**POLITECHNIKA BIAŁOSTOCKA**

**WYDZIAŁ INFORMATYKI**

**KATEDRA** ………………….

PRACA DYPLOMOWA INŻYNIERSKA

TEMAT: APLIKACJA INTERNETOWA WSPOMAGAJĄCA PROWADZENIE ZAJĘĆ Z PRZEDMIOTU WPROWADZENIE DO INFORMATYKI

WYKONAWCA:

MICHAŁ KIERZKOWKI

PODPIS: ..................................

PROMOTOR:

DR INŻ. IRENA BUŁATOWA

PODPIS: ..................................

**BIAŁYSTOK 2020 ROK**

**KARTA DYLOMOWA**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| POLITECHNIKA BIAŁOSTOCKA | | |  | | | Nr albumu studenta 101335 | |
| Wydział Informatyki | | | Studia stacjonarne  I stopnia | | | Rok akademicki 2019/2020 | |
| Kierunek studiów  Informatyka | |
| Katedra/Zakład................................... | | |  | | | Specjalność...................................  ...................................................... | |
| **Imię i nazwisko studenta**: Michał Kierzkowski  **Temat pracy dyplomowej:**  Aplikacja internetowa wspomagająca prowadzenie zajęć z przedmiotu Wprowadzenie do informatyki  Zakres pracy:   1. Przegląd technologii tworzenia aplikacji internetowych. 2. Opracowanie koncepcji aplikacji, analiza wymagań. 3. Wykonanie projektu aplikacji internetowej. 4. Implementacja i testowanie aplikacji.   **Słowa kluczowe (max 5):** aplikacja internetowa, komputerowe wspomaganie nauczania, e-learning | | | | | | | |
| .........................................................................  *Imię i nazwisko, stopień/ tytuł promotora - podpis* | | | | ......................................................................................  *Imię i nazwisko kierownika katedry - podpis* | | | |
| .....................................................*Data wydania tematu pracy dyplomowej*  *- podpis promotora* | | ...................................................  *Regulaminowy termin złożenia pracy dyplomowej* | | | ........................................................  *Data złożenia pracy dyplomowej*  *- potwierdzenie dziekanatu* | | |
|  | .....................................  *Ocena promotora* | | | ..................................  *Podpis promotora* | | |  |
| ...................................................  *Imię i nazwisko, stopień/ tytuł recenzenta* | | ...........................................  *Ocena recenzenta* | | | | ................................................  *Podpis recenzenta* | |

Michał Kierzkowski Białystok, dnia………………………

*Imiona i nazwisko studenta*

101335  
*Nr albumu*

Informatyka, Studia stacjonarne I stopnia

*Kierunek i forma studiów*

dr inż. Irena Bułatowa

*Promotor pracy dyplomowej*

**OŚWIADCZENIE**

Przedkładając w roku akademickim 2019/2020 Promotorowi dr inż. Irena Bułatowa pracę dyplomową pt.: Aplikacja internetowa wspomagająca prowadzenie zajęć z przedmiotu Wprowadzenie do Informatyki, dalej zwaną pracą dyplomową, **oświadczam, że:**

1. praca dyplomowa stanowi wynik samodzielnej pracy twórczej;
2. wykorzystując w pracy dyplomowej materiały źródłowe, w tym w szczególności: monografie, artykuły naukowe, zestawienia zawierające wyniki badań (opublikowane, jak i nieopublikowane), materiały ze stron internetowych, w przypisach wskazywałem/am ich autora, tytuł, miejsce i rok publikacji oraz stronę, z której pochodzą powoływane fragmenty, ponadto w pracy dyplomowej zamieściłem/am bibliografię;
3. praca dyplomowa nie zawiera żadnych danych, informacji i materiałów, których publikacja nie jest prawnie dozwolona;
4. praca dyplomowa dotychczas nie stanowiła podstawy nadania tytułu zawodowego, stopnia naukowego, tytułu naukowego oraz uzyskania innych kwalifikacji;
5. treść pracy dyplomowej przekazanej do dziekanatu Wydziału Informatyki jest jednakowa w wersji drukowanej oraz w formie elektronicznej;
6. jestem świadomy/a, że naruszenie praw autorskich podlega odpowiedzialności na podstawie przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. z 2017 r. poz. 880), jednocześnie na podstawie przepisów ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (Dz. U. z 2016 r. poz. 1842, z późn. zm.) stanowi przesłankę wszczęcia postępowania dyscyplinarnego oraz stwierdzenia nieważności postępowania w sprawie nadania tytułu zawodowego;
7. udzielam Politechnice Białostockiej nieodpłatnej licencji niewyłącznej na umieszczenie elektronicznej wersji pracy dyplomowej w repozytorium uczelnianym/Bazie Wiedzy PB i do korzystania z utworu bez ograniczeń czasowych i terytorialnych przez jego udostępnienie on-line dla użytkowników sieci wewnętrznej Politechniki Białostockiej. Upoważniam Politechnikę Białostocką do przechowywania i archiwizowania pracy dyplomowej na nośnikach cyfrowych oraz jej zwielokrotniania i udostępniania w formie elektronicznej w zakresie koniecznym do weryfikacji autorstwa pracy i ochrony przed przywłaszczeniem autorstwa, w tym na przekazanie pracy dyplomowej do Ogólnopolskiego Repozytorium Prac Dyplomowych;
8. zostałem poinformowany/a, że na podstawie art. 24 ust. 1 ustawy z dnia 29 sierpnia 1997 roku o ochronie danych osobowych (Dz. U. z. 2016 r. poz. 922, z późn. zm.) administratorem danych jest rektor Politechniki Białostockiej, ul. Wiejska 45A, 15-351 Białystok. Dane będą przetwarzane w celach realizacji procedury antyplagiatowej przyjętej na Politechnice Białostockiej i nie będą udostępniane odbiorcom danych w rozumieniu art. 7 pkt. 6 ustawy o ochronie danych osobowych. Osobie, której dane dotyczą, przysługuje prawo dostępu do treści swoich danych oraz ich poprawiania. Podanie danych jest obowiązkowe (art. 167b ustawy z dnia 27 lipca 2005 roku Prawo o szkolnictwie wyższym Dz. U. z 2016 r. poz. 1842 z późn. zm.). Dane będą przetwarzane w Politechnice Białostockiej przez okres 50 lat.

.………………………………………. *czytelny podpis studenta*

Thesis topic:

A web application supporting the teaching of "Introduction to Computer Science"

Summary

The purpose of this thesis was to create an Internet application that helps in conducting classes of Introduction to Computer Science at the Bialystok University of Technology. The project was initiated by the growing demand for a reliable source of knowledge and the author's desire to develop skills of using technologies that appears in the project.

The main function of the project includes solutions such as courses, conducted by the author, explaining IT attitudes and quizzes that are checking the state of acquired knowledge. It was extremely important to prepare the site so the knowledge is provided in a nice and clear way. Therefore, in courses, beyond just theory, are also given YouTube’s videos explaining problems presented in the topic of the course. After learning from each chapter, the user has the opportunity to test himself in the interactive quizzes. The results of each quiz are written in statistics to allow users and the administrator to follow the learning progress, and if necessary, the administrator can add more material to increase understanding of the topic. Additional function includes a forum with access allowed for every user. It includes adding or commenting the topic related with courses or quizzes putted on the site.

The basic part of the project was developed in Python using the Django Framework, based on the MVT (Model-View-Template) architecture. The site uses server-side code technology to dynamically display various data when needed. Thanks to the usage of AJaX technology, user interaction with the server takes place without reloading the entire document. The entire presentation layer was developed with integrated front-end technology, i.e. HTML, CSS and JavaScript. In addition, the UiKit framework, which is the equivalent of Bootstrap, was used. It significantly increased the appearance of the page, including the available modules and extensible components.

Creating the entire project required extraordinary workload and collecting universal knowledge about Computer Science took a lot of time. The result of this work is a simple and friendly website that helps in learning.

Spis Treści

[1. Wstęp 1](#_Toc26370588)

[2. Rozwój technologii a nauczanie zdalne 3](#_Toc26370589)

[2.1 Zarys historyczny nauczania na odległość 3](#_Toc26370590)

[2.2 Przegląd najważniejszych zagadnień związanych z e-learningiem 5](#_Toc26370591)

[2.3 Zalety i wady zdalnej edukacji 8](#_Toc26370592)

[2.4 Przegląd dostępnych systemów e-learningowych 10](#_Toc26370593)

[3. Projekt systemu 14](#_Toc26370594)

[3.1 Główne założenia i funkcjonalności 14](#_Toc26370595)

[3.2 Ustandaryzowanie systemu 18](#_Toc26370596)

[3.3 Zasoby dydaktyczne 20](#_Toc26370597)

[3.4 Baza danych 21](#_Toc26370598)

[3.5 Podstawowe założenia interfejsu 26](#_Toc26370599)

[4. Implementacja projektu 29](#_Toc26370600)

[4.1 Języki programowania, framworki oraz środowisko programistyczne 29](#_Toc26370601)

[4.2 Tworzenie modeli i funkcjonalności 32](#_Toc26370602)

[4.3 Implementacja szablonów 37](#_Toc26370603)

[4.4 Testy 41](#_Toc26370604)

[5. Interface witryny 43](#_Toc26370605)

[5.1 Logowanie i rejestracja 44](#_Toc26370606)

[5.2 Panel użytkownika 45](#_Toc26370607)

[5.3 Zarządzanie i administracja systemem 47](#_Toc26370608)

[6. Podsumowanie 49](#_Toc26370609)

[Literatura 51](#_Toc26370610)

[Spisy 53](#_Toc26370611)

[Spis tabel 53](#_Toc26370612)

[Spis rysunków 54](#_Toc26370613)

[Listing 55](#_Toc26370614)

[Instrukcja instalacji 56](#_Toc26370615)

# 1. Wstęp

W dzisiejszym świecie coraz większą rolę w systemie nauczania zaczynają odgrywać różnorakie witryny e-learningowe. Umożliwiają one kształcenie się, ukończenie różnego rodzaju kursów i szkoleń, bez przymusu wychodzenia z domu. Wystarczy, że posiadamy komputer lub smartfon z dostępem do Internetu i już możemy stać się uczestnikami różnego typu zajęć prowadzonych na płaszczyźnie wirtualnej.

Platformy e-learningowe niosą za sobą wiele korzyści, wynikających głównie   
z łatwości i szybkości dostępu do nich. Jednakże największą zaletą tego typu rozwiązania jest fakt, iż osoba, która zdecyduje się na taki rodzaj kursu sama ustala tempo przyswajania wiedzy, jak i czas podjęcia nauki. Mamy tutaj do czynienia z indywidualizacją procesu nauczania, a przez to większą skutecznością kształcenia. Jednocześnie z korzyściami dla uczestników szkolenia, witryny e-learningowe są znacznym ułatwieniem dla instruktorów poprzez jednolity przekaz dla wszystkich słuchaczy, możliwość prowadzenia kursu dla bardzo dużej grupy odbiorców, łatwość nadzoru nad postępami nauczania, jak również sprawdzanie zdobytej wiedzy.

Niniejsza praca inżynierska dotyczy oprogramowania wspomagających prowadzenie kursów i szkoleń. Głównym celem pracy jest stworzenie (opracowanie koncepcji, analiza wymagań, projekt, implementacja, testowanie) aplikacji internetowej wspomagającej prowadzenie zajęć z przedmiotu Wprowadzenie do Informatyki. Witryna ma umożliwiać wyjaśnienie zagadnień dotyczących tematów realizowanych na zajęciach, poprzez stworzone struktury strony tak, aby przekazywała użytkownikowi wiedzę w miły i czytelny sposób. Aplikacja ma również pozwalać na przygotowanie zestawów pytań sprawdzających stan zdobytej wiedzy oraz obliczanie i przechowywanie wyników rozwiązanych testów. Innymi zadaniami postawionymi przed autorem poniższej pracy jest: zapoznanie się z obecnym stanem wiedzy na temat zdalnego nauczania, przegląd dostępnych na rynku platform służących kształceniu na odległość oraz umieszczenie w witrynie zebranych materiałów dydaktycznych pomagających w nauce.

Autora pracy do podjęcia tematu skłonił fakt, że zauważył on zapotrzebowanie na stworzenie platformy zawierającej zebrane w jednej witrynie wszystkie zagadnienia związane z podstawami informatyki potrzebnymi do zaliczenia przedmiotu Wprowadzenie do Informatyki na Politechnice Białostockiej. Dotychczas dostępne witryny nie zawierały całego zakresu materiału, a jedynie poruszały pojedyncze aspekty tego jakże niebagatelnego tematu. Dodatkowym aspektem motywacyjnym była chęć rozwój wiedzy autora na temat technologii użytych w projekcje, o których mowa w późniejszych częściach pracy.

Praca podzielona jest na sześć rozdziałów. W drugim rozdziale przedstawionej pracy inżynierskiej mieści się omówienie historii i zagadnień ogólnej charakterystyki kursów zdalnych oraz porównanie dostępnych na rynku platform e-learningowych. Rozdział ten skupia się głównie na przedstawieniu wad i zalet oraz na ogólnym rozeznaniu i analizie rynku e-nauczania.

Trzeci rozdział zawiera projekt kompletnej witryny wspomagającej prowadzenie zajęć. Wyróżnione są w nim funkcjonalności oraz poszczególne założenia, jakie ma spełnić system. Rozdział skupia również uwagę na strukturze bazy danych, zasobów dydaktycznych w niej zawartych oraz krótkim zarysie interfejsu strony.

Czwarty rozdział jest to przedstawienie i omówienie użytych technologii oraz opis implementacji wybranych funkcjonalności. W rozdziale zawarto główne zalety oraz korzyści, jakie niosą za sobą użyte frameworki. Omówiono również użyte rozwiązania podczas kodowania określonych założeń.

W kolejnym rozdziale przedstawiono instrukcję użytkownika zawierającą prezentację interfejsu witryny. Pokazany został proces rejestracji użytkownika w witrynie, jak również omówione zostały poszczególne widoki strony oraz sposób administracji nad zawartymi w niej treściami jak i poszczególnymi użytkownikami.

Ostatni - szósty rozdział zawiera podsumowanie oraz wnioski wynikające zarówno z części teoretycznej jak praktycznej pracy nad aplikacją. Ta część przedstawia również dalsze perspektywy rozwoju stworzonej aplikacji.

W zakończeniu pracy przestawiony został spis literatury wykorzystanej do napisania niniejszej pracy inżynierskiej, a także spis tabel i rysunków oraz krótka instrukcja instalacji aplikacji.

# 2. Rozwój technologii a nauczanie zdalne

Pierwszym krokiem, jaki wykonano podczas tworzenia niniejszej pracy było zebranie odpowiedniej literatury oraz wyszukanie w niej informacji dotyczących zarówno procesu rozwoju zdalnego nauczania, jaki i współczesnych technik oraz technologii wykorzystywanych w systemach e-nauczania. Wiedza ta pozwoliła autorowi na szerokie spektrum podejścia do dalszych etapów tworzenia pracy, jakimi są projektowanie i implementacja systemu.

## 2.1 Zarys historyczny nauczania na odległość

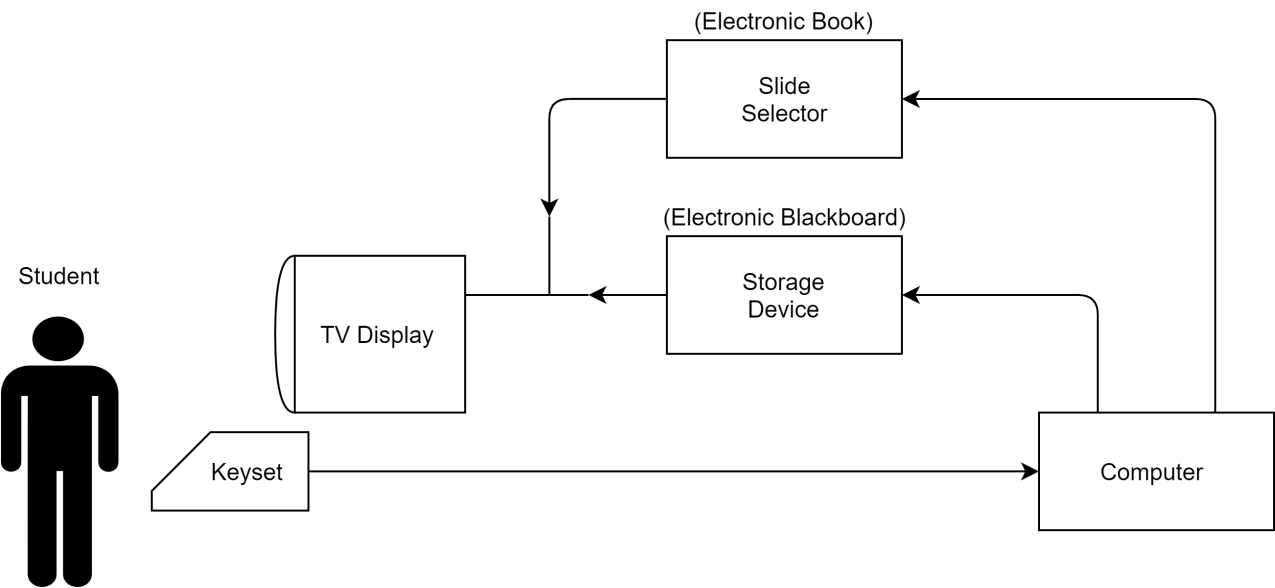
W dzisiejszych czasach nikogo nie dziwi już stosowanie nowoczesnych metod nauczania, jakim jest e-edukacja. Trzeba mieć świadomość tego, że e-nauczanie zalicza się do jednych z większych postępów w dziedzinie nauczania, który niesie za sobą mnóstwo możliwości.

Pierwsze wzmianki dotyczące zdalnego nauczania pojawiły się w 1728 roku. Właśnie wtedy w Stanach Zjednoczonych ukazało się ogłoszenie nawiązujące do odbycia kursu korespondencyjnego. Celem kursu było wysyłanie pocztą wybranych materiałów szkoleniowych (między instruktorem a kursantem) [1].

Dziewiętnasty wiek przyniósł ze sobą nieznaczny rozwój tej metody nauczania, poprzez coraz większe zainteresowanie kursami językowymi, jak również kursami stenografii na odległość. Jednakże nadal wszystko odbywało się bardzo długo, przy użyciu tradycyjnej poczty [1].

Przełom przyniósł wiek XX, który poprzez rozwój nowych technologii pozwolił na powstanie nowych technik nauczania zdalnego. Na początku wieku zaczęły funkcjonować pierwsze audycje w radiu, tzw. radio edukacyjne. Na wstępie było ono wykorzystywane głównie w słabo zaludnionych częściach Australii, zaś z biegiem lat wiele krajów zaczęło stosować je do nauczania ludności na obszarach wiejskich. Można było też spotkać pierwsze możliwości uzyskania stopnia naukowego poprzez studia korespondencyjne. 1945 rok był początkiem telewizji edukacyjnej, która już nie tylko poprzez sam dźwięk czy też tekst, ale także proste animacje pozwalała przyswoić określony materiał.   
W późniejszych latach również przekaz satelitarny był wykorzystywany do procesu nauczania [1].

Niewątpliwie największym przełomem w zdalnym nauczaniu był rozwój telekomunikacji cyfrowej oraz multimedialnej, dzięki którym stało się możliwe nauczanie z zastosowaniem Internetu, co dało niemal nieograniczone (w kontekście poprzednich rozwiązań) możliwości komunikowania się [1]. Pierwsza na świecie platforma nauczania komputerowego powstała w 1960 roku nosiła nazwę Programmed Logic for Automatic Teaching Operations (PLATO), została zaprojektowana i zbudowana przez Uniwersytet w Illinois. Początkowo była to platforma typu CBT, czyli computer based learning. Już pod koniec lat 70 obsługiwała kilka tysięcy terminali graficznych dystrybuowanych na całym świecie. Z biegiem lat opracowano na niej wiele nowoczesnych koncepcji obliczeń dla wielu użytkowników, ich przykładami są fora dyskusyjne, testy online, języki obrazkowe czy komunikatory internetowe [15]. Jeden z pierwszych uproszczonych schematów systemu PLATO przedstawia Rys. 2.1.

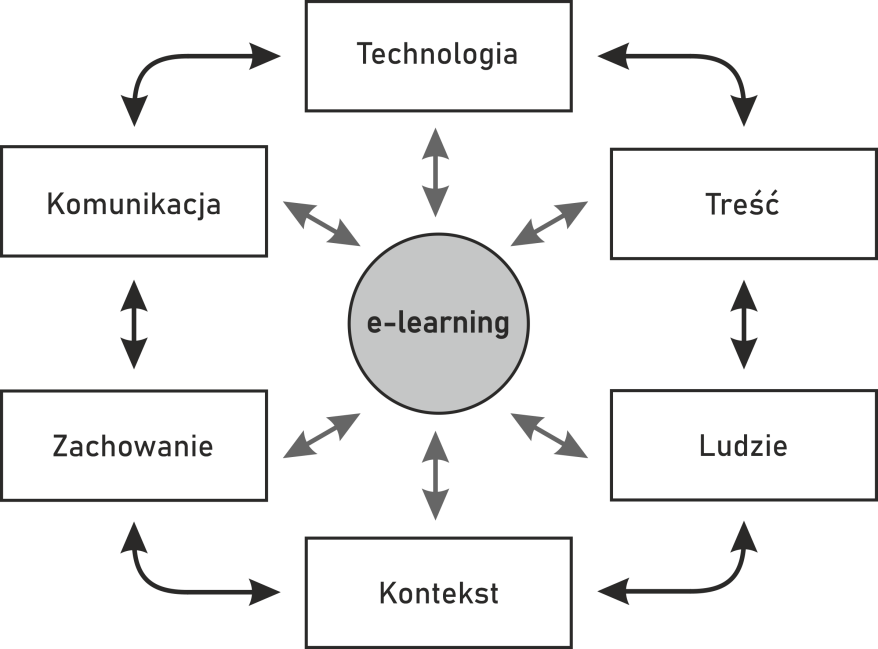


Rys. 2.1 Uproszczony schemat blokowy pierwszego systemu PLATO z 1960 roku [15]

Mówiąc o edukacji zdalnej, nie można pominąć polskich rozwiązań w tej dziedzinie. Jednym z nich był tzw. Uniwersytet Latający powstały w 1885 roku. Był on tajną szkołą wyższą powstałą dla kobiet, dla których edukacja na poziomie uniwersyteckim była zamknięta. Nie posiadał on konkretnego ośrodka akademickiego, dlatego też możemy uznać go za edukację w formie zdalnej. Jednak największym przedsięwzięciami tego typu w Polsce była Politechnika Telewizyjna, działająca w latach 1966-1971, gdzie w sposób zdalny kształcono inżynierów poprzez realizację dydaktycznych programów telewizyjnych z zakresu pierwszego i drugiego roku zaocznych studiów technicznych. Było to pionierskie, pierwsze w Europie przedsięwzięcie obejmujące całe terytorium kraju, wprowadzające nową formę kształcenia w obszarze szkolnictwa wyższego [1, 7].

## 2.2 Przegląd najważniejszych zagadnień związanych z e-learningiem

Jak podaje definicja znajdująca się na stronie Głównego Urzędu Statystycznego, e-learning jest to „nauczanie lub szkolenia przy użyciu technologii informatycznej. Oznacza wspomaganie procesu dydaktyki za pomocą komputerów osobistych, smartfonów, tabletów i Internetu. Pozwala na ukończenie kursu, szkolenia, a nawet studiów bez konieczności fizycznej obecności w sali wykładowej. Uzupełnia również tradycyjny proces nauczania, budując blended learning” [11].



Rys. 2.2 Powiązania w obszarze e-leraningu (opracowanie własne na podstawie [16])

Sama nazwa e-learning, jak i technika edukacji, odnosi się do o wiele szerszych ram niż tradycyjne szkolenia komputerowe (Rys. 2.2). Obejmuje swoim zasięgiem kilka znaczących, wzajemnie powiązanych obszarów, takich jak [16]:

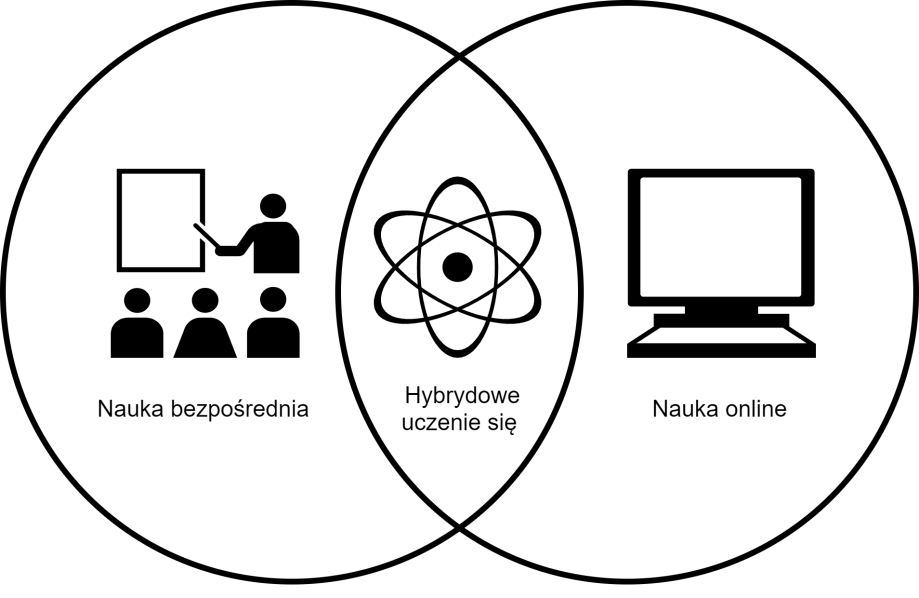
* treść – udostępnione zasoby edukacyjne w wersji elektronicznej;
* technologia – wszystkie rodzaje oprogramowania i sprzętu wykorzystywane do tworzenia oraz udostępniania treści;
* kontekst – uwarunkowania, w jakich wdraża się e-szkolenie;
* ludzie – ściśle scharakteryzowani adresaci kursu;
* komunikacja – sposób porozumiewania się uczestników szkolenia zarówno na poziomie nauczyciel-kursant jak i kursant-kursant
* zachowania – określenie podstawowych typów zachowań psychospołecznych występujących poda czad szkoleń

Rozpoczęcie nauczania poprzez e-learningi stało się swoistą rewolucją, której skutki możemy porównać do wpływu, jaki wcześniej wywarło wynalezienie druku   
i masowa produkcja książek. Obecnie większość szkół wyższych posiada platformy e-learningowe w celu zwiększenia dostępności materiałów dla studentów, a także możliwości łatwego śledzenia postępu w zdobywaniu wiedzy oraz kontrolowaniu jej poziomu. Zazwyczaj studenci zobowiązani są do uczestnictwa w zajęciach, jednakże wiele kursów zawierających scenariusze lekcji czy też arkusze ćwiczeniowe udostępnionych jest online.

Odnosząc się do stopnia wykorzystania technik i metod kształcenia zdalnego   
w procesie dydaktycznym, możemy wyróżnić [17]:

* mieszane uczenie się (inaczej uczenie się hybrydowe, ang. *blended learning*) – proces dydaktyczny jest realizowany częściowo w sposób tradycyjny, częściowo zaś za pośrednictwem Internetu;
* nauka online (ang. *online learning*) – proces dydaktyczny jest prowadzony w pełni za pośrednictwem Internetu, nauczyciel i uczący się nie spotykają się w tradycyjnej klasie;
* nauka bezpośrednia (ang. *face to face learning*) – w tym przypadku nowoczesne metody i techniki kształcenia służą jedynie uzupełnieniu i wsparciu tradycyjnego procesu nauczania.

Zależność pomiędzy wyodrębnionymi procesami dydaktycznymi ukazuje Rys.2.3.



Rys. 2.3 Zależności pomiędzy rodzajami procesów dydaktycznych (opracowanie własne na podstawie [17])

Dodatkowo możemy wyodrębnić podział procesów dydaktycznych wykorzystujących kształcenie zdalne odnoszący się do stopnia ich interaktywności, a więc intensywności oraz ilości interakcji zachodzących pomiędzy kursantem a nauczycielem i innymi uczestnikami kursu. Biorąc pod uwagę to kryterium wyróżniamy [2]:

* proces dydaktyczny, w którym następuje praca tylko z materiałem dydaktycznym umieszczonym w Internecie;
* kursy z e-mentorem, w których uczący nadal realizuje poszczególne treści dydaktyczne, ma jednak możliwość uzyskania pomocy mentora w dowolnym momencie kursu;
* kursy z nauczycielami odbywają się w grupach, tzw. wirtualnych klasach, w tym przypadku kontrola nad przebiegiem całego procesu nauczania należy do nauczyciela prowadzącego, który konstruuje zadania, stawia pytania dyskusyjne   
  i przypomina o terminowości.

Warto również pamiętać, że w systemach nauczania na odległość standardowe podejście do roli nauczyciela, jako specjalisty z danej dziedziny wiedzy i będącej jej przekaźnikiem ulega zmianie na przewodnika czy też doradcę, którego głównym zadaniem jest stosowanie odkrywczego modelu nauczania, w którym nie przekazuje gotowej wiedzy, lecz przyczynia się do jej samodzielnego zdobycia poprzez udostępnione treści, które mają wzbudzić w odbiory chęć dociekania, odkrywania i dyskutowania [10].

Od około 20 lat zdalne uczenie się zyskało dodatkową pomoc poprzez sieć drugiej generacji, czyli Web 2.0. Umożliwia ona każdemu użytkownikowi aktywną współpracę   
w tworzeniu określonych zasobów wiedzy. W ten sposób ukształtował się niehierarchiczny model pozyskiwania, jak i przyswajania wiedzy, w którym wszyscy są jednocześnie odbiorcami i twórcami treści, mogą je dodawać, usuwać, komentować, współdzielić itp.[4]. Zasoby, które powstały na skutek oddolnej aktywności użytkowników Internetu, mają we współczesnym świecie niemal równorzędne znaczenie w stosunku do opracowań naukowych czy też materiałów znajdujących się na platformach e-learninowych. Dzięki temu użytkownicy uzyskali łatwy dostęp do treści i wiedzy współtworzonych przez wielu autorów, innymi słowy korzystają z wiedzy społecznej [17].

Web 2.0 ma swoje podłoże w różnego rodzaju aplikacjach. Spośród nich możemy wyróżnić: narzędzia Wiki (szeroko znane w formie encyklopedii on-line), blogi, media społecznościowe, publikacje dźwiękowej lub filmowe (podcasty, screemcasty, webcasty).

## 2.3 Zalety i wady zdalnej edukacji

Od kiedy w obszarze nauczania pojawił się e-learning, wyróżnia się kilka jego głównych atutów, mających znaczenie zarówno z punktu widzenia osoby uczącej się, jak   
i nauczycieli czy organizacji nauczających. Najczęściej do zalet kształcenia zdalnego zalicza się [17, 5, 7]:

* elastyczność i mobilność – możliwość prowadzenia kursu w dowolnym miejscu   
  i czasie, teoretycznie 24 godziny na dobę przez 7 dni w tygodniu;
* brak ograniczeń terytorialnych – możliwość brania udziału w kursach prowadzonych przez uczelnie czy firmy szkoleniowe mające siedzibę w dowolnym miejscu na ziemi;
* możliwość lepszego i bardziej efektywnego zarządzania czasem nauki i pracy – uczeń, który jest aktywny zawodowo lub rodzinnie nie musi opuszczać miejsca pracy czy też domu, aby brać udział w kursie. Może również dopasować godziny nauki do własnego rozkładu dnia. Analogicznie, zdalne nauczanie tworzy szansę dla prowadzących zajęcia na wykonywanie obowiązków dydaktycznych równocześnie z innymi zadaniami zawodowymi czy rodzinnymi;
* szeroki dostęp do wiedzy – wzięcie udziału w edukacji zdalnej otwiera dla uczestnika pełniejsze spożytkowanie informacyjnych zasobów Internetu;
* możliwość indywidualizacji tempa i sposobu nauki – kursanta, co prawda obowiązuje pewien terminarz, co do realizacji zadań czy też rozwiązania testów, jednakże poza tym nie jest mu narzucone rygorystyczne tępo nauki i może   
  w dowolny sposób zarządzać swoim tokiem uczenia. E-learning jest także rozwiązaniem dla uczniów o zróżnicowanych stylach przyswajania informacji. Dzięki szerokiemu wachlarzowi funkcjonalności, co do kontroli postępów, nauczyciel ma możliwość obserwacji i kontroli postępów ucznia oraz dopasowywania materiałów dydaktycznych do jego potrzeb;
* oszczędność finansowa – poprzez zdalne kursy niwelowane są koszty czy to dojazdu, czy też wynajmu sal szkoleniowych. Poza tym ta forma nauczania szanuje przede wszystkim czas ucznia i nauczyciela, dzięki czemu nie rezygnują oni   
  z ważnych wydarzeń życia codziennego, jak i nie jest marnotrawią godzin poświęconych dojazdowi na kurs;
* rozwój kreatywności, samodzielności i umiejętności pracy w grupie – zdalne kształcenie wymusza na uczącym się przejawianie aktywności i inicjatywy. Jednocześnie wymuszając wielostronną i ścisłą współpracę pomiędzy uczniami   
  a nauczycielem oraz kształci umiejętności komunikacji w zespole, a także sprzyja wymianie informacji.

Jednocześnie z rozwojem zdalnego nauczania zaczęły konkretyzować się ograniczenia i trudności związane z tą formą uczenia się i nauczania. Do głównych należą [17, 5]:

* pośredni kontakt – realizowany zazwyczaj w formie pisemnej niesie za sobą prawdopodobieństwo wyniknięcia zakłóceń w komunikacji i trudności w ich wyjaśnianiu. Nauczanie na odległość wiążę się z ograniczeniem komunikacji niewerbalnej, będącej podstawowym źródłem informacji na temat zaangażowania ucznia, jego emocji czy chociażby braku zrozumienia danego materiału;
* brak możliwości utrzymania stałej aktywności ucznia – brak kontaktu bezpośredniego między uczniem a nauczycielem może spowodować zniechęcenie tego pierwszego. Niepoprawny plan pracy własnej ucznia może skutkować poczuciem izolacji, narastającym stresem a co za tym idzie doprowadzeniem do sytuacji kryzysowej, w której uczeń ma utrudnione przyswajanie materiału, czego skutkiem jest pogłębianie się zaległości i uczucie coraz większej niechęci do kursu;
* trudności w utrzymaniu motywacji ucznia do nauki – w tradycyjnym toku nauczania uczeń ma obowiązek uczestnictwa na zajęciach a niekiedy aktywnego udziału w nich. Pomaga to w skoncentrowaniu się na konkretnej treści i zadaniach, inaczej jak w edukacji zdalnej gdzie uczeń realizuje zadnia sam niekiedy wieczorami lub podczas dni wolnych, gdzie jego uwaga skupia się głównie na tym, by szybko wykonać zadnia, niekiedy nie przykładając wagi do poprawności. Dlatego też najtrudniejszym zadaniem w projektowaniu aplikacji do e-nauczania jest systematyczne motywowanie do nauki. Niewątpliwie duże w tym znaczenie mają czynniki techniczne zachęcanie do angażowania się w określone działania;
* plagiat i ściąganie – procesy zdalne w edukacji znacznie ułatwiają niesamodzielność ucznia, poprzez ściąganie, kopiowanie cudzych prac czy przesyłanie sobie wzajemnie odpowiedzi przez uczestników kursu. Problem ten jest dużym wyzwaniem dla prowadzących i projektujących zajęcia zdalne, dlatego też aktywności przewidziane w kursie powinny wymagać pracy twórczej   
  i maksymalnej samodzielności;
* konieczność dużego nakładu pracy, jak i posiadania różnorodnych i złożonych kompetencji przy projektowaniu kursów – przed rozpoczęciem kursu zajęcia zdalne wymagają przygotowania wielu zasobów dydaktycznych taki jak opracowanie tematów czy zadania do samodzielnego wykonania. Samo projektowanie zasobów edukacyjnych niesie za sobą zaangażowania sporej liczby osób. Należy pamiętać, że w pełni wartościowe materiały powinni przygotowywać doświadczeni eksperci tacy jak: eksperci dziedzinowi, graficy i programiści. Niezwykle rzadko spotykany jest ekspert w konkretnej dziedzinie, a zarazem mający wysokie umiejętności komunikacyjne czy graficzne, a jednocześnie znający technologię informatyczne, by samodzielnie samemu opracować dobrej, jakości kurs e-learningowy.

Podsumowując atuty i ograniczenia zdalnego nauczania, można wyodrębnić, jakie wyzwania stoją przed twórcami, którzy chcą osiągnąć wysoki wynik w edukacji zdalnej. Należą do nich przede wszystkim: umiejętność stworzenia materiałów i aktywności dopasowanych do zdalnego procesu dydaktycznego, a także znajomość technik budowania stron społecznościowych oraz utrzymywania motywacji ucznia na wysokim poziomie [21].

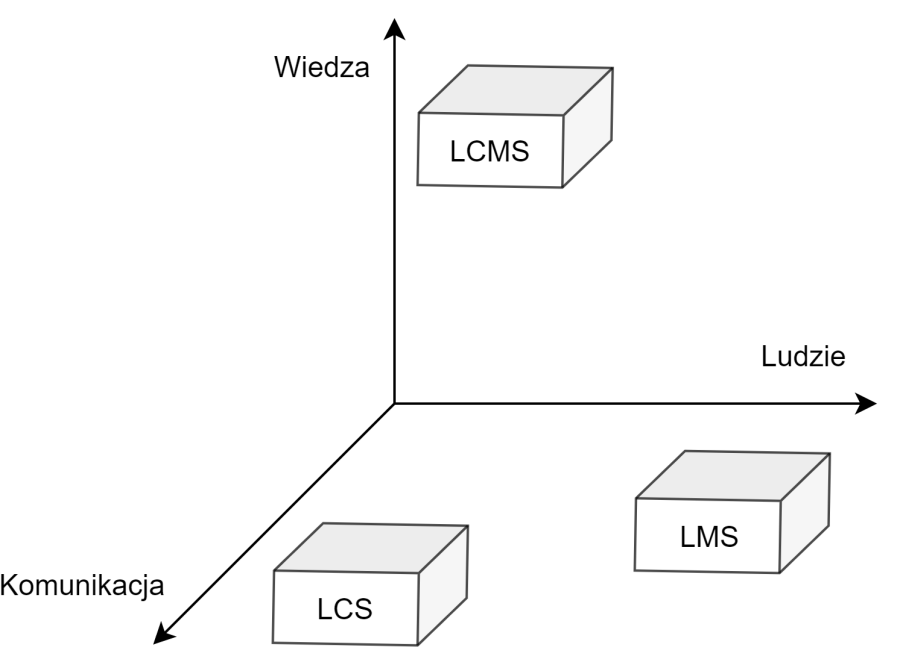
Wykorzystanie metod i technik zdalnego kształcenia oraz realizowanych przez nią postulatów, czyli: Just for me, Just enough, Just in time (spersonalizowane, elastyczne,   
w dogodnym czasie) wymuszają konieczność stworzenia przez nauczycieli i instytucje edukacyjne oferty dydaktycznej dostosowanej do subiektywnych potrzeb odbiorcy, jak również optymalizacji przekazywanej wiedzy, niezbędnej do osiągnięcia danego celu naukowego [17,13].

## 2.4 Przegląd dostępnych systemów e-learningowych

Ze względu na dużą ilość standardów e-learningowych, jaki i instytucji je stanowiących systemy zdalnego nauczania najczęściej dzieli się według rodzaju systemu   
i jego funkcjonalności. Możemy rozróżnić systemy [20, 5]:

* Learning Management System (LMS) – jego zadaniem jest pomoc w śledzeniu, zarządzaniu i raportowaniu aktywności szkoleniowych w organizacji. W zależności od rodzaju kursu aktywności LMS różnią się, jednak z punktu widzenia użytkownika końcowego zapewnia dynamiczne obserwowanie kompetencji   
  i umiejętności indywidualnego użytkownika, oraz posiada proste metody lokalizacji oraz rejestracji na określony kurs. Wartością dodaną do LMS jest szeroki zakres funkcjonalności;
* Life Communication System (LCS) – wspiera edukację zdalną synchroniczną. Jego funkcjonalności są ściśle związane z prowadzeniem zajęć na żywo poprzez wykorzystanie wideokonferencji. Prowadzone przez LCS zajęcia są archiwizowane a następnie wykorzystywane, jako obiekty szkoleniowe;
* Learning Content Management System (LCMS) – zadaniem systemu jest wspomaganie w tworzeniu, lokalizacji, zarządzaniu oraz ulepszaniu zawartości szkoleń. Materiały szkoleniowe są zwykle zarządzane w scentralizowanym archiwum. Ich forma jest różna, jednak najczęściej są one niewielkie, samo opisujące i identyfikowalne lub przedstawione, jako jednostki szkoleniowe,   
  z których każda satysfakcjonuje jeden lub więcej dobrze zdefiniowanych obiektów szkoleniowych. System potrafi odnaleźć i dostarczyć do danego użytkownika końcowego indywidualną jednostkę naukową, aby zaspokoić pojedynczy cel, lub dostarczać elementy większego obiektu edukacyjnego, zdefiniowanego w systemie LMS. Ponadto LCMS udostępnia osobom nadzorującym przejrzyste raporty, które mogą posłużyć do doskonalenia obiektów szkoleniowych. Systemy nie posiadają jednak dodatkowych funkcji administracyjnych i nie służą do zarządzania szkoleniami ani logistyką.

Na podstawie specyfikacji wyżej wymienionych systemów możemy określić ich ukierunkowania oraz aspekt rozwoju (Rys.2.4).



Rys. 2.4 Ukierunkowania poszczególnych rodzaj systemów zdalnego nauczania (opracowanie własne na podstawie [7])

Aktualnie na rynku znajduje się wiele systemów e-learnigowych działających głównie za zasadzie LMS. Do najpopularniejszych niekomercyjnych platform tego typu zaliczają się: Moodle, ILIAS, Caroline. Początkowo stosowane były głównie do ogólnego zapoznania się z e-learningiem i funkcjonalnościami oferowanymi przez witryny. Obecnie stosowane są na równi z platformami komercyjnymi. Ich kod źródłowy jest ogólnodostępny i może być dowolnie modyfikowany.

Biorąc pod uwagę polski obszar platform edukacyjnych na rynku dominuje platforma Moodle. Swój sukces zawdzięcza głównie darmowej licencji GNU General Public Licence, dzięki której zainteresowane osoby czy też instytucje mogą pobrać ją za darmo. Moodle charakteryzuje się szerokim wachlarzem dostępnych funkcjonalności zarówno w zakresie administracji, jak i zarządzania treściami. Ponadto jest jednym   
z najstarszych systemów tego typu. Posiada ona ponad 176 milionów użytkowników w 231 krajach na całym świecie [14].

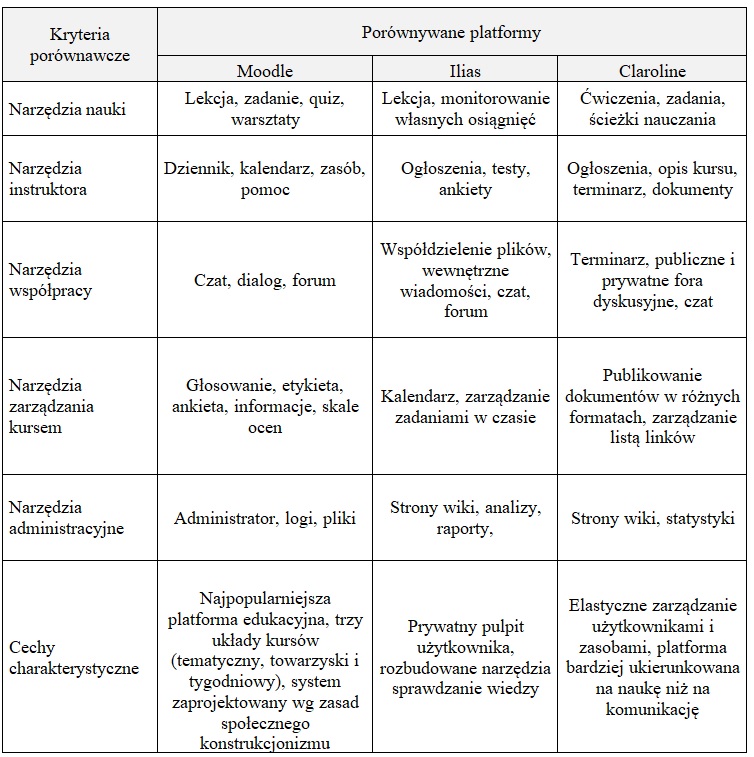
W krajach Europy Zachodniej, a przede wszystkim w krajach niemieckojęzycznych, ale nie tylko, coraz większą popularność zyskuje platforma ILIAS, która posiada ponad 25 tys. instalacji na całym świecie, i jest wykorzystywana zarówno   
w publicznych, jak i niepublicznych uczelniach. ILIAS podobnie jak platforma Moodle, posiada modułową budowę, jednak daje możliwości szerszej rozbudowy i dostosowania platformy do indywidualnych potrzeb użytkowników. Ponadto platforma jest zintegrowana z systemami niezbędnymi na uczelni, takimi jak moduł obsługi dziekanatu, co stanowi dodatkowy atut tego systemu [12].

Trzecim równie popularnym narzędziem e-learningowym jest Caroline. Powstała w Uniwersytecie w Louvain platforma obecnie posiada tłumaczenie na 35 języków i ma wielu użytkowników w około 100 krajach świata. Caroline tak jak dwie wymienione wcześniej platformy może elastycznie dostosowywać się do indywidualnych potrzeb organizacji, zaś same kursy w witrynie są projektowane na zasadzie tzw. ścieżki nauczania, która ściśle określa porządek przechodzenia przez poszczególne jego   
fragmenty [6].

Cechą wspólną zaprezentowanych platform jest ich darmowość, elastyczność oraz wysoka popularność. Mimo to każda z nich jest na swój sposób wyjątkowa poprzez oferowane użytkownikowi narzędzia oraz funkcje, co zostało zobrazowane w tabeli 2.1.

Równie ważną grupę w systemach zdalnego nauczania stanowią komercyjne platformy edukacyjne, jednakże z uwagi na ponoszenie dodatkowych kosztów za licencje są one stosunkowo rzadko spotykane w różnego typu firmach szkoleniowych czy też na uczelniach wyższych. Jednym z przykładów takiej platformy jest Blackboard.

Tabela 2.1 Porównanie wybranych platform e-learningowych (opracowanie własne na podstawie [16])



Możemy również zaobserwować, iż pojedyncze organizacje tworzą własne systemy informatyczne wyłącznie na indywidualne potrzeby. Przykładem tego typu rozwiązania jest platforma EDUX, która została stworzona w celu wspomagania e-learningu w Polsko-Japońskiej Akademii Technik Komputerowych [19].

Jednakże, jak wynika z badań uczelni w Polsce, wybór platformy edukacyjnej nie ma większego wpływu, na jakość publikowanych materiałów edukacyjnych. Zaś przeważającą formą materiałów udostępnianych w ramach kursów e-learningowych są pliki, co oznacza, że szkoły wyższe w Polsce nie wykorzystują interaktywnej roli kursów, a jedynie stosują platformy, jako repozytoria materiałów dla studentów. Wynika to głównie z braku odpowiedniego szkolenia wśród osób prowadzących kursy [19].

# 3. Projekt systemu

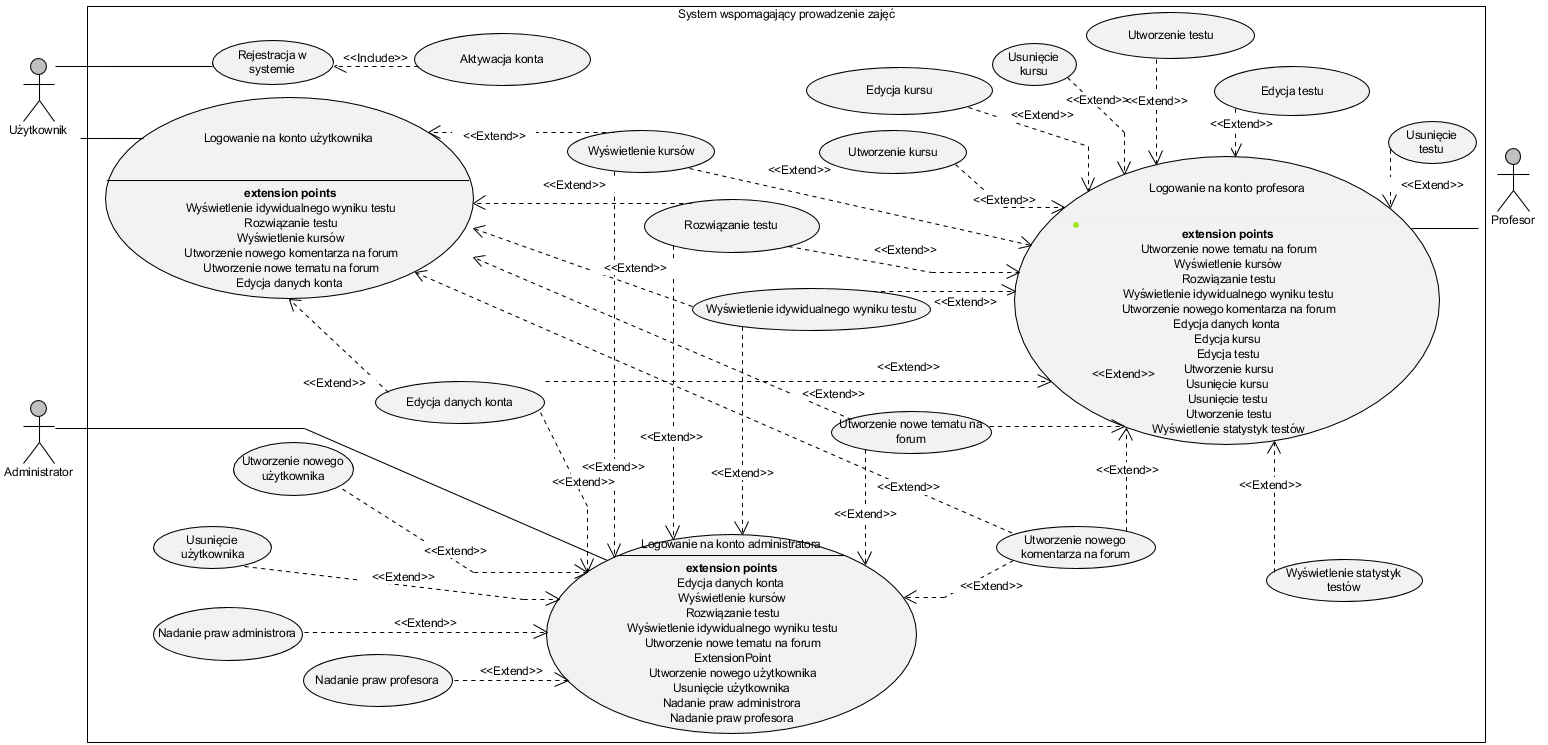
Nawiązując do przytoczonych uprzednio wad i zalet zdalnej edukacji oraz wyników badań przeprowadzonych na kilku uczelniach wyższych w Polsce, jednym z celów autora było zaprojektowanie systemu e-learningowego, który udostępni prowadzącym zajęcia łatwy interfejs tak, aby w prosty sposób mogli tworzyć interaktywne kursy przekazujące uczniowi wiedzę w miły i czytelny sposób.

## 3.1 Główne założenia i funkcjonalności

Aby osiągnąć zakładaną łatwość obsługi oraz interaktywność przekazu, autor już na starcie postawił kilka wymagań, które powinien spełniać projekt. Głównymi są:

* przejrzystość – łatwy dostęp do zawartości merytorycznej kurów a także dogodne przemieszczanie się po nich;
* dostęp z poziomu przeglądarki – do skorzystania z zasobów witryny nie będzie potrzebna instalacja dodatkowego oprogramowania;
* łatwość przyswajania wiedzy – moduły nie powinny być przeładowane treścią, ale także nie powinny być tylko hasłami jak w przypadku prezentacji;
* dostęp do materiałów wspomagających – moduły poza zwykłym, tekstowym wyjaśnieniem zagadnienia powinny zawierać dodatkowe źródła, jakimi są filmiki czy też linki do stron;
* sprawdzanie poziomu wiedzy – quizy powinny w znacznym stopniu ułatwić określenie stanu wiedzy użytkownika;
* uczenie się na błędach – po rozwiązaniu quizu użytkownik powinien posiadać możliwość wglądu do niego by móc określić, jakie błędy popełnił;
* możliwość komunikacji pomiędzy użytkownikami – umieszczenie w projekcie forum, jako dodatkowego źródła wymiany informacjami.

Pierwszymi krokiem, jaki wykonano, aby spełnić przedstawione założenia było zaprojektowanie diagramu przypadków użycia w celu uzyskania wglądu w to, jakie funkcjonalności zostaną zawarte w przyszłych fazach projektu. Stworzono diagram (Rys.3.1) oraz podstawowe przypadki użycia, czyli rejestracja nowego konta oraz logowanie na konto użytkownika.



Rys. 3.1 Diagram przypadków użycia

|  |  |
| --- | --- |
| Nazwa aktora | Opis |
| Użytkownik | Jest uczestnikiem kursów. Może on przeglądać dostępne dla niego kursy, rozwiązywać testy oraz sprawdzać błędy własne w już zakończonych przez siebie testach. |
| Profesor | Pełni rolę osoby zarządzającej kursami i testami. Dodaje on nowe materiały dydaktyczne do systemu, zarządza ich wyglądem i dostępnością. Posiada także możliwość podglądu statystyk oraz indywidualnych wyników testów. |
| Administrator | Pełni rolę osoby zarządzającej użytkownikami oraz profesorami. Ma możliwość dodania lub usunięcia użytkownika, zarządza również prawami w systemie. Ma on także dostęp do rozszerzonego panelu administracyjnego. |

Tabela 3.2 Opis tekstowy aktorów

Opis przypadku użycia „Rejestracja w systemie”:

1. Uczestniczący aktorzy:

* Użytkownik

1. Podstawowy ciąg zdarzeń:

* System wyświetla formularz rejestracji
* Użytkownik wpisuje dane nowego strażnika
* System weryfikuje kompletność danych
* System zapisuje dane do bazy
* System dokonuje wysłania wiadomości e-mail na adres nowego użytkownika w celu aktywacji konta

1. Alternatywne ciągi zdarzeń:
   1. System stwierdza niekompletność danych lub ich nieprawidłowość

* System ponownie wyświetla formularz
  1. Kierownik rezygnuje z tworzenia nowego konta
* W systemie nie są dokonywane żadne akcje

1. Zależności czasowe:
   1. częstotliwość wykonania: około 300 razy rocznie
   2. przewidywane spiętrzenia: podczas początku roku akademickiego
   3. typowy czas realizacji: ~5 min.
   4. maksymalny czas realizacji: nieokreślony
2. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia:

* Założenie nowego konta
* Komunikat informujący o powodzeniu lub niepowodzeniu operacji

Opis przypadku użycia „Logowanie na konto użytkownika”:

1. Uczestniczący aktorzy:

* Użytkownik

1. Podstawowy ciąg zdarzeń:

* System wyświetla formularz logowania
* Użytkownik wpisuje swój login i hasło oraz naciska przycisk ‘Zaloguj’
* System weryfikuje wprowadzone dane
* System dokonuje zalogowania użytkownika
* System wyświetla interfejs z wyborem wszystkich możliwych czynności do wykonania przez użytkownika
* Użytkownik dokonuje wyboru czynności, którą chce wykonać
* System uruchamia wybraną akcję

1. Alternatywne ciągi zdarzeń:
   1. System stwierdza niekompletność danych lub ich nieprawidłowość

* System ponownie wyświetla formularz logowania
  1. Użytkownik rezygnuje z zalogowania
* W systemie nie są dokonywane żadne akcje

1. Zależności czasowe:
   1. częstotliwość wykonania: w zależności od okresu semestru od ponad 3 razy dziennie do żadnego
   2. przewidywane spiętrzenia: po południu (około godziny 15)
   3. typowy czas realizacji: ~1 min.
   4. maksymalny czas realizacji: nieokreślony
2. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia:

* Dostęp do konta
* Komunikat informujący niepowodzeniu operacji

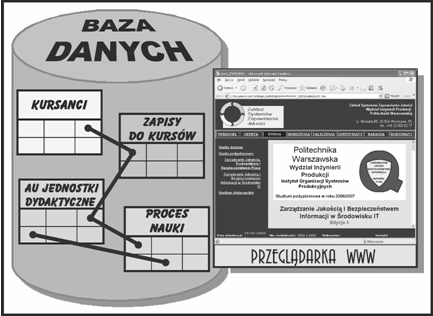
## 3.2 Ustandaryzowanie systemu

Po przeglądzie wszystkich przypadków użycia, a także analizie założeń określono, że projekt zostanie wykonany jako system LMS, spełniający podstawowe funkcje tego typu rozwiązania, czyli:

* wsparcie dla tworzenia szkoleń;
* stanowienie narzędzia administracyjnego dla szkoleń i użytkowników;
* raportowanie i analizy wyników szkoleń.

Ogólna charakterystyka zastosowania systemu LMS została już opisana na stronie 10 niniejszej pracy inżynierskiej.

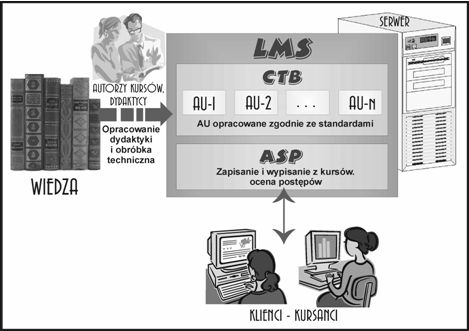
Z technologicznego punktu widzenia system zarządzania nauczaniem (LMS) to bazodanowa aplikacja działająca z wykorzystaniem interfejsu webowego, czyli w oparciu o przeglądarkę internetową. Posługuje się ona relacyjnym modelem baz danych, w którym informacje zapisywane są w tabelach zawierających wzajemne odniesienia do siebie. Interfejs między systemem e-learning a bazą danych stanowi standardowa przeglądarka internetowa. Przy jej zastosowaniu system LMS pobiera i zapisuje odpowiednie typy danych. Schemat tej aplikacji przedstawiono na Rys. 3.2. Widzimy na nim, iż system LMS zapisuje dane o historii procesu w odpowiedniej tabeli bazy danych, przyporządkowuje kursantów oraz rejestruje wiele inny informacje potrzebnych do poprawnego wykonania funkcji systemu.



Rys. 3.2 Schemat aplikacji LMS [20]

Struktura kursów w systemie jest bardzo podobna do struktury zwykłych stron internetowych. Każdy kursu składa się z materiałów dydaktycznych przedstawionych według konkretnego schematu zapisanego m.in. w języku HTML. Do takiej kompozycji dodano jednak specjalne skrypty oraz tagi, których zadaniem jest umożliwienie systemowi klasy LMS zarządzania procesem nauczania, który składa się z odbieranie informacji dotyczących na przykład wyników testów. Cały system działa w myśl modelu przedstawionego na Rys. 3.3, a jego głównymi składowymi są [20]:

* Assignable Unit (AU) - komputerowa jednostka dydaktyczna, uruchamiana   
  i śledzona przez system LMS;
* Computer Based Training (CTB) - grupa jednostek dydaktycznych (AU)   
  o określonej strukturze i funkcjonalności tworząca kurs;
* Application Service Provider (ASP) - system informatyczny udostępniający uczestnikom kursu usługi sytemu LMS.



Rys. 3.3 Model systemu e-learning klasy LMS [20]

Podsumowując wszystkie aspekty systemu zarządzania nauczaniem jego najistotniejszym aspektem z punktu widzenia projektowania jest stworzenie odpowiedniego modelu baza danych, który umożliwi szybki dostęp do informacji o wynikach szkoleń oraz umożliwienie łatwego zarządzanie użytkownikami a także łatwy odczyt i zapisu materiałów edukacyjnych.

## 3.3 Zasoby dydaktyczne

Podczas projektowania bazy wiedzy do witryny autor zwrócił uwagę na dwie główne cechy zasobów, jakie powinien posiadać odpowiedni materiał dydaktyczny. Należą do nich:

* estetyczny wygląd elektronicznej postaci zasobu;
* ścisły związek zasobu z zamierzonym celem szkoleniowym.

Logiczne zestawienie elektronicznych materiałów szkoleniowych wraz z planem aktywności ucznia wykorzystano jako podstawę struktury dydaktycznej.

W technicznym podziale zasobów autor wziął pod uwagę rodzaj przekazywanej treści wraz z formą realizacji procesu dydaktycznego oraz posiadanymi narzędziami technicznymi. W ten sposób wyodrębniono trzy zasadnicze zasoby wchodzące w skład witryny:

* tekstowe – jako lektura z ekranu komputera;
* pliki – jako format PDF w wersji do pobrania na komputer lub urządzenie mobilne;
* wideo – jako wybrane filmy z platformy YouTube wspomagające, w sposób wizualny, zdobywanie wiedzy.

Wzięto również pod uwagę zasoby sprawdzające osiągniętą wiedzę i umiejętność poprzez skonstruowanie testów w formie pytań z 4 wariantami odpowiedzi. Będącymi nawiązaniem do poszczególnych tematów realizowanych podczas kursu. Według zasady „uczenie się na błędach” po rozwiązaniu danego zasobu testowego użytkownik posiada możliwość wglądu do niego by móc określić, jakie błędy popełnił.

Do stworzenia zasobów dydaktycznych jak i materiałów do testów w witrynie wykorzystano udostępnione, dzięki uprzejmości promotora, prywatne materiały dydaktyczne jak również zebrane przez autora dodatkowe materiały oraz linki do wybranych filmów wspomagających, które wspólnie tworzą kompendium wiedzy wspomagającej edukację z zakresu przedmiotu Wprowadzenie do Informatyki.

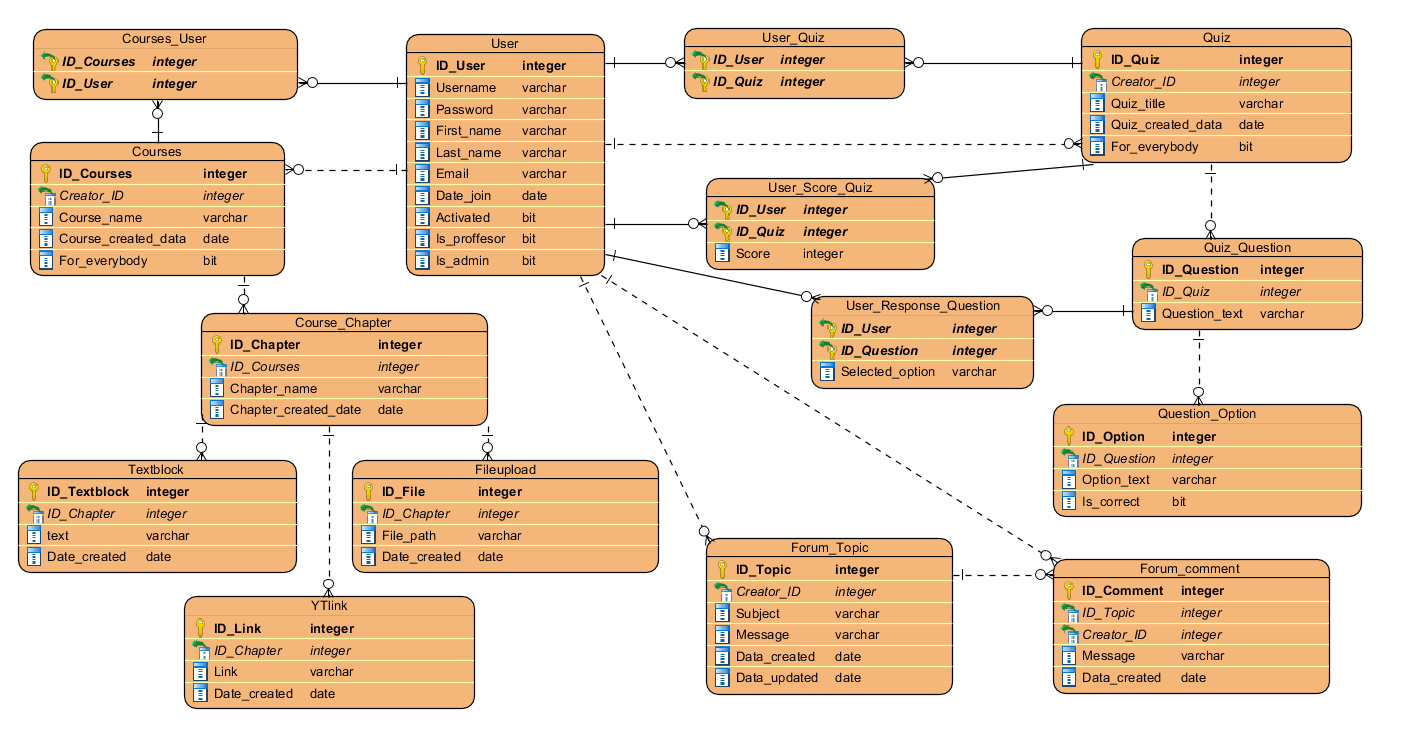
Elementem uzupełniającym, który wspomaga uczniów w zdobywaniu wiedzy przy pomocy witryny jest forum dyskusyjne. Umożliwia ono każdemu użytkownikowi aktywną współpracę w tworzeniu dodatkowych zasobów wiedzy lub dyskusje na temat już istniejących.

## 3.4 Baza danych

W celu stworzenia funkcjonalnej baza danych zawierającej wszystkie opracowane uprzednio założenia, co do użytkowników jak i zasobów wiedzy, sporządzono listę istotnych założeń potrzebnych do skonstruowania odpowiedniej struktury bazy:

* kurs może być udostępniony dla wszystkich użytkowników lub tylko dla wybranych jednostek;
* kurs może mieć tylko jednego autora i zawierać dowolną ilość rozdziałów;
* rozdział jest przypisany tylko do jednego kursu;
* wszystkie tematy mogą zawierać każdy z trzech zasobów:
  + tekst;
  + link do platformy YouTube;
  + plik.
* pojedyncze zasoby dydaktyczne są przyporządkowane do tematu;
* użytkownik ma możliwość rozwiązania quizu tylko raz;
* po rozwiązaniu quizu użytkownik ma możliwość sprawdzić jego wynik lub poprawność odpowiedzi;
* każde pytanie posiada tylko jedną prawidłową odpowiedź;
* każdy użytkownik ma możliwość dodanie tematu jak i komentarza na forum;
* temat może zawierać dowolną ilość komentarzy.

Na ich podstawie zaprojektowano Entity Relationship Diagram (ERD) będący odzwierciedleniem zawartych danych oraz relacji występujących pomiędzy nimi. Sposób organizacji informacji w bazie został określony w formie tabel zawierających rekordy podzielone na pola, w których są przechowywane informacje poszczególnych kategorii. W poszczególnych polach zapisywane są dane oddzielnych kategorii. Dzięki temu baza danych umożliwia szybkie sortowanie rekordów według poszczególnych kategorii lub wyszukiwanie informacji w obrębie tylko wybranych pól. Poniżej przedstawiono schemat relacyjnej bazy danych (Rys.3.4).



Rys. 3.4 Schemat bazy danych

Dla ułatwienia odczytu struktury bazy sporządzono specyfikację poszczególnych tabel oraz zawartych w nich pól:

* tabela User będąca podstawą systemu, przechowywane są w niej dane na temat wszystkich użytkowników systemu, zawiera podstawowe informacje takie jak nazwę użytkownika, zaszyfrowane hasło, czy też mail. Dodatkowo znajdują się w niej informacje na temat aktywacji konta oraz uprawnień użytkownika w systemie (Tabela 3.2);

Tabela 3.3 Struktura oraz typy danych w tabeli User

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_User | integer | Unikalny identyfikator każdego użytkownika |
| Username | varchar | Nazwa użytkownika |
| Password | varchar | Zaszyfrowane hasło |
| First\_name | varchar | Imię użytkownika |
| Last\_name | varchar | Nazwisko użytkownika |
| Email | varchar | Email użytkownika |
| Date\_join | Date | Data dodania użytkownika |
| Activated | Bit | Wartość określająca czy konto zostało aktywowane |
| Is\_proffesor | Bit | Wartość określająca czy użytkownik jest profesorem |
| Is\_admin | Bit | Wartość określająca czy użytkownik jest administratorem |

* tabela Courses\_User jest wiążąca dla tabel Courses oraz User w celu zapewnienia relacji wiele do wielu. Zawiera informacje na temat kluczy głównych wiązanych tabel (Tabela 3.3);

Tabela 3.4 Struktura oraz typy danych w tabeli Courses\_User

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Courses | integer | Klucz obcy powiązany z tabelą Couses |
| ID\_User | integer | Klucz obcy powiązany z tabelą User |

* tabela Courses przechowująca dane na temat wszystkich kursów w systemu, zawiera podstawowe informacje takie jak nazwa kursu, dane o twórcy czy też datę utworzenia. Dodatkowo znajduje się w niej informacja czy kurs jest dostępny dla wszystkich użytkowników (Tabela 3.4);

Tabela 3.5 Struktura oraz typy danych w tabeli Courses

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Courses | integer | Unikalny identyfikator każdego kursu |
| Creator\_ID | integer | Klucz obcy powiązany z tabelą User |
| Course\_name | varchar | Tytuł kursu |
| Course\_created\_data | date | Data powstania kursu |
| For\_everybody | Bit | Wartość określająca czy kurs jest dostępny dla wszystkich użytkowników |

* tabela Courses\_Chapter zawierająca dane na temat działów zawartych w kursach, przechowuje informacje o nazwie działu, powiązaniu z kursem oraz dacie utworzenia (Tabela 3.5);

Tabela 3.6 Struktura oraz typy danych w tabeli Courses\_Chapter

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Chapter | integer | Unikalny identyfikator każdego działu |
| ID\_Courses | integer | Klucz obcy powiązany z tabelą Courses |
| Chapter\_name | varchar | Nazwa działu |
| Chapter\_created\_date | Date | Data stworzenia działu |

* tabela Textblock przechowuje dane zasoby tekstowego zawartego w dziale oraz powiązanie z działem i datę utworzenia (Tabela 3.6);

Tabela 3.7 Struktura oraz typy danych w tabeli Textblock

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Textbloc | Integer | Unikalny identyfikator każdego zasobu tekstowego |
| ID\_Chapter | Integer | Klucz obcy powiązany z tabelą Chapter |
| Text | varchar | Tekst zasobu |
| Date\_created | Date | Data stworzenia zasobu |

* tabela YTlink przechowuje link do filmu z platformy YouTube oraz powiązanie z działem i date utworzenia (Tabela 3.7);

Tabela 3.8 Struktura oraz typy danych w tabeli YTlink

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Link | integer | Unikalny identyfikator każdego linku |
| ID\_Chapter | integer | Klucz obcy powiązany z tabelą Chapter |
| Link | varchar | Link do zasobu |
| Date\_created | Date | Data stworzenia zasobu |

* tabela Fileupload przechowuje dane pliku zawartego w dziale oraz powiązanie z działem i datę utworzenia (Tabela 3.8);

Tabela 3.9 Struktura oraz typy danych w tabeli Fileupload

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_File | integer | Unikalny identyfikator każdego pliku |
| ID\_Chapter | integer | Klucz obcy powiązany z tabelą Chapter |
| File\_path | varchar | Ścieżka do pliku |
| Date\_created | Date | Data stworzenia zasobu |

* tabela User\_Quiz jest wiążąca dla tabel User oraz Quiz w celu zapewnienia relacji wiele do wielu. Zawiera informacje na temat kluczy głównych wiązanych tabel (Tabela 3.9);

Tabela 3.10 Struktura oraz typy danych w tabeli User\_Quiz

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_User | Integer | Klucz obcy powiązany z tabelą User |
| ID\_Quiz | Integer | Klucz obcy powiązany z tabelą Quiz |

* tabela Quiz przechowująca dane na temat wszystkich testów w systemie, zawiera podstawowe informacje takie jak nazwa testu oraz dane o twórcy czy też datę utworzenia. Dodatkowo znajduje się w niej informacja czy test jest dostępny dla wszystkich użytkowników (Tabela 3.10);

Tabela 3.11 Struktura oraz typy danych w tabeli Quiz

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Quiz | integer | Unikalny identyfikator każdego testu |
| Creator\_ID | integer | Klucz obcy powiązany z tabelą User |
| Quiz\_title | varchar | Tytuł testu |
| Quiz\_created\_data | Date | Data stworzenia testu |
| For\_everybody | Bit | Wartość określająca czy test jest dostępny dla wszystkich użytkowników |

* tabela Quiz\_Question zawierająca dane na temat pytania zawartego w teście, przechowuje tekst pytania oraz powiązanie z testem oraz dacie utworzenia  
   (Tabela 3.11);

Tabela 3.12 Struktura oraz typy danych w tabeli Quiz\_Question

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Question | integer | Unikalny identyfikator każdego pytania |
| ID\_Quiz | integer | Klucz obcy powiązany z tabelą Quiz |
| Question\_text | varchar | Treść pytania |

* tabela Quiz\_Option zawierająca dane na temat opcji odpowiedzi zawartego w pytaniu, przechowuje tekst opcji i powiązanie z pytaniem oraz informację czy opcja jest poprawna odpowiedzią na pytanie (Tabela 3.12);

Tabela 3.13 Struktura oraz typy danych w tabeli Quiz\_Option

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Option | integer | Unikalny identyfikator każdego opcji |
| ID\_Question | integer | Klucz obcy powiązany z tabelą Questiont |
| Option\_text | varchar | Treść opcji odpowiedzi |
| Is\_correct | Bit | Wartość określająca czy opcja jest prawidłową odpowiedzią na pytanie |

* tabela User\_Score\_Quiz zawierająca dane o wyniku użytkownika osiągniętego w teście, przechowuje wynik użytkownika oraz podwójny klucz główny składający się z powiązania klucza tabeli User oraz Quiz (Tabela 3.13);

Tabela 3.14 Struktura oraz typy danych w tabeli User\_Score\_Quiz

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_User | integer | Klucz główny (Klucz obcy powiązany z tabelą User) |
| ID\_Quiz | integer | Klucz główny (Klucz obcy powiązany z tabelą Quiz) |
| Score | integer | Wartość wyniku |

* tabela User\_Response\_Question zawierająca dane o odpowiedzi użytkownika na pojedyncze pytanie, przechowuje wybraną opcję przez użytkownika oraz klucz podwójny wiążący tabele User oraz Quiz\_Question (Tabela 3.14);

Tabela 3.15 Struktura oraz typy danych w tabeli User\_Response\_Question

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_User | integer | Klucz główny (Klucz obcy powiązany z tabelą User) |
| ID\_Question | integer | Klucz główny (Klucz obcy powiązany z tabelą Question) |
| Selected\_option | varchar | Wybrana odpowiedz |

* tabela Forum\_Topic przechowująca dane na temat wszystkich tematów rozpoczętych na forum: tytuł, zawartość tematu, informację o twórcy oraz daty stworzenia i ostatniej aktywności w temacie (Tabela 3.15);

Tabela 3.16 Struktura oraz typy danych w tabeli Forum\_Topic

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Topic | integer | Unikalny identyfikator każdego tematu |
| Creator\_ID | integer | Klucz obcy powiązany z tabelą User |
| Subject | varchar | Tytuł tematu |
| Massage | varchar | Zawartość tematu |
| Date\_created | Date | Data stworzenia tematu |
| Date\_updated | Date | Data ostatniej aktywności w temacie |

* tabela Forum\_Comment zawierające dane na temat komentarzy pod tematami na forum, zawiera informacje takie jak treść komentarza, powiązanie z tematem, informację o twórcy oraz daty stworzenia wpisu (Tabela 3.16);

Tabela 3.17 Struktura oraz typy danych w tabeli Forum\_Comment

|  |  |  |
| --- | --- | --- |
| NAZWA POLA | TYP | OPIS |
| ID\_Comment | integer | Unikalny identyfikator każdego komentarza |
| ID\_Topic | integer | Klucz obcy powiązany z tabelą Topic |
| Creator\_ID | integer | Klucz obcy powiązany z tabelą User |
| Message | varchar | Zawartość komentarza |
| Date\_created | date | Data stworzenia komentarza |
|  |  |  |

## 3.5 Podstawowe założenia interfejsu

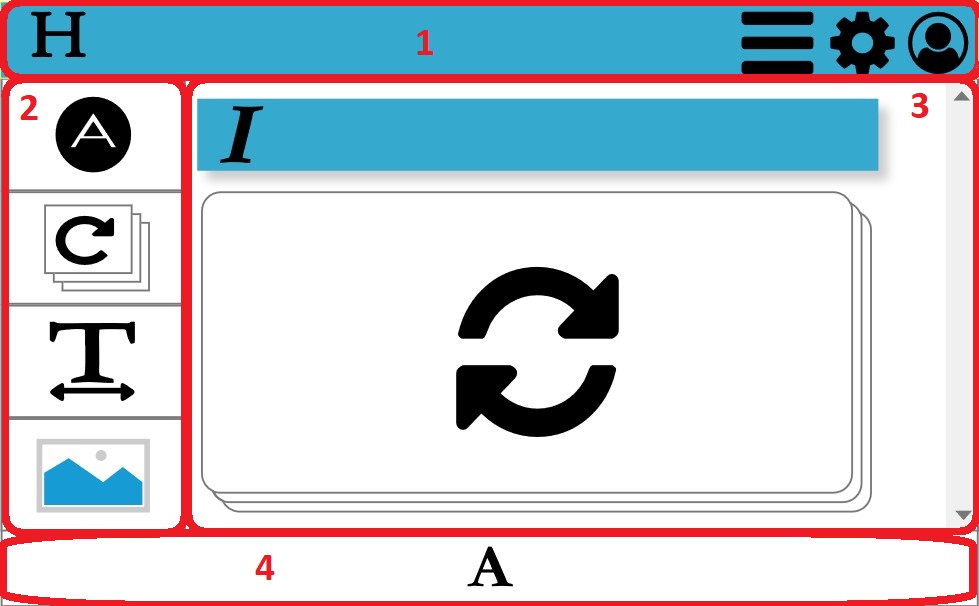
Ostatnim aspektem, na jakim skupił się autor podczas tworzenia projektu systemu było stworzenie koncepcji przejrzystego interfejsu. Zwrócono uwagę na kilka skutecznych sposobów zwiększających intuicyjność oraz czytelności strony, które ułatwiają zarówno uczniowi jak i nauczycielowi realizację zadań postawionych przed nimi. Są to miedzy innymi:

* zastosowanie standaryzowanej topografii strony – rozmieszczenie głównych elementów strony takich jak pasek nawigacyjny w standardowych miejscach przyczynia się do łatwiejszego poruszania po witrynie;
* zastosowanie ikon - pozwala na połączenie wykonywanej przez użytkownika czynność z jej symbolem wizualnym, co ułatwia mu orientację w interfejsie i uczenie się go;
* graficzne odseparowanie fragmentów strony – umożliwia wizualne odizolowanie poszczególnych fragmentów strony, co wykorzystywano, jako metodę zwiększenia użyteczności serwisu;
* użycie technologii szybkiej zmiany kontekstu – mechanizm ten oferuje użytkownikowi błyskawiczny dostęp do kilku różnych treści lub funkcji, bez konieczności przeładowywania całej zawartości strony;
* wykorzystanie zróżnicowanego rozmiaru elementów – pozwala na ustalenie hierarchii ważności poszczególnych elementów strony dla użytkownika.

Na postawie przytoczonych wyżej założeń opracowano prototypowy wygląd interfejsu strony (Rys.3.5) oraz jego główne składowe:

1. pasek nawigacji;
2. dynamiczny pasek boczny;
3. główny widok strony działający w mechanizmu szybkiej zmiany kontekstu;
4. stopka.

Ponadto zaprojektowano również wygląd strony kursów oraz forum (Rys. 3.6).



Rys. 3.5 Projekt ogólnego wyglądu interfejsu strony



Rys. 3.6 Projekt głównego widoku strony: a) kursu b) forum

Kończąc proces projektowania wyglądu aplikacji autor skupił się na wymyśleniu nazwy oraz logotypu dla tworzonego systemu tak, aby był on łatwo kojarzony oraz zapamiętywany przez użytkowników. Biorąc pod uwagę aspekt, iż platforma ma służyć wspomaganiu nauczania z przedmiotu Wprowadzenie do Informatyki (ang. *Introduction to Computer Science*) postanowiono wykorzystań pierwsze litery angielskiej nazwy przedmiotu i stworzono nazwę ICoS, którą następnie umieszczono w logo (Rys.3.7).



Rys. 3.7 Logo sytemu

# 4. Implementacja projektu

Na podstawie wszystkich stworzonych uprzednio schematów oraz założeń zarówno funkcjonalnych jak i niefunkcjonalnych autor przystąpił do implementacji systemu. Proces ten rozpoczęto od wyboru łatwo interpretowalnego języka programowania posiadającego szeroko rozwiniętą platformę programistyczną

## 4.1 Języki programowania, framworki oraz środowisko programistyczne

Po porównaniu kilku dostępnym na rynku rozwiązań programistycznych autor do tworzenia aplikacji wybrał platformę Django będącą frameworkiem języka wysokiego poziomu Python. Język ten jest językiem obiektowym pozwalającym na szybkie i łatwe tworzenie wielomodułowych aplikacji, łączy ze sobą niezwykłą moc z bardzo przejrzystą składnią, posiada interfejsy do wielu wywołań systemowych, a także różnych systemów okienkowych. Dodatkowo jest użytecznym językiem rozszerzeń dla aplikacji napisanych w innych językach wymagających użycia łatwych interfejsów skryptów lub   
automatyzacji [18].

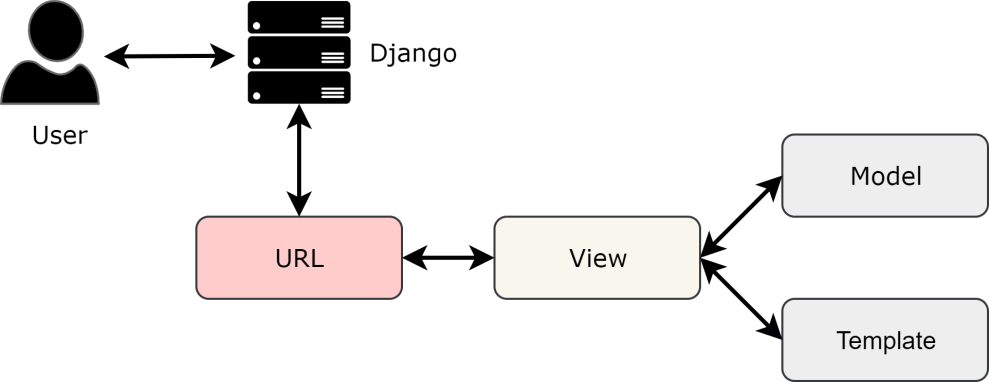
Sama platforma internetowa Django została stworzona przez doświadczonych programistów i już na starcie niweluje podstawowe problemy związane z tworzeniem stron internetowych, dzięki czemu od początku procesu implementacji można skupić się na pisaniu aplikacji bez konieczności tworzenia skomplikowanych systemów do obsługi np. zapytań w bazie danych.

Głównymi cechami frameworku Django są:

* kompletność – zapewnienia prawie wszystko, czego programista może potrzebować w procesie programowania, dzięki czemu wszystko jest spójne i posiada obszerną dokumentację;
* wszechstronność – może być wykorzystywany do budowy niemal każdego rodzaju strony internetowej. Może współpracować z dowolnym frameworkiem po stronie klienta i dostarczyć treści w prawie każdym formacie;
* bezpieczeństwo – wspomaga programistę w uniknięciu wielu błędów bezpieczeństwa poprzez włączenie ochrony przed wieloma lukami w zabezpieczeniach.

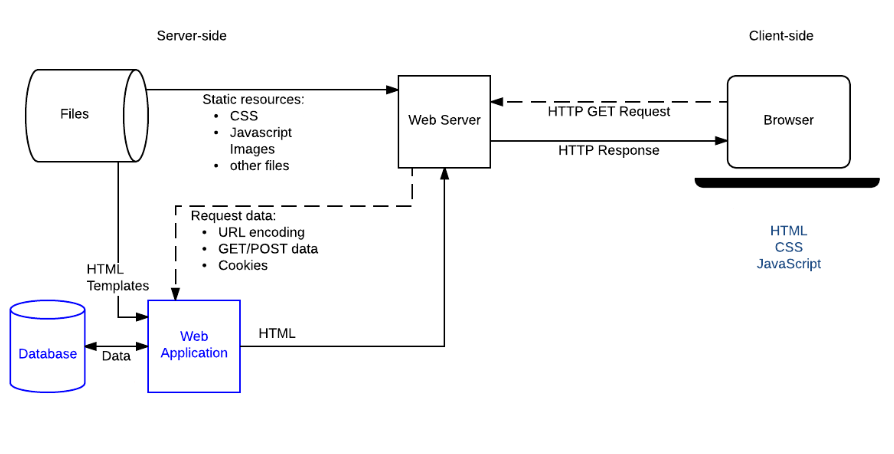
Dodatkowo posiada on aktywną społeczność, świetną dokumentację oraz wiele opcji bezpłatnego wsparcia [9].

Django oparte jest na architekturze Model-View-Template (MVT, Rys. 4.1) , która jest pokrewna klasycznej architektury Model-View-Controller (MVC) jednakże rolę Controlera przejmuje sam Framework, zaś View jest rozbite na dwie części współpracujące ze sobą, View odpowiadające za przygotowanie danych do prezentacji zgodne z wskazanym Uniform Resource Locator (URL) oraz Template odpowiadające za prezentację danych przygotowanych w View.



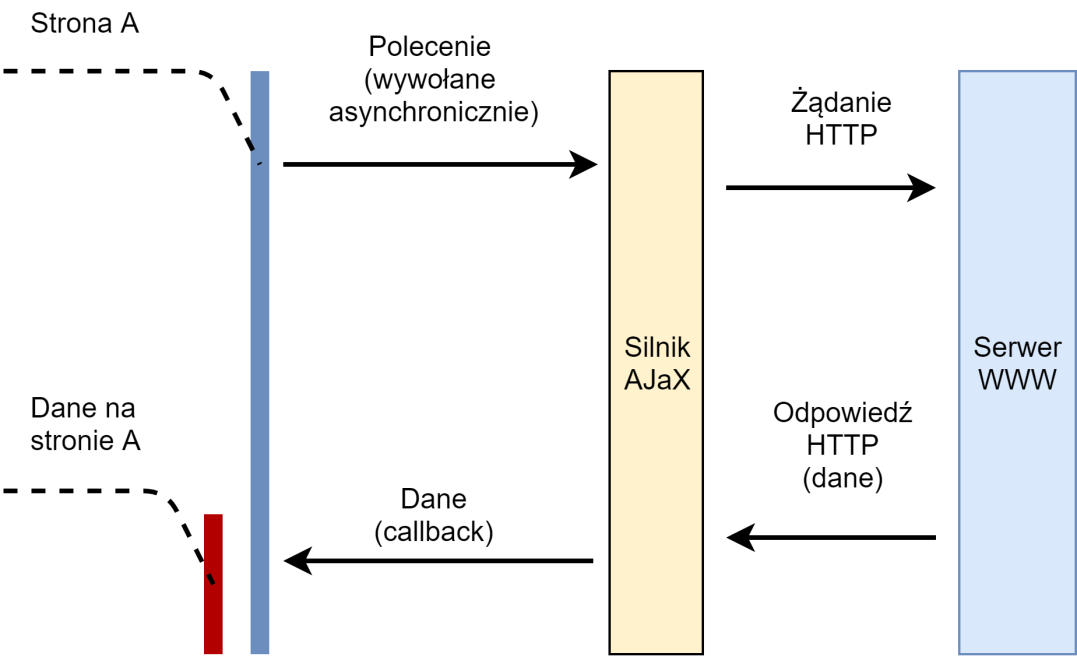
Rys. 4.1 Struktura architektury Model-View-Template (opracowanie własne nana podstawie [8])

Użyta platforma dzięki swej architekturze pozwoliła na stworzenie dynamicznej strony, której kod do obsługi znajduje się na serwerze. Utworzenie takiej właśnie struktury witryny znane jest, jako programowanie po stronie serwera i pozwala na aktywne wyświetlanie danych w razie potrzeby, poprzez wysłanie żądania zasobów dynamicznego do serwera, gdzie odbywa się odczytywanie odpowiednich danych z bazy, łączenie ich z szablonem a następnie odsyłanie odpowiedzi zawierającej wygenerowany HTML (Rys.4.2)



Rys. 4.2 Schemat działania kodu po stronie serwera (opracowanie własne na postawie [8])

Najbardziej znaczącą zaletą takiego kodowania jest to, że pozwala on na dostosowanie zawartości witryny do indywidualnych użytkowników oraz umożliwia jego większą interakcję ze stroną. Ponadto podczas tworzenia aplikacji zastosowano technologię asynchronicznego JavaScript i XML (AJaX), która pozwoliła stworzyć przyjazne środowisko interakcji użytkownika z serwerem, które odbywa się bez konieczności przeładowywania całej strony HTML jak ma to miejsce w tradycyjnym modelu strony (Rys.4.3).



Rys. 4.3 Schemat działania technologii AJaX (opracowanie własne na postawie [8])

Do stworzenia warstwy prezentacji w witrynie użyto języków stanowiących jej podstawę, czyli: HTML, CSS i JavaScript, oraz powiązanej z nim biblioteki jQuery. Wykorzystano również Framework UiKit, będący odpowiednikiem Bootstrapa, który poprzez oferowane moduły i rozszerzalne komponenty znacznie uatrakcyjnił wygląd witryny.

Jako środowisko programistyczne autor wybrał darmowy i uniwersalny edytor Visual Studio Code. Oprogramowanie to zostało stworzone prze firmę Microsoft i posiada wsparcie dla użytych w projekcie języków programowania oraz pozwala na zarządzanie wersjami kodu z systemu Git, co znacznie ułatwia pracę podczas tworzenia jak i eksperymentowania z nowo powstałym oprogramowaniem.

## 4.2 Tworzenie modeli i funkcjonalności

Tworzenie struktury programu rozpoczęto od implementacji poszczególnych modeli w bazie danych będącej podstawą projektu oraz architektury implementacyjnej. Do stworzenia struktury modelu użytkownika została użyta udostępniona przez Django moduł django.contrib.auth zawierający podstawowe pola i metody potrzebne do obsługi użytkowników w systemie. Rozszerzono ją jednak o dwa pola określające uprawnienia w systemie (Listing 4.1).

Listing 4.1 Budowa modelu użytkownika \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

from django.contrib.auth.models import AbstractUser

class UserProfile(AbstractUser):

    is\_professor = models.BooleanField(default=False)

    is\_site\_admin = models.BooleanField(default=False)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dzięki kompletności użytej platformy programistycznej autor został niejako wyręczony w tworzeniu tabel tworzących relację wiele do wielu będącą postawą połączenia tabeli użytkownika z kursem (Listing 4.2) czy też testem (Listing 4.3).

Listing 4.2 Budowy modelu kursu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

class Course(models.Model):

    course\_name = models.CharField(unique=True, max\_length=40)

    course\_created\_date = models.DateTimeField(auto\_now\_add=True)

    user = models.ForeignKey(UserProfile, default=1)

    student = models.ManyToManyField(UserProfile, related\_name='student\_to\_course')

    for\_everybody = models.BooleanField(default=True)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Listing 4.3 Budowa modelu testów \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

class Test(models.Model):

   quiz\_title = models.CharField(max\_length=100)

   course\_created\_date = models.DateTimeField(auto\_now\_add=True)

   student = models.ManyToManyField(UserProfile, related\_name='student')

   teacher = models.ForeignKey(UserProfile, on\_delete=models.CASCADE,   
 related\_name='teacher')

   for\_everybody = models.BooleanField(default=True)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Pozostałe modele budowane były na podstawie schematu bazy stworzonego na etapie projektowania witryny. Jedynym aspektem, jaki uległ zmianie w stosunku do projektu był typ pól użytych w bazie, i tak: typ bit został zastąpiony boolean, a typ varchar w niektórych przypadkach zastoiny został typem char.

W późniejszy etapach projektowania w poszczególnych modelach pojawiły się metody rozszerzające ich funkcjonalność, przykładem tego jest zestaw metod zawracający całkowitą ścieżkę (URL) konkretnego działu w kursie (Listing 4.4).

Listing 4.4 Funkcja zwracająca ścieżkę do kursu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

def get\_absolute\_url(self):

    return reverse("chapter", kwargs={"course\_name": self.course,

                                      "slug": self.slug})

def slug\_default(self):

    slug = create\_slug(new\_slug=self.chapter\_name)

    return slug

def create\_slug(instance=None, new\_slug=None):

    slug = slugify(instance.chapter\_name)

    if new\_slug is not None:

        slug = new\_slug

    qs = Chapter.objects.filter(slug=slug).order\_by("-id")

    exists = qs.exists()

    if exists:

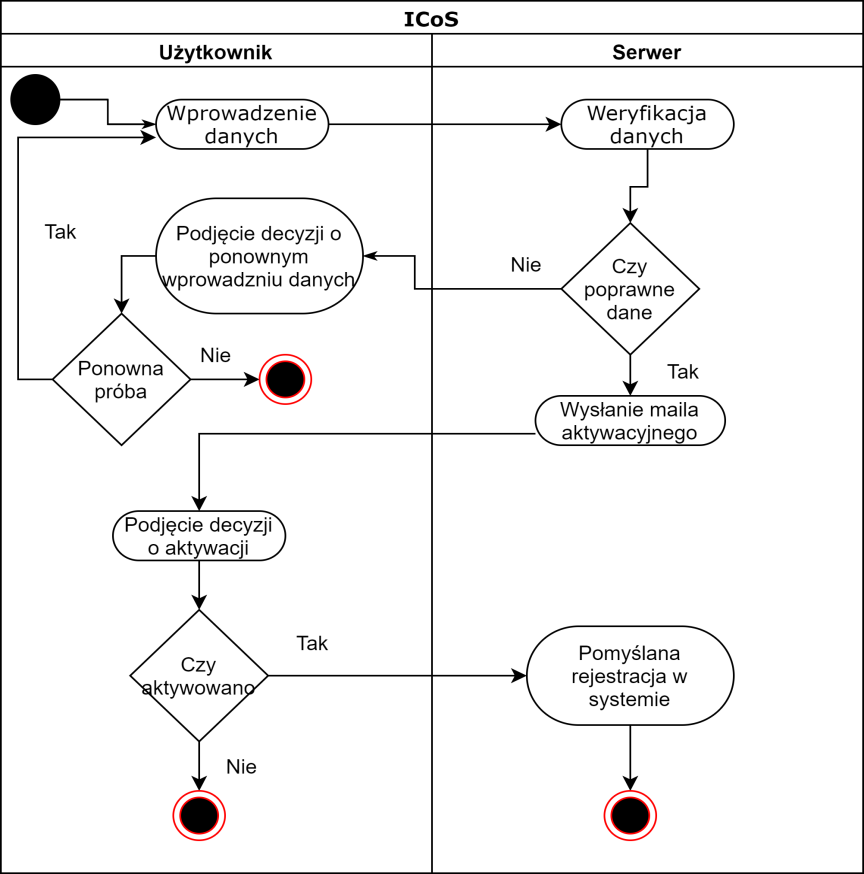
        new\_slug = "%s-%s" % (slug, qs.first().id)

        return create\_slug(instance, new\_slug=new\_slug)

return slug

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Następną fazą implementacji było zdefiniowanie poszczególnych funkcji w View odpowiadających za pobieranie danych z modeli bazy i przesyłanie ich do szablonów oraz za funkcjonalności w nich występujące.



Rys. 4.4 Rejestracja w systemie – diagram czynności

Jako pierwszy stworzony został cały moduł obsługi użytkownika posiadający funkcjonalności tak jak rejestracja, logowanie, zmiana szczegółów konta czy też przypominanie hasła. Dzięki zastosowaniu rozszerzenia django-registration autor stworzył dwufazową rejestrację składająca się z początkowej rejestracji, a następnie wiadomości e-mail z potwierdzeniem i instrukcjami aktywacji nowego konta (Rys. 4.4). Podczas tworzenia maila generowany jest token bez utrwalania go w bazie danych, do generowania wykorzystywany jest SECRET\_KEY aplikacji, więc jest to bezpieczna i niezawodna metoda uwierzytelniająca.

Do tworzenia formularzy zarówno tych do obsługi użytkownika jak i zbudowanych w późniejszym etapie modułów użyte zostało rozszerzenie django-crispy-form. Jest to aplikacja Django, która pozwala łatwo budować, dostosowywać i ponownie wykorzystywać formularze przy użyciu ulubionego narzędzia CSS, bez pisania kodu szablonu. Dodatkowo chipsy-form pozwala na tworzenie funkcji walidacji oraz wyświetlania komunikatów o błędach. Poniższy przykład prezentuje budowę przykładowej komórki formularza oraz jej funkcje walidacji.

Listing 4.5 Budowa komórki formularza oraz jej funkcja walidacyjna \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

class UserDetailsChangeForm(forms.Form):

    username = forms.RegexField(regex=r'^[\w.@+-]+$',

       max\_length=150,

       label=mark\_safe('<i class="uk-icon-user-secret"></i> Username'),

       help\_text="Username can contain only letters, numbers and @/./+/-/\_ characters.",

       error\_messages={'invalid': "This value may contain only letters, numbers and @/./+/-/\_ characters."},

       widget=forms.TextInput(attrs={'placeholder': 'Username', 'class': "uk-width-1-1"}))

...

def clean\_username(self):

   username = self.cleaned\_data['username']

   if User.objects.filter(username=username).exclude(id=self.user.id).exists():

       raise ValidationError("A user with that username already exists.")

   return username

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Kolejnym etapem implementacji było utworzenie funkcjonalności pozwalających na tworzenie kursów w witrynie, w tym celu zbudowano panel profesora pozwalający na zarządzanie kursami. Do głównych funkcjonalności tego obszaru należą: dodanie, usuwanie oraz edycję zasobów tekstowych, plikowych jak i linków do platformy YouTube. Utworzenie materiału w kursie odbywa się poprzez sprawdzanie poprawności wprowadzanych danych, a następnie dodawania nowej instancji do bazy. Listing 4.6 ilustruje nam dodanie linku filmu YouTube do bazy poprzez wprowadzanie adresu URL wybranego wideo. Na starcie tworzona jest instancja, następnie sprawdzane jest czy ścieżka posiada wzmiankę „osadź”(ang. *embed*), dzięki której film będzie mógł być wyświetlany w materiałach strony, jeśli link spełnia postawiony warunek instancja jest zapisywana do bazy.

Listing 4.6 Metoda walidacyjna linku do platformy YouTube \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

if add\_link\_form.is\_valid() and 'add\_link' in request.POST:

     instance = add\_link\_form.save(commit=False)

     instance.yt\_link\_fk = Chapter.objects.get(id=place.id)

     key = add\_link\_form.cleaned\_data.get("link")

     if 'embed' not in key and 'youtube' in key:

         key = key.split('=')[1]

         instance.link = 'https://www.youtube.com/embed/' + key

         instance.yt\_link\_fk = Chapter.objects.get(id=place.id)

         instance.save()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

W panelu profesora zaimplementowano również obsługę testów znajdujących się w witrynie, stworzono możliwości dodania, usuwania, edycji testu oraz wglądu w wyniki uzyskane przez użytkowników w teście (Listing 4.7). Przechodząc do konkretnego testu możemy utworzyć lub usunąć pytania w nim zawarte.

Listing 4.7 Funkcja zwracająca wyniki wszystkich użytkowników uzyskanych w teście \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

def view\_result(request, quiz\_id):

    quiz = Test.objects.get(quiz\_id=quiz\_id)

    all\_students = quiz.student.all()

    dict = {}

    for t in all\_students:

        q = Test.objects.get(quiz\_title=quiz\_title, student=t)

        dict[t] = t.score\_set.get(quiz=q)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dodatkowo podczas tworzenia kursów profesor może dokonać wyboru czy dany zasób będzie dostępny dla wszystkich czy tylko dla wybranych użytkowników w systemie. Jeśli wybrany zostanie drugi wariant, autor kursu ma możliwość wyboru z listy wszystkich studentów tych, dla których kurs będzie dostępny i zostają oni dodani do listy uczestników kursu (Listing 4.8).

Listing 4.8 Podział studentów na listy z dostępem do kursu i bez dostępu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

query\_first = request.GET.get("q1")

 if query\_first:

   excluded\_students = excluded\_students.filter(username\_\_icontains=query\_first)

query\_second = request.GET.get("q2")

 if query\_second:

  added\_students =added\_students.filter(username\_\_icontains=query\_second)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Po stworzeniu wszystkich funkcjonalności do konstruowania jak i zarządzania kursami oraz testami, autor opracował strukturę narzędzi dla ucznia, w której znalazły się użyteczności takie jak przeglądanie udostępnionych kursów, przeglądanie wyników oraz rozwiązywanie testów, jak również komunikowanie się z innymi uczniami, jak   
i profesorem na forum. Zaimplantowane tu funkcje opierają się głównie na odczycie zasobów z bazy danych przy użyciu odpowiednich warunków. Listing 4.9 prezentuje funkcję rozwiązywania quizu, po zatwierdzaniu odpowiedzi jest ona zapisywana, a następne pytanie wczytywane z bazy.

Listing 4.9 Funkcja obsługi rozwiązywania quizu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

def quiz\_home(request, quiz\_id, question\_id):

    quiz = Test.objects.get(id=quiz\_id)

    question = Question.objects.get(quiz=quiz, id=question\_id)

    next\_id = question\_id

    qid= int (question\_id)

    for q in Question.objects.filter(quiz=quiz).order\_by('-id'):

        if q.id > qid:

            next\_id = q.id

    quiz.student.add(request.user)

    if request.method == "POST":

        selected\_option = request.POST.get('option')

        res = Response.objects.create(question=question, selected\_option=selected\_option)

        res.student.add(request.user)

        qid= int (question\_id)

        if qid < quiz.question\_set.last().id:

            return redirect('/quiz/' + str(quiz\_id) + '/' + str(next\_id))

        else:

            return redirect('/quiz/' + str(quiz\_id)+'/score')

   else:return render(request, 'quiz/quiz.html', {"title":quiz.quiz\_title,'question': question})

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Kończąc prace twórczą nad funkcjonalnościami autor zaimplementował prosty panel administracyjny do zarządzania użytkownikami w systemie. Dzięki niemu osoba uprawniona może tworzyć oraz usuwać wybrane profile, a także nadawać im prawa w witrynie. Dodatkowo administrator witryny otrzymał dostęp do rozszerzonego panelu administracyjnego udostępnionego przez Django, do którego dodano możliwość modyfikowania zawartości wszystkich tabel w witrynie (Listing 4.10).

Listing 4.10 Kod umożliwiający zarządzanie tabelami z bazy \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

from django.contrib import admin

from .models import Test, Question, Response

admin.site.register(Test)

admin.site.register(Question))

admin.site.register(Response)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aby zapewnić odpowiednie odseparowanie funkcjonalności dostępnych dla poszczególnych użytkowników systemu oraz osób zalogowanych od niezalogowanych, w witrynie użyte zostały dekoratory Django takie jak:

* [login\_required()](https://docs.djangoproject.com/en/2.2/topics/auth/default/#django.contrib.auth.decorators.login_required) – sprawdza czy użytkownik jest zalogowany, jeśli nie, przekierowuje na kartę logowania.
* user\_passes\_test(lamba user: user.is\_professor) – sprawdza czy użytkownik ma uprawnienie profesora i może mieć dostęp do treści strony
* user\_passes\_test(lamba user: user.is\_side\_admin) – sprawdza czy użytkownik ma uprawnienia administratora i może mieć dostęp do treści strony

Warto również wspomnieć, iż jednocześnie z budową wszystkich funkcji konstruowana była mapa URL całej witryny. Dla większej czytelności adresy zostały podzielone według przypisania modułowego oraz funkcji (Listing 4.11).

Listing 4.11 Podstawy mapowania URL w witrynie \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

urlpatterns = [

    url(r'^$', user\_views.home, name='home'),

    url(r'^about/$', user\_views.about, name='about'),

    url(r'^contact/$', user\_views.contact, name='contact'),

    url(r'^admin/', admin.site.urls),

    url(r'^courses/', include('courses.urls')),

    url(r'^quiz/', include('quiz.urls')),

    url(r'^forum/', include('forum.urls')),

    url(r'^profile/', include('users.urls'))

]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 4.3 Implementacja szablonów

Analogicznie z pracami nad funkcjonalnościami zawartymi w View postępowały prace nad tworzeniem szablonów (Template). Początkowo były one tworzone głównie na zasadzie sprawdzenia czy dany widok wyświetla poprawny wynik. Zaś po zakończeniu procesu twórczego nad wewnętrzną stroną aplikacji, autor skupił się nad dopracowaniem warstwy prezentacji.

Jak już wcześniej wspomniano do stworzenia przyjaznej integracji użytkownika z systemem wykorzystano dynamikę kodu po stornie serwera jak również technologię AJaX, zaś do stworzenia przejrzystego i intuicyjnego wyglądu framework UIkit. Główny szablon strony został podzielony według założonego w procesie projektowym schematu na kilka bloków:

* navbar – pasek nawigacji strony, który dzięki wykorzystaniu kodu po stronie serwera dynamicznie zmieniał zawartość w zależności od tego czy użytkownik był zalogowany i jakie posiadał prawa w witrynie (Listing 4.12);

Listing 4.12 Fragment kodu budującego pasek nawigacji \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<ul class="uk-navbar-flip uk-navbar-nav">

            <li class="active"><a href="/"><span class="glyphicon glyphicon-home"></span> Home</a></li>

            <li><a href="{% url 'about' %}"><span class="glyphicon glyphicon-info-sign"></span> About</a></li>

            <li><a href="{% url 'contact' %}"><span class="glyphicon glyphicon-earphone"></span> Contact</a></li>

            {% if request.user.is\_authenticated %}

              <li><a href="{% url 'courses' %}"><span class="glyphicon glyphicon-education"></span> Global courses</a></li>

              <li><a href="{% url 'quizhome' %}"><span class="glyphicon glyphicon-check"></span> Quizzes</a></li>

              <li><a href="{% url 'forum' %}"><span class="glyphicon glyphicon glyphicon-comment"></span> Forum</a></li>

            {% endif %}

...

                      {% if  user.is\_site\_admin or user.is\_professor %}

                      <li><a href="{% url 'profile' %}"><span class="glyphicon glyphicon-briefcase"></span> Management</a>

                      </li>

                      {% else %}

                      <li><a href="{% url 'profile' %}"><span class="glyphicon glyphicon-paperclip"></span> Individual course</a>

                      </li>

...

</ul>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* sitemenu – menu boczne witryny, w zależności od kontekstu zmieniające wyświetlane wartości np. w zakładce forum wyświetla ostatnie aktualizowane wpisy (Listing 4.13);

Listing 4.13 Kod wyświetlający ostatnio aktualizowane wpisy na forum \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<h4>Recent activities:</h4>

  <dl class="uk-description-list-line">

    {% for instance in topic\_list|dictsortreversed:"stamp\_updated"|slice:"0:3" %}

    <dt>

      <a class="uk-link-muted"  href="{% url 'forum' %}{{ instance.slug }}">{{ instance.subject }}</a>

    </dt>

      <dd><p>Modified: {{ instance.stamp\_updated|timesince }}</p></dd>

    {% endfor %}

  </ul>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* titlebar –pasek tytułowy wyświetlający tytuł modułu, w którym się znajdujemy oraz ścieżkę dostępu (Listing 4.14);

Listing 4.14 Budowa paska tytułowego \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<h1 class="title-header">{{ title }} </h1>

      <ul class="breadcrumb breadcrumb\_\_t">

...

        {% url 'student\_course' course\_name slug as student\_course\_url %}

        {% if request.get\_full\_path ==  student\_course\_url %}

          <li><a href="{% url 'courses' %}">Courses</a></li>

        {% endif %}

...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* content – główny blok strony w nim w zależności od kontekstu wyświetlane są wszystkie zawartości strony. Zawiera szereg rozszerzeń jak np. dynamiczny skrypt (technologii AJaX) dodający nowy temat do forum (Listing 4.15);

Listing 4.15 Dynamiczny skrypt dodawania tematu na forum \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

$(document).ready(function () {

    $('.add-topic form').submit(function () {

        var $subject = $.trim($('#id\_subject').val());

        var formData = tinyMCE.activeEditor.getContent({format: 'raw'});

        var regex = /(<([^>]+)>)/ig;

        var editorContent = formData.replace(regex, "");

        if ($subject === '' || editorContent === '') {

            alert("Can't submit empty fields");

            return false;

        }

    });

    var $topic = $(".add-btn");

    var $returnLink = $(".return-link");

    var $block = $(".add-topic");

    $topic.on("click", handleAddTopicClick);

    $returnLink.on("click", handleAddTopicClick);

    function handleAddTopicClick() {

        $block.toggle();

        $topic.toggle();

        $returnLink.toggle();

        $('html,body').animate({

            scrollTop: $(window).scrollTop() + block.height() / 2

        })

    }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* footer – stopka witryny zawierająca stałe dane o twórcy oraz mail kontaktowy.

Dodatkowo, aby spełnić założoną w projekcie interaktywność przekazu, autor stworzył strukturę kursów, w której użytkownik może dynamicznie przechodzić pomiędzy kolejnymi rozdziałami, a każdy z nich posiada formę blokową, aby uporządkować wszystkie zawarte zasoby w kursie. Do struktury działu zostały również wbudowane ramki służące do wyświetlania dodanych do kursu filmów platformy YouTube (Listing 4.16).

Listing 4.16 Wyświetlanie ramki z filmikiem platformy YouTube \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 <div class="container student-course">

    <div class="row">

      <div class="col-md-2">

        {% if chapter\_list %}

          <ul class="sidebar-nav">

            <li>

              {% for instance in chapter\_list %}

                <a class="a-target" href="{% url 'student\_course' course\_name instance.slug %}">

                  {{ instance.chapter\_name }}

                </a>

              {% endfor %}

            ...

      </div>

      <div class="col-md-10">

        {% if result\_list %}

          {% for item in result\_list %}

            {% if item.link %}

              {% if 'youtube.com' in item.link %}

                <div class="course-inner-content video">

                  <iframe width="640" height="360" src="{{ item.link }}" frameborder="0" allowfullscreen></iframe>

                </div>

              ...

  </div>

  <script>

    $(document).ready(function () {

      var rightColumn = $(".col-md-10");

      var parentHeight = rightColumn.parent().height();

      rightColumn.css("height", parentHeight);

      var pathname = window.location.pathname;

      pathname = pathname.substring(0, pathname.length - 1);

      $('.sidebar-nav > li a[href="' + pathname + '"]').addClass('active-sidebar');

    });

  </script>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 4.4 Testy

Jednocześnie z pracami nad witryną sprawdzana była poprawność poszczególnych modułów. Poprzez zastosowanie Python Unit Testing tworzone były testy poszczególnych funkcjonalności w wewnętrznej strukturze witryny. Listing 4.17 przedstawia przykładowe testy stworzone dla modułu obsługi użytkownika. W całym projekcie stworzono łącznie 42 testy jednostkowe, które na etapie implementacji funkcjonalności pomogły ustrzec się większych błędów programistycznych, zaś po zakończeniu procesu twórczego pomogły upewnić się, że wszystko jest „OK” (Rys. 4.5).

Listing 4.17 Przykładowe testy jednostkowe\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

class UserProfileTestCase(TestCase):

    def setUp(self):

        self.factory = RequestFactory()

        self.user = UserProfile.objects.create\_user(

            username='Hodor', email='test@gmail.com', password='top\_secret', is\_active=True)

    def test\_user\_can\_register(self):

        UserProfile.objects.create(username="user\_name\_sample", email='t@t.com', password=make\_password('abcd'))

        user = UserProfile.objects.get(username="user\_name\_sample")

        self.assertEqual(user.username, 'user\_name\_sample')

    def test\_user\_can\_login(self):

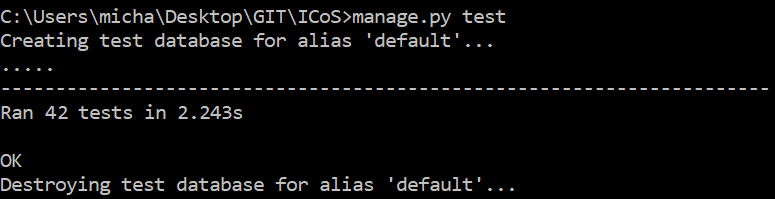
        self.c = Client()

        self.user = authenticate(username='Hodor', password='top\_secret')

        login = self.c.login(username='Hodor', password='top\_secret')

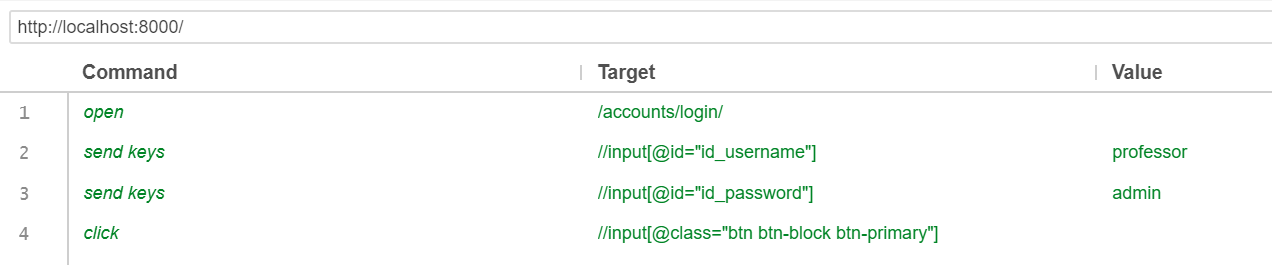
        self.assertTrue(login)

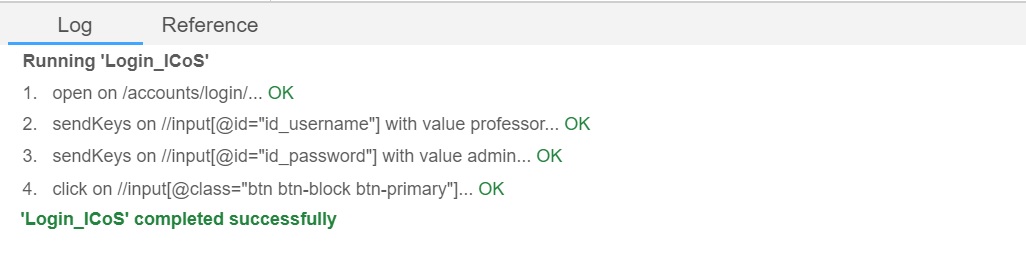
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Rys. 4.5 Wynik uruchomienia wszystkich testów jednostkowych po zakończeniu implementacji

Po dopracowaniu części wizualnej projektu została ona przetestowana dzięki zainstalowanemu jako rozszerzenie przeglądarki programowi Selenium IDE, który pomaga w generowaniu testów i eliminuje potrzebę ręcznego wykonywania powtarzalnych czynności, organizowane testy można łączyć w zestawy dla łatwego zarządzania.   
Rysunek 4.6 przedstawia przykładowy wygląd projektu testu z programu Selenium IDE oraz logi z jego przebiegu. Witryna została również przetestowana w sposób ręczny dla całkowitego upewnienia się o braku błędów.

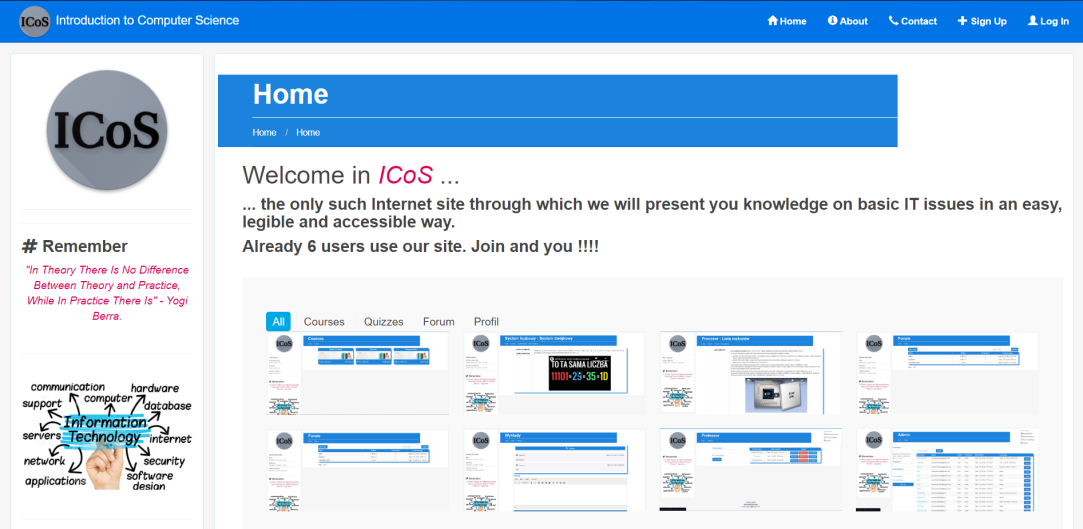




Rys. 4.6 Projekt testu oraz logi z jego przebiegu – program Selenium IDE

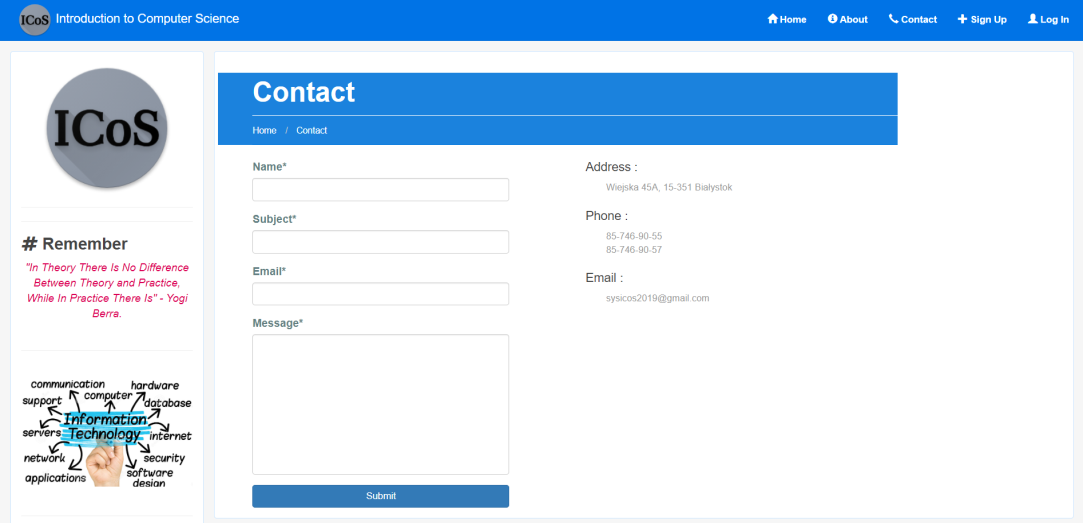
Zastosowanie zarówno testów automatycznych jak i manualnych pozwoliło autorowi na wykrycie problemu na wczesnym etapie cyklu rozwojowego oprogramowania. Testy objęły zarówno błędy w implementacji programisty, jak i braki w warstwie prezentacji. Pisanie testów zmusiło autora, aby dokładnie przeanalizował dane wejściowe i wyjściowe oraz precyzyjnie określił pożądane zachowanie danej jednostki. Dzięki znalezieniu i naprawieniu błędów na początku kodowania ich waga w całym procesie implementacyjnym była niewielka.

# 5. Interface witryny



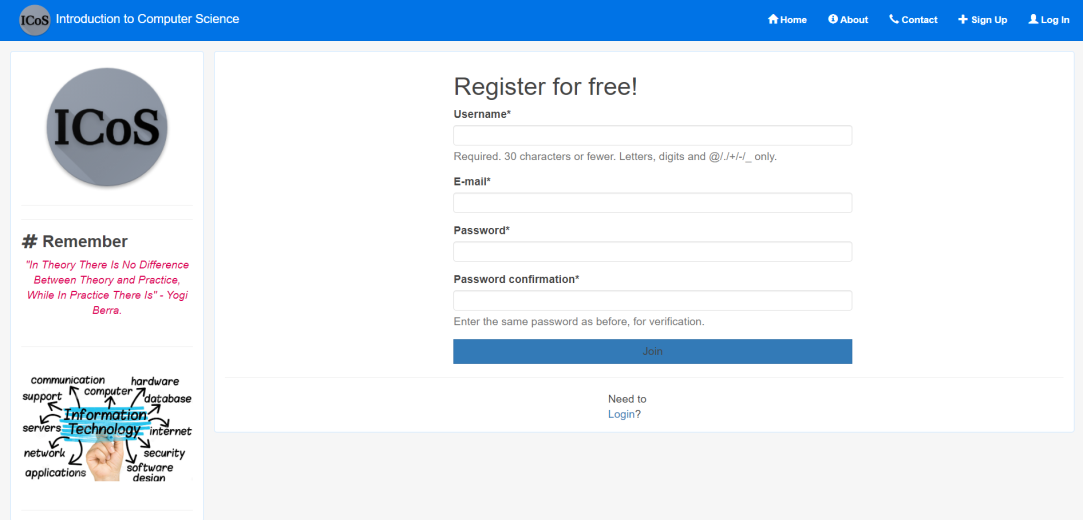
Rys. 5.1 Strona startowa witryny

Dzięki zastosowaniu nowoczesnych technik programistycznych wygląd witryny jest przyjazny a jej interakcja z użytkownikiem szybka i przejrzysta. Już przy pierwszym wejściu na stronę jej widok startowy (Rys.5.1) ukazuje nam jak została podzielona logika nawigacji w witrynie, dzięki czemu użytkownik może dostrzec jej zalety. Dodatkowo umieszczono zdjęcia poszczególnym modułów, co ma zachęcić oglądającego do zarejestrowania się w witrynie. Przed zarejestrowaniem i zalogowaniem użytkownik ma dodatkowo możliwość przeczytania krótkiej notki informacyjnej o witrynie jak i o samej Politechnice Białostockiej (zakładka About), a także możliwość skontaktowania się z administratorami witryny w celu zadania pytania luz zgłoszenia błędu w działaniu strony   
(zakładka Contact, Rys.5.2).



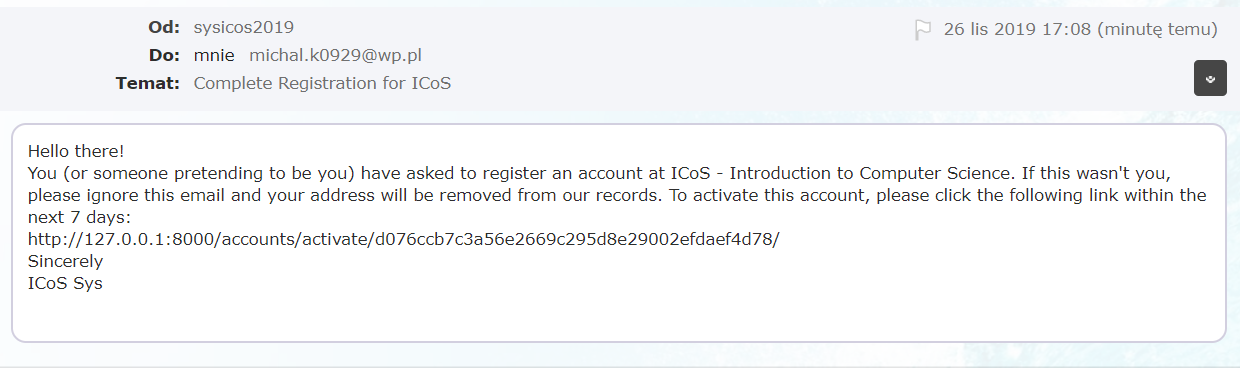
Rys. 5.2 Strona umożliwiająca kontakt z administratorem witryny

## 5.1 Logowanie i rejestracja

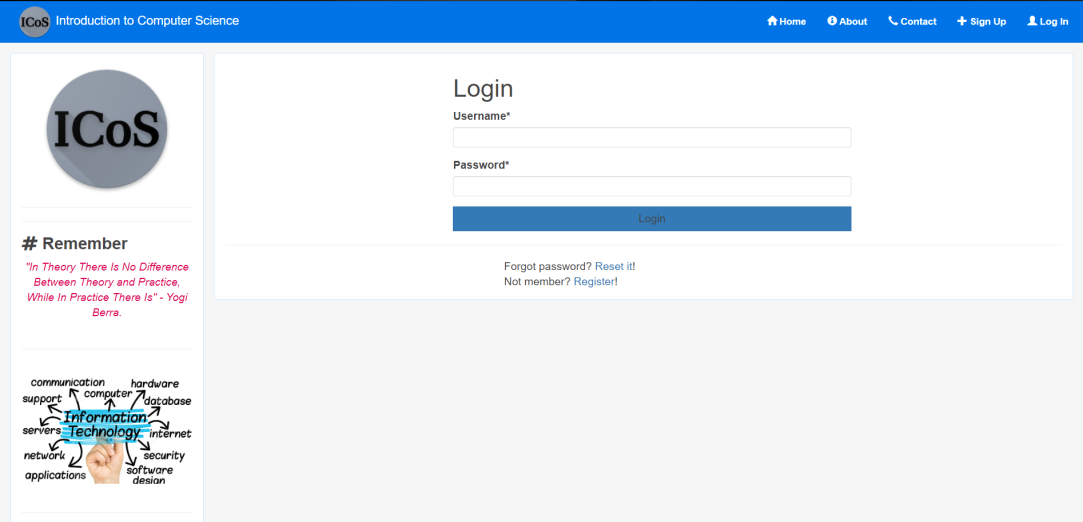


Rys. 5.3 Widok ekranu rejestracji w witrynie

Aby użytkownik mógł rejestrować się do systemu musi przejść w zakładkę „Sing up”, następnie wprowadzić dane takie jak login, mail oraz hasło (Rys. 5.3). Po wprowadzeniu wszystkich danych należy wcisnąć przycisk „Join” następnie system przeprowadzi walidacja danych, jeśli wszystkie wprowadzone wartości są poprawne wyświetli się komunikat ukończeniu rejestracji oraz wysłanie maila aktywującego konto. Dokończenie procesu rejestracji wymaga od użytkownika przejścia do indywidualnej poczty email podanej w formularzu rejestracji i otworzenia linku umieszczonego w wiadomości otrzymaj od systemu (Rys. 5.4). Link przekieruje bezpośrednio do zalogowanego konta w witrynie oraz wyświetli się komunikat o poprawnej aktywacji konta. Po zakończeniu procesu rejestracji użytkownik ma dostęp do wszystkich zasobów strony przeznaczonych dla niego, aby otrzymać dodatkowe uprawnienia należy zgłosić się do administratora strony.



Rys. 5.4 Mail aktywujący wysłany z systemu

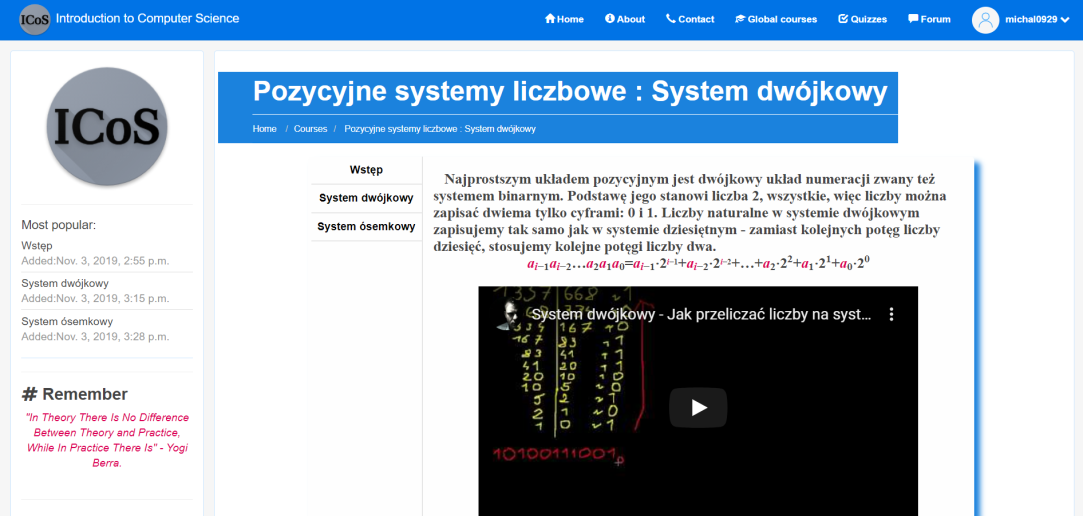


Rys. 5.5 Ekran logowania do witryny

Procedura logowania do systemu polega na przejściu do zakładki „Log in” i wpisaniu nazwy użytkownika oraz hasła stworzonego uprzednio konta (Rys. 5.5). Jeśli jednak użytkownik zapomni hasła do systemu może go odzyskać dzięki funkcji „Reset”. Polega ona na wpisaniu maila podanego podczas rejestracji, na który zostanie przysłana wiadomość z linkiem do zresetowania starego hasła i wprowadzenia nowego. Ponadto każdy użytkownik może zmienić swoje hasło, nazwę użytkownika czy maila w ustawieniach konta po zalogowaniu, zakładka ta znajduje się pasku nawigującym, należy wybrać nazwa użytkownika a następnie z pojawiającej się listy pozycje „Account Settings”.

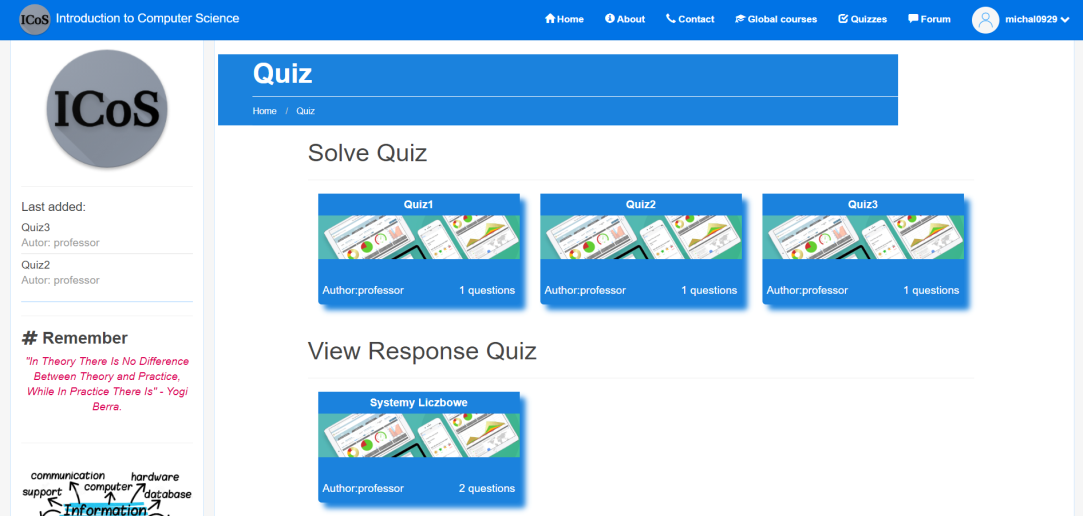
## 5.2 Panel użytkownika

Zalogowany użytkownik posiada dostęp do modułów kursów, testów oraz forum. Aby przejść do tego pierwsze należy wybrać zakładkę „Global courses” w nim zawarte są wszystkie zasoby nauczania udostępnione dla ogółu użytkowników (Rys. 5.6). Następnie uczeń może wybrać dział, który go interesuje i przejść do jego zawartości. Dodatkowo profesor może udostępnić wybrane materiały dydaktyczne tylko dla niektórych jednostek, w tym przypadku, aby przejść do tych zasobów należy wejść w zakładkę z nazwą użytkownika i na rozwijanym pasku wybrać pozycje „Individual course”.

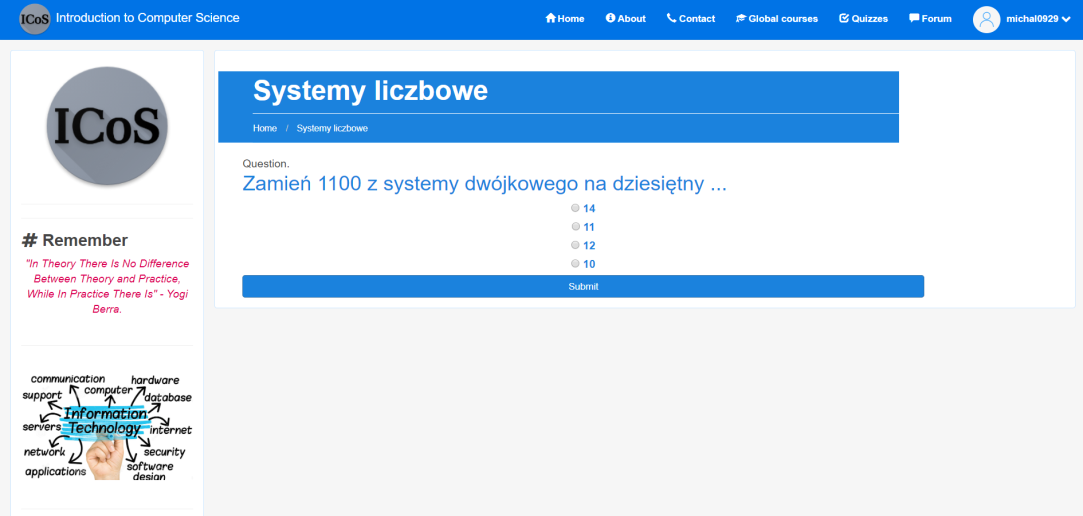


Rys. 5.6 Widok działu w kursie

Podobnie jak w modułem kursów należy postępować z quizami, analogicznie przejść w zakładkę Quizzes (Rys. 5.7) wybrać interesujący nas temat z działu „Solve Quiz” i po kolei odpowiadać na pytania zaznaczając tylko jedną odpowiedz (Rys. 5.8). Na koniec testu wyświetlony zostanie wynik podejścia. Po rozwiązaniu quiz zostanie przeniesiony do działu „View Response Quiz”. Znajdujące się tam quizy oznaczają, że były one już rozwiązywane i możemy jedynie zobaczyć, jakie odpowiedzi udzielono na konkretne pytanie i sprawdzić jej poprawność. Użytkownik posiada również wgląd w wyniki wszystkich rozwiązanych testów, aby je sprawdzić należy przejść w zakładkę z nazwą użytkownika a następnie z rozwiniętej listy wybrać „Quiz Scoreboard”.

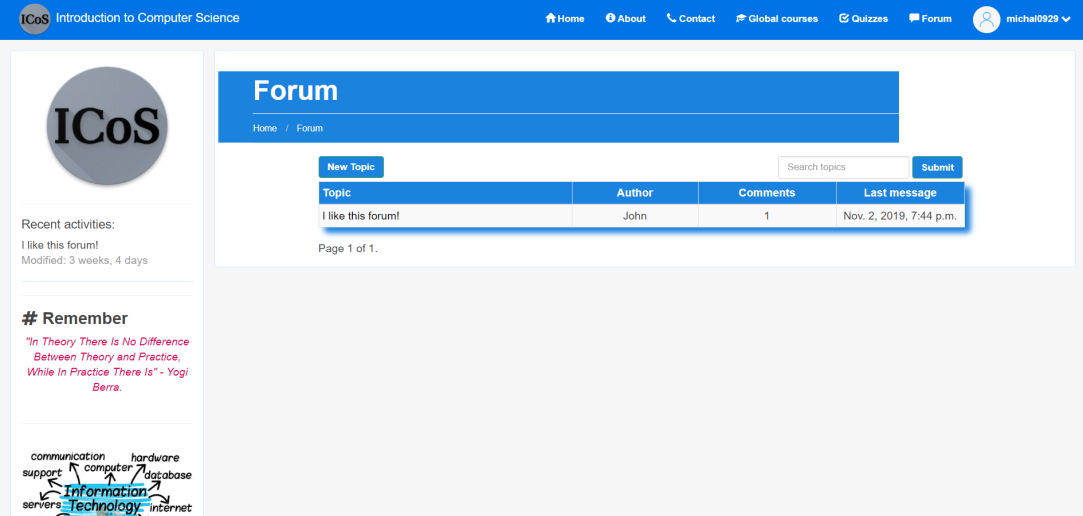


Rys. 5.7 Widok ekranu głównego quizów



Rys. 5.8 Widok pojedynczego pytania w quzie

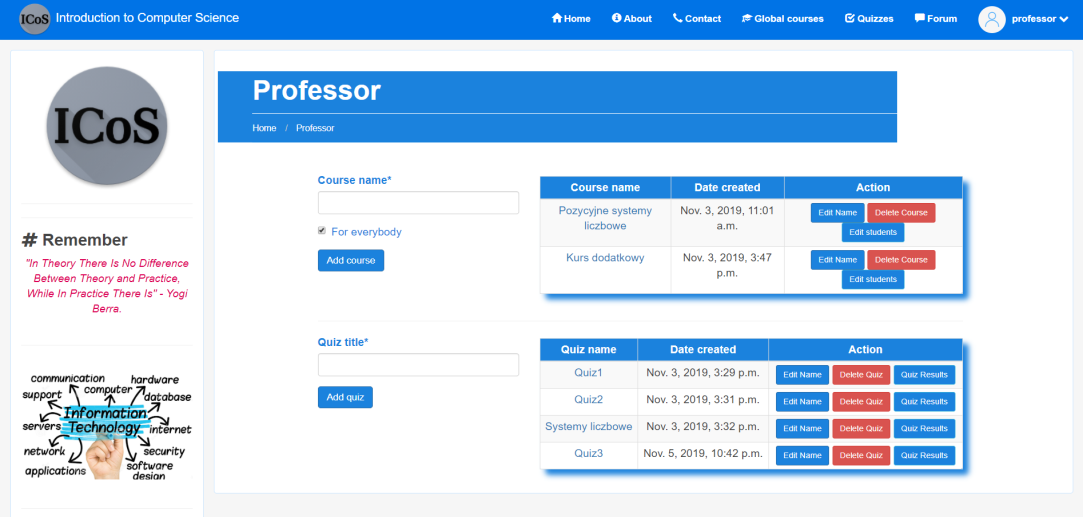
Ostatnim elementem dostępnym osoby korzystającej z witryny jest forum, dzięki któremu może komunikować się zarówno z innymi uczniami jak i profesorami. W zakładce „Forum” znajdują się wszystkie wylistowane tematy (Rys. 5.9). Każdy użytkownika może utworzyć nowy temat z przy użyciu przycisku „New Topic” oraz dodać komentarz do istniejących tematów, w tym celu musi wejść w interesujący go temat wpisać komentarz w pole „Message” i wybrać przycisk „Submit”.

****

Rys. 5.9 Widok ekranu głównego forum

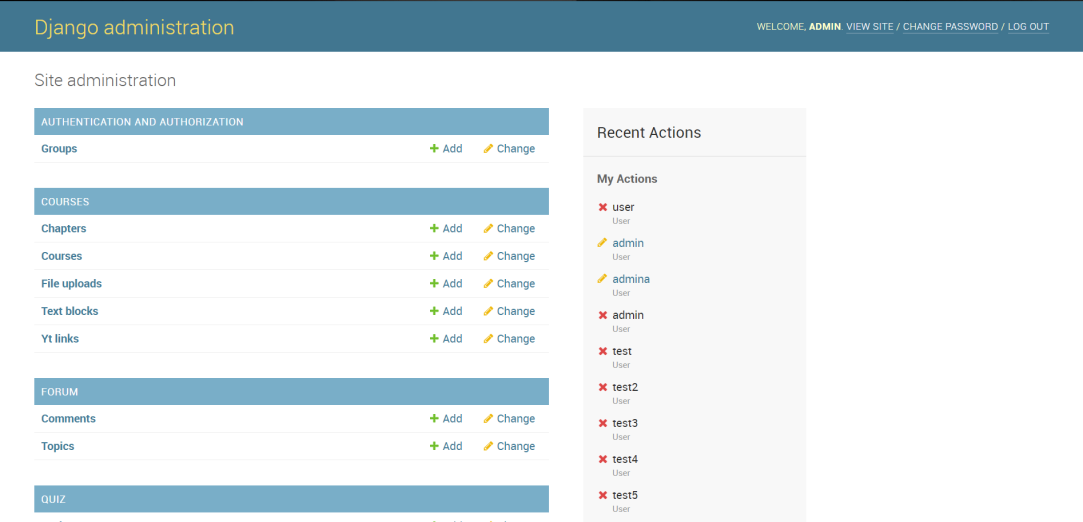
## 5.3 Zarządzanie i administracja systemem

Użytkownicy tacy jak profesor czy też administrator posiadają dodatkowe zakładki w liście znajdującej się pod nazwą użytkownika. Nauczyciel w zakładach posiada panel zarządzający, w którym może dodać, usunąć lub edytować kursy, quizy oraz działy istniejące w kursach (Rys. 5.10). Ponadto po wejściu konkretny dział wyświetlany jest widok umożliwiający dodawanie materiałów do kursu, zaś po otworzeniu quizu można dodawać lub usuwać pytania.



Rys. 5.10 Widok panelu zarządzającego profesora

Konto administratora posiada zakładkę „Management” umożliwiającą mu zarządzanie wszystkimi użytkownikami w witrynie oraz nadawanie im poszczególnych praw. Dodatkowo panel administracyjny posiada przekierowanie („Extended admin”) do panelu „Django administration” gdzie umożliwione jest gospodarowanie wszystkimi tabelami bazy danych w witrynie (Rys. 5.11).



Rys. 5.11 Widok panelu „Django administration”

# 6. Podsumowanie

Głównym celem pracy było stworzenie (opracowanie koncepcji, analiza wymagań, projekt, implementacja, testowanie) aplikacji internetowej wspomagającej prowadzenie zajęć z przedmiotu Wprowadzenie do Informatyki, a także zapoznanie się z obecnym stanem wiedzy na temat zdalnego nauczania oraz przegląd dostępnych na rynku platform służących kształceniu na odległość. Do zakresu pracy należało również zebranych materiałów dydaktycznych pomagających w nauczaniu podstaw informatyki i umieszczenie ich w witrynie.

Cele pracy zostały zrealizowane. Z przeglądu obecnego stanu wiedzy przedstawionej w rozdziale pierwszym wynika, że nauczanie zdalne towarzyszy człowiekowi już od ponad 250 lat, jednakże przełomem w tej dziedzinie był rozwój Internetu. Obecnie na rynku znajduje się wiele platform e-larningowych posiadających coraz więcej funkcjonalności, dzięki czemu zdalne nauczanie posiada coraz więcej zalet, które można wymieniać w setkach. Jednakże e-nauczanie nie jest bez wad, zaś największą z nich jest brak osobistego kontaktu pomiędzy nauczycielem a uczniem.

Przedstawiony w rozdziale trzecim projekt aplikacji opiera się na założeniu, iż witryna ma być łatwa w obsłudze oraz posiadać interaktywny przekaz będący ułatwieniem w przyswajaniu wiedzy a także w jej przekazywaniu. Ustalone funkcjonalności i założenia w systemie pozwoliły jasno określić kierunek implementacji oraz ustandaryzować system w ramach ogólnie istniejących podziałów. Zaś widoki zostały zaprojektowane w taki sposób aby były intuicyjne dla potencjalnego użytkownika.

Faza implementacji ukazana w rozdziale czwartym dzięki użytej platformie internetowej Django oraz językowi wysokiego poziomu Python została przeprowadzona w sposób profesjonalny, a witryna spełnia wszystkie postawione przed nią w fazie projektowania zadania. Dzięki użytym rozwiązaniom takim jak dwuetapowa rejestracja w witrynie jak i wbudowanymi w framework systemami bezpieczeństwa witryna jest chroniona przed wieloma typami ataków w tym XSS (Cross-site scripting). Ponadto użyte w implementacji technologie takie jak AJaX czy UIKit uatrakcyjniły wygląd witryny oraz stworzyły przyjemne środowisko interakcji dla użytkownika. Prowadzone równocześnie z implementacją testy jednostkowe pozwoli autorowi na ustrzeżenie się przed większymi błędami programistycznymi, zaś stworzone w późniejszym etapie testy automatyczne pozwoliły na wykrycie błędów w warstwie prezentacji.

Dokładna instrukcja poruszania się po warstwie frontendowej została ukazana w rozdziale piątym będącym jednocześnie zwieńczeniem pracy twórczej nad aplikacją. Przedstawione w nim instrukcję tyczą się zarówno użytkownika podstawowego jak i użytkowników posiadających dodatkowe prawa w witrynie, co pozwala uzyskać pełen wgląd w możliwości systemu.

Należy przypuszczać, że cała sfera systemów e-learnigowych rozwijać się będzie w kierunku pełnego odseparowania nauczyciela od kursanta, tzn. w stronę tak zwanego d-learningu (ang. *distance learning*). Jest to ukierunkowane ogólną tendencją do podnoszenia elastyczności narzędzi edukacyjnych, a także dążeniem do indywidualizacji podejścia do ucznia. Również obserwowana na całym świecie globalizacja oraz ułatwiony dostęp do Internetu sprzyja rozwojowi tej formy nauczania. Dodatkowym argumentem sprzyjającym rozwojowi nauki przez Internet jest aspekt redukcji kosztów przeznaczonych na edukację poprzez ograniczenie stosowanych dotychczas zasobów szkoleniowych, jakimi są chociażby budynki uczelni czy też stosowane podczas zajęć pomoce dydaktyczne [3]. Dlatego też w przyszłości witryna może zostać rozszerzona w wiele różnych kierunków np. poprzez dodatkowe funkcjonalności związane z quizami czy też kontaktem ucznia z profesorem.

# Literatura

1. Banachowski L., *Postęp e-edukacji*, Wydawnictwo PJWSTK, Warszawa, 2013
2. Bednarek J., Lubina E., *Kształcenie na odległość*, Wydawnictwo Naukowe PWN, Warszawa, 2008
3. Belsky L., *Where Online Learning Goes Next*, „Harvard Business Review”, 2019
4. Brzostek-Pawłowska J., *Zmiany w modelach i technologiach informacyjnych w dobie Web 2.0 i Web 3.0*, „Elektronika” 2011, nr 3.
5. Clarke A., *E-learning - nauka na odległość*, Wydawnictwo Komunikacji i Łączności, Warszawa, 2007
6. Claroline. Witryna internetowa. www.claroline.net stan z dnia 10.11.2019
7. Czarkowski J.J., *E-learning dla dorosłych*, Wydawnictwo Difin, Warszawa, 2012
8. Developer Mozilla. Witryna internetowa. www.developer.mozilla.org stan z dnia 20.11.2019
9. Django. Documentacja internetowa. www.docs.djangoproject.com/pl/ stan z dnia 20.11.2019
10. Gachago D., Morkel J., Hitge L., *Developing eLearning champions: a design thinking approach*, „Springer International Publishing”, 2017
11. Główny Urząd Statystyczny. Witryna internetowa. www.stat.gov.pl, stan z 9.11.2019.
12. ILIAS. Witryna internetowa. www.docu.ilias.de stan z 10.11.2019
13. Mon Mon The, Tsuyoshi Usagawa*, Effectiveness of E-learning Experience through Online Quizzes: A Case Study of Myanmar Students,* „iJET”, 2018, nr 13
14. Moodle. Witryna internetowa. www.stats.moodle.org stan z dnia 10.11.2019
15. PLATO History. Witryna internetowa. www.platohistory.org , stan z 8.11.2019
16. Plebańska M., *Tajniki edukacji na odległość*, Wydawnictwo C.H. Beck, Warszawa, 2011
17. Przybyła W., Ratalewska M., *Poradnik dla projektujących kursy e-learningowe*, Wydawnictwo Naukowe Instytutu Technologii Eksploatacji - Państwowego Instytutu Badawczego, Warszawa, 2012
18. Python. Documentacja internetowa. www.docs.python.org stan z dnia 20.11.2019
19. Redlarski K., Garnik I., *Zastosowanie systemów e-learningu w szkolnictwie wyższym*, Politechnika Gdańska, Gdańsk, 2014
20. Waćkowski K., Chmielewski J. M., *Rola standaryzacji platform w e-learningu*, „E-mentor”, 2007, nr 2
21. Zając M., *Model aktywności w kursach online*, „E-mentor” 2009, nr 4

# Spisy

## Spis tabel

[Tabela 2.1 Porównanie wybranych platform e-learningowych (na podstawie [14]) 13](#_Toc25850784)

[Tabela 3.1 Opis tekstowy aktorów 16](#_Toc25850785)

[Tabela 3.2 Struktura oraz typy danych w tabeli User 23](#_Toc25850786)

[Tabela 3.3 Struktura oraz typy danych w tabeli Courses\_User 23](#_Toc25850787)

[Tabela 3.4 Struktura oraz typy danych w tabeli Courses 23](#_Toc25850788)

[Tabela 3.5 Struktura oraz typy danych w tabeli Courses\_Chapter 24](#_Toc25850789)

[Tabela 3.6 Struktura oraz typy danych w tabeli Textblock 24](#_Toc25850790)

[Tabela 3.7 Struktura oraz typy danych w tabeli YTlink 24](#_Toc25850791)

[Tabela 3.8 Struktura oraz typy danych w tabeli Fileupload 24](#_Toc25850792)

[Tabela 3.9 Struktura oraz typy danych w tabeli User\_Quiz 24](#_Toc25850793)

[Tabela 3.10 Struktura oraz typy danych w tabeli Quiz 25](#_Toc25850794)

[Tabela 3.11 Struktura oraz typy danych w tabeli Quiz\_Question 25](#_Toc25850795)

[Tabela 3.12 Struktura oraz typy danych w tabeli Quiz\_Option 25](#_Toc25850796)

[Tabela 3.13 Struktura oraz typy danych w tabeli User\_Score\_Quiz 25](#_Toc25850797)

[Tabela 3.14 Struktura oraz typy danych w tabeli User\_Response\_Question 26](#_Toc25850798)

[Tabela 3.15 Struktura oraz typy danych w tabeli Forum\_Topic 26](#_Toc25850799)

[Tabela 3.16 Struktura oraz typy danych w tabeli Forum\_Comment 26](#_Toc25850800)

## Spis rysunków

Rys. 2.1 Uproszczony schemat blokowy pierwszego systemu PLATO z 1960 roku [0] 4

Rys. 2.2 Powiązania w obszarze e-leraningu (opracowanie własne na podstawie [14]) 5

Rys. 2.3 Zależności pomiędzy rodzajami procesów dydaktycznych (opracowanie własne na podstawie [3]) 6

Rys. 2.4 Ukierunkowania poszczególnych rodzaj systemów zdalnego nauczania (opracowanie własne na podstawie [13]) 11

Rys. 3.1 Diagram przypadków użycia 15

Rys. 3.2 Schemat aplikacji LMS [11] 18

Rys. 3.3 Model systemu e-learning klasy LMS [11] 19

Rys. 3.4 Schemat bazy danych 22

Rys. 3.5 Projekt ogólnego wyglądu interfejsu strony 27

Rys. 3.6Projekt głównego widoku strony: a) kursu b) forum 28

Rys. 3.7 Logo sytemu 28

Rys. 4.1 Struktura architektury Model-View-Template (opracowanie własne na podstawie [18]) 30

Rys. 4.2 Schemat działania kodu po stronie serwera (opracowanie własne na postawie [18]) 30

Rys. 4.3 Schemat działania technologii AJaX (na postawie [18]) 31

Rys. 4.4 Rejestracja w systemie – diagram czynności 33

Rys. 4.5 Wynik uruchomienia wszystkich testów jednostkowych po zakończeniu implementacji 41

Rys. 4.6 Projekt testu oraz logi z jego przebiegu – program Selenium IDE 42

Rys. 5.1 Strona startowa witryny 43

Rys. 5.2 Strona umożliwiająca kontakt z administratorem witryny 43

Rys. 5.3 Widok ekranu rejestracji w witrynie 44

Rys. 5.4 Mail aktywujący wysłany z systemu 44

Rys. 5.5 Ekran logowania do witryny 45

Rys. 5.6 Widok działu w kursie 46

Rys. 5.7 Widok ekranu głównego quizów 46

Rys. 5.8 Widok pojedynczego pytania w quzie 47

Rys. 5.9 Widok ekranu głównego forum 47

Rys. 5.10 Widok panelu zarządzającego profesora 48

Rys. 5.11 Widok panelu „Django administration” 48

## Listing

[Listing 4.1 Budowa modelu użytkownika 32](#_Toc26369719)

[Listing 4.2 Budowy modelu kursu 32](#_Toc26369720)

[Listing 4.3 Budowa modelu testów 32](#_Toc26369721)

[Listing 4.4 Funkcja zwracająca ścieżkę do kursu 33](#_Toc26369722)

[Listing 4.5 Budowa komórki formularza oraz jej funkcja walidacyjna 34](#_Toc26369723)

[Listing 4.6 Metoda walidacyjna linku do platformy YouTube 35](#_Toc26369724)

[Listing 4.7 Funkcja zwracająca wyniki wszystkich użytkowników uzyskanych w teście 35](#_Toc26369725)

[Listing 4.8 Podział studentów na listy z dostępem do kursu i bez dostępu 35](#_Toc26369726)

[Listing 4.9 Funkcja obsługi rozwiązywania quizu 36](#_Toc26369727)

[Listing 4.10 Kod umożliwiający zarządzanie tabelami z bazy 36](#_Toc26369728)

[Listing 4.11 Podstawy mapowania URL w witrynie 37](#_Toc26369729)

[Listing 4.12 Fragment kodu budującego pasek nawigacji 38](#_Toc26369730)

[Listing 4.13 Kod wyświetlający ostatnio aktualizowane wpisy na forum 38](#_Toc26369731)

[Listing 4.14 Budowa paska tytułowego 39](#_Toc26369732)

[Listing 4.15 Dynamiczny skrypt dodawania tematu na forum 39](#_Toc26369733)

[Listing 4.16 Wyświetlanie ramki z filmikiem platformy YouTube 40](#_Toc26369734)

[Listing 4.17 Przykładowe testy jednostkowe 41](#_Toc26369735)

# Instrukcja instalacji

Aby uruchomić aplikację należy:

* Posiadać zainstalowany Python w wersji 3.7.4
* pobrać na komputer plik ICoS znajdujący się na płytce

Następnie wejść do folderu ICoS\_Sys i uruchomić wiersz poleceń i wpisywać niniejsze polecenia:

* virtualenv venv
* \venv\Scripts\activate
* pip install -r requirements
* manage.py makemigrations
* manage.py migrate

Jeśli cały proces instalacji przebiegł poprawnie uruchamiamy aplikację poleceniem „*manager.py runserver*”. Teraz możemy przeglądać jej zawartość lokalnym serwerze.

Uwaga: Witryna po uruchomieniu nie zawiera bazy wiedzy, należy ją ręczenie wprowadzić z pliku ICoS\_Edu.