

*Krzysztof Wilaszek  
Tomasz Wójcik  
Piotr Leśniak*

## SPIS TREŚCI

<b>Wymagania funkcjonalne.....</b>	<b>3</b>
Wstęp.....	3
Zarządzanie kontem użytkownika .....	3
Tworzenie Konta .....	3
Logowanie .....	3
Wylogowanie .....	4
Wyświetlenie grup użytkownika .....	4
Zarządzanie grupami.....	5
Utworzenie grupy.....	5
Dodanie członka do grupy.....	6
Opuszczenie grupy .....	6
Wyświetlenie informacji o grupie .....	7
Rachunki i opłaty .....	7
Dodanie opłaty i pozycji opłaty .....	7
Zgłoszenie i zatwierdzenie spłaty długu .....	8

<b>Projekt systemu.....</b>	<b>10</b>
Diagram klas .....	10
Diagramy sekwencji .....	12
<b>Postęp implementacji.....</b>	<b>15</b>
Warstwa danych .....	15
Warstwa serwisowa i prezentacji .....	15
<b>Konfiguracja nowych narzędzi .....</b>	<b>16</b>
Jenkins .....	16
Github – wiki i bug tracker.....	16

## WYMAGANIA FUNKCJONALNE

### WSTĘP

W tym rozdziale opisane zostaną wymagania funkcjonalne stawiane aplikacji Cyberwej do zarządzania rachunkami i opłatami w grupach znajomych. Wymagania zostały podzielone na trzy podstawowe obszary, zarządzanie kontem użytkownika, zarządzanie grupami oraz rachunki i opłaty.

### ZARZĄDZANIE KONTEM UŻYTKOWNIKA

#### TWORZENIE KONTA

Każdy użytkownik w ramach aplikacji Cyberwej będzie posiadał własne konto z spersonalizowanym widokiem strony głównej. Użytkownik będzie miał możliwość utworzenia konta, wypełniając elektroniczny formularz z podstawowymi informacjami oraz hasłem.

#### SCENARIUSZ UŻYCIA

Aktor: Niezalogowany użytkownik

Cel przypadku użycia: Utworzenie konta

1. Użytkownik wybiera opcję utworzenia konta
2. System wyświetla formularz tworzenia nowego konta
3. Klient wprowadza wartości dla wszystkich wymaganych pól (identyfikator, imię, nazwisko?) i opcjonalnie wartości dla pozostałych pól
4. Klient wprowadza hasło dostępu do konta
5. Klient zatwierdza wprowadzone dane
6. System tworzy konto

#### LOGOWANIE

Po wybraniu w przeglądarce adresu aplikacji Cyberwej, użytkownik będzie miał możliwość zalogowania się na wcześniej utworzone przez siebie konto, podając login i hasło.

## SCENARIUSZ UŻYCIA

---

Aktor: Niezalogowany użytkownik

Cel przypadku użycia: Zalogowanie się

1. Użytkownik wybiera opcję zalogowania się
2. Użytkownik wprowadza swój identyfikator i hasło
3. Użytkownik naciska przycisk logowania
4. System loguje użytkownika do jego konta

---

## WYLOGOWANIE

Po zakończeniu pracy z aplikacją, użytkownik będzie mógł wylogować się z poziomu głównej strony Cyberweji.

## SCENARIUSZ UŻYCIA

---

Aktor: Zalogowany użytkownik

Cel przypadku użycia: Wylogowanie się

1. Użytkownik naciska przycisk wylogowania
2. System wylogowuje użytkownika, przenosi użytkownika na główną stronę dostępną dla niezalogowanych użytkowników.

---

## WYŚWIETLENIE GRUP UŻYTKOWNIKA

Na głównej stronie aplikacji, użytkownik będzie miał podgląd grup do których należy oraz bilans swoich wydatków. Dzięki temu każdy użytkownik będzie wiedział jaki jest jego status rozliczenia opłat względem innych członków grup do których należy.

Na głównej stronie będą także dostępne notyfikacje (takie jak zaproszenia do grupy, zgłoszenia spłaty długu) oraz odpowiednie mechanizmy do wyszukania grup oraz użytkowników a także stworzenia nowej grupy.

## SCENARIUSZ UŻYCIA

---

Aktor: Zalogowany użytkownik

Cel przypadku użycia: wyświetlenie grup, do których się należy

1. Użytkownik wybiera link wyświetlania swoich grup
2. System wyświetla wszystkie grupy użytkownika – ich nazwy wraz z bilansem wpłat użytkownika względem grup

## ZARZĄDZANIE GRUPAMI

### UTWORZENIE GRUPY

Każdy użytkownik będzie miał możliwość tworzenia grup w ramach których dokonywane będą rozliczenia za rachunki. Każda grupa składać się może z wielu użytkowników, którzy dokonują wspólnego rozliczenia się z opłat.

Użytkownik, chcąc utworzyć grupę podaje jej nazwę oraz listę członków, których do niej zaprasza. Każdy użytkownik zaproszony do danej grupy, otrzyma zaproszenie wyświetlane na jego głównej stronie aplikacji. Dopiero po potwierdzeniu zaproszenia użytkownik zostanie dodany do grupy.

## SCENARIUSZ UŻYCIA

---

Aktor: Zalogowany użytkownik

Cel przypadku użycia: utworzenie nowej grupy oraz określenie jej członków

1. Użytkownik naciska link tworzenia nowej grupy
2. Użytkownik wprowadza nazwę grupy
3. Użytkownik wprowadza członków grupy
4. System tworzy grupę i zapisuje informację o niej.
5. System dodaje użytkownika do nowo utworzonej grupy
6. System przekazuje informację o zaproszeniu do grupy do dodanych użytkowników

## SCENARIUSZ UŻYCIA II

---

Aktor: Zalogowany użytkownik

Cel przypadku użycia: Odpowiedź na zaproszenie do grupy i ewentualne dołączenie się do grupy

1. System wyświetla użytkownikowi informację o zaproszeniu do grupy(nazwę grupy, osobę zapraszającą)
2. Użytkownik zatwierdza zaproszenie do grupy
3. System dodaje nowego członka do grupy

---

### DODANIE CZŁONKA DO GRUPY

Każdy członek grupy może zaprosić innego użytkownika do bycia członkiem grupy. Wtedy taki użytkownik otrzymuje zaproszenie wyświetlane na głównej stronie, po potwierdzeniu którego staje się członkiem grupy.

## SCENARIUSZ UŻYCIA

---

Aktor: Członek grupy

Cel przypadku użycia: Dodanie nowego członka do grupy

1. Członek grupy wybiera opcję dodania nowej osoby do grupy
2. Użytkownik wybiera osobę, która powinna być dodana do grupy
3. System przekazuje informację o zaproszeniu do grupy do dodanego użytkownika

---

### OPUSZCZENIE GRUPY

Możliwe jest opuszczenie grupy. Użytkownik który nie posiada zobowiązań względem innych członków grupy (jego bilans opłat wynosi 0) może opuścić grupę. Po opuszczeniu grupy użytkownik nie jest już brany pod uwagę w rozliczeniach odbywających się w ramach tej grupy.

## SCENARIUSZ UŻYCIA

---

Aktor: Członek grupy

Cel przypadku użycia: Odejście z grupy

1. Użytkownik naciska przycisk opuszczenia grupy.
2. System wyświetla okno potwierdzenia decyzji o opuszczeniu grupy
3. Użytkownik potwierdza swój wybór
4. System sprawdza brak zobowiązań użytkownika wobec grupy
5. System usuwa użytkownika z grupy

---

## **WYŚWIETLENIE INFORMACJI O GRUPIE**

Każdy użytkownik będzie mógł wyszukać grupę po jej nazwie oraz wyświetlić jej podstawowe informacje (data, użytkownicy). Członkowie grupy będą mogli wyświetlić szczegółowe informacje dotyczące grupy, bilans opłat, szczegółowe informacje na temat stanu opłat każdego z użytkowników.

### **SCENARIUSZ UŻYCIA**

---

Aktor: Członek grupy

Cel przypadku użycia: prezentacja informacji o grupie

1. Użytkownik naciska na link grupy, dla której chce zobaczyć widok szczegółowych informacji
2. System wyświetla szczegółowe informacje o grupie – nazwę, listę członków, listę wspólnych opłat

## **RACHUNKI I OPŁATY**

---

### **DODANIE OPŁATY I POZYCJI OPŁATY**

W ramach danej grupy, użytkownicy będą mieli możliwość rozliczania się za wspólne wydatki. System będzie przechowywał informację o zarejestrowanych opłatach i wyświetlał status każdego użytkownika względem innych.

Użytkownik, w ramach grupy której jest członkiem będzie miał możliwość zarejestrowania wspólnego rozliczenia. W tym celu użytkownik stworzy nową opłatę, do której doda odpowiednie pozycje odzwierciedlające wydarzenie wspólnego rozliczenia. Każda pozycja będzie składała się z ceny, ilości oraz osób płacących za dane dobro lub usługę. Każda opłata składać się będzie z listy pozycji. Po zarejestrowaniu opłaty system automatycznie obliczy, jaki jest status rozliczenia pomiędzy członkami grupy.

## SCENARIUSZ UŻYCIA

---

Aktor: Członek grupy

Cel przypadku użycia: Wprowadzenie nowej wspólnej opłaty do systemu

1. Użytkownik wybiera grupę
2. Użytkownik wybiera opcję dodania opłaty
3. Użytkownik wprowadza datę opłaty
4. Użytkownik wprowadza pozycje opłaty
5. Użytkownik wybiera opcję zapisu opłaty
6. System dodaje opłatę do grupy

## SCENARIUSZ UŻYCIA II

---

Aktor: Członek grupy

Cel przypadku użycia: Dodanie do opłaty nowej pozycji oraz osób, które z niej korzystały

1. Użytkownik wybiera opcję dodania pozycji do opłaty
2. Użytkownik wprowadza nazwę pozycji
3. Użytkownik określa liczbę zakupionych towarów/usług
4. Użytkownik określa użytkowników konsumujących zakupione dobra
5. Użytkownika zatwierdza dodawaną pozycję
6. System dodaje pozycję do opłaty

---

## ZGŁOSZENIE I ZATWIERDZENIE SPŁATY DŁUGU

Użytkownik, który jest winny pieniądze innym użytkownikom, po oddaniu należności będzie miał możliwość zarejestrować ten fakt w aplikacji Cyberwej. Użytkownik, który oddawał pieniądze wybierze członka grupy, któremu zwrócił pieniądze oraz sumę. Użytkownik, któremu pieniądze zostały oddane, zostanie poinformowany przez odpowiednią notyfikację o wydarzeniu zwrotu pieniędzy. Po zaakceptowaniu przez niego zwrotu, system odnotuje i przeliczy status rozliczenia pomiędzy członkami grupy.



## SCENARIUSZ UŻYCIA

---

Aktor: Członek grupy z niedopłatą

Cel przypadku użycia: odnotowanie spłaty zobowiązań wobec członka grupy

1. Użytkownik wybiera grupę
2. Użytkownika wybiera osobę której był winny pieniądze
3. Użytkownik wybiera opcję poinformowania o spłacie długu
4. Użytkownik wprowadza kwotę spłaty
5. Użytkownik potwierdza zgłoszenie spłaty długu
6. System przekazuje zgłoszenie do osoby zatwierdzającej

## SCENARIUSZ UŻYCIA II

---

Aktor: Członek grupy z nadpłatą

Cel przypadku użycia: zatwierdzenie spłaty zobowiązań, wprowadzenie do systemu zmian wartości nadpłat i zadłużenia

1. System wyświetla użytkownikowi prośbę o zatwierdzenie zgłoszenia spłaty długu(wraz z informacją o osobie spłacającej i wysokości spłaty)
2. Użytkownik zatwierdza zgłoszenie
3. System odnotowuje zwrot(zmniejsza wartość zadłużenie osoby spłacającej i zmniejsza nadpłatę osoby akceptującej zgłoszenie)

#### OPIS

---

**User** – klasa reprezentująca użytkownika systemu. Pola opisujące użytkownika to name, surname, mail, login, location. Pola name i login powinny być unikalne w obrębie aplikacji. Użytkownik ma również dostęp do następujących obiektów:

- groupMemberships – wykaz przynależności do grup
- userInvitations – wykaz zaproszeń kierowanych do użytkownika
- paybacksForUser – zgłoszenia spłaty długu oczekujące na akceptację użytkownika
- paybacksForOthers – spłaty zgłoszone przez użytkownika oczekujące na akceptację

**Group** – klasa reprezentuje grupy użytkowników w systemie. Grupa posiada nazwę unikalną w obrębie aplikacji. Grupa posiada również wykaz następujących obiektów ze sobą powiązanych:

- groupMembers – wykaz przynależności do grupy
- invitations – wykaz oczekujących zaproszeń do grupy
- payments – wykaz opłat grupy
- paybacks – wykaz spłat w obrębie grupy

**GroupMembership** – klasa reprezentująca przynależność do grupy. Zawiera wskazanie na osobę i grupę. Przechowuje informacje o statusie wpłat użytkownika względem grupy.

**Payment** – klasa reprezentuje opłatę grupy. Przechowuje informacje o opłacie – datę oraz opis. Opłata składa się z pozycji opłaty – paymentItems. Opłata przechowuje również listę osób, które złożyły się na tę opłatę.

**Product** – klasa reprezentuje wprowadzone do systemu dobra i usługi, które mogą znaleźć się liście pozycji opłaty. Produkt identyfikowany jest poprzez unikalną nazwę.

**PaymentItem**—obiekty tej klasy odpowiadają jednej pozycji opłaty. Zawierają informację o tym jakiego produktu dotyczą, posiadają cenę i liczbę zakupionych produktów(usług). Obiekt tej klasy posiada wykaz osób, które korzystały z zakupionego produktu bądź usługi.

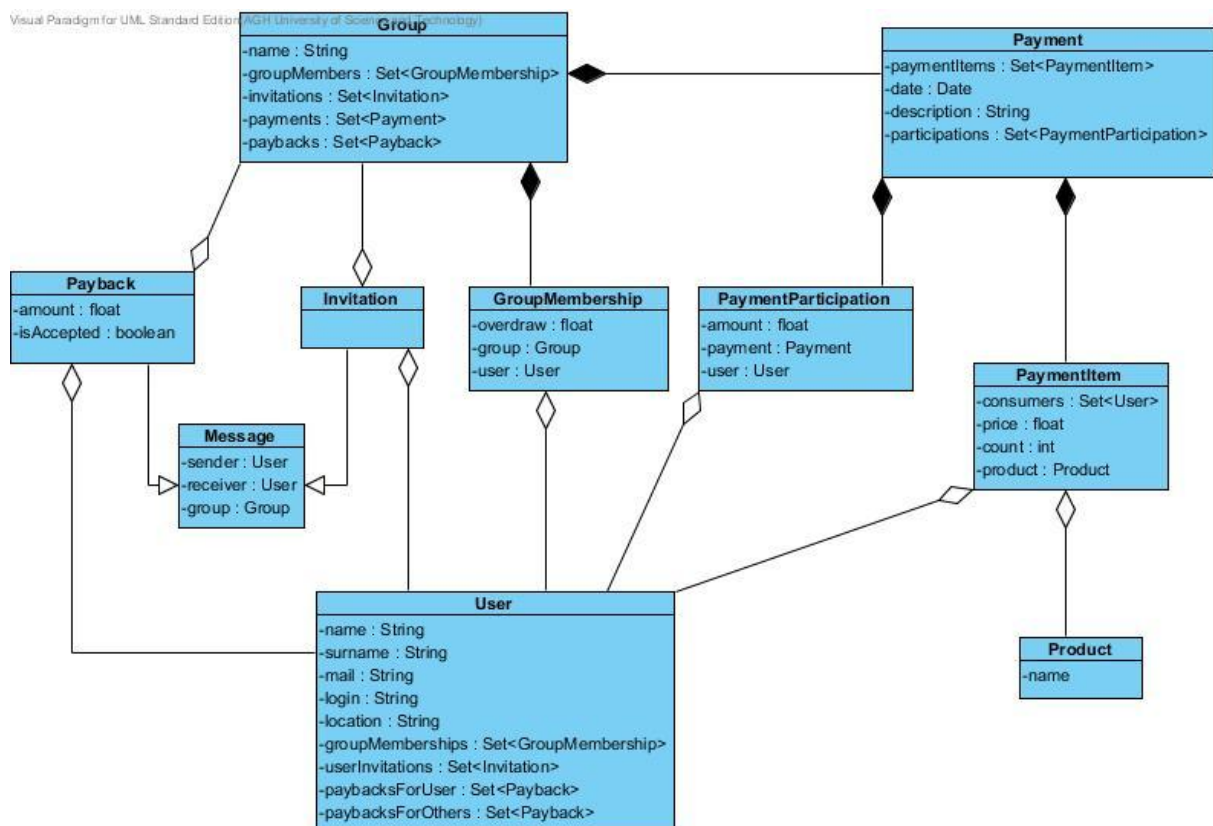
**PaymentParticipation** – obiekt klasy reprezentuje udział w spłacie rachunku. Odnotowuje, że użytkownik przekazał określoną kwotę na rzecz opłacenia danego rachunku.

**Message** – klasa bazowa dla komunikatów przekazywanych użytkownikowi. Posiada informację o nadawcy i odbiorcy komunikatu oraz grupie, w ramach której przesyłana jest informacja

**Payback** – klasa reprezentuje zgłoszenie spłaty długu wobec użytkownika. Pozwala określić wysokość spłaty oraz przechowuje informację o statusie zgłoszenia(zaakceptowane lub nie).

**Invitation** – klasa reprezentuje zaproszenie użytkownika do grupy, do której należy skierowane do innego użytkownika systemu.

## DIAGRAM



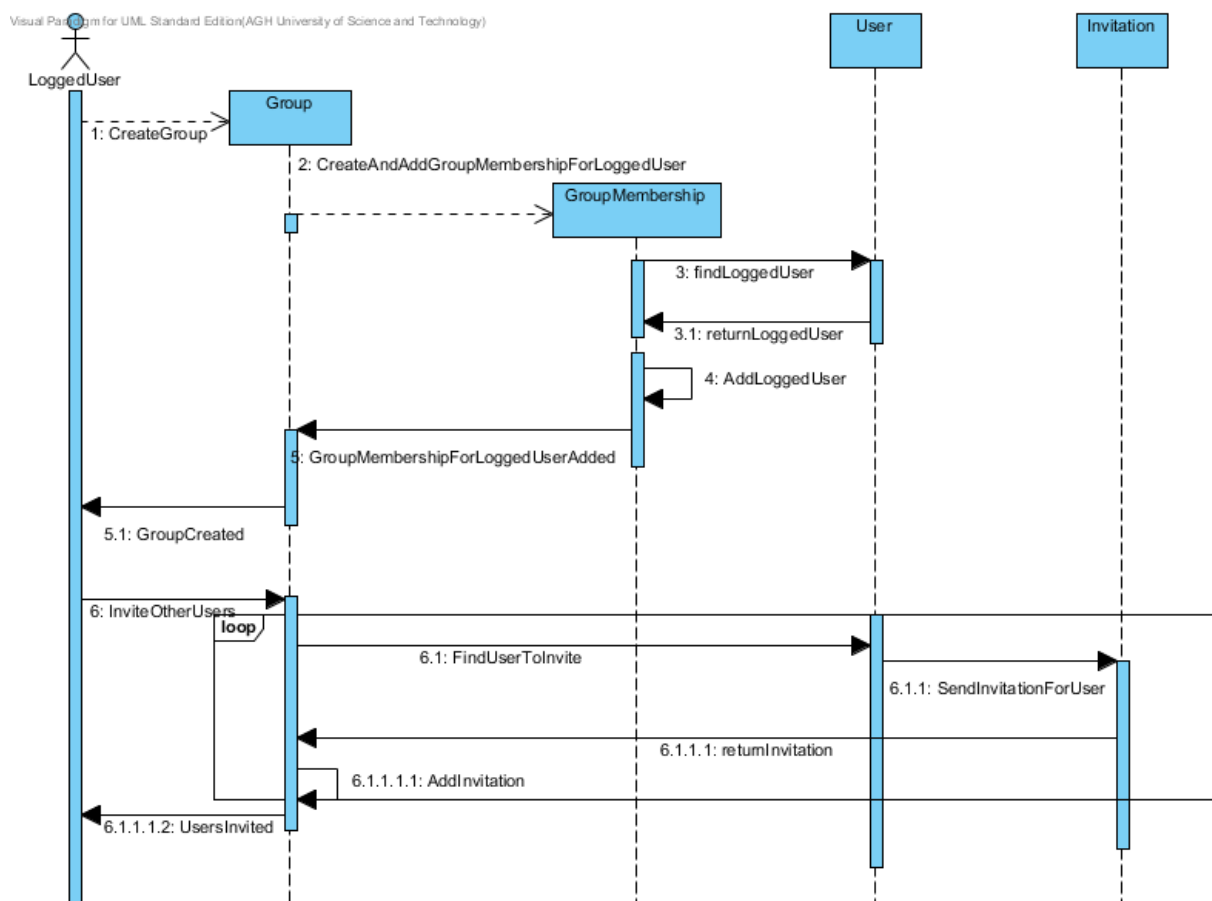
## DIAGRAMY SEKWENCJI

Stworzone zostało pięć diagramów sekwencji reprezentujących pięć najważniejszych zachowań systemu. W diagramach nie zostały użyte klasy operujące na danych (gdyż nie zostały one jeszcze stworzone) lecz klasy reprezentujące dane.

### DODANIE GRUPY

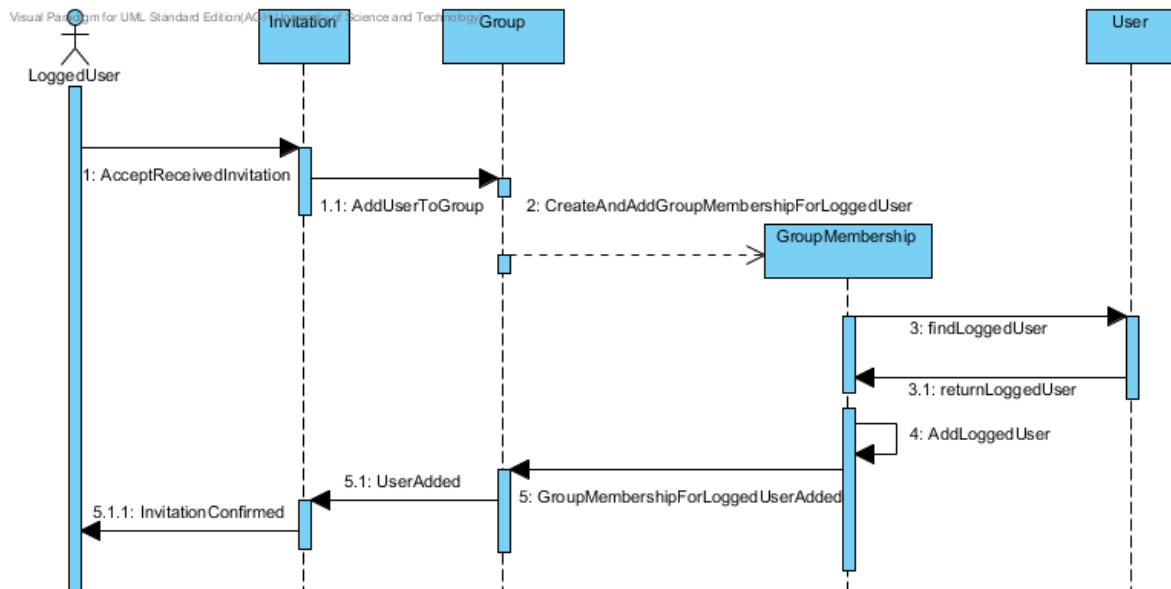
Diagram reprezentuje tworzenie nowej grupy użytkowników w systemie. Użytkownik, tworząc nową grupę dodaje do niej innych użytkowników. Użytkownik tworzący grupę automatycznie zostaje dodany jako członek grupy. Po zaproszeniu innych użytkowników do grupy, wysyłane są im zaproszenia.

Dla użytkownika dodawanego do grupy tworzony jest obiekt GroupMembership reprezentujący użytkownika będącego członkiem grupy. W obiekcie tym przetrzymywany są m. in. status pieniężny użytkownika względem grupy.



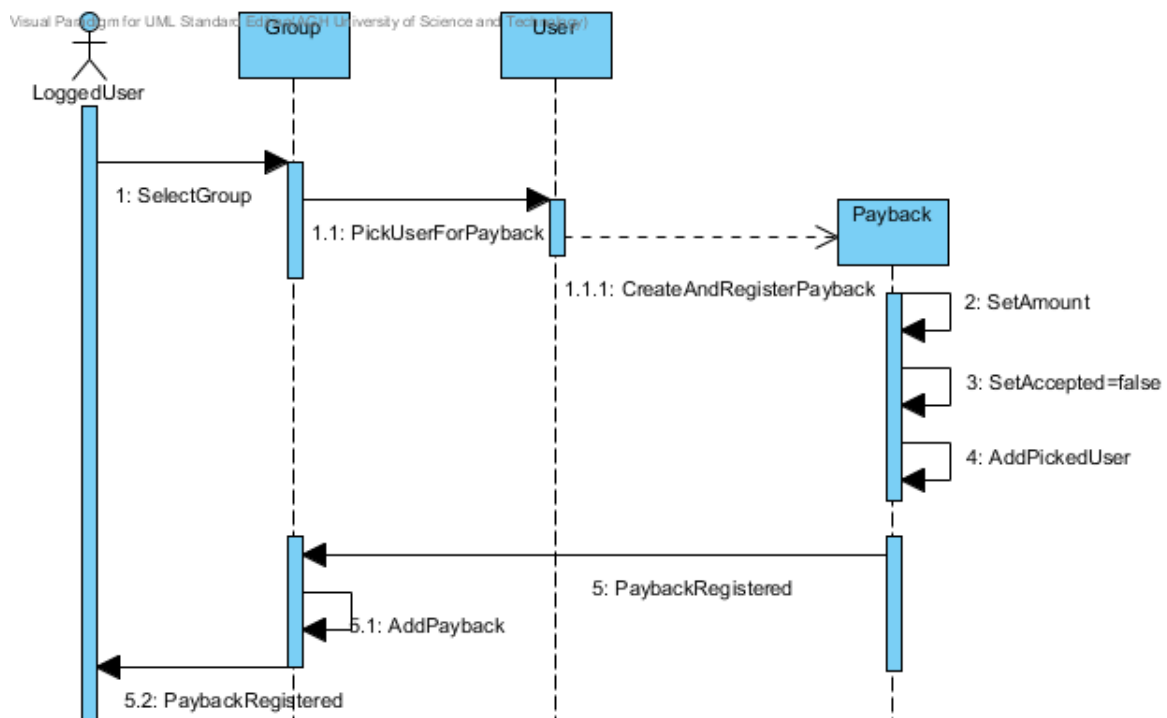
## ZATWIERDZENIE ZAPROSZENIA DO GRUPY

Użytkownik który dostanie zaproszenie do grupy, może stać się jej członkiem akceptując je. Dla takiego użytkownika tworzony jest obiekt GroupMember, który reprezentuje użytkownika będącego członkiem grupy.



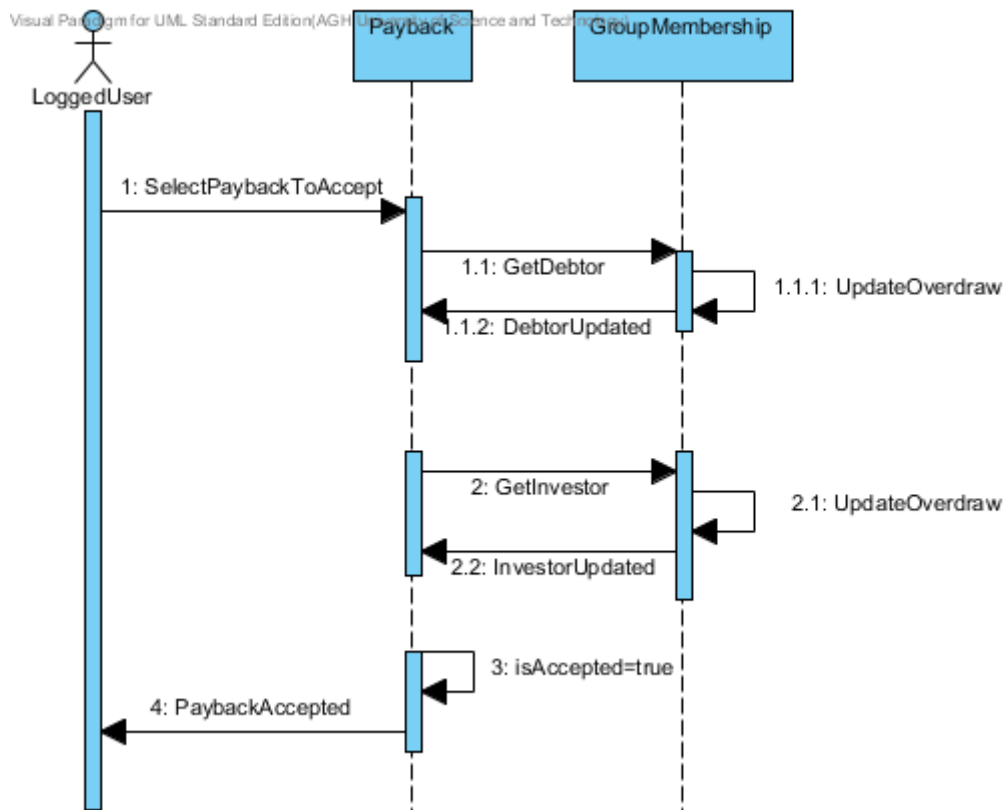
## REJESTRACJA ZWROTU DŁUGU

Aby zarejestrować zwrot długu, użytkownik wybiera grupę a następnie drugiego użytkownika, któremu chce zwrócić pieniądze. Tworzony jest Payback zawierający odniesienia do obydwu użytkowników, jednak kwota zostanie zarejestrowana po akceptacji go przez drugiego użytkownika.



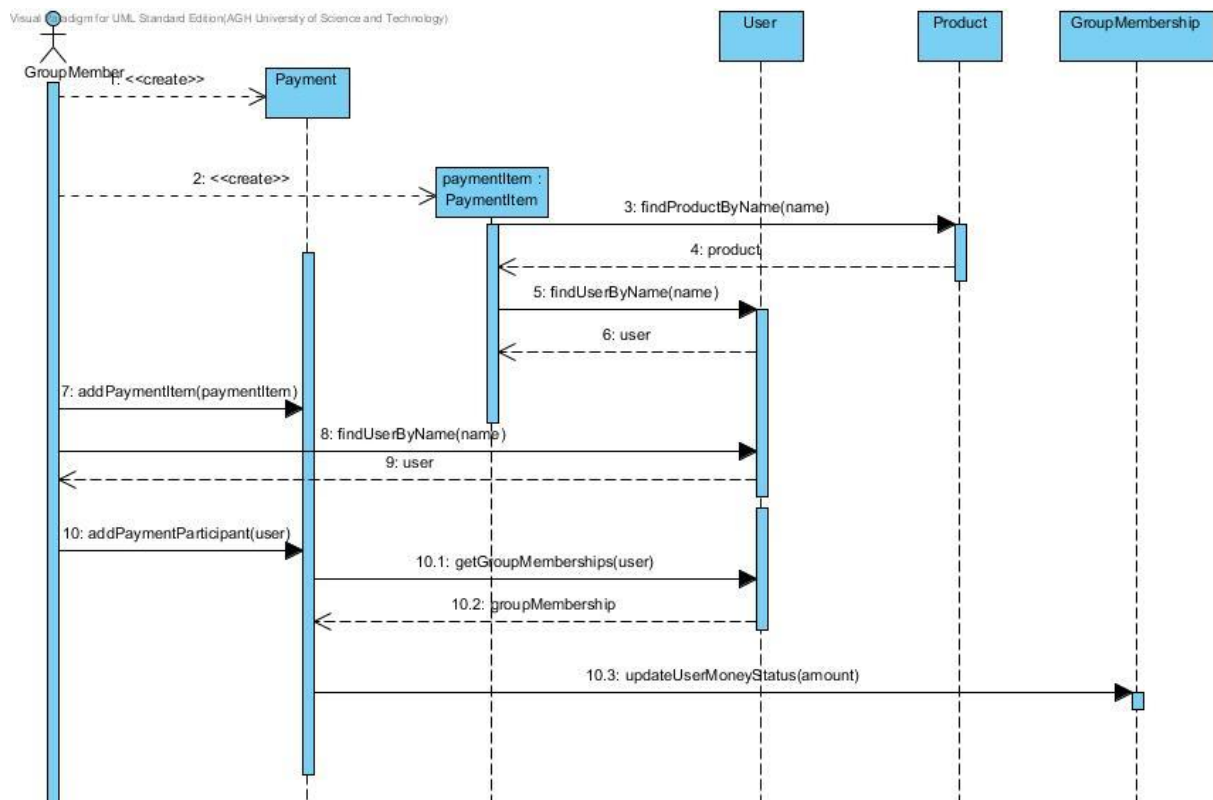
## AKCEPTACJA ZWROTU DŁUGU

Użytkownik, któremu zwrócone zostały pieniądze, może zaakceptować wstawiony na niego przez oddającego użytkownika zwrot. W takiej sytuacji dla obydwu użytkowników przeliczony zostaje ich status finansowy.



## DODANIE PŁATNOŚCI

Chcąc dodać nową płatność, użytkownik musi najpierw wybrać jedną z grup do których należy. Następnie, w ramach nowej płatności dodaje do niej nowe pozycje, wraz z produktem, którego dotyczyły, jego ceną, oraz użytkownikami, którzy go skonsumentali. Następnie dla danej pozycji, przypisany zostaje członek grupy, który za nią zapłacił. Lista takich pozycji zostaje zapisana jako płatność, a status pieniężny (przechowywany w GroupMembership) wszystkich użytkowników płacących za dane pozycje, zostaje uaktualniony.



## POSTĘP IMPLEMENTACJI

### WARSTWA DANYCH

W ramach drugiej fazy utworzony został komplet klas – encji, reprezentujących przechowywane dane. Do każdej takiej klasy, utworzone zostało DAO umożliwiające przeprowadzanie na nich podstawowych operacji CRUD.

Wszystkie DAO zostały przetestowane, do testów skonfigurowana została baza danych hsqldb tworzona w pamięci.

### WARSTWA SERWISOWA I PREZENTACJI

W ramach drugiej fazy, skonfigurowany został serwer tomcat na którym umieszczona została zbudowana aplikacja. Utworzony został pierwszy serwis oraz widok (w technologii JSF) umożliwiające tworzenie i usuwanie użytkowników.

W ten sposób, uruchomiony został pierwszy przykład, który wykorzystuje wszystkie warstwy oraz wszystkie technologie użyte w projekcie.

## KONFIGURACJA NOWYCH NARZĘDZI

### JENKINS

Serwer ciągłej integracji **Jenkins** został uruchomiony na uczelnianym serwerze ernie. Jest on dostępny pod adresem <http://ernie.icslab.agh.edu.pl:32123/> Zainstalowane zostały następujące wtyczki rozszerzające funkcjonalność aplikacji:

- Locale plugin
- GitHub API Plugin
- Git Plugin
- GitHub plugin
- Static Analysis Utilities
- PMD Plug-in
- Checkstyle Plug-in
- ChuckNorris Plugin
- Jenkins Cobertura Plugin

Na szczególną uwagę zasługują:

1. Narzędzia PMD oraz checkstyle służące do statycznej analizy kodu, mają na celu wymuszenie jednolitego standardu kodowania oraz wykrycie potencjalnych błędów.
2. Plugin Cobertura badający pokrycie kodu testami jednostkowymi.

### GITHUB – WIKI I BUG TRACKER

Serwis Github został przez nasz zespół wykorzystany nie tylko jako system kontroli wersji, ale także jako Wiki i Bug Tracker.

Na Wiki stworzone zostały strony zawierające dokumenty dotyczące kolejnych faz projektu, minetek ze spotkań zespołowych oraz How to's zawierających rozwiązania konkretnych problemów projektowych.

Na Bug Trakerze tworzone są milestone, w ramach których zespół rejestruje zadania którymi aktualnie się zajmuje oraz Bugi, które wychwyci u siebie lub u innych.