

Uniwersytet Gdańskie Wydział Matematyki, Fizyki i
Informatyki Instytut Informatyki

Oliver Gruba, Maciej Nasiadka

5 listopada 2025

Imię i Nazwisko (nr indeksu)	Oliver Gruba (292583) Maciej Nasiadka (292574)
Nazwa uczelni	Uniwersytet Gdańskie
Kierunek	Informatyka (profil praktyczny)
Prowadzący	dr inż. Stanisław Witkowski
Specjalność	—
Nazwa ćwiczenia	Diagram przypadków użycia — modelowanie w UML UML
Numer sprawozdania	2
Data zajęć	23.10.2025
Data oddania	05.11.2025
Miejsce na ocenę	

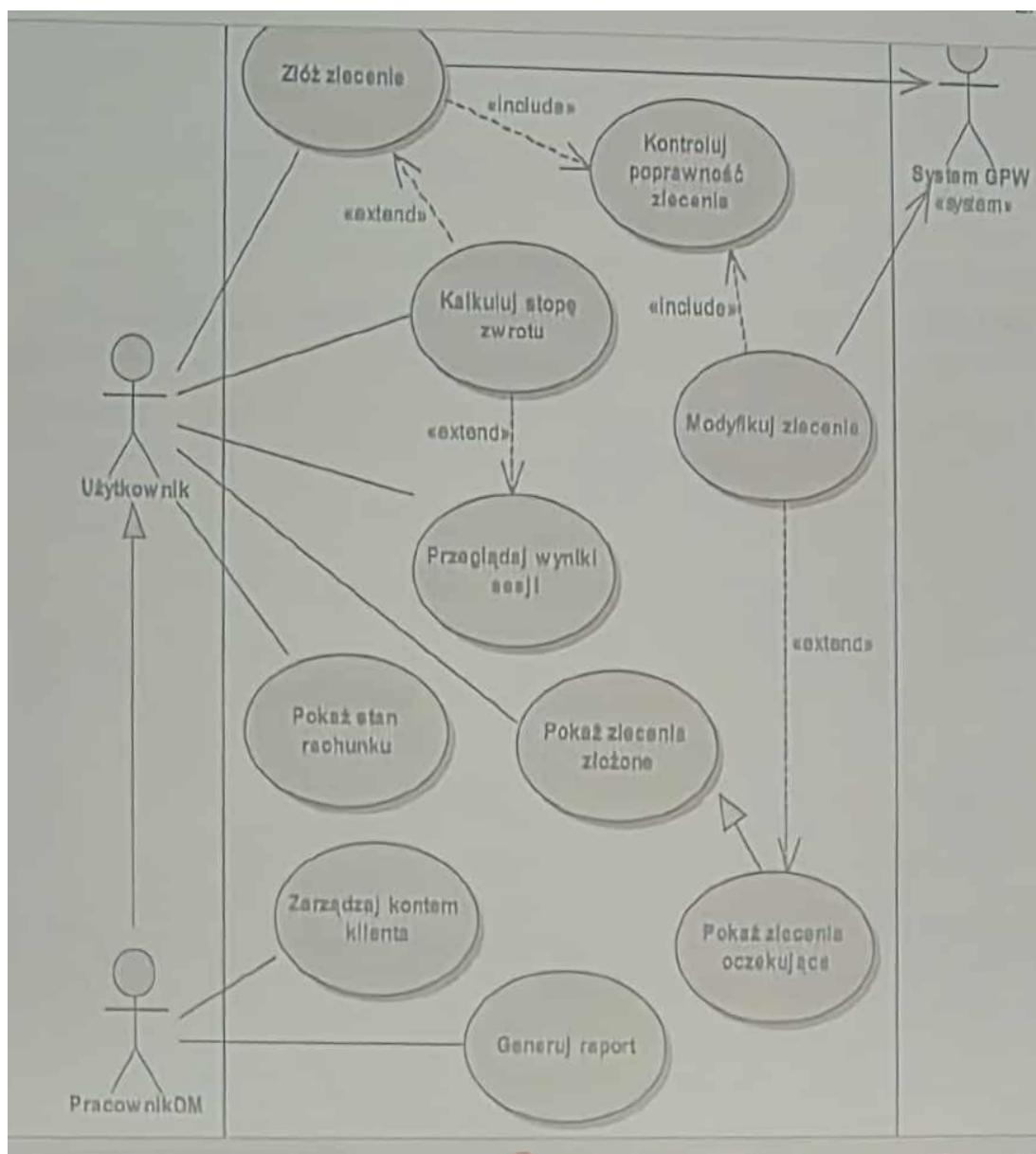
Spis treści

1 Zadanie 1: Przykład prowadzącego	3
1.1 Opis systemu	3
1.2 Wymagania	4
1.3 Struktura diagramu	4
1.4 Opis elementów diagramu	4
1.5 Przypadki użycia (Use Cases)	5
1.6 Typy relacji na diagramie	6
1.7 Podsumowanie	6
2 Zadanie 2: Przykład z pracy dyplomowej (czytelnia)	6
2.1 Wymagania funkcjonalne	6
2.2 Struktura diagramu	9
3 Zadanie 3: Realizacja pracy zespołowej	9
3.1 Temat pracy zespołowej	9
3.2 Cele pracy	9
3.3 Założenia pracy	9
3.4 Wymagania funkcjonalne i niefunkcjonalne	10
3.5 Dokumentowanie przypadków użycia	11
3.6 Przypisanie aktorów i nadanie im działań i powiązań	12
3.7 Budowa diagramu	12
3.8 Omówienie założeń diagramu przypadków użycia	13

1. Zadanie 1: Przykład prowadzącego

1.1. Opis systemu

Diagram przedstawia relacje między użytkownikami (aktorami) a funkcjonalnościami systemu (przypadekami użycia). W tym przypadku dotyczy to systemu obsługi zleceń w GPW (Giełdzie Papierów Wartościowych) lub podobnego systemu finansowego, gdzie użytkownik może składać i modyfikować zlecenia inwestycyjne, przeglądać ich stan oraz wyniki sesji.



Rysunek 1: Przykład diagramu od prowadzącego

1.2. Wymagania

System ma umożliwić użytkownikom przeglądanie, składanie i ewentualne modyfikowanie zleceń inwestycyjnych na Giełdzie Papierów Wartościowych.

System powinien automatycznie kontrolować poprawność składanych zleceń oraz komunikować się z systemem Giełdy Papierów Wartościowych w celu realizacji zleceń lub ich ewentualnej modyfikacji.

Użytkownik powinien mieć dostęp do podglądu własnych wyników sesji, podglądu własnych zleceń złożonych, podglądu własnego stanu rachunku oraz wyliczenia stopy procentowej zwrotu.

Pracownik DM powinien mieć możliwość zarządzania kontami klientów oraz możliwość generowania raportów.

1.3. Struktura diagramu

Diagram przypadków użycia przedstawia strukturę interakcji pomiędzy trzema różnymi aktorami: Użytkownikiem, Pracownikiem Domu Maklerskiego oraz Systemem Giełdy Papierów Wartościowych. Aktorzy korzystają z różnych przypadków użycia reprezentujących funkcjonalności systemu. Relacje typu *include* oznaczają, że dana czynność zawsze obejmuje inną (np. modyfikuj zlecenie zawsze obejmuje kontroluj poprawność zlecenia). Relacje typu *extend* wskazują na możliwość rozszerzenia głównego przypadku o dodatkowe działania (np. modyfikuj zlecenie może być rozszerzone o pokaz zleceń oczekujące). Diagram odzwierciedla logiczną strukturę funkcji systemu oraz pokazuje, które elementy są współdzielone lub zależne między aktorami.

1.4. Opis elementów diagramu

Aktorzy

- **Użytkownik:** podstawowy aktor korzystający z systemu do składania i zarządzania zleceniami inwestycyjnymi (np. inwestor indywidualny). Może wykonać następujące czynności:
 - Przeglądać zlecenia
 - Składać zlecenia
 - Modyfikować zlecenia
 - Przeglądać wyniki sesji
 - Przeglądać stan rachunku
 - Wyliczać stopę procentową zwrotu

- **Pracownik Domu Maklerskiego:** aktor odpowiedzialny za zarządzanie kontami klientów oraz generowanie raportów. Może wykonać następujące czynności:
 - Zarządzać kontami klientów
 - Generować raporty
- **System Giełdy Papierów Wartościowych:** zewnętrzny system, z którym komunikuje się nasz system w celu realizacji lub modyfikacji zleceń. Jest powiązany z przypadkiem użycia Modyfikuj zlecenie.

1.5. Przypadki użycia (Use Cases)

Przypadek użycia	Znaczenie
Złoż zlecenie	Użytkownik wprowadza nowe zlecenie do systemu.
Kontroluj poprawność zlecenia («include»)	System automatycznie sprawdza, czy zlecenie jest poprawne (np. dane, format, limity).
Kalkuluj stopę zwrotu («extend»)	Użytkownik może wyliczyć zwrot z inwestycji jako rozszerzenie procesu składania zleceń.
Przeglądaj wyniki sesji	Użytkownik analizuje rezultaty notowań.
Pokaż stan rachunku	Wyświetlenie aktualnego salda i wartości portfela.
Pokaż zlecenia złożone / oczekujące	Użytkownik ma wgląd w swoje aktywne i historyczne zlecenia.
Modyfikuj zlecenie	Użytkownik (lub system) może zmienić parametry złożonego zleceń.
Zarządzaj kontem klienta	Pracownik DM ma dostęp administracyjny do kont klientów.
Generuj raport	Tworzenie raportów z działania systemu lub kont klientów.

1.6. Typy relacji na diagramie

typ relacji	opis	Przykład
Association (asocjacja)	Linia łącząca aktora z przypadkiem użycia - pokazuje, kto korzysta z danej funkcji.	Użytkownik → Złoż zlecenie
«include» (zawiera)	Jeden przypadek użycia zawsze wywołuje inny, bo jest jego częścią.	Złoż zlecenie → Kontroluj poprawność zlecenia
«extend» (rozszerza)	Przypadek użycia czasami wywołuje inny (opcjonalnie, np. zależnie od sytuacji).	Złoż zlecenie → Kalkuluj stopę zwrotu
Dependency (zależność)	Pokazuje wpływ jednego elementu na inny, często między systemami.	System GPW ↔ Modyfikuj zlecenie

1.7. Podsumowanie

Diagram przypadków użycia przedstawia kompleksowy obraz interakcji między użytkownikami a systemem obsługi zleceń inwestycyjnych. Pokazuje, jakie funkcje są dostępne dla różnych aktorów oraz jak te funkcje są ze sobą powiązane poprzez różne typy relacji. Dzięki temu można lepiej zrozumieć wymagania systemu oraz sposób jego działania. Diagram ten może służyć jako podstawa do dalszego projektowania systemu, implementacji oraz testowania jego funkcjonalności lub jako dokumentacja dla zespołu deweloperskiego.

2. Zadanie 2: Przykład z pracy dyplomowej (czytelnia)

2.1. Wymagania funkcjonalne

Określenie wymagań funkcjonalnych to bardzo ważny etap podczas powstawania różnego rodzaju oprogramowania. Definiują one w jaki sposób będzie działał tworzony system i jak będzie się on zachowywał. Przedstawia cechy i funkcje, które powinna posiadać aplikacja, tak aby potrzeby i oczekiwania użytkownika zostały spełnione w jak największym stopniu. W uproszczeniu można przyjąć, że są to cechy produktu, które są wykrywane przez jego użytkownika. Przykładowo, może to być przycisk na stronie pełniący konkretną, określoną funkcję. System taki nie będzie mógł działać, jeśli nie będą spełnione wszystkie określone wcześniej wymagania funkcjonalne.

Na podstawie analizy istniejących rozwiązań bibliotecznych wybrano najważniejsze cechy, które powinna posiadać tworzona aplikacja. Cechy te zostały podzielone ze względu na stan zalogowania użytkownika oraz jego rolę w systemie.

Dla użytkownika niezalogowanego system umożliwia

- zarejestrowanie się i stworzenie konta podając imię, nazwisko, unikalny adres e-mail oraz hasło spełniające wymogi bezpieczeństwa
- zalogowanie się do systemu za pomocą adresu e-mail oraz hasła podanego przy rejestracji.

Dla użytkownika zalogowanego system umożliwia

- przeglądanie materiałów bibliotecznych (książki, czasopisma, prace dyplomowe) w formie papierowej udostępnione do wypożyczenia.
- przeglądanie materiałów bibliotecznych w formie cyfrowej udostępnione do pobrania.
- przeglądanie materiałów bibliotecznych w formie papierowej aktualnie niedostępnych (wypożyczonych przez innych użytkowników).
- filtrowanie katalogu stosując kryteria: autor, tytuł, wydawca lub kategoria. Kryteria filtrowania powinny móc być zastosowane osobno lub kilka jednocześnie.
- zarezerwowanie materiałów bibliotecznych dostępnych do wypożyczenia.
- prolongowanie aktualnie trwającego wypożyczenia.
- pobranie materiału bibliotecznego w formie cyfrowej dostępnego do pobrania.
- dodanie za pomocą formularza nowych rekordów do katalogu (bazy danych) materiałów bibliotecznych wymagających zaakceptowania przez administratora.
- dodanie autora do bazy danych za pomocą formularza.
- dodanie wydawnictwa do bazy danych za pomocą formularza.
- otrzymywanie powiadomień o wydarzeniach związanych z wypożyczeniami lub materiałami bibliotecznymi (zaakceptowanie lub odrzucenie dodanego materiału bibliotecznego, zaakceptowanie lub odrzucenie wypożyczenia, przypomnienie o zbliżającym się terminie zwrotu materiału, przypomnienie o upływie terminie zwrotu materiału).
- oznaczanie powiadomień jako odczytane.
- przeglądanie trwających, oczekujących, zakończonych oraz odrzuconych wypożyczeń.

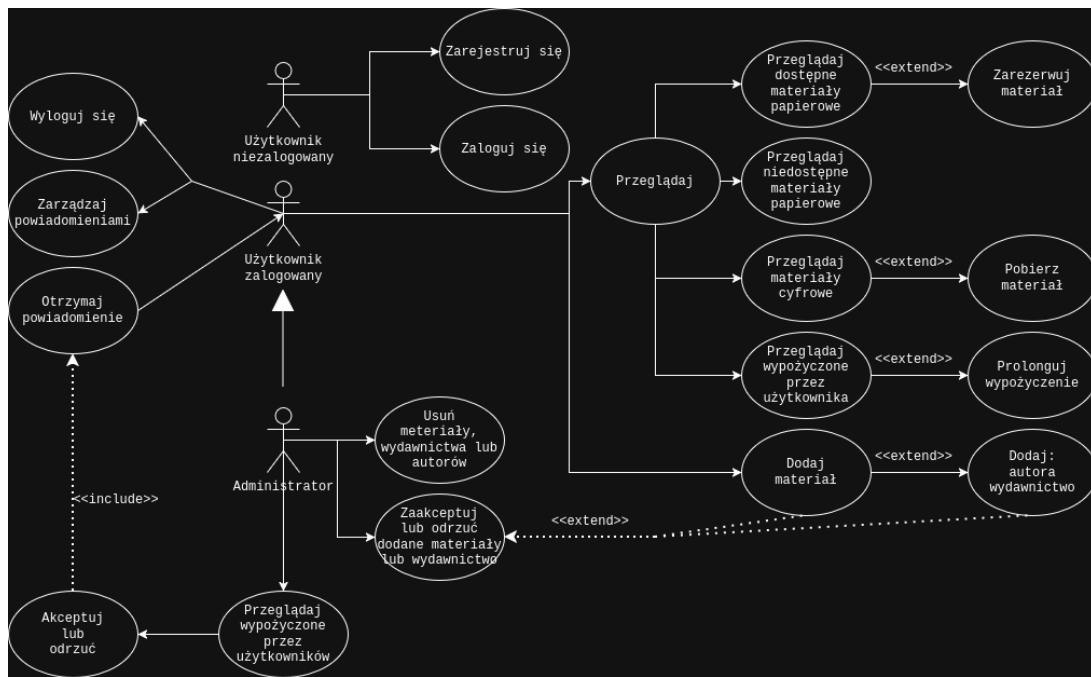
- wylogowanie się z systemu.

Dla użytkownika administratora system umożliwia

- przeglądanie materiałów bibliotecznych oczekujących na zatwierdzenie.
- zaakceptowanie lub odrzucenie oczekującego materiału bibliotecznego.
- przeglądanie trwających lub zakończonych wypożyczeń materiałów bibliotecznych wszystkich użytkowników.
- filtrowanie wypożyczeń użytkowników według ich imienia i nazwiska.
- wysyłanie powiadomień przypominających o terminie zwrotu materiału bibliotecznego.
- oznaczanie wypożyczeń jako zakończone.
- przeglądanie wypożyczeń oczekujących na zaakceptowanie.
- zatwierdzanie lub odrzucanie oczekujących wypożyczeń.
- edytowanie informacji o materiałach bibliotecznych w bazie danych.
- usuwanie materiałów bibliotecznych z bazy danych.
- dodanie za pomocą formularza nowych rekordów do katalogu (bazy danych) materiałów bibliotecznych bez potrzeby zatwierdzania.

Projekt zakłada, że niektóre funkcje użytkownika zalogowanego i administratora są zbieżne. Dotyczą one przeglądania katalogu dostępnych materiałów bibliotecznych czy dodawania autorów i wydawnictw do bazy danych podczas uzupełniania formularza do dodawania nowego rekordu bibliotecznego.

2.2. Struktura diagramu



Rysunek 2: Struktura diagramu przypadków użycia dla czytelni

3. Zadanie 3: Realizacja pracy zespołowej

3.1. Temat pracy zespołowej

Kairo Habit App

Aplikacja mobilna do monitorowania i utrwalania nawyków z elementami grywalizacji i społeczności.

3.2. Cele pracy

Celem pracy jest stworzenie funkcjonalnej aplikacji mobilnej umożliwiającej użytkownikom monitorowanie własnych nawyków oraz obserwowanie postępów znajomych. Projekt ma angażować użytkowników poprzez gamifikacje: system punktów, osiągnięć i rankingów. Dążymy do promowania zdrowych i produktywnych zachowań oraz budowania społeczności motywujących się nawzajem. Ostatecznym celem jest dostarczenie estetycznej, intuicyjnej i wydajnej aplikacji dostępnej dla systemów Android i iOS.

3.3. Założenia pracy

- System ma być aplikacją mobilną (Android/iOS), wykonaną z użyciem frameworka wieloplatformowego – React Native

- Użytkownicy tworzą konta i definiują swoje nawyki (np. czytanie, ćwiczenia, nauka).
- Aplikacja śledzi postępy użytkownika, generuje statystyki (streak, wykresy, oś czasu).
- System wspiera funkcje społecznościowe: dodawanie znajomych, przeglądanie ich aktywności, wysyłanie powiadomień (pingów).
- Backend oparty na REST API + baza danych – PHP

3.4. Wymagania funkcjonalne i niefunkcjonalne

Wymagania funkcjonalne

- Rejestracja i logowanie użytkownika.
- Dodawanie, edycja i usuwanie nawyków (CRUD).
- Przeglądanie własnych postępów (streak, wykresy, oś czasu).
- Dodawanie znajomych i podgląd ich aktywności.
- System osiągnięć i powiadomień.
- Tryb skupienia i możliwość zamrożenia streaka.
- Możliwość pingowania znajomych.
- Dodawanie powiadomień do kalendarza.
- Personalizacja motywów aplikacji.

Wymagania niefunkcjonalne

- Intuicyjny i estetyczny interfejs użytkownika (UI/UX).
- Wysoka dostępność i stabilność aplikacji.
- Bezpieczeństwo danych (szyfrowanie, uwierzytelnianie).
- Responsywność (działanie na różnych rozdzielczościach ekranów)
- Skalowalność backendu.
- Czas reakcji interfejsu < 1 sekundy.
- Zgodność z wytycznymi platform Android i iOS.

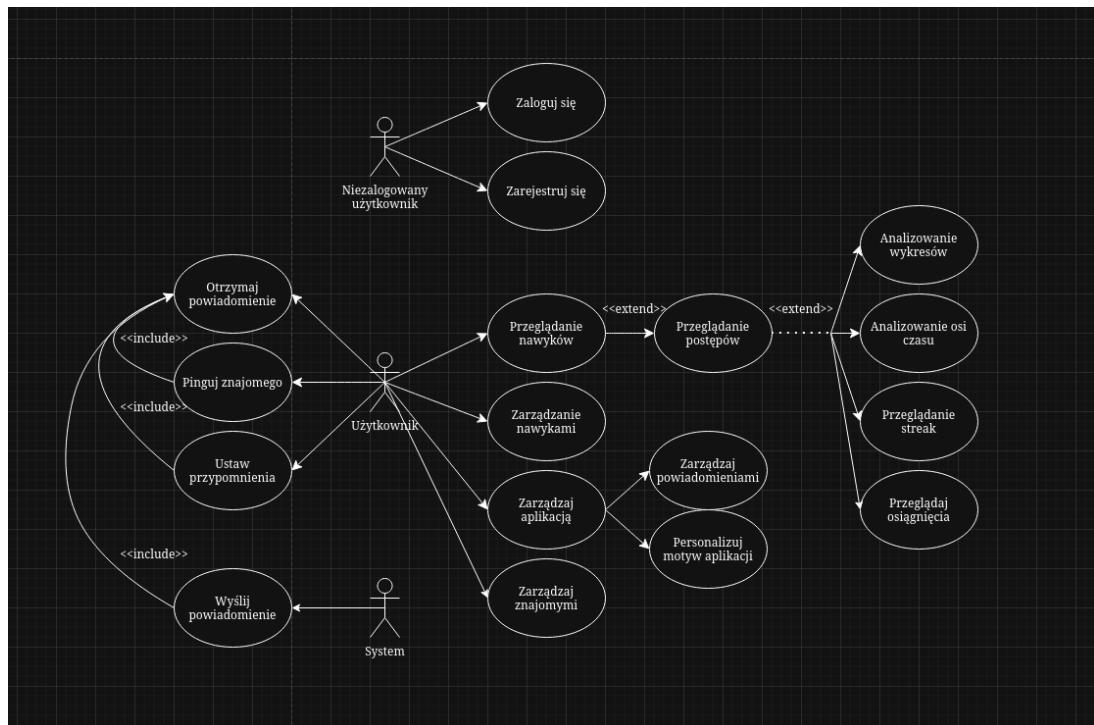
3.5. Dokumentowanie przypadków użycia

ID	Nazwa przypadku użycia	Aktor	Opis	Warunek początkowy	Wynik końcowy
UC1	Rejestracja użytkownika	Użytkownik	Użytkownik wypełnia formularz rejestracji i zakłada konto.	Brak konta	Konto utworzone i zapisane w bazie.
UC2	Dodanie nowego nawyku	Użytkownik	Użytkownik dodaje nowy nawyk do śledzenia.	Użytkownik zalogowany	Nawyk zapisany w systemie.
UC3	Przegląd postępów	Użytkownik	Użytkownik przegląda wykres i streak swoich aktywności.	Użytkownik posiada zapisane dane	Wyświetlenie statystyk.
UC4	Dodanie znajomego	Użytkownik	Użytkownik wysyła zaproszenie do znajomego.	Obaj użytkownicy posiadają konta	Znajomość zapisana w systemie.
UC5	Pingowanie znajomego	Użytkownik	Użytkownik motywuje znajomego do działania.	Obaj użytkownicy są znajomymi	Znajomy otrzymuje powiadomienie.

3.6. Przypisanie aktorów i nadanie im działań i powiązań

Aktor	Opis roli	Powiązane przypadki użycia
Użytkownik	Główny aktor systemu. Rejestruje się, zarządza nawykami, przegląda statystyki.	UC1, UC2, UC3, UC4, UC5
System	Przechowuje dane, generuje statystyki, obsługuje powiadomienia.	UC2, UC3, UC4
Znajomy	Interaktywny element społeczny – odbiera powiadomienia i motywuje użytkownika.	UC4, UC5
Administrator (opcjonalny)	Zarządza kontami i monitoruje poprawność działania aplikacji.	UC1–UC5

3.7. Budowa diagramu



Rysunek 3: Diagram przypadków użycia aplikacji mobilnej do śledzenia nawyków użytkownika

3.8. Omówienie założeń diagramu przypadków użycia

Diagram przypadków użycia przedstawia główne interakcje pomiędzy użytkownikami systemu a jego funkcjonalnościami. System ten służy do wspierania użytkowników w zarządzaniu nawykami, śledzeniu postępów oraz utrzymywaniu motywacji poprzez powiadomienia i elementy społecznościowe. Na diagramie wyróżniono trzech aktorów: *Niezalogowany użytkownik*, *Użytkownik* oraz *System*.

Niezalogowany użytkownik Aktor ten ma dostęp wyłącznie do podstawowych funkcjonalności, umożliwiających rozpoczęcie pracy z aplikacją:

- **Zaloguj się** - umożliwia autoryzację i uzyskanie dostępu do pełnej funkcjonalności systemu.
- **Zarejestruj się** - pozwala na utworzenie nowego konta użytkownika.

Po wykonaniu jednej z tych czynności użytkownik staje się aktorem *Użytkownik* i może korzystać z pozostałych przypadków użycia.

Użytkownik Zalogowany użytkownik posiada pełny dostęp do możliwości aplikacji. Może on:

- **Przeglądać nawyki** - uzyskać wgląd w listę swoich nawyków oraz ich aktualny stan.
- **Zarządzać nawykami** - dodawać, edytować lub usuwać nawyki.
- **Przeglądać postępy** - analizować swoje osiągnięcia i wyniki. Ten przypadek użycia jest rozszerzany (*extend*) o:
 - **Analizowanie wykresów** - wizualizacja postępów w formie graficznej.
 - **Analizowanie osi czasu** - śledzenie zmian w kontekście czasowym.
 - **Przeglądanie streaków** - obserwacja ciągłości realizacji nawyków.
 - **Przeglądanie osiągnięć** - prezentacja zdobytych rezultatów i odznak.
- **Zarządzać aplikacją** - dostosować ustawienia systemu do własnych preferencji, w tym:
 - **Zarządzać powiadomieniami** - konfigurować przypomnienia i alerty.
 - **Personalizować motyw aplikacji** - zmieniać wygląd interfejsu użytkownika.
- **Zarządzać znajomymi** - dodawać i usuwać znajomych oraz utrzymywać z nimi interakcję.

- **Pingować znajomego** - wysyłać krótkie przypomnienia lub motywacyjne wiadomości.
- **Ustawiać przypomnienia** - tworzyć i planować automatyczne alerty.
- **Otrzymywać powiadomienia** - odbierać informacje o nadchodzących zadaniach, aktywnościach lub wiadomościach od znajomych.

Przypadki użycia *Pinguj znajomego*, *Ustaw przypomnienia* oraz *Otrzymaj powiadomienie* zawierają (*include*) przypadek *Wyślij powiadomienie*, realizowany przez system.

System *System* jest aktorem automatycznym, odpowiedzialnym za wykonywanie działań w tle, w szczególności za:

- **Wysyłanie powiadomień** - informowanie użytkowników o zaplanowanych przypomnieniach, aktywnościach znajomych lub osiągnięciach.

Relacje między przypadkami użycia Na diagramie występują dwa typy relacji:

- **Relacja *include*** - oznacza obowiązkowełączenie jednego przypadku do innego (np. wysyłanie powiadomienia jako część przypomnienia).
- **Relacja *extend*** - wskazuje na opcjonalne rozszerzenie funkcjonalności, np. przeglądanie postępów może być rozszerzone o analizę wykresów lub osi czasu.

Założenia ogólne systemu System został zaprojektowany z myślą o wspieraniu użytkowników w budowaniu pozytywnych nawyków i utrzymywaniu motywacji. Zakłada się:

- integrację funkcji społecznościowych (współpraca i rywalizacja ze znajomymi),
- automatyzację przypomnień i powiadomień,
- wykorzystanie wizualizacji danych do motywowania użytkownika,
- możliwość personalizacji interfejsu i ustawień aplikacji.

Diagram ukazuje kompleksową strukturę interakcji między użytkownikami a systemem, wskazując główne obszary funkcjonalne aplikacji oraz zależności między nimi.