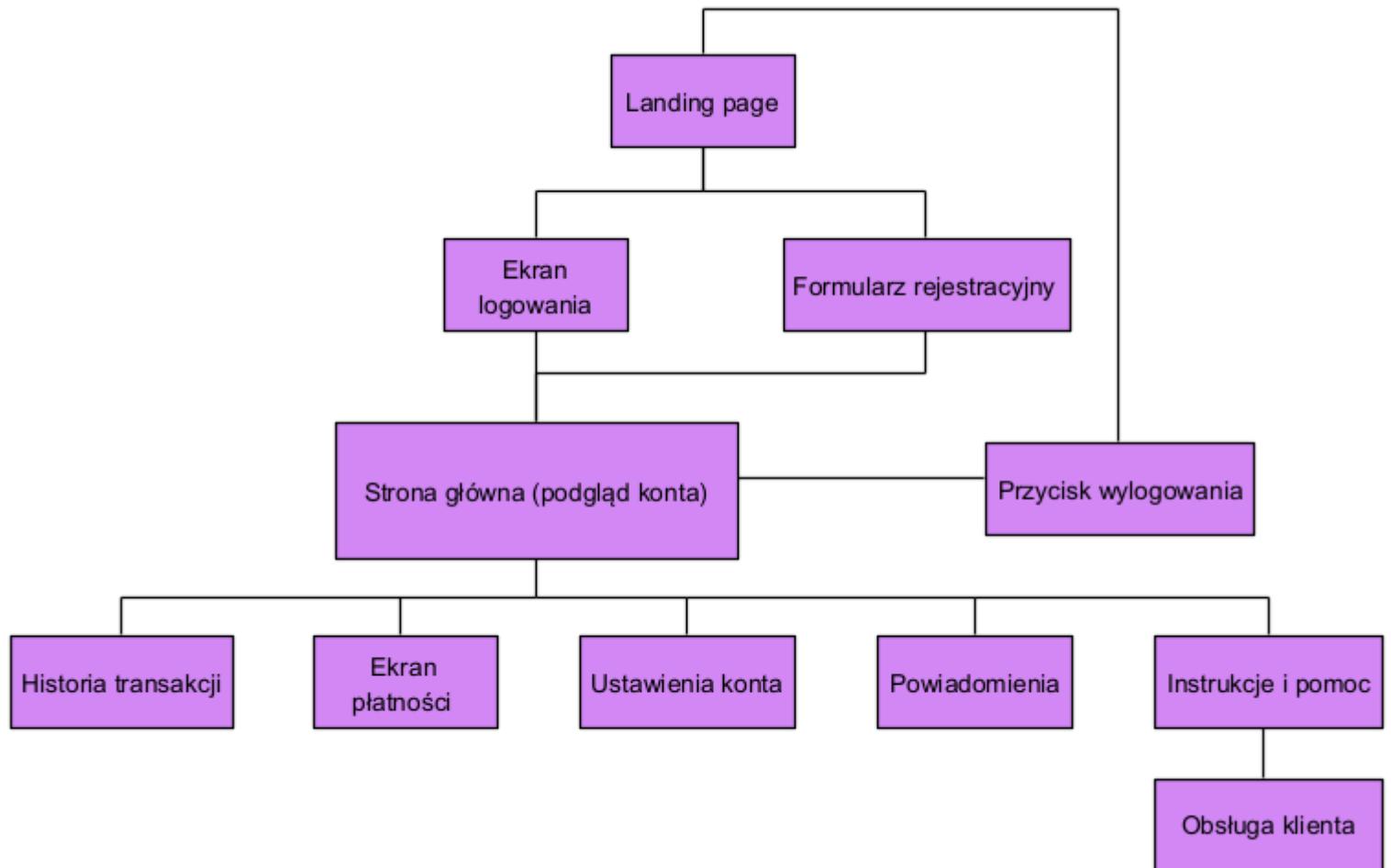
Instrukcje i pomoc: Zapewniają użytkownikom dostęp do instrukcji obsługi aplikacji, często zadawanych pytań oraz innych zasobów pomocy technicznej.

Przycisk wylogowania: Pozwala użytkownikom bezpiecznie wylogować się z aplikacji, aby zabezpieczyć swoje konto.

Formularz rejestracyjny: Użytkownicy muszą wypełnić formularz, podając swoje dane osobowe, takie jak imię, nazwisko, adres e-mail, numer telefonu, adres zamieszkania, data urodzenia itp.

Strona główna (landing page): pierwsza strona, którą użytkownicy zobaczą po uruchomieniu aplikacji.

9.6.2 Przejścia między głównymi elementami



9.6.3 Projekty szczegółowe poszczególnych elementów

1. Dashboard

Welcome Back, [USERNAME]!

Total Balance
\$ 27,777,777.04

Recent Transaction

Date	Information	Value	Details
25 FEB	Gas Station	-50.45\$...
25 FEB	Gas Station	-50.45\$...
25 FEB	Gas Station	-50.45\$...
25 FEB	Gas Station	-50.45\$...

My Card

Platinum

1234 1234

FirsName SecondName 11/24

Dashboard

Accounts

Cards

Transfers

Services

Logout

A user interface for a mobile banking application. It features a header with a welcome message and three icons: gear, bell, and profile. Below the header is a large button labeled "My Card" showing a sample credit card with a balance of \$27,777,777.04. To the left is a sidebar with navigation links: Dashboard, Accounts, Cards, Transfers, and Services. The main content area displays a "Recent Transaction" history with four entries, each showing a transaction at a gas station for -\$50.45. An illustration of a person holding a smartphone is positioned next to a large, stylized bank card icon.

2. Logowanie

Welcome to
ParaBank Online Banking

Client Number or Username

Password

Log In

or

Register

Recover Username or password
Contact with Support

The ParaBank logo is visible on the left side of the page.

A large, stylized ParaBank logo is on the left. The login form on the right includes fields for "Client Number or Username" (with placeholder "123415345436") and "Password". A large orange "Log In" button is prominent. Below it are links for "or", "Register", "Recover Username or password", and "Contact with Support".

3. Rejestracja



Create Your Account
Join **ParaBank** today!

First Name Last Name
Day Month Year
Phone Number E-mail
Password Confirm Password
 I have read and agree to the terms and conditions.

Sign In — or —
Already have an account? [Log In](#)

Contact with Support

4. Konta



Dashboard
Accounts
Cards
Transfers
Services



Main Account

Savings Account

New



5. Karty

Platinum

1234 1234

FirstName SecondName 11/24

New

Dashboard

Accounts

Cards

Transfers

Services

→

6. Przelewy

Transfer

Own transfer

Phone top-up

BLIK phone transfer

Tax transfer

Currency transfer

Dashboard

Accounts

Cards

Transfers

Services

→

7. Usługi



Dashboard
Accounts
Cards
Transfers
Services



8

Usługi

Safety zone

Stop-listing and blocking

Security strengthening

Security advice

Payments

Recipients

Upcomming and recurring payments

Transfer request

Finance management

Expense analysis

Budget

Smart Saver



9.7 Procedura wdrożenia

jeśli informacje w harmonogramie nie są wystarczające (a zapewne nie są)

10 Dokumentacja dla użytkownika

Opcjonalnie – dla chętnych

Na podstawie projektu docelowej aplikacji, a nie zaimplementowanego prototypu architektury

4-6 stron z obrazkami (np. zrzuty ekranowe, polecenia do wpisania na konsoli, itp.)

- pisana językiem odpowiednim do grupy odbiorców – czyli najczęściej nie do informatyków
- może to być przebieg krok po kroku obsługi jednej głównej funkcji systemu, kilku mniejszych, instrukcja instalacji lub innej pomocniczej czynności.

11 Podsumowanie

11.1 Szczegółowe nakłady projektowe członków zespołu

tabela (kolumny to osoby, wiersze to działania) pokazująca, kto ile czasu poświęcił na projekt oraz procentowy udział każdej osoby w danym zadaniu oraz wiersz podsumowania – procentowy udział każdej osoby w skali całego projektu

12 Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach