**UNIWERSYTET WARMIŃSKO MAZURSKI W OLSZTYNIE**

**WYDZIAŁ MATEMATYKI I INFORMATYKI**

**IGOR KAWCZYŃSKI**

**Kierunek: Informatyka**

**Aplikacja webowa wspomagająca przeprowadzanie kursów na żeglarstwo wodne i rezerwację rejsów – panel klienta.**

**Praca inżynierska**

**wykonana w Katedrze Metod Matematycznych Informatyki**

**pod kierunkiem dr Pawła Drozdy**

**Olsztyn 2024**

**UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN**

**FACULTY OF MATHMATICS AND COMPUTER SCIENCE**

**IGOR KAWCZYŃSKI**

**Field of study: Computer Science**

**Web application supporting the conduct of water sailing courses and booking cruises – customer panel**

**Engineering Thesis**

**Written in Katedra Metod Matematycznych Informatyki**

**under supervision of dr Paweł Drozda**

**Olsztyn 2024**

**Spis treści**

Spis treści

[Streszczenie 4](#_Toc156025315)

[Streszczenie w języku polskim 4](#_Toc156025316)

[Abstract 4](#_Toc156025317)

[1. Wstęp 5](#_Toc156025318)

[1.1 Wprowadzenie 5](#_Toc156025319)

[1.1 Cel i zakres pracy dyplomowej – dodać wzmiankę o panelu administratora pisanym przez Grzegorza Krycha 6](#_Toc156025320)

[1.2 Wykorzystane technologie – rozszerzyć opis niektórych technologii 6](#_Toc156025321)

[2. Specyfikacja systemu 11](#_Toc156025322)

[2.1 Opis analityczny 11](#_Toc156025323)

[2.2 Lista funkcjonalności (Przypadki użycia) 11](#_Toc156025324)

[2.3 Grupy użytkowników 12](#_Toc156025325)

[2.4 Diagramy przypadków użycia 13](#_Toc156025326)

[2.5 Scenariusze 16](#_Toc156025327)

[Diagram 1 - Przypadków Użycia - Ogólny 13](#_Toc156025364)

[Diagram 2 - Przypadków Użycia - PU Wyświetlenie informacji ze strony 14](#_Toc156025365)

[Diagram 3 - Przypadków Użycia - PU Zarządzanie rejsami 14](#_Toc156025366)

[Diagram 4 - Przypadków Użycia - PU Zarządzanie kursami 15](#_Toc156025367)

[Diagram 5 - Przypadków Użycia - PU E-Learning 15](#_Toc156025368)

[Diagram 6 - Przypadków Użycia - PU Zarządzanie profilem 16](#_Toc156025369)

Nie można odnaleźć pozycji dla spisu ilustracji.

# Streszczenie

## Streszczenie w języku polskim

Głównym zadaniem pracy dyplomowej jest usprawnienie oraz ułatwienie nauki i zapisów na kursy żeglarskie oraz rezerwacji na rejsy – wszystko przy pomocy stworzonej aplikacji webowej.

Strona umożliwia przegląd nadchodzących kursów i eventów oraz zapis na konkretny kurs dla zalogowanego klienta oraz rezerwację na rejs przy pomocy wypełnionego formularza. Użytkownik, który opłacił kurs ma możliwość korzystania z nauki (e-learning) w celu przygotowania się do egzaminu kończącego kurs. E-learning pozwala na wylosowanie pytania z konkretnego działu oraz sprawdzenie poprawnej odpowiedzi, wygenerowanie egzaminu próbnego, lub przejrzenie statystyk nauki.

Oprócz głównych założeń, aplikacja pozwala nam również przejrzeć zbiór informacji o firmie świadczącej usługi oraz odpowiedzi na najczęściej zadawane pytania w formie FAQ. Do dyspozycji na stronie znajduje się także cennik z ofertą żeglarską, gdzie widoczne są poszczególne terminy kursów lub rejsów wraz z zapisaną liczbą uczestników. Cenną funkcjonalnością strony jest również profil klienta, gdzie możemy edytować swoje dane dla konta bądź sprawdzić przypisane dla nas kursy lub rejsy i nimi zarządzać.

## Abstract

The primary objective of the dissertation is to enhance and simplify the process of learning, enrolling in sailing courses, and booking voyages, all through a specially developed web application.

This site allows users to view upcoming courses and events, and enables registered clients to sign up for a specific course or reserve a spot in an event via form. Users who have paid for a course gain access to e-learning resources, designed to help them prepare for the final examination. The e-learning system offers functionalities such as selecting questions from specific categories, verifying correct answers, generating mock exams, and check learning statistics.

In addition to its core features, the application also provides comprehensive information about the service-providing company and a FAQ section for commonly asked questions. The website also includes a detailed price list showcasing various course offerings, along with their scheduled dates and the current number of registered participants. Another valuable aspect of the site is the client profile section, where users can edit their account details, and manage their enrolled courses or booked voyages.

# Wstęp

## 1.1 Wprowadzenie

W erze cyfryzacji, wiele branż doświadcza rewolucji w sposobie zarządzania i organizacji pracy. Pomimo tego, wiele jednostek i instytucji w Polsce, zajmujących się organizacją rejsów i kursów żeglarskich, nadal opiera się na tradycyjnych metodach, przede wszystkim opierając swój biznes na papierowej kartce i komunikacji telefonicznej.

W kontekście żeglarstwa, sektorze z bogatą historią, wprowadzenie nowoczesnej technologii umożliwiającej szybkie i wygodne zapisy na kursy bądź rejsy żeglarskie w połączeniu z możliwością nauki do uzyskania konkretnego patentu żeglarskiego może znacząco wpłynąć na popularność tej branży i usprawnić przeprowadzanie tych procesów.

Niniejsza praca skupia się wytworzeniu aplikacji webowej, która usprawni wymienione procesy i zaoferuje szybki oraz wygodny interfejs, który otworzy nowe możliwości dla klientów oraz organizatorów, jednocześnie podnosząc standardy obsługi i efektywność zarządzania.

Ponadto, aplikacja dzięki wprowadzeniu interaktywnych elementów e-learningu dla kilkunastu rodzajów kursów jest w stanie znacząco wpłynąć na pozytywne wyniki uzyskiwane w egzaminach kończących, niezbędnych do uzyskania konkretnego patentu żeglarskiego. Implementując łatwy dostęp do praktycznych narzędzi naukowych, aplikacja ta ma potencjał, aby w większym stopniu zniwelować barierę wejścia dla nowych entuzjastów żeglarstwa i możliwie wpłynąć na popularność tego hobby lub profesji.

Przez zaprojektowanie i wdrożenie innowacyjnych rozwiązań, niniejsza praca ma na celu nie tylko przedstawienie funkcjonalnego narzędzia online, ale także zobrazowanie, w jaki sposób nowoczesna aplikacja webowa jest w stanie przyczynić się do rozwoju i ulepszenia konkretnych sektorów życia.

Ważnym aspektem aplikacji była ścisła współpraca i liczne konsultacje z właścicielem firmy zajmującej się wyżej wymienioną organizacją kursów i rejsów. To pozwoliło na większe zrozumienie wszelkich wymagań i oczekiwań zarówno potencjalnego klienta, jak i organizatora. Dzięki tej współpracy, projekt aplikacji był stale modyfikowany i dopasowywany tak, aby zapewnić, że finalny produkt będzie w stanie spełnić oczekiwania obu stron.

Realizacja tego projektu stanowiła cenną okazję do zastosowania i rozwinięcia zaawansowanych umiejętności w dziedzinie projektowania, implementacji i testowania oprogramowania. Praca ta wymagała dogłębnej analizy i wykorzystania nowoczesnych technik programowania oraz zwrócenia na aspekty takie jak wydajność, skalowalność i utrzymanie dobrej jakości kodu. Przez zastosowanie standardów clean code oraz wykorzystanie nowych narzędzi, projekt ten nie tylko przyczynił się do optymalizacji procesów zapisu na kursy bądź rezerwacji rejsów żeglarskich, ale także stanowił platformę do nagłego rozwoju zawodowego, przyczyniając się do pogłębienia wiedzy technicznej i praktycznych umiejętności w zakresie inżynierii oprogramowania.

## Cel i zakres pracy dyplomowej

Celem omawianej pracy dyplomowej jest przede wszystkim zaprojektowanie oraz wdrożenie oprogramowania aplikacji internetowej, która będzie w stanie wydajnie i bezbłędnie funkcjonować, dostarczając przy tym szereg możliwości w obrębie kursów i rejsów żeglarskich oraz przedstawienie procesu tworzenia omawianej aplikacji. W ramach projektu powstał także panel administracyjny pozwalający na zarządzanie bazą danych, którym zajął się Pan Grzegorz Krych, w niniejszej pracy priorytetem będzie jednak część kliencka, którą opracowałem ja, Igor Kawczyński. Kluczowym elementem jest omówienie funkcjonalności, jakie dostarcza aplikacja, w tym zarówno te bezpośrednio widoczne dla użytkownika, jak i te działające w tle, zapewniające płynność i efektywność systemu. Praca dostarcza również informacji o analitycznym aspekcie aplikacji, czyli omawia uwzględnione w niej scenariusze, przypadki użycia, diagramy, a także sposób implementacji, czyli przedstawienie wykorzystanych technologii i narzędzi, które zostały użyte do stworzenia oprogramowania, wraz z uzasadnieniem ich wyboru i opisem, w jaki sposób przyczyniły się one do realizacji założonych celów projektowych.

Zakres głównych zadań :

* Projekt relacyjnej bazy danych aplikacji
* Utworzenie dokumentu analitycznego uwzględniając podejście UML – przypadki użycia, scenariusze, aktorzy biznesowi, diagramy
* Utworzenie listy zadań z podziałem pracy na panel administratora i panel klienta
* Sporządzenie opisu funkcjonalności aplikacji
* Wykonanie logotypu aplikacji
* Zaprojektowanie szablonów widoków stron aplikacji
* Implementacja aplikacji od strony backendowej i frontendowej
* Przeprowadzenie testów jednostkowych

Warto dodatkowo nadmienić, że aplikacja powstała w sposób umożliwiający dalszy jej rozwój, który mógłby przyczynić się do wykorzystania jej w szerszym spektrum w zakresie branży żeglarskiej.

## 1.2 Wykorzystane technologie

Aplikacje webowe w ostatnich latach zyskały w dużym stopniu na popularności, przede wszystkim ze względu na łatwą i uniwersalną dostępność, gdzie możemy z poziomu przeglądarki na naszych urządzeniach mobilnych (smartfon, tablet) lub urządzeniach stacjonarnych (laptop, komputer osobisty) przejść na konkretną stronę i korzystać z jej funkcjonalności. Takie rozwiązanie ma zdecydowaną przewagę nad aplikacjami desktopowymi, które musimy instalować na naszym sprzęcie i używać przede wszystkim własnej mocy obliczeniowej do korzystania z ich funkcjonalności – w aplikacji webowej natomiast większe znaczenie ma sam serwer, który daną aplikację utrzymuje. Pełnoprawne aplikacje webowe składają się z trzech kluczowych dla nich części, frontendu, backendu oraz bazy danych.

Baza danych:

* Przechowuje – Baza danych przechowuje całe struktury danych, najczęściej w postaci tabel i odpowiadających im rekordów o danym typie
* Integruje – Baza danych odpowiada także za utrzymanie bazy spójnej i poprawnej, chodzi konkretnie o sprawdzanie ograniczeń spowodowanych kluczami obcymi lub specjalistycznymi typami danych
* Operuje – Baza danych jest w stanie wykonywać zapytania, użytkownik dzięki temu jest w stanie przeprowadzać operacje CRUD (Create, Read, Update, Delete), które są kluczowe do przetwarzania wszelkich informacji

Backend:

* Przetwarzanie wszelkich żądań użytkownika – Backend odbiera wysłane żądania HTTP przesłane przez frontend lub innych klientów i przetwarza je według napisanej logiki systemu
* Zarządzanie bazą danych – Backend jest odpowiedzialny za komunikację pomiędzy logiką systemu, a bazą danych, dzięki czemu jesteśmy w stanie tworzyć, pobierać, aktualizować oraz usuwać dane w niej przechowywane. To zazwyczaj zapytania napisane w języku SQL lub jego pochodne
* Logikę biznesową – Backend obejmuje całą logikę biznesową aplikacji internetowej, która nakreśla jak cały system ma działać. To wszelkie reguły, algorytmy i wyliczenia niezbędne do korzystania z funkcjonalności aplikacji
* Udostępnienie API (interfejsu programowania aplikacji) – Backend umożliwia frontendowi lub innym klientom komunikację z logiką aplikacji, najczęściej w formie REST API, GRAPHQL lub WebSocket
* Bezpieczeństwo – Backend powinien obsłużyć wszelkie sytuacje zagrażające bezpieczeństwu danych, między innymi są to ataki typu Cross Site Request Forgery, Cross Site Scripting czy SQL Injection
* Wydajność – Backend dba o szybkość działania aplikacji i tempo w jakim przetwarzane są wszelkie informacje w jej obrębie

Frontend:

* Wygląd i układ interfejsu strony – Frontend odpowiada za generowanie widoków, czyli między innymi pasek nawigacyjny, nagłówki, tekst, przyciski, obrazy i ich ostylowanie, najlepiej w sposób responsywny, czyli taki, który obsługuje ekrany o różnych proporcjach i rozmiarach, chodzi przede wszystkim o urządzenia mobilne (mniejsze ekrany), a także komputery stacjonarne i ich większe wyświetlacze
* Interakcje z aplikacją – Frontend zapewnia możliwość interakcji klienta z całą aplikacją, mianowicie obsługą myszki i klawiatury
* Wysyłanie żądań do backendu – Frontend odpowiada także za wysyłanie wyżej wymienionych żądań do serwera backendowego, który obsługuje je dzięki swojej logice biznesowej

Istnieją oczywiście odstępstwa od tych reguł, bywa nawet tak, że to na frontendzie znajduje się lwia część walidacji, którą także można nazywać logiką systemu. Wszystko jest zależne od rodzaju i przeznaczenia konkretnej aplikacji internetowej. W przypadku aplikacji stworzonej w celach pracy dyplomowej, zostały zachowane jednak główne konwencje integracji frontendu z backendem.

Do zbudowania wszystkich trzech komponentów aplikacji użyte zostały technologie takie jak:

Baza danych:

* PostgreSQL - Jako język zarządzania bazą danych wybrano PostgreSQL ze względu na jego wydajność, niezawodność, wsparcie dla dużych zbiorów danych i wygodną aplikację w postaci PgAdmin – wersja 15
* Flyway – Narzędzie do zarządzania migracjami baz danych. Pozwala na uspójnienie skryptów i ich automatyczne wdrażanie. Szczególnie przydatny w przypadku implementacji kodu dla większej liczby osób – stąd został użyty w aplikacji wymienionej w pracy dyplomowej, aby skrypty z panelu administratora nie konfliktowały z tymi od strony klienta i na odwrót – wersja 9

Technologie Backendowe:

* Java – Główny język programistyczny wykorzystany w aplikacji. Wysokopoziomowy, kompilowany, silnie obiektowy i z restrykcyjną kontrolą typów, gdzie składnia wzorowana jest na C++. Java razem z Pythonem jest najczęściej używanym językiem do tworzenia nowoczesnych systemów aplikacji webowych – stąd też wybór języka nie był szeroki i padł właśnie na Javę, która względem Pythona charakteryzuje się kompilacją do kodu bajtowego, co korzystnie wpływa na wydajność i wymienioną wcześniej silną kontrolą typów, co wpływa równie korzystnie na intuicyjność pisanego kodu. Java użytkowana jest także do tworzenia oprogramowania desktopowego czy gier. Ponadto, system operacyjny Android działa w środowisku Javy, co pozwala na intuicyjne tworzenie aplikacji lub gier na Androida. Jest preferowanym językiem w wielu korporacjach i dużych firmach. Często wykorzystywana w systemach bankowych i ubezpieczeniowych ze względu na oferowane przez ten język solidne rozwiązania w zakresie zabezpieczeń. Wersja Javy użyta w projekcie to 17
* Spring – Framework[[1]](#footnote-1) dla języka Javy, stworzony z myślą o implementowaniu aplikacji wysokiej jakości, które są bezpieczne, skalowalne i stosunkowo łatwe do zarządzania. Spring oferuje mnóstwo dodatków i narzędzi, które ułatwiają rozwijanie wielu aspektów aplikacji, takich jak obsługa danych czy ich zabezpieczenie. Framework ten oferuje także bardzo dobrą praktykę programowania jaką jest wstrzykiwanie zależności, które szeroko zostało użyte w wymienionej pracy inżynierskiej
* Spring Boot – Projekt Springa, którego główną ideą jest zminimalizowanie implementacji licznej, czasami zbędnej konfiguracji, co pozwala na dużo szybszy start dla stworzenia aplikacji względem klasycznego Springa. Oprócz zmian względem konfigracji, Spring Boot dostarcza także serwer aplikacjny Tomcat, który obsługuje nasze aplikacje oraz masę innych udogodnień, które pozytywnie wpływają na możliwości i szybkość rozwoju kodu backendowego. Wersja Springa użyta w projekcie to 3.1.5
* Spring Security – Wysoce konfigurowalny framework służący do zabezpieczenia aplikacji. Wspiera nowoczesne protokoły i standardy zabezpieczeń. Użyty przede wszystkim do zabezpieczenia wszelkich żądań http poprzez filtrowanie i autoryzacji z uwierzytelnieniem. Wykorzystany w wersji 6.1.5
* Java Persistence API – Specyfikacja dostarczająca Javie sposób mapowania obiektów na rekordy w bazie danych (ORM – Object Relational Mapping), co pozwala na tworzenie obiektów w kodzie, aby następnie przekonwertować je na rekordy pochodzące z konkretnych encji. JPA definiuje zestaw interfejsów, reguł i adnotacji, które można w łatwy sposób wykorzystać, dzięki czemu programista poświęca mniej czasu na komunikację pomiędzy kodem backendowym, a bazą danych i może skupić się na logice biznesowej aplikacji. Wersja JPA użyta w projekcie to 3.1.0
* Hibernate – Framework implementujący JPA i dostarczający własne zapytania typu HQL (Hibernate Query Language), a także własny sposób zarządzania sesją i pamięcią cache. Wersja użyta w projekcie to 6.2.13
* JUnit – Framework przeznaczony do automatycznego testowania jednostkowego – użyty w celu przetestowania konkretnych części aplikacji. Wersja to 5.9.3
* Mockito – Framework umożliwiający tworzenie symulacji obiektów, czyli tak zwanych „mocków”. Takie sztuczne obiekty pozwalają na odtworzenie prawdziwych obiektów i tym samym rzeczywistych scenariuszy, które mogą wystąpić w samej aplikacji. Użyto w wersji 5.3.1
* Jakarta Validation API – Framework zapewniający mechanizm automatycznego walidowania obiektów, oparty na adnotacjach i modyfikowalnych regułach. Wersja frameworka to 3.0.2
* Lombok – Biblioteka automatyzująca pisanie schematycznego kodu w Javie i tym samym narzędzie redukujące tak zwany „boilerplate code”. Wersja to 1.18.30
* JSON Web Token (JWT) – Sposób bezpiecznego przekazywania informacji pomiędzy dwoma stronami w formie obiektu JSON. Użyty w projekcie do zabezpieczenie autoryzacji użytkownika w wersji 0.11.5

Technologie Frontendowe:

* HTML – Fundament każdej strony internetowej, język znaczników wykorzystywany do ich tworzenia i ustrukturyzowania. Użyty w wersji 5
* CSS – Arkusz styli, który służy do kontrolowania wyglądu tworzonych widoków na stronie internetowej. Użyty w wersji 3
* JavaScript – Język skryptowy tworzący dynamikę i interaktywność aplikacji. Użyty w wersji ES2022
* TypeScript – Język rozszerzający JavaScript o statyczne typowanie i lepsze wsparcie dla środowisk programistycznych. Użyty w wersji 4.7.5
* NPM – System zarządzania pakietami dla języka JavaScript, pozwala na łatwe dodanie kolejnych zależności do frontendu. Użyta wersja to 9.6.7
* Angular – Framework opracowany przez Google używany do implementowania dynamicznych aplikacji internetowych. Oferuje bogaty zestaw funkcji począwszy od zabezpieczonych przekierowań do dwustronnego wiązania danych. Obok frameworka React najpopularniejszy w kontekście tworzenia aplikacji webowych. Wybrany ze względu na dużą intuicyjność i większe podobieństwo do struktury projektu napisanego w Spring Boot. Użyty w wersji 16
* Angular CLI – Narzędzie rozszerzające framework Angular o linię poleceń pozwalającą na szybkie tworzenie, modyfikację i testowanie komponentów frontendowych. Używana wersja w projekcie to 16.2.10
* Jasmine – Framework umożliwiający pisanie testów jednostkowych po stronie frontendowej w czytelny, jasny sposób. Użyty w wersji 4.6.0
* Bootstrap – Framework do tworzenia nowoczesnych i skalowalnych styli stron internetowych. Znacząco ułatwia projektowanie interfejsu użytkownika dzięki gotowym komponentom i klasom stylowym. Używany w wersji 5.3.2
* PrimeNG – Kolekcja gotowych komponentów dostarczających interfejs i funkcjonalności. Wersja to 16.1
* Angular Material – Podobnie jak PrimeNG, narzędzie dostarczające komponenty, które umożliwiają przyspieszenie tworzenia widoków na stronie. Dedykowana kolekcja styli dla frameworka Angular, również stworzona przez Google. Używana wersja 16.2.12

Inne:

* Trello – Aplikacja pomocna w zarządzaniu projektem zespołowym poprzez tworzenie listy tablic zawierających zadania do wykonania. Użyta w celu podziału prac dla panelu administratora i panelu klienta i dla śledzenia wszelkich zmian w projekcie aplikacji
* Figma – Strona internetowa, w której zaprojektowany został UI/UX[[2]](#footnote-2) dla całej aplikacji

# 2. Specyfikacja systemu

## 2.1 Opis analityczny

Celem opracowanej aplikacji webowej było przede wszystkim usprawnienie procesu zapisów na kursy żeglarskie oraz rezerwacji na rejsy. System ten, zaprojektowany z myślą o intuicyjnym użytkowaniu, umożliwia użytkownikom łatwe przeglądanie dostępnych kursów i rejsów, wraz z ich terminami i cenami, co znacząco ułatwia proces wyboru odpowiedniej oferty. Sam wybór oferty polega na wybraniu konkretnego rodzaju kursu bądź rejsu, a następnie terminu i miejsca wydarzenia. Warto dodatkowo podkreślić, że zapis na kursy odbywa się w kontekście konta klienta, czyli konkretny kurs przypisany jest do konkretnego użytkownika, rejsy natomiast rezerwujemy na podany w formularzu e-mail, co pozwala na rezerwację bez posiadania konta użytkownika.

Kluczowym aspektem aplikacji jest również wprowadzenie modułu e-learningowego, który pozwala użytkownikom na zdalne przygotowanie się do egzaminu na patent żeglarski. Ten segment aplikacji zawiera funkcje umożliwiające dostęp do losowo generowanych pytań z konkretnego działu żeglarskiego, generowanie egzaminów próbnych oraz analizę postępów w nauce, co stanowi istotne wsparcie w procesie edukacyjnym.

Aplikacja została także wyposażona w funkcje logowania i rejestracji, co zapewnia bezpieczne i personalizowane środowisko dla każdego użytkownika. Dodatkowo, z myślą o wsparciu użytkowników i ułatwieniu dostępu do ważnych informacji, system zawiera sekcję FAQ, odpowiadającą na najczęściej zadawane pytania związane z rejsami, kursami oraz procesem nauczania żeglarskiego, a także sekcję kontaktową, która może być wykorzystywana przez firmę świadczącą usługi żeglarskie do komunikacji z potencjalnymi klientami. Oprócz tego, aplikacja oferuje możliwość zarządzania profilem użytkownika, co obejmuje edycję danych osobowych oraz przeglądanie statystyk aktywności w ramach systemu.

## 2.2 Lista funkcjonalności (Przypadki użycia)

* Rejestracja konta użytkownika
* Logowanie
* Wylogowanie
* Wyświetlenie statystyk profilu
* Edycja profilu
* Wyświetlenie dostępnych kursów
* Wyświetlenie dostępnych rejsów
* Nauka ogólna (wygenerowanie losowego pytania)
* Wygenerowanie egzaminu próbnego
* Rozwiązanie egzaminu próbnego
* Podgląd zbliżających się kursów użytkownika
* Wycofanie się z kursu
* Podgląd zbliżających się rejsów użytkownika
* Wycofanie się z rejsu
* Wyświetlenie informacji kontaktowych
* Wyświetlenie najczęściej zadawanych pytań o kursy i rejsy

## 2.3 Grupy użytkowników

W ramach opracowanego systemu, zdefiniowane zostały trzy główne grupy użytkowników, każda z różnym poziomem dostępu do głównych funkcjonalności systemu. Hierarchia dostępu i uprawnień jest kluczowym elementem zapewniającym efektywność i bezpieczeństwo operacji w systemie.

|  |  |  |  |
| --- | --- | --- | --- |
| Dostępy | Gość[[3]](#footnote-3) | Użytkownik zalogowany | Administrator |
| Strona główna | TAK | TAK | TAK |
| Przegląd kursów i rejsów | TAK | TAK | TAK |
| Zapis na kursy | NIE | TAK | TAK |
| Rezerwacja rejsów | TAK | TAK | TAK |
| Zarządzanie kursami i rejsami | NIE | TAK | TAK |
| E-learning | NIE | TAK | TAK |
| Zarządzanie profilem | NIE | TAK | TAK |
| Zakładka kontaktowa | TAK | TAK | TAK |
| Najczęściej zadawane pytania | TAK | TAK | TAK |
| Panel administracyjny | NIE | NIE | TAK |

## 2.4 Diagramy przypadków użycia

Diagram przypadków użycia (DPU), jest graficznym przedstawieniem funkcji systemu widzianych z perspektywy używających go osób. Stosuje się go, aby zilustrować interakcje i zależności pomiędzy użytkownikami (zwanymi dalej aktorami), a systemem, które nazywane są przypadkami użycia. Jest to narzędzie wykorzystywane głównie w analizie i projektowaniu systemów w celu określenia wymagań biznesowych. Poniższy diagram przedstawia ogólny zarys funkcjonalności systemu, które zostały dalej rozbite na konkretne przypadki użycia (PU).

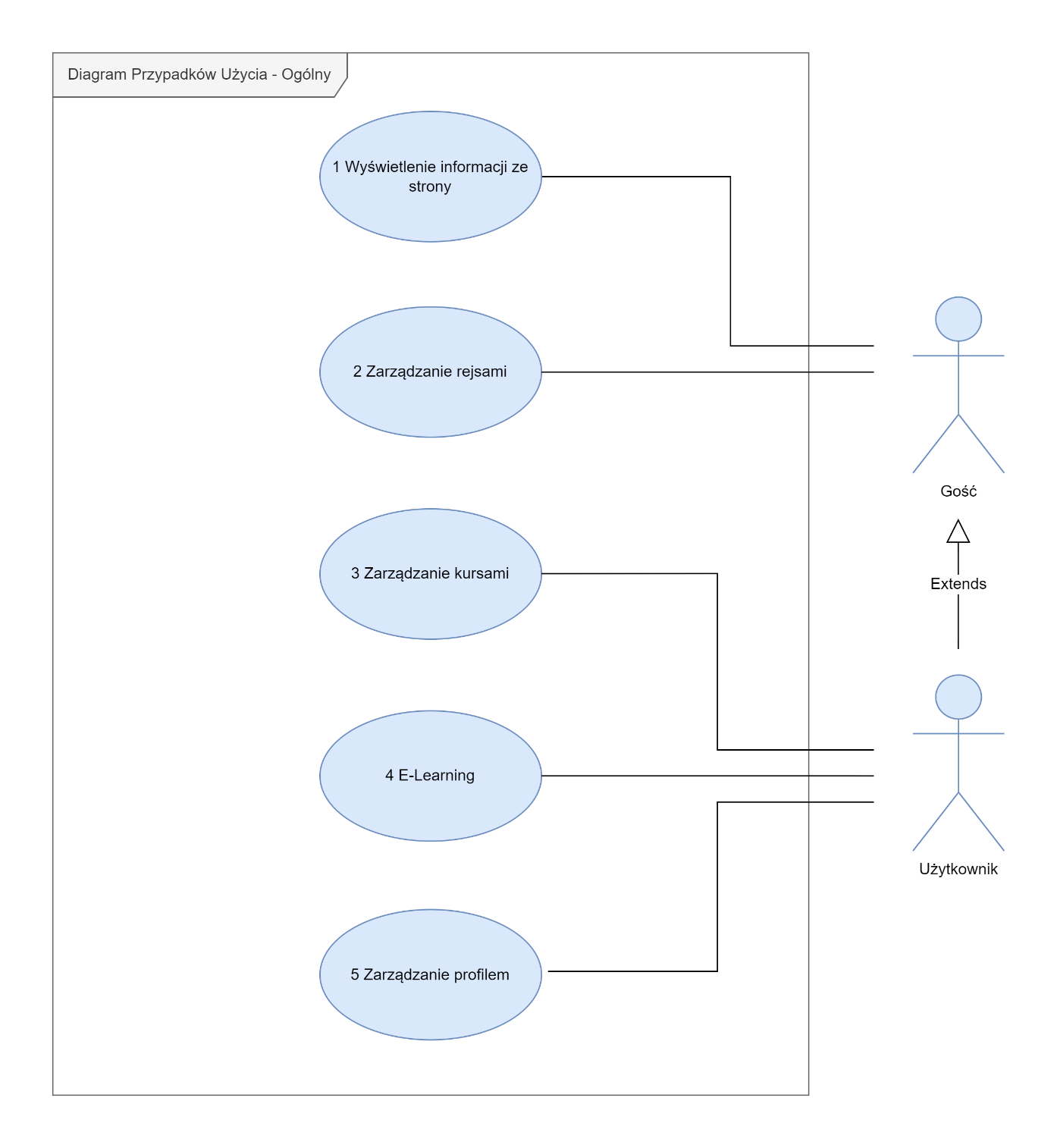


Diagram 1 - Przypadków Użycia - Ogólny

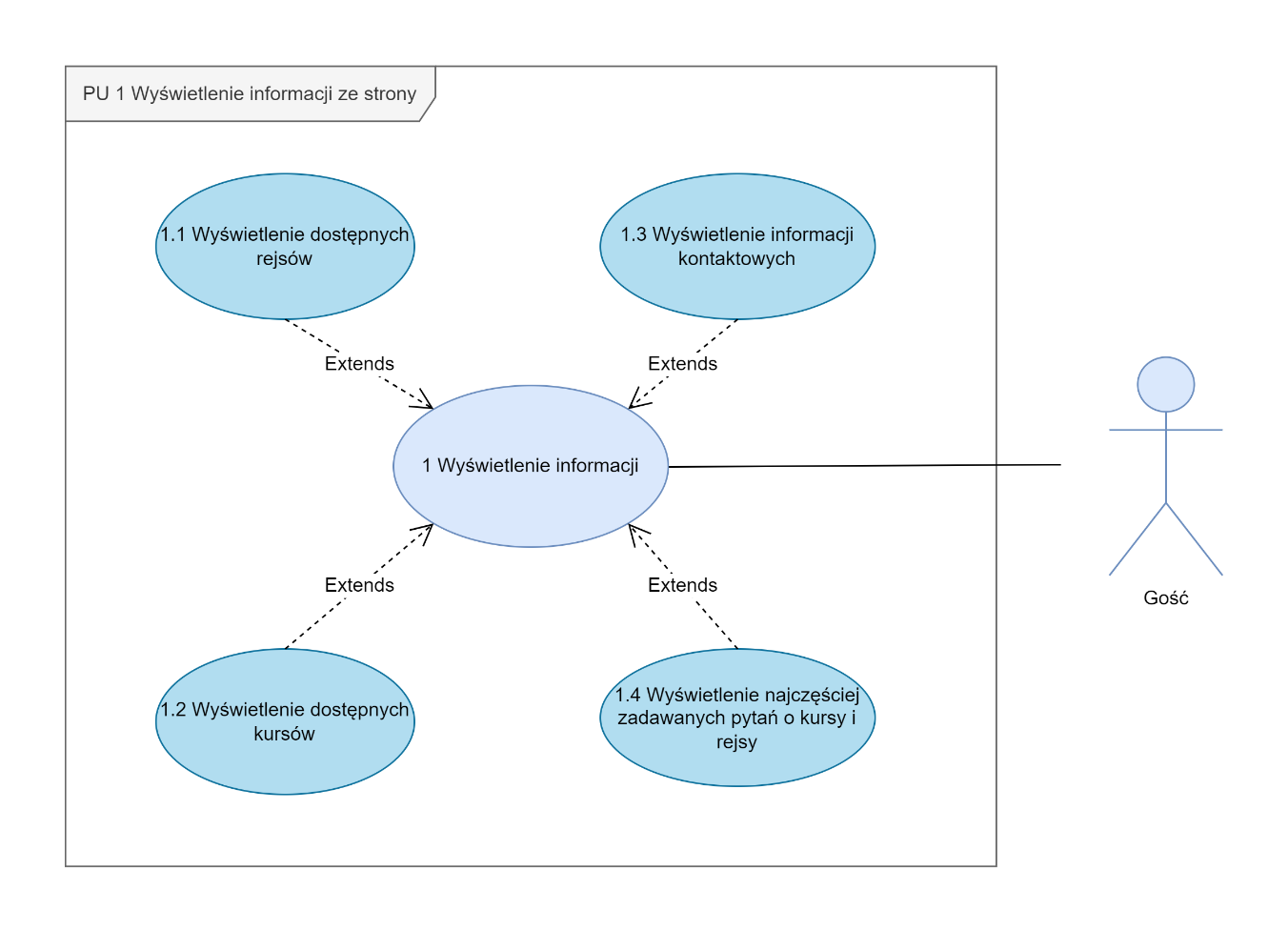
****

Diagram 2 - Przypadków Użycia - PU Wyświetlenie informacji ze strony

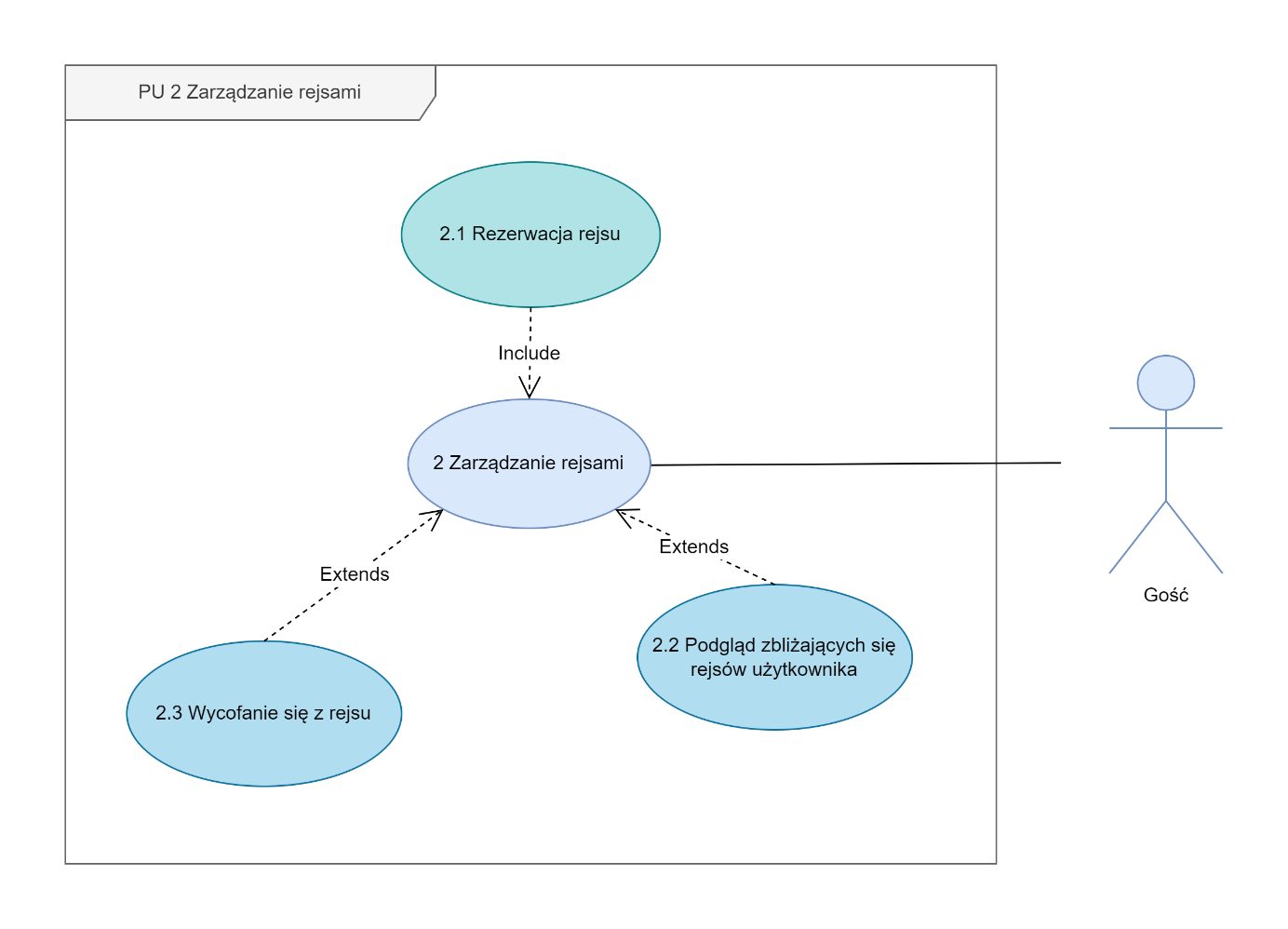
****

Diagram 3 - Przypadków Użycia - PU Zarządzanie rejsami

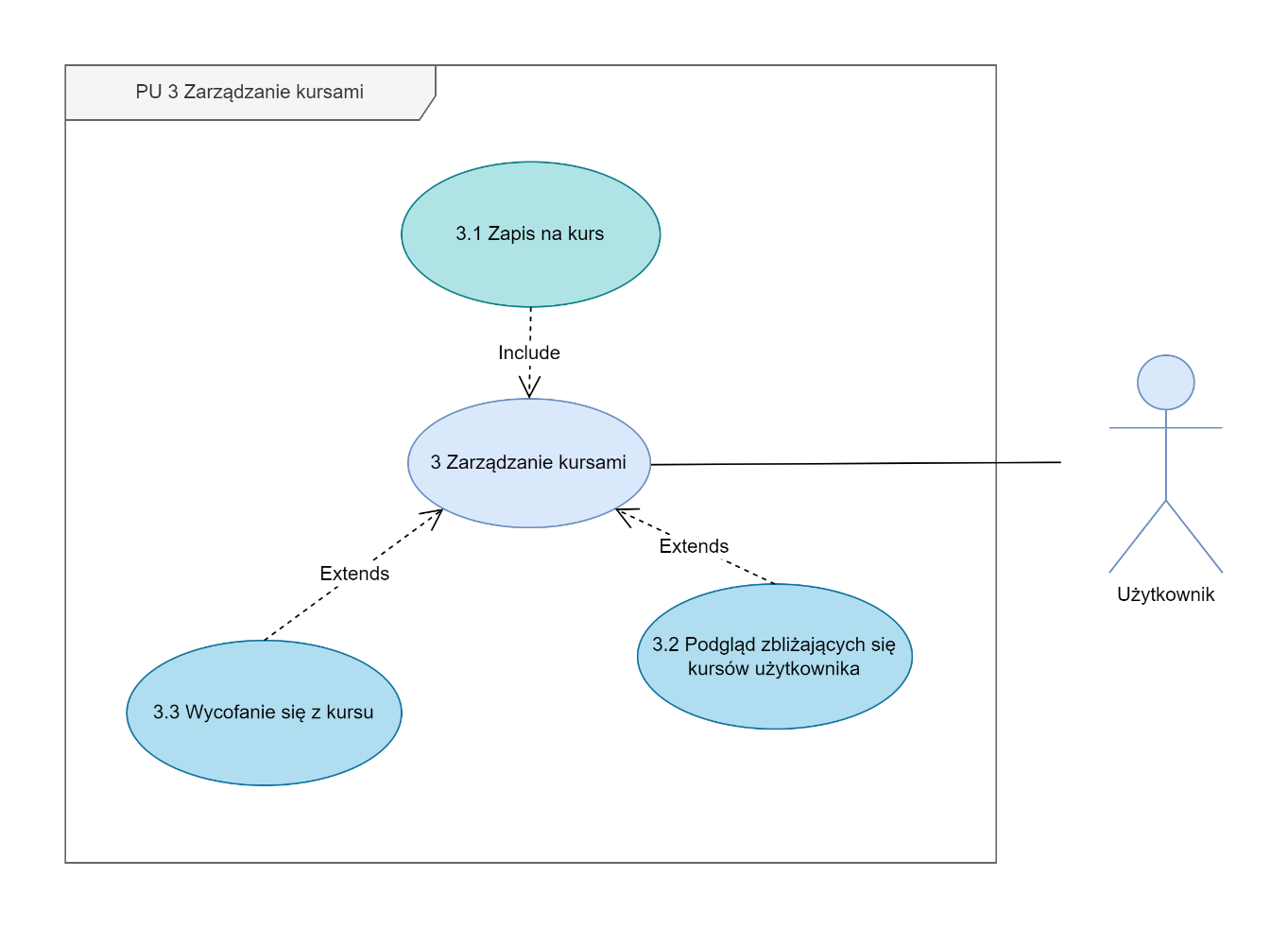


Diagram 4 - Przypadków Użycia - PU Zarządzanie kursami

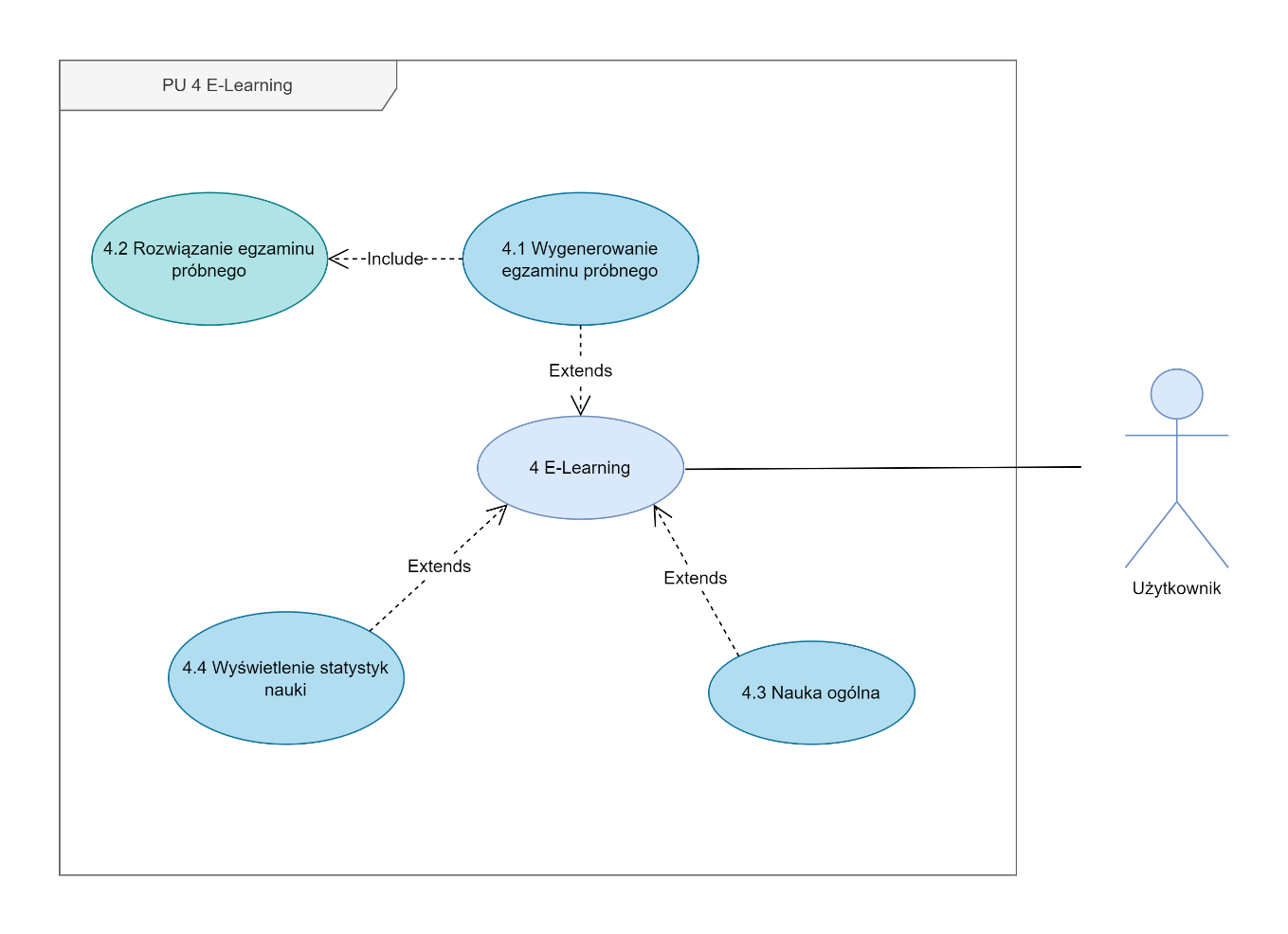


Diagram 5 - Przypadków Użycia - PU E-Learning

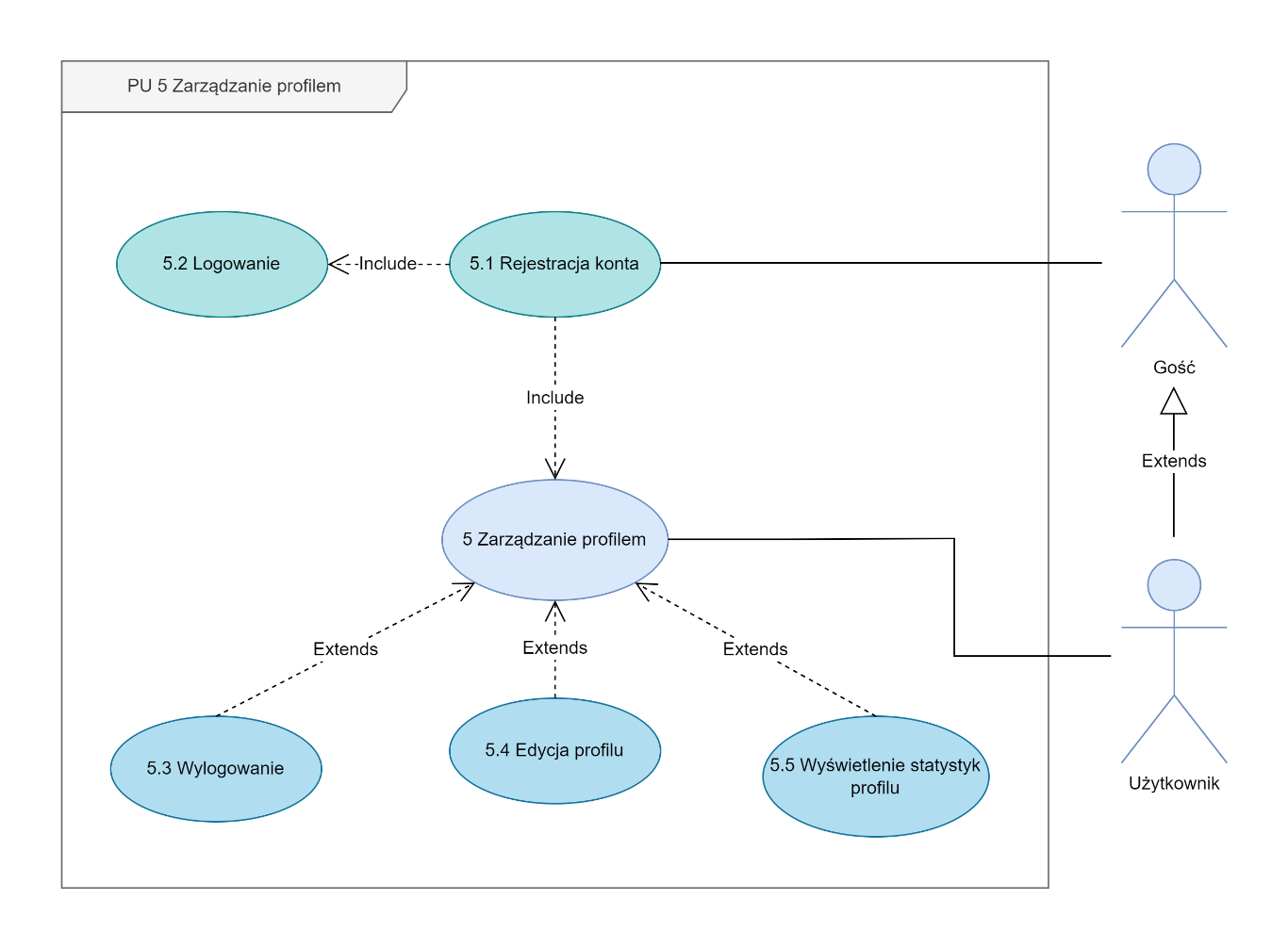
****

Diagram 6 - Przypadków Użycia - PU Zarządzanie profilem

## 2.5 Scenariusze

Scenariusze są istotnym elementem podejścia analitycznego, ze względu na:

* **Projektowanie i Prototypowanie**: Służą jako podstawa do projektowania interfejsów użytkownika i prototypowania systemu. Przykładowe ścieżki użytkowników w scenariuszach mogą być bezpośrednio użyte do tworzenia prototypów i mockupów[[4]](#footnote-4).
* **Definiowanie Wymagań:** Pozwalają dokładnie zdefiniować i zrozumieć wymagania użytkowników względem systemu. Przez to, że scenariusze skupiają się na interakcjach użytkownika z systemem, pomagają określić, jakie funkcjonalności są potrzebne.
* **Planowanie Rozwoju**: Scenariusze pomagają w planowaniu procesu rozwoju oprogramowania, umożliwiając programistom skupienie się na konkretnych funkcjach i zapewnienie, że wszystkie scenariusze użytkowania są obsługiwane.
* **Testowanie:** Scenariusze stanowią podstawę dla przypadków testowych w procesie zapewniania jakości. Ewentualni testerzy mogą używać scenariuszy do weryfikacji, czy wszystkie ścieżki użytkownika są prawidłowo obsługiwane przez system. Szczególnie przydatne dla testów End to End.

Przypadek użycia: 1.1 Wyświetlenie dostępnych rejsów

|  |  |
| --- | --- |
| Nazwa | 1.1 Wyświetlenie dostępnych rejsów |
| Aktor | Gość |
| Krótki opis | Wyświetlenie konkretnego rodzaju rejsu wraz ze szczegółowymi informacjami, cennikiem i dostępnymi terminami |
| Warunki wstępne | Brak |
| Warunki końcowe | System wyświetla rejs wraz z dostępnymi terminami |
| Główny przepływ wydarzeń | 1. Gość przechodzi na stronę główną aplikacji 2. System prezentuje panel powitalny   3a. Gość wybiera przycisk ‘Eventy’   1. System prezentuje listę dostępnych rodzajów rejsów 2. Gość wybiera interesujący go rodzaj rejsu i przechodzi do jego szczegółów poprzez przycisk ‘Zobacz więcej’ 3. System prezentuje szczegóły wybranego rodzaju rejsu, cennik oraz dostępne terminy |
| Alternatywny przepływ wydarzeń | 3b. Gość przechodzi na zakładkę ‘Oferta’ i wybiera ‘Eventy’ |

Przypadek użycia: 1.2 Wyświetlenie dostępnych kursów

|  |  |
| --- | --- |
| Nazwa | 1.2 Wyświetlenie dostępnych kursów |
| Aktor | Gość |
| Krótki opis | Wyświetlenie konkretnego rodzaju kursu wraz ze szczegółowymi informacjami, cennikiem i dostępnymi terminami |
| Warunki wstępne | Brak |
| Warunki końcowe | System wyświetla kurs |
| Główny przepływ wydarzeń | 1. Gość przechodzi na stronę główną aplikacji 2. System prezentuje panel powitalny   3a. Gość wybiera przycisk ‘Kursy’   1. System prezentuje listę dostępnych rodzajów kursów 2. Gość wybiera interesujący go rodzaj kursu i przechodzi do jego szczegółów poprzez przycisk ‘Zobacz więcej’ 3. System prezentuje szczegóły wybranego rodzaju kursu, cennik oraz dostępne terminy |
| Alternatywny przepływ wydarzeń | 3b. Gość przechodzi na zakładkę ‘Oferta’ i wybiera ‘Kursy’ |

Przypadek użycia: 1.3 Wyświetlenie Informacji kontaktowych

|  |  |
| --- | --- |
| Nazwa | 1.3 Wyświetlenie informacji kontaktowych |
| Aktor | Gość |
| Krótki opis | Wyświetlenie informacji kontaktowych potencjalnego usługodawcy |
| Warunki wstępne | Brak |
| Warunki końcowe | System wyświetla numer telefonu oraz email usługodawcy |
| Główny przepływ wydarzeń | 1. Gość przechodzi na stronę główną aplikacji 2. System prezentuje panel powitalny 3. Gość przechodzi na zakładkę ‘O firmie’ i wybiera ‘Kontakt’ 4. System prezentuje informacje kontaktowe |
| Alternatywny przepływ wydarzeń | Brak |

Przypadek użycia: 1.4 Wyświetlenie najczęściej zadawanych pytań o kursy i rejsy

|  |  |
| --- | --- |
| Nazwa | 1.4 Wyświetlenie najczęściej zadawanych pytań o kursy i rejsy |
| Aktor | Gość |
| Krótki opis | Wyświetlenie rozwijalnej listy wszystkich pytań i odpowiedzi na temat kursów bądź rejsów |
| Warunki wstępne | Brak |
| Warunki końcowe | System wyświetla listę pytań oraz odpowiedzi |
| Główny przepływ wydarzeń | 1. Gość przechodzi na stronę główną aplikacji 2. System prezentuje panel powitalny 3. Gość przechodzi na zakładkę ‘O firmie’ i wybiera ‘Najczęściej zadawane pytania’ 4. System prezentuje listę pytań 5. Gość wybiera interesujące go pytanie i klika na nie 6. System prezentuje odpowiedź na wybrane pytanie |
| Alternatywny przepływ wydarzeń | Brak |

Przypadek użycia: 2.1 Rezerwacja rejsu

|  |  |
| --- | --- |
| Nazwa | 2.1 Rezerwacja rejsu |
| Aktor | Gość |
| Krótki opis | Rezerwacja rejsu o konkretnym terminie na podane w formularzu dane osobowe |
| Warunki wstępne | Brak |
| Warunki końcowe | Zarezerwowanie konkretnego rejsu, na konkretny termin, dla konkretnego emaila, na podaną liczbę uczestników |
| Główny przepływ wydarzeń | 1. Wywołanie PU 1.1 Wyświetlenie dostępnych rejsów   2a. Gość wybiera przycisk ‘Wybierz termin’  3a. System prezentuje dostępne terminy  4a. Gość wybiera interesujący go termin i klika ‘Przejdź dalej’  5a. System wyświetla informacje potwierdzające rezerwowany rejs, formularz do wypełnienia i finalną kwotę do zapłaty  6a. Gość wprowadza stosowne dane do formularza oraz klika w przycisk ‘Rezerwuj’  7a. System powiadamia klienta o pomyślnie zarezerwowanym rejsie |
| Alternatywny przepływ wydarzeń | 2b. System wyświetla szczegóły wybranego rejsu, który nie ma dostępnych terminów  3b. Gość wraca do poprzedniego widoku w celu wybrania innego rodzaju rejsu lub porzuca proces  4b. Gość nie wybiera żadnego terminu i anuluje proces poprzez przycisk ‘Anuluj’ bądź ‘X’  5b. System chowa wyświetlone okno dialogowe i wraca do poprzedniego widoku  6b. Gość wprowadza błędne dane do formularza  7b. System przeprowadza walidację i powiadamia gościa o błędnych wartościach  6c. Gość wycofuje rezerwację rejsu poprzez przycisk ‘Anuluj’ bądź ‘X’  7c. System chowa wyświetlony moduł i wraca do poprzedniego widoku |

Przypadek użycia: 2.2 Podgląd zbliżających się rejsów użytkownika

|  |  |
| --- | --- |
| Nazwa | 2.2 Podgląd zbliżających się rejsów użytkownika |
| Aktor | Użytkownik |
| Krótki opis | Wyświetlenie listy wszystkich eventów przypisanych do emaila użytkownika |
| Warunki wstępne | Użytkownik jest poprawnie zalogowany |
| Warunki końcowe | System wyświetla listę eventów użytkownika |
| Główny przepływ wydarzeń | 1. Gość przechodzi na stronę główną aplikacji 2. System prezentuje panel powitalny 3. Gość przechodzi na zakładkę ‘Moje eventy’ 4. System prezentuje listę eventów |
| Alternatywny przepływ wydarzeń | Brak |

Przypadek użycia: 2.3 Wycofanie się z rejsu

|  |  |
| --- | --- |
| Nazwa | 2.3 Wycofanie się z rejsu |
| Aktor | Użytkownik |
| Krótki opis | Wycofanie użytkownika z wybranego rejsu |
| Warunki wstępne | Użytkownik jest poprawnie zalogowany i jest zarejestrowany na przynajmniej 1 rejs, który jest jeszcze nieopłacony |
| Warunki końcowe | System usuwa powiązanie pomiędzy użytkownikiem, a rejsem i odświeża listę rejsów użytkownika |
| Główny przepływ wydarzeń | 1. Wywołanie PU 2.2 Podgląd zbliżających się rejsów użytkownika 2. Wybiera interesujący go rejs i używa przycisku ‘Anuluj’   3a. System powiadamia użytkownika o pomyślnym wycofaniu się z rejsu oraz odświeża listę |
| Alternatywny przepływ wydarzeń | 3b. System powiadamia użytkownika o niemożliwości wycofania się z kursu ze względu na przekroczony dopuszczalny czas do wycofania |

Przypadek użycia: 3.1 Zapis na kurs

|  |  |
| --- | --- |
| Nazwa | 3.1 Zapis na kurs |
| Aktor | Użytkownik |
| Krótki opis | Użytkownik wpisuje się na konkretny rodzaj kursu i konkretny termin |
| Warunki wstępne | Użytkownik jest poprawnie zalogowany |
| Warunki końcowe | Wpisanie użytkownika na dany kurs |
| Główny przepływ wydarzeń | 1. Wywołanie PU 1.2 Wyświetlenie dostępnych kursów   2a. Użytkownik przechodzi do procesu zapisu na kurs poprzez przycisk ‘Zapisz się na kurs’   1. System prezentuje dostępne terminy   4a. Użytkownik wybiera interesujący go termin i klika ‘Zapisz się’  5a. System wyświetla informacje potwierdzające zapis na kurs |
| Alternatywny przepływ wydarzeń | 2b. System wyświetla szczegóły wybranego kursu, który nie ma dostępnych terminów  3b. Użytkownik wraca do poprzedniego widoku w celu wybrania innego rodzaju kursu lub porzuca proces  4b. Użytkownik nie wybiera żadnego terminu i anuluje proces poprzez przycisk ‘Anuluj’ bądź ‘X’  5b. System chowa wyświetlone okno dialogowe i wraca do poprzedniego widoku  4c. Użytkownik wybiera termin, na który już się zapisał i klika ‘Zapisz się’  5c. System wyświetla stosowny komunikat i odrzuca zapis na kurs |

Przypadek użycia: 3.2 Podgląd zbliżających się kursów użytkownika

|  |  |
| --- | --- |
| Nazwa | 3.2 Podgląd zbliżających się kursów użytkownika |
| Aktor | Użytkownik |
| Krótki opis | Wyświetlenie listy wszystkich kursów przypisanych do konta użytkownika |
| Warunki wstępne | Użytkownik jest poprawnie zalogowany |
| Warunki końcowe | System wyświetla listę kursów użytkownika |
| Główny przepływ wydarzeń | 1. Gość przechodzi na stronę główną aplikacji 2. System prezentuje panel powitalny 3. Gość przechodzi na zakładkę ‘Moje kursy’ 4. System prezentuje listę kursów |
| Alternatywny przepływ wydarzeń | Brak |

Przypadek użycia: 3.3 Wycofanie się z kursu

|  |  |
| --- | --- |
| Nazwa | 3.3 Wycofanie się z kursu |
| Aktor | Użytkownik |
| Krótki opis | Wypisanie użytkownika z wybranego kursu |
| Warunki wstępne | Użytkownik jest poprawnie zalogowany i jest zapisany na przynajmniej 1 kurs, który jest jeszcze nieopłacony |
| Warunki końcowe | System usuwa powiązanie pomiędzy użytkownikiem, a kursem i odświeża listę kursów użytkownika |
| Główny przepływ wydarzeń | 1. Wywołanie PU 3.2 Podgląd zbliżających się kursów użytkownika 2. Wybiera interesujący go kurs i używa przycisku ‘Anuluj’   3a. System powiadamia użytkownika o pomyślnym wycofaniu się z rejsu oraz odświeża listę |
| Alternatywny przepływ wydarzeń | 3b. System powiadamia użytkownika o niemożliwości wycofania się z kursu ze względu na jego opłacenie |

Przypadek użycia: 4.1 Wygenerowanie egzaminu próbnego

|  |  |
| --- | --- |
| Nazwa | 4.1 Wygenerowanie egzaminu próbnego |
| Aktor | Użytkownik |
| Krótki opis | Wygenerowanie użytkownikowi wszystkich pytań do egzaminu oraz rozpoczęcie odliczania czasu |
| Warunki wstępne | Użytkownik jest poprawnie zalogowany i jest zapisany na przynajmniej 1 kurs, który jest opłacony |
| Warunki końcowe | System generuje użytkownikowi losowy egzamin próbny |
| Główny przepływ wydarzeń | 1a. Wywołanie PU 3.2 Podgląd zbliżających się kursów użytkownika   1. Użytkownik wybiera interesujący go kurs i używa przycisku ‘Nauka’ 2. System wyświetla listę kursów i dostępnych przycisków do e-learningu[[5]](#footnote-5) 3. Użytkownik używa przycisku do wygenerowania próbnego egzaminu dla interesującego go opłaconego kursu 4. System generuje egzamin próbny oraz rozpoczyna odliczanie do jego zakończenia |
| Alternatywny przepływ wydarzeń | 1b. Użytkownik przechodzi na zakładkę ‘Nauka’ |

Przypadek użycia: 4.2 Rozwiązanie egzaminu próbnego

|  |  |
| --- | --- |
| Nazwa | 4.2 Rozwiązanie egzaminu próbnego |
| Aktor | Użytkownik |
| Krótki opis | Użytkownik odpowiada na pytania oraz kończy egzamin próbny |
| Warunki wstępne | Użytkownik jest poprawnie zalogowany i jest zapisany na przynajmniej 1 kurs, który jest opłacony |
| Warunki końcowe | System wyświetla wyniki z rozwiązanego egzaminu oraz dopisuje go do statystyk |
| Główny przepływ wydarzeń | 1. Wywołanie PU 4.1 Wygenerowanie egzaminu próbnego 2. Użytkownik kończy wybrany egzamin poprzez przycisk ‘Zakończ egzamin’ 3. System wyświetla wyniki z rozwiązanego egzaminu |
| Alternatywny przepływ wydarzeń | Brak |

Przypadek użycia: 4.3 Nauka ogólna

|  |  |
| --- | --- |
| Nazwa | 4.3 Nauka ogólna |
| Aktor | Użytkownik |
| Krótki opis | Użytkownik dostaje losowe pytanie do rozwiązania z konkretnego działu żeglarskiego |
| Warunki wstępne | Użytkownik jest poprawnie zalogowany i jest zapisany na przynajmniej 1 kurs, który jest opłacony |
| Warunki końcowe | System generuje losowe pytanie |
| Główny przepływ wydarzeń | 1a. Wywołanie PU 3.2 Podgląd zbliżających się kursów użytkownika   1. Użytkownik wybiera interesujący go kurs i używa przycisku ‘Nauka’ 2. System wyświetla listę kursów i dostępnych przycisków do e-learningu 3. Użytkownik używa przycisku do nauki ogólnej dla interesującego go opłaconego kursu 4. System generuje losowe pytanie   6a. Użytkownik odpowiada na pytanie i sprawdza odpowiedź  7a. System wyświetla poprawną odpowiedź |
| Alternatywny przepływ wydarzeń | 1b. Użytkownik przechodzi na zakładkę ‘Nauka’  6b. Użytkownik przechodzi do następnego pytania  7b. System generuje kolejne pytanie |

Przypadek użycia: 4.4 Statystyki nauki

|  |  |
| --- | --- |
| Nazwa | 4.4 Statystyki nauki |
| Aktor | Użytkownik |
| Krótki opis | Użytkownik otrzymuje spersonalizowane statystyki nauki z odpowiedzi na pytania i z egzaminów próbnych |
| Warunki wstępne | Użytkownik jest poprawnie zalogowany i jest zapisany na przynajmniej 1 kurs, który jest opłacony |
| Warunki końcowe | System generuje wyniki z e-learningu |
| Główny przepływ wydarzeń | 1a. Wywołanie PU 3.2 Podgląd zbliżających się kursów użytkownika   1. Użytkownik wybiera interesujący go kurs i używa przycisku ‘Nauka’ 2. System wyświetla listę kursów i dostępnych przycisków do e-learningu 3. Użytkownik używa przycisku do statystyki nauki dla interesującego go opłaconego kursu 4. System generuje statystyki |
| Alternatywny przepływ wydarzeń | Brak |

Przypadek użycia: 5.1 Rejestracja konta

|  |  |
| --- | --- |
| Nazwa | 5.1 Rejestracja konta |
| Aktor | Gość |
| Krótki opis | Utworzenie konta poprzez formularz danych osobowych |
| Warunki wstępne | Brak |
| Warunki końcowe | System tworzy nowego użytkownika |
| Główny przepływ wydarzeń | 1. Gość przechodzi na stronę główną aplikacji 2. System prezentuje panel powitalny 3. Gość przechodzi na zakładkę ‘Zarejestruj’ z poziomu górnego paska nawigacyjnego 4. System prezentuje panel rejestracyjny z formularzem   5a. Gość wprowadza poprawne dane i klika przycisk ‘Zarejestruj się’ |
| Alternatywny przepływ wydarzeń | Brak |

1. Framework – zestaw narzędzi, bibliotek i konwencji, które przyspieszają proces wytwarzania oprogramowania [↑](#footnote-ref-1)
2. UI/UX – User Interface/User Experience – odpowiada za projektowanie interfejsu użytkownika na podstawie jego doświadczeń [↑](#footnote-ref-2)
3. Gość – Użytkownik niezalogowany (nieautoryzowany) [↑](#footnote-ref-3)
4. Mockup – wizualna, dość szczegółowa reprezentacja interfejsu graficznego aplikacji, która ma na celu ukazanie wyglądu i układu widoku strony [↑](#footnote-ref-4)
5. Przyciski do e-learningu to statystyki nauki, nauka ogólna (wygenerowanie losowego pytania) oraz egzamin próbny prowadzący do jego wygenerowania [↑](#footnote-ref-5)