



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

kierunek studiów: Informatyka

Praca dyplomowa - inżynierska

Aplikacja E-Learningowa

Jakub Amanowicz

słowa kluczowe:

aplikacja, web, rest, json, angular.js

node.js, express.js, socket.io, mongo.db

e-learning, edukacja, nauczanie, web-rtc

krótkie streszczenie:

Przedmiotem pracy jest wykonanie projektu oraz implementacja webowej aplikacji E-Learningowej mającej na celu umożliwienie studentom i prowadzącym zdalną realizację jednostek edukacyjnych. Podczas realizacji wykorzystano architekturę klient-serwer opartą o zestaw najnowszych technologii oraz metody przetwarzania multimediów. W pracy przedstawiono istniejące już rozwiązania tego typu, kryteria i wytyczne projektowe oraz niektóre fazy implementacji.

opiekun pracy dyplomowej	dr inż. Marek Kopel		
	<i>Tytuł/stopień naukowy/imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>

Do celów archiwalnych pracę dyplomową zakwalifikowano do:*

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

* niepotrzebne skreślić

pieczęć wydziałowa

Wrocław 2016

Spis Treści

Streszczenie pracy	4
Abstract	5
1. Wstęp	7
1.1 Wprowadzenie do problematyki	7
1.2 Geneza pracy	7
1.3 Cel pracy	8
2. Kryteria projektowe	9
2.1 Wstępny opis i funkcjonalności	9
2.2 Grono odbiorców	9
2.3 Środowisko programowe i sprzętowe	10
3. Istniejące rozwiązania w zakresie tematyki pracy	11
3.1 Wprowadzenie	11
3.2 Zestawienie rozwiązań realizujących podobne założenia	11
3.2.1 Coursera	11
3.2.2 Udacity	12
3.2.3 edX	13
3.3 Zestawienie przydatnych technologii i rozwiązań	14
3.4 Podsumowanie i wnioski	15
4. Wytyczne projektowe	17
4.1 Zakres prac	17
4.2 Wymagania funkcjonalne	17
4.3 Wymagania нефunkcjonalne	18
4.4 Zarys architektury systemu	18
4.5 Technologie i metodyki wykorzystane w projekcie	18
5. Projekt	21
5.1 Wprowadzenie	21
5.2 Przypadki użycia	22
5.3 Interfejs	28
5.3.1 Omówienie	28
5.3.2 Makiety	28
5.4 Baza danych	34
5.5 Bezpieczeństwo	35
5.6 Przetwarzanie mediów	36
5.7 Podsumowanie	36
6. Implementacja	37
6.1 Wprowadzenie	37
6.2 Wykorzystane środowiska i narzędzia programistyczne	37
6.3 Wykorzystane biblioteki i frameworki	37
6.4 Interfejs	39

6.5 Bezpieczeństwo	46
6.6 Przetwarzanie mediów	46
6.7 Instalacja oprogramowania	47
6.8 Instrukcja użytkowania	47
6.9 Archiwizacja danych	49
6.10 Podsumowanie	49
7. Podsumowanie	51
7.1 Ewaluacja i wnioski	51
7.2 Możliwe ścieżki rozwoju	51
7.2.1 Strumieniowanie i komunikacja	52
7.2.2 Interfejs	52
Bibliografia	55
Spis ilustracji	57
Spis tabel	58

Streszczenie pracy

W dzisiejszych czasach, w których technologia, zwłaszcza rozwiązania internetowe, staje się powszechniejsza z dnia na dzień, a ludziom coraz ciężiej o wolny czas, branża E-Learningu, zdalnego nauczania, znacznie zyskała na znaczeniu. Edukacja i rozwój to usługi, na które popyt od zawsze był ogromny, a przy stale rosnącej konkurencyjności na rynku pracy, zapotrzebowanie na ten segment będzie tylko rosło.

Naturalnie, rynek zareagował na ten proces w sposób bardzo aktywny. Przez co w sieci można na dzień dzisiejszy znaleźć ogromną liczbę usług E-Learningowych. Co ważne, ta branża nadal bardzo dynamicznie się rozwija i wciąż dostępne są nisze, które można w jej ramach zagospodarować.

Niestety, na dzień dzisiejszy środowisko akademickie, przynajmniej to w Polsce, nie wykorzystuje jeszcze w pełni możliwości jakie niosą ze sobą nowoczesne technologie webowe w zakresie nauczania. Uczelnie wyższe nie chcą przykładowo udostępniać swoim studentom wykładów i kursów w formie zdalnej ze względu na niższą frekwencję na zajęciach.

Ze względu na powyższe argumenty, powstała poniższa praca. Jej celem było zaprojektowanie aplikacji webowej wykorzystującej najnowsze technologie i umożliwiającej studentom i prowadzącym realizację kursów w sposób zdalny, bez konieczności fizycznej interakcji, która wymaga często niepotrzebnych komplikacji logistycznych i nakładów czasowych.

Projekt zrealizowano przy użyciu standardowych metodologii, opierając się na popularnych i sprawdzonych modelach oraz odpowiednio przygotowanych przypadkach użycia. W fazie implementacji użyto zespołu technologii MEAN, opartego na warstwie danych obsługiwanej przez nie relacyjną bazę danych MongoDB, serwerze aplikacji zaprogramowanej z użyciem Node.js oraz Express.js, a także warstwie interfejsu klienta wspomaganą przez framework Angular.js.

Abstract

Nowadays, when technology, with the emphasis on web based solutions, is becoming more and more common and people are struggling to find some free time in their schedules, the branch of E-Learning has gained momentum. Education and self-development are the services that have been in demand for many years now. This trend is not expected to stop in the nearest future.

Naturally, the market has already actively reacted to this process. Therefore, there are numerous examples of E-Learning services to be found in the Internet. However, this area is still to develop and expand.

Unfortunately, the academic society, at least the one in Poland, hasn't yet fully used the resources and possibilities offered by the newest trends in web technology when it comes to remote teaching. An example for that is the fact that universities don't publish the course content and recorded lectures to students. The reason behind this might be the concern of low attendance.

Because of the mentioned aspects, this project and paper have been made. The goal behind the project was to plan and implement web application, based on the newest web technologies, that would aim to enable students and lecturers remote education process.

The whole project was prepared with the respect to standard methodologies that guarantee high level of reliability. Therefore, many well known models have been used. The implementation process was fulfilled with use of the MEAN stack technologies that bases on MongoDB being responsible for non relational database, application server programmed with Node.js and Express.js as well as client side interface built with help of Angular.js framework.

1. Wstęp

1.1 Wprowadzenie do problematyki

E-Learning to zagadnienie, które na przestrzeni ostatnich lat znacznie zyskało na popularności [34]. Odnosi się ono do ogółu rozwiązań edukacyjnych wykorzystujących różnorodne technologie informacyjne, wśród których wyróżnić należy Internet oraz sieci komputerowe. Zgodnie z danymi przedstawionymi przez Christophera Pappasa, założyciela portalu “eLerning Industry”, popularność rozwiązań E-Learningowych rośnie w bardzo szybkim tempie. W przypadku Polski, w roku 2015 segment rynku poświęcony tego typu edukacji wzrósł o 28%, co jest piątym wynikiem w skali świata. Jednocześnie całkowity przemysł światowy związany z E-Learningiem wycenia się obecnie na ponad 107 miliardów dolarów.

Powyższe dane jasno pokazują panujący aktualnie trend na edukację wspomaganą nowoczesnymi rozwiązaniami technologicznymi. Tendencję tę można wyjaśnić za pomocą licznych argumentów. Część z nich opiera się na dzisiejszej strukturze społeczeństwa, w którym coraz trudniej o czas przy jednocześnie rosnącym zapotrzebowaniu na wykształcenie. Z tego powodu nauka w określonym przez siebie czasie, bez konieczności fizycznej interakcji i z dostępem do licznych materiałów, jaką zapewniają rozwiązania E-Learningowe, ma bardzo duży popyt. Kolejnym czynnikiem wpływającym pozytywnie na zainteresowanie edukacją tego typu jest przystępność informacji, która jest efektywnie przyswajana przy użyciu różnorodnych multimediów.

1.2 Geneza pracy

Do wybrania tego konkretnego zagadnienia jako tematu pracy inżynierskiej przekonały mnie doświadczenia, które zdobyłem podczas nauki na RWTH Aachen w Akwizgranie w ramach wymiany studenckiej. Uczelnia ta, w ramach niektórych kursów, oferowała bardzo rozwinięte rozwiązania E-Learningowe, które znacznie usprawniały proces edukacyjny. Jednym z nich, który najbardziej zapadł w moją pamięć, był system nagrywania wykładów i udostępniania ich studentom zapisanych na dany moduł edukacyjny. Sam proces nie był szczególnie złożony, gdyż opierał się na organizacji studenckiej, której członkowie nagrywali wybrane wykłady po czym udostępniali je na serwerze HTTP, do którego dostęp mieli jedynie studenci wspomnianej uczelni. Skutkowało to tym, że uczestnicy kursu nie musieli stawiać się osobiście na niektórych zajęciach. Dzięki temu organizacja tych jednostek lekcyjnych przebiegała sprawniej, a sami zainteresowani mieli dostęp do całości przekazywanej wiedzy, nie tylko w postaci slajdów i dokumentów, ale również faktycznych komentarzy wykładowcy, w dowolnym czasie.

Mimo licznych zalet system ten posiada kilka wad. Jedną z nich jest bez wątpienia fakt, że wykładowca zawsze musi stawić się w określonym miejscu i czasie, aby poprowadzić zajęcia podczas, nawet przy zerowej frekwencji. Dodatkowo proces zależny jest od obecności, motywacji i sprzętu organizacji studenckiej.

Ze względu na powyższe powody pojawił się pomysł alternatywnego rozwiązania,

które nie wymagałoby od pracownika dydaktycznego fizycznej obecności w konkretnie określonym miejscu i czasie oraz nie zależałoby od osób trzecich, takich jak organizacje studenckie. Środkiem, który byłby w stanie spełniać wszystkie opisane powyżej wymagania, jest aplikacja webowa, której implementację zrealizowałem w ramach pracy inżynierskiej.

1.3 Cel pracy

Celem pracy jest wykonanie projektu oraz implementacja E-Learningowej aplikacji webowej. Program ten powinien zapewniać prowadzącym i studentom wygodny i w pełni funkcjonalny mechanizm zastępujący regularny kurs poprzez umożliwienie zdalnej edukacji. Pobocznym zamysłem przedsięwzięcia jest dotarcie do społeczności akademickiej, a więc rozwijany produkt musi mieć na uwadze związane z tym wymagania. Konkretnie kryteria, funkcjonalności i wytyczne związane z aplikacją zostały opisane w dalszych rozdziałach pracy, zwłaszcza trzecim, czwartym oraz piątym.

2. Kryteria projektowe

2.1 Wstępny opis i funkcjonalności

Zgodnie z założeniami wspomnianymi w poprzedniej sekcji dokumentu, celem projektu jest zaprojektowanie i zaimplementowanie webowej aplikacji E-Learningowej, która ma za zadanie usprawnienie procesu edukacyjnego zarówno dla prowadzących jak i studentów.

Największy nacisk stawiany jest na umożliwienie prowadzącemu nagrywania wykładów w dowolnym miejscu i czasie. Ponadto, twórca jednostki lekcyjnej powinien być w stanie wzbogacić ją o dowolnego rodzaju materiały dodatkowe, w szczególności w formie slajdów zsynchronizowanych z nagraniem materiałem wideo. Takie podejście do problemu najlepiej odpowiada klasycznemu wykładowi, podczas którego studenci mają do czynienia jednocześnie z komentarzami prowadzącego oraz materiałami przez niego przygotowanymi.

Poza wspomnianymi wyżej funkcjonalnościami aplikacja powinna również zawierać system powiadomień, które służyłyby informowaniu studenta o nowo powstałym zasobie edukacyjnym, przykładowo w formie nagranego wykładu.

Ponadto, wymagane jest, aby program spełniał podstawowe wymagania dotyczące nowoczesnych aplikacji typu CRUD. Z tego powodu konieczne jest podłączenie i zarządzanie odpowiednio przygotowaną bazą danych.

Z racji zarządzania kontami użytkowników, potrzebne jest również odpowiednie podejście do kwestii bezpieczeństwa cyfrowego i implementacja procesów identyfikacji, autoryzacji i uwierzytelniania.

Interfejs aplikacji webowej powinien być jak najbliższy temu, z którym spotykamy się w przypadku programów desktopowych. Efekt ten można osiągnąć implementując aplikację spełniającą model "Single Page Application".

Poza aspektami około-technicznymi bardzo ważnym wymogiem budowanej aplikacji jest utrzymanie jej w charakterze otwartego i wolnego oprogramowania. Aplikacja powinna być zaimplementowana zgodnie z myślą, iż edukacja oraz wiedza powinna być dobrem ogólnie dostępnym i darmowym. Z tego powodu należy ją zaprogramować z myślą o zachowaniu braku wszelkich opłat oraz umożliwieniu publikacji swoich materiałów oraz wiedzy przez każdego, kto będzie miał ochotę i czas, aby to zrobić.

2.2 Grono odbiorców

Przez użytkowników aplikacji rozumiemy szeroko pojęte grono akademickie oraz wszelkie inne osoby zainteresowane udzielaniem darmowych lekcji i dostępem do materiałów edukacyjnych autorstwa innych zarejestrowanych w serwisie ludzi. Od żadnego z potencjalnych użytkowników nie powinno wymagać się ponadprzeciętnych umiejętności komputerowych - aplikacja powinna być przystępna w użyciu dla każdego, kto nie jest osobą predysponowaną technologicznie.

2.3 Środowisko programowe i sprzętowe

Aplikacja powinna mieć możliwość zostania zainstalowaną na komputerze świadczącym usługi serwera oraz działać na nim nieprzerwanie. Na tej samej maszynie konieczna jest instalacja i konfiguracja bazy danych, która będzie użytkowana przez zainstalowaną wcześniej aplikację w celu archiwizacji i przechowywania danych.

Zainstalowany na serwerze program powinien być zdolny do udostępniania użytkownikowi interfejsu oraz zasobów rozumianych i interpretowanych przez przeglądarkę zainstalowaną po stronie klienta.

Serwer, na którym docelowo ma działać aplikacja musi być zarządzany przez System Operacyjny z rodziny Windows, Linux lub macOS. Pod uwagę brane są dystrybucje nie starsze niż pięć lat w momencie publikacji powyższego dokumentu.

Przeglądarka internetowa, zainstalowana po stronie klienta i reprezentująca dane i interfejs dostarczone przez aplikację działającą na serwerze, powinna być jedną z wymienionych: Google Chrome, Internet Explorer, Mozilla Firefox, Safari, Opera. Jednocześnie wersja używanej przeglądarki, na której aplikacja działa bezproblemowo, nie powinna być starsza niż pięć lat w momencie publikacji powyższego dokumentu.

3. Istniejące rozwiązania w zakresie tematyki pracy

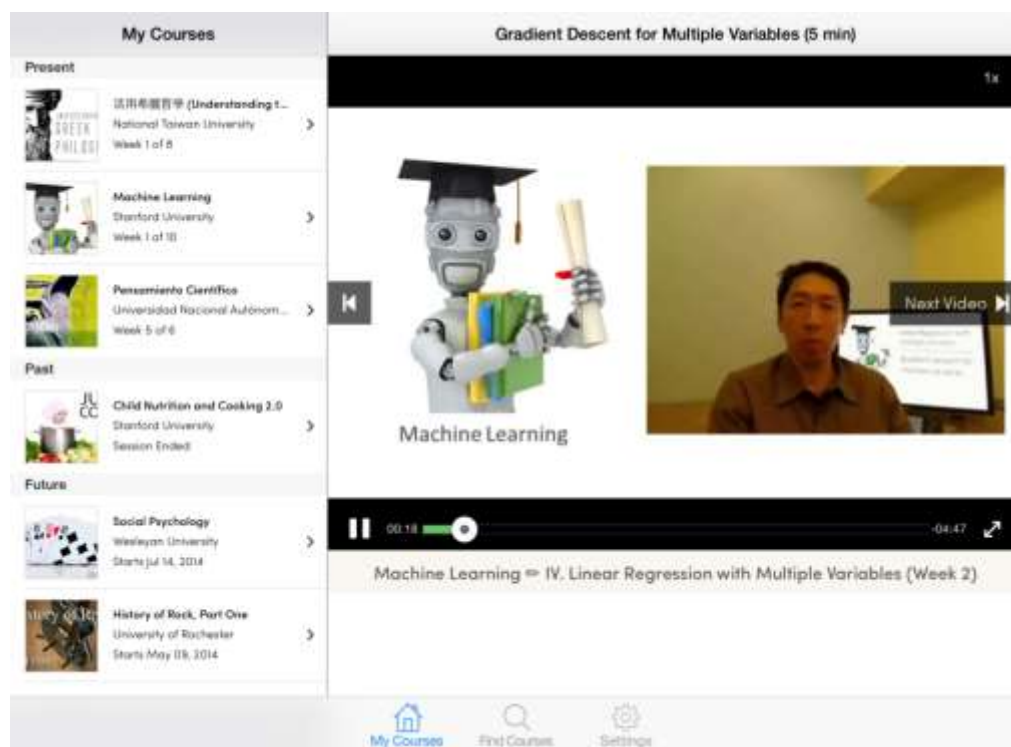
3.1 Wprowadzenie

W poniższym rozdziale omówione i przeanalizowane zostaną istniejące już aplikacje E-Learningowe, które zyskały popularność na rynku oraz oferują funkcjonalności zbliżone do tych, które zdefiniowane są dla projektu będącego przedmiotem tej pracy. Na podstawie ewaluacji wspomnianych serwisów wyciągnięte zostaną wnioski, dzięki którym proces projektowania i tworzenia aplikacji będzie efektywniejszy.

3.2 Zestawienie rozwiązań realizujących podobne założenia

Poniżej zaprezentowano i przeanalizowano kilka z wiodących, wybranych arbitralnie serwisów E-Learningowych. Naturalnie, usług internetowych tego typu jest znacznie więcej, jednak to te wiodą prym na rynku i były opisywane w niejednym artykule poświęconym rozwiązaniom nauczania zdalnego, między innymi w “Digital culture clash: ‘massive’ education in the *E-learning and Digital Cultures MOOC*” autorstwa Jeremy’ego Knoxa [31].

3.2.1 Coursera



Rysunek 3.1: Zrzut ekranu przedstawiający serwis webowy Coursera

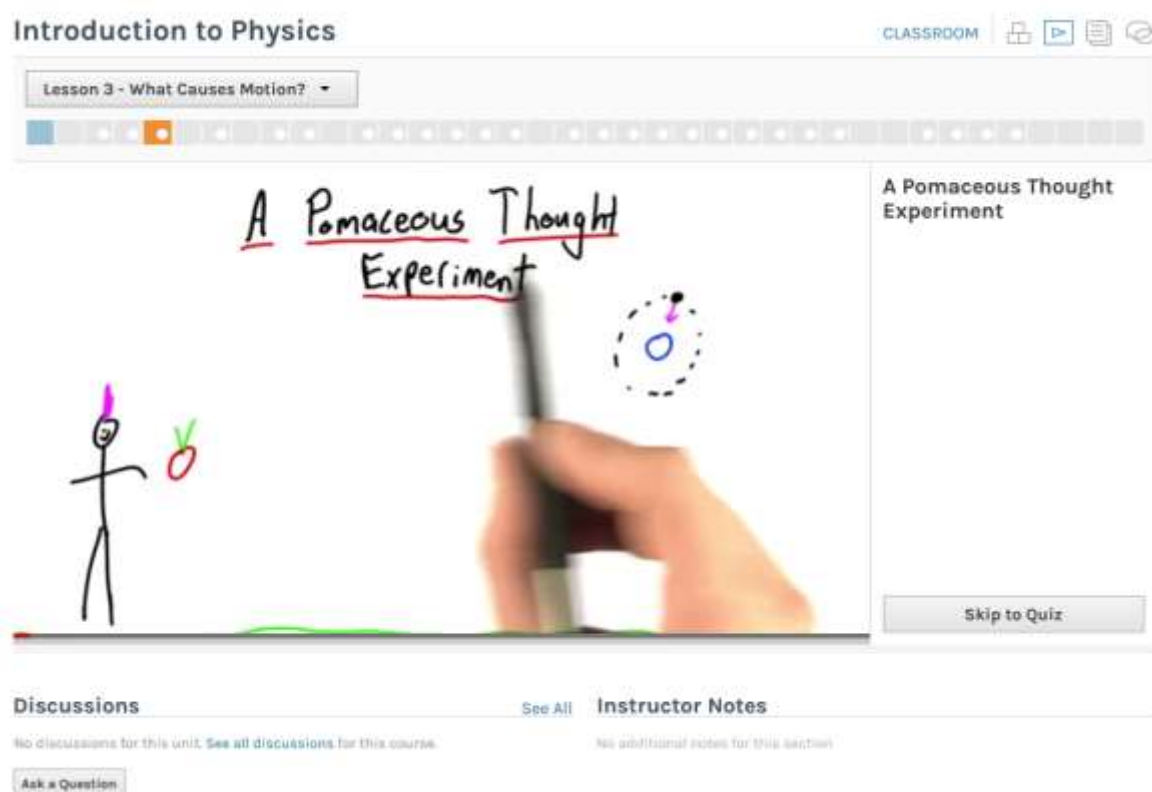
Coursera jest platformą edukacyjną mającą swoje korzenie na Uniwersytecie Stanforda w Kalifornii [1]. Serwis ten adresowany jest głównie do społeczności akademickiej, mimo to każdy może się w nim zarejestrować.

Filozofia tej organizacji opiera się na umożliwieniu prowadzenia kursów jedynie wykładowcom uniwersytetów, z którymi współpracuje. Z tego powodu, spore grono twórców i ludzi nauki niezrzeszonych z żadną uczelnią nie ma możliwości szerzenia swojej wiedzy za pomocą tego portalu.

Ponadto, nie wszystkie materiały oferowane w ramach tego serwisu są darmowe. Niektóre z kursów posiadają sekcje, do których dostęp możliwy jest dopiero po wniesieniu odpowiedniej opłaty. W pewnych przypadkach tego typu regulacje dotyczą nawet całego kursu. Z punktu widzenia ekonomicznego jest to w pełni uzasadnione, gdyż portal ten oferuje rzeczywistą alternatywę do standardowej rejestracji w ramach niektórych uczelni i kierunków studiów. Jest to jednak sprzeczne z otwartością i darmowością serwisu opisanego w wymaganiach projektowych.

Z punktu widzenia użytkowego i technologicznego Coursera realizuje wszystkie określone wcześniej wymagania. Każdy kurs podzielony jest na jednostki lekcyjne, z których każda posiada korespondujący materiał wideo oraz zasoby edukacyjne innego rodzaju. Wszystkie z tych danych mogą być w dowolnym momencie przeglądane i stworzone. Ponadto, serwis ten zapewnia świetną komunikację z użytkownikiem dzięki istniejącemu modułowi komunikacyjnemu w postaci studenckiego forum oraz bezpośrednich pytań do prowadzących.

3.2.2 Udacity

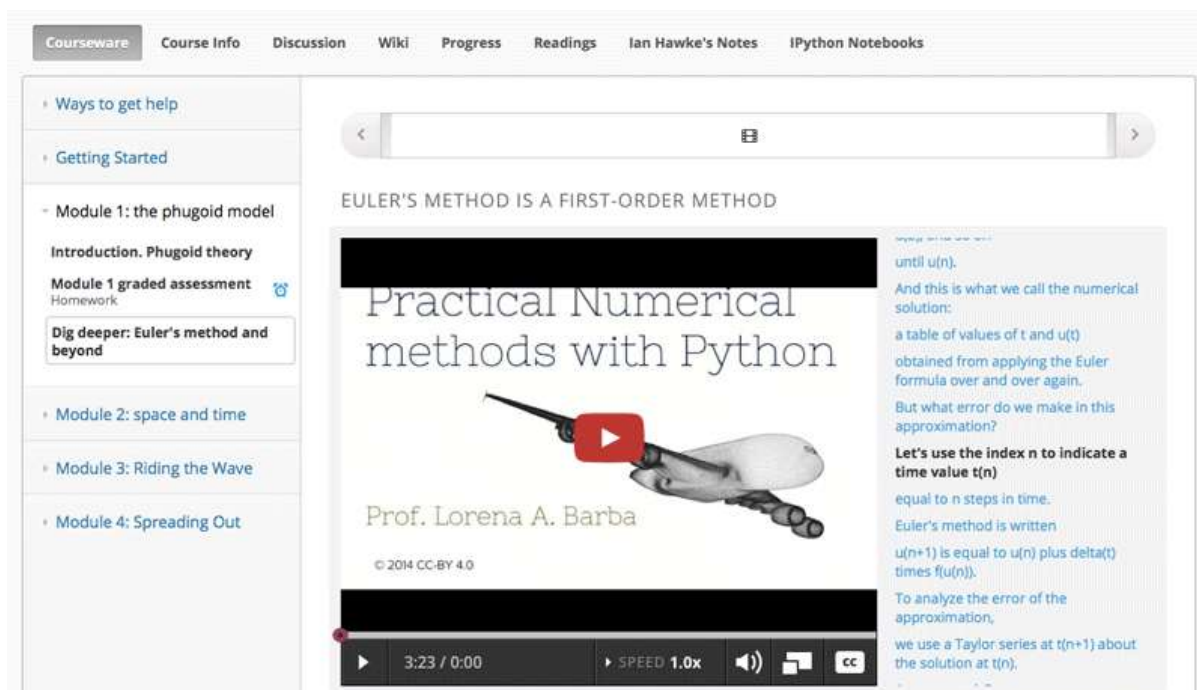


Udacity, podobnie jak Coursera, jest serwisem E-Learningowym wywodzącym się ze środowiska akademickiego Uniwersytetu Stanforda w Kaliforni [2]. Jest to jednak organizacja znacznie ściślej współpracująca z sektorem prywatnym oraz oferująca kursy komercyjne, których ukończenie bywa, w zależności od jednostki edukacyjnej, gwarantem zatrudnienia.

Udacity z punktu widzenia czysto technologicznego oferuje praktycznie ten sam wachlarz usług co wcześniej opisywana Coursera. Niemniej jest coś, co ją wyróżnia - mianowicie program "Nanodegree", który oferuje specjalną ścieżkę edukacyjną, po skończeniu której użytkownik zyskuje specjalną certyfikację zapewniającą zatrudnienie w jednej z firm partnerskich organizacji - jest to część umowy między klientem, a serwisem. Naturalnie, porozumienie to nie jest darmowe, a uczestnictwo w tym modelu edukacyjnym wynosi 199\$ miesięcznie. Całość programu trwa zazwyczaj od pół do całego roku.

Podsumowując, Udacity ze zwykłego serwisu E-Learningowego, stało się pewnego rodzaju płatną akademią webową. Warto zwrócić uwagę, że z punktu widzenia edukacyjnego model biznesowy przyjęty przez tę organizację przynosi wiele benefitów, przykładowo w postaci środków na rozwój technologiczny usprawniający działanie portalu.

3.2.3 edX



Rysunek 3.3: Zrzut ekranu przedstawiający serwis webowy edX

Ostatnią z opisywanych E-Learningowych aplikacji webowych jest edX [3]. Jest to jednocześnie organizacja non-profit, stawiająca na darmową rejestrację i uczestnictwo w oferowanych na swojej stronie internetowej kursach. Jediną opłatą jaką można ponieść podczas korzystania z tego serwisu jest związana z otrzymaniem certyfikacji od uczelni odpowiadającej za dany kurs.

Można śmiało stwierdzić, że edX i Coursera są bliźniaczymi rozwiązaniami. Zarówno aspekty technologiczne jak i podejście do użytkowników, w tym twórców zasobów edukacyjnych, jest niemalże takie samo.

EdX również stawia na środowisko akademickie, pozwalając w ramach swojego portalu publikować jedynie oficjalne kursy pod patronatami uczelni partnerskich. Tak samo jak w przypadku Coursery jest to krzywdzące w stosunku do niezrzeszonych twórców, którzy swoją wiedzę zmuszeni są przekazywać za pomocą aplikacji nie przystosowanych tak dobrze jak te wymienione powyżej - można tu wymienić edukacyjne vlogi w serwisach typu YouTube czy cykle postów na forach specjalistycznych, tracących zresztą stale na popularności.

3.3 Zestawienie przydatnych technologii i rozwiązań

Na wstępie przeglądu technicznego wymienionych wyżej aplikacji należy zwrócić uwagę na podstawowe ich cechy, które są zgodne z wymaganiami projektowymi dotyczącymi oprogramowania będącego przedmiotem tej pracy.

Naturalnie, każdy z opisanych systemów jest oparty na modelu klient-serwer. Ponadto, są one aplikacjami typu CRUD posiadającymi moduły odpowiadające za zarządzanie użytkownikami i bezpieczeństwo. Z punktu widzenia interakcji z użytkownikiem, podobieństwa widoczne są już na zamieszczonych w poprzednim podpunkcie zrzutach ekranu. Należy tu wymienić podział większych jednostek edukacyjnych, zazwyczaj kursów, na mniejsze moduły. Całość wspomagana jest przez materiały w postaci wideo, notatek oraz, czasami, slajdów dostępnych na stronach.

Na podstawie powyższych obserwacji widoczna jest bardzo duża rola multimediów na każdej z opisanych stron. Służą one lepszemu przyswajaniu informacji naukowych przez użytkownika. Bardzo istotne jest tu zastosowanie najnowszej wersji HTML w wersji piątej, która umożliwia bardzo efektywną obsługę wszelkiego rodzaju multimediów.

Dobrym tego przykładem jest chociażby element Canvas, który umożliwia sprawne i dynamiczne renderowanie grafik bez użycia wszelkiego rodzaju wtyczek i bibliotek. Jednym z możliwych jego zastosowań jest tworzenie kształtów dwuwymiarowych na odtwarzanym wideo lub animacji. Proces ten widać na zrzucie ekranu zamieszczonym w sekcji 3.2.2 poświęconej Udacity.

Poza wspomnianym elementem Canvas, w ramach HTML5 pojawiły się również elementy pozwalające na umieszczenie wielu mediów w ramach dokumentu, jak chociażby `<audio>` czy `<video>`.

Inną technologią, która jest używana przez wspomniane wyżej aplikacje oraz świetnie nadaje się do obsługi multimediów i komunikacji typu P2P, a więc zarazem wspomagania nauczania w sieci, jest kolejny standard HTML5 - WebRTC [13].

Jest to projekt realizowany na zasadzie wolnego oprogramowania, rozpoczęty przez Google, a z czasem budowany przez naprawdę spore grono osadzone w środowisku webowym - wliczając takie organizacje jak Mozilla Foundation czy Opera.

WebRTC jest zbiorem różnorodnych rozwiązań w postaci interfejsu programistycznego aplikacji wbudowanego we współczesnych przeglądarkach, wśród elementów którego wyróżnić należy

- metodę ***getUserMedia***, pozwalającą im na dostęp do audiowizualnych urządzeń wejściowych oraz możliwość zapisywania rejestrowanych przez nie danych,

- **RTCPeerConnection** będące interfejsem połączenia WebRTC między lokalną maszyną, a zdalnym hostem - jest on wykorzystywany przykładowo w nawiązywaniu połączeń audiowizualnych,
- **RTCDataChannel** będące interfejsem dwukierunkowego połączenia WebRTC typu P2P, za pomocą którego możliwy jest dwustronny przesył danych.

Konkludując wzmiankę o WebRTC należy wspomnieć jak rozwiązania komunikacyjne oferowane przez ten projekt mogą wspomóc procesy E-Learningowe. Najlepszym przykładem jest tu możliwość nawiązywania bezpośredniej komunikacji między nauczycielem, a uczniem bez konieczności używania aplikacji trzecich jak na przykład Skype czy internetowe czaty wideo.

Ponadto, dzięki dostępowi do audiowizualnych urządzeń peryferyjnych twórca kursów edukacyjnych jest w stanie nagrać wykład w dowolnym miejscu i czasie tak długo jak będzie dysponował urządzeniem z przeglądarką zgodną ze standardem HTML5.

3.4 Podsumowanie i wnioski

Dzięki analizie funkcjonalnej oraz technologicznej istniejących już na rynku aplikacji E-Learningowych poznane zostały dobre praktyki i rozwiązania, które mogą być zastosowane również w programie będącym tematem tej pracy.

Z punktu widzenia oferowanych funkcjonalności i odbioru treści przez klienta końcowego bardzo ważną rolę okazały się odgrywać multimedia, których obsługę zapewniają elementy będące składową standardu HTML5.

Jednym z takich elementów jest projekt WebRTC, dzięki któremu przeglądarki mają dostęp do urządzeń peryferyjnych i mogą inicjować połączenia typu P2P, co w niektórych przypadkach może być bardzo pomocne w kontekście internetowego nauczania [13].

4. Wytyczne projektowe

4.1 Zakres prac

Za cel pracy postawiono realizację aplikacji E-Learningowej, w ramach której użytkownicy są w stanie tworzyć kursy i jednostki lekcyjne oraz do nich dołączać. Ponadto twórcy w ramach lekcji będą mogli tworzyć własne audiowizualne pomoce naukowe w postaci nagrywanych przez aplikację wykładów zsynchronizowanych z wcześniej przygotowanymi prezentacjami.

W porównaniu z konkurencją, system ten będzie charakteryzował się otwartością i nie będzie wymagał od użytkowników żadnych opłat oraz bycia pracownikiem akademickim lub zarejestrowanym na jakiejkolwiek uczelni studentem.

4.2 Wymagania funkcjonalne

W ramach systemu jesteśmy w stanie rozróżnić dwa typy użytkowników - zalogowanego oraz niezalogowanego. Użytkownik niezalogowany nie ma żadnej możliwości korzystania poza dostępem do strony rejestracji i logowania.

Użytkowników zalogowanych można z punktu widzenia kursu podzielić na trzy kategorie: twórcę, uczestnika oraz użytkownika nieuczestniczącego.

Możliwości użytkownika niezalogowanego to:

- rejestracja nowego konta w ramach systemu,
- zalogowanie na istniejące konto w ramach systemu.

Możliwości użytkownika zalogowanego to:

- wylogowanie z systemu,
- zmiana danych profilowych takich jak email, pseudonim oraz hasło,
- przeglądanie wszystkich kursów,
- filtrowanie listy kursów,
- wyszukiwanie konkretnego kursu po nazwie,
- tworzenie nowych kursów,
- tworzenie lekcji w ramach stworzonego przez siebie kursu,
- wgrywanie gotowych materiałów w ramach stworzonej przez siebie lekcji,
- nagrywanie wykładu w formie audiowizualnej za pomocą urządzeń peryferyjnych podłączonych do komputera w ramach stworzonej przez siebie lekcji,
- synchronizacji nagrywanego przez siebie wykładu w formie audiowizualnej z wcześniej przygotowaną prezentacją w ramach stworzonej przez siebie lekcji,
- dołączania do dostępnych kursów,
- przeglądania lekcji stworzonych w ramach kursów, do których dołączył, lub które stworzył,

- pobierania materiałów dostępnych w ramach lekcji kursu, do którego dołączył, lub który stworzył,
- odtwarzania wykładu w formie nagrania audiowizualnego zsynchronizowanego z prezentacją multimedialną w ramach lekcji kursu, do którego dołączył, lub który stworzył,
- otrzymywania powiadomień w czasie rzeczywistym na temat nowo powstałych lekcji w ramach kursu, do którego dołączył.

4.3 Wymagania niefunkcjonalne

- System powinien być oparty na trzech warstwach - serwera bazy danych, serwera aplikacji oraz interfejsu graficznego aplikacji wyświetlanego w przeglądarce internetowej,
- dane przekazywane między warstwami systemu powinny być przedstawiane w formacie JSON,
- dostęp do zasobów aplikacji powinien być autoryzowany - materiały zawarte w konkretnym kursie powinny być udostępniane jedynie jego autorowi oraz użytkownikom na niego zapisanym,
- interfejs webowy aplikacji powinien być skalowalny - poprawnie wyświetlany na ekranach o różnych rozdzielczościach tak, aby aplikacja była przystępna niezależnie od używanego urządzenia,
- interfejs webowy aplikacji powinien być zrealizowany zgodnie z modelem aplikacji jednostronicowej, co przypomina interfejs aplikacji pulpitowych i jest przyjazne użytkownikowi.

4.4 Zarys architektury systemu

Zgodnie z wymaganiem niefunkcjonalnym system zrealizowany będzie zgodnie ze standardem trójwarstwowej aplikacji webowej. Warstwa danych opierać się będzie na bazie danych, z której dane pobierane i przetwarzane będą przez serwer aplikacji, który tak przygotowane informacje udostępniać będzie użytkownikowi za pośrednictwem interfejsu webowego działającego na przeglądarce internetowej.

4.5 Technologie i metodyki wykorzystane w projekcie

Systemem zarządzającym nie relacyjną i bardzo bazą danych będzie MongoDB [10]. Dostęp do tego repozytorium oraz wszelka logika biznesowa realizowane będą przez aplikację napisaną w środowisku uruchomieniowym Node.js, w której dużą rolę odgrywał będzie również framework do tworzenia aplikacji webowych Express.js. Aby zapewnić powiadomienia w czasie rzeczywistym, wykorzystane zostaną połączenia typu WebSocket, które obsługiwane będą przez kolejny framework - Socket.IO [7]. Tak zrealizowana aplikacja webowa będzie udostępniać zasoby zgodnie ze standardem REST API.

Po stronie klienta zastosowany zostaną frameworki Angular.js oraz Bootstrap, które pozwolą na stworzenie przyjaznego wizualnie, jednostronicowego i skalowalnego interfejsu webowego [5][8].

Przesyłane w systemie dane będą w formacie JSON, co jest domyślnym rozwiązaniem dla wszystkich technologii wykorzystywanych w projekcie.

Tak wybrany zestaw technologii umotywowany jest faktem ich popularności, sporymi zasobami dokumentacji oraz spójnością języka programowania - wszystkie trzy warstwy,

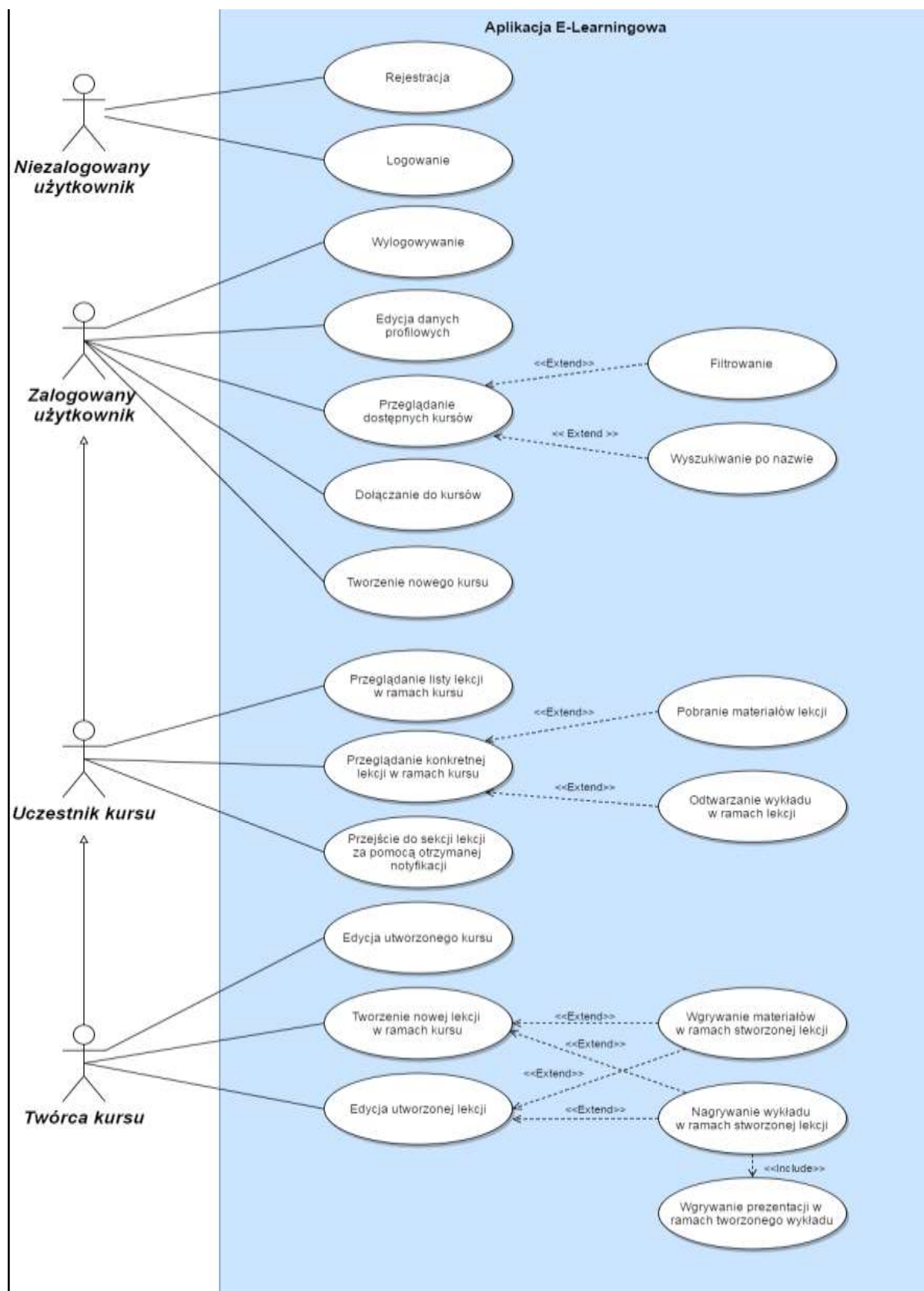
nawet powłoka MongoDB, używają JavaScriptu, co znacznie ułatwia proces implementacji oraz integracji poszczególnych poziomów [10].

Ponadto, oprogramowanie wykonane będzie zgodnie z zachowaniem zasad tak zwanego “czystego kodu”. Funkcjonalności serwisu będą podzielone na hermetyczne moduły, co pozwoli na uniwersalność i skalowalność, dzięki czemu w przyszłości projekt będzie miał szansę na zostanie rozbudowanym bez konieczności ingerencji w już powstały kod.

5. Projekt

5.1 Wprowadzenie

Poprzednie sekcje pracy definiowały podstawowe konceptualne i funkcjonalne kwestie dotyczące rozwijanej aplikacji. Zwłaszcza poprzedni rozdział skupił się na definicji wymagań oraz technologii użytych w systemie. W tej części pracy przedstawiony zostanie szeroko rozumiany projekt aplikacji - określone zostaną przypadki użycia, interfejs oraz inne ważne aspekty takie jak kwestie bezpieczeństwa i przetwarzania multimediów. Powstała w ten sposób dokumentacja jest elementem bazowym implementacji aplikacji.



Rysunek 5.1: Diagram przypadków użycia

1. Rejestracja

Ścieżka główna

- 1.1. System wyświetla niezalogowanemu użytkownikowi formularz rejestracyjny.
- 1.2. Użytkownik wypełnia formularz rejestracyjny.
- 1.3. System waliduje unikalność danych podanych przez użytkownika.
- 1.4. Walidacja kończy się pozytywnie, system tworzy nowe konto użytkownika i przekierowuje użytkownika do dalszej sekcji aplikacji, dostępnej tylko dla zalogowanych.

Ścieżki alternatywne

- 1.3.a Walidacja kończy się negatywnie, następuje powrót do punktu 1.1.

2. Logowanie

Ścieżka główna

- 2.1. System wyświetla niezalogowanemu użytkownikowi formularz do logowanie.
- 2.2. Użytkownik wypełnia formularz do logowanie.
- 2.3. System waliduje poprawność danych podanych przez użytkownika.
- 2.4. Walidacja kończy się pozytywnie, system przekierowuje użytkownika do dalszej sekcji aplikacji, dostępnej tylko dla zalogowanych.

Ścieżki alternatywne

- 2.3.a Walidacja kończy się negatywnie, użytkownik zostaje poinformowany o zaistniałym błędzie, następuje powrót do punktu 2.1.

3. Wylogowywanie

Ścieżka główna

- 3.1. Będąc w sekcji dla zalogowanych, użytkownik naciska przycisk odpowiedzialny za wylogowywanie.
- 3.2. System wylogowuje użytkownika.

4. Zmiana danych profilowych

- 4.1. Będąc w sekcji dla zalogowanych, użytkownik naciska przycisk profilu użytkownika.
- 4.2. System wyświetla informacje na temat profilu użytkownika oraz formularz edycji danych.
- 4.3. Użytkownika wypełnia formularz edycji danych.
- 4.4. System waliduje unikalność danych podanych przez użytkownika.
- 4.5. Walidacja kończy się pozytywnie, nowe dane użytkownika zostają wprowadzone do bazy danych.

Ścieżki alternatywne

- 4.5.a Walidacja kończy się negatywnie, użytkownik zostaje poinformowany o zaistniałym błędzie, następuje powrót do punktu 4.2.

5. Przeglądanie dostępnych kursów
Ścieżka główna
5.1. Będąc w sekcji dla zalogowanych, użytkownik naciska przycisk dostępnych kursów.
5.2. System wyświetla listę wszystkich kursów stworzonych w aplikacji.
6. Filtrowanie dostępnych kursów
Ścieżka główna
6.1. Będąc w sekcji wyświetlającej listę wszystkich kursów stworzonych w aplikacji, użytkownik naciska przycisk filtrowania wyników.
6.2. System wyświetla możliwe opcje filtrowania kursów dostępnych w ramach aplikacji.
6.3. Użytkownik wybiera jedną z dostępnych opcji poprzez naciśnięcie odpowiedniego przycisku.
6.4. System wyświetla kursy przefiltrowane zgodnie z wybranym przez użytkownika w punkcie 6.3 kryterium
7. Wyszukiwanie dostępnych kursów po nazwie
Ścieżka główna
7.1. Będąc w sekcji wyświetlającej listę wszystkich kursów stworzonych w aplikacji, użytkownik naciska pole wyszukiwania kursów i wpisuje nazwę szukanego kursu lub jej część.
7.2. System wyświetla wszystkie kursy których nazwa pasuje do wpisanego przez użytkownika w punkcie 7.1 zapytania.
8. Dołączanie do kursu
Ścieżka główna
8.1. Będąc w sekcji wyświetlającej listę wszystkich kursów stworzonych w aplikacji, użytkownik naciska przycisk dołączenia do jednego z wyświetlanych kursów.
8.2. System zapisuje użytkownika jako uczestnika kursu.
8.3. System przekierowuje użytkownika na stronę kursu.
9. Przeglądanie dostępnych lekcji w ramach kursu
Ścieżka główna
9.1. Będąc w sekcji wyświetlającej konkretny kurs użytkownikowi wyświetlone są wszystkie stworzone w jego ramach lekcje.
10. Przeglądanie konkretnej lekcji w ramach kursu
Ścieżka główna

- 10.1. Będąc w sekcji wyświetlającej konkretny kurs wraz z utworzonymi w jego ramach lekcjami, użytkownik wybiera jedną z nich poprzez naciśnięcie jej nazwy.
 - 10.2. System przekierowuje użytkownika na stronę lekcji.
 - 10.3. System wyświetla użytkownikowi wszystkie materiały wgrane w ramach danej lekcji oraz przycisk przekierowujący do wykładu.
11. Odtwarzanie wykładu w ramach konkretnej lekcji
- Ścieżka główna**
- 11.1. Będąc w sekcji wyświetlającej konkretną lekcję użytkownik naciska przycisk przejścia do wykładu.
 - 11.2. System przekierowuje użytkownika do sekcji z wykładem.
 - 11.3. System pobiera odpowiednie multimedia z serwera, w postaci slajdów oraz pliku audiowizualnego, w międzyczasie użytkownikowi wyświetlana jest animacja ładowania.
 - 11.4. Pliki zostają załadowane, system wyświetla użytkownikowi pierwszy ze slajdów oraz odtwarzacz wideo.
 - 11.5. Użytkownik naciska przycisk odtwarzania, odtwarzacz wideo wyświetla nagrany wykład. Jednocześnie slajdy ulegają tranzycji wraz z postępem wyświetlanego wideo.
- Ścieżki alternatywne**
- 11.5.a Podczas odtwarzania użytkownik naciska przycisk pauzy, zatrzymując odtwarzanie wideo oraz tranzycje slajdów.
 - 11.5.b Po załadowaniu plików użytkownik naciska przycisk odpowiadający za tryb synchronizacji i wyłącza go umożliwiając sterowanie slajdami samemu za pomocą dwóch przycisków przesuwających slajdów w przód i w tył.
 - 11.5.c Po wyłączeniu synchronizacji użytkownik po raz kolejny naciska przycisk odpowiadający za tryb synchronizacji i włącza go po czym następuje tranzycja slajdów zgodna z synchronizacją.
12. Tworzenie nowego kursu
- Ścieżka główna**
- 12.1. Będąc w sekcji wyświetlającej listę wszystkich kursów stworzonych w aplikacji, użytkownik naciska przycisk tworzenia nowego kursu.
 - 12.2. System przekierowuje użytkownika do sekcji tworzenia nowego kursu.
 - 12.3. System wyświetla formularz tworzenia nowego kursu.
 - 12.4. Użytkownik wypełnia formularz tworzenia nowego kursu.
 - 12.5. System waliduje unikalność danych wprowadzonych przez użytkownika.
 - 12.6. Walidacja przebiega pozytywnie, nowy kurs zostaje zapisany do bazy danych, a użytkownik zostaje przekierowany do sekcji wyświetlającej kursy.
- Ścieżki alternatywne**

- 12.6.a Walidacja przebiega negatywnie, użytkownik zostaje poinformowany o błędach, następuje powrót do punktu 12.3.
13. Edycja istniejącego kursu
- Ścieżka główna**
- 13.1. Będąc w sekcji wyświetlającej listę wszystkich kursów stworzonych w aplikacji, twórca kursu naciska przycisk edycji danego kursu.
- 13.2. System przekierowuje użytkownika do sekcji edycji kursu.
- 13.3. System wyświetla formularz edycji kursu.
- 13.4. Użytkownik wypełnia formularz edycji kursu.
- 13.5. System waliduje poprawność danych wprowadzonych przez użytkownika.
- 13.6. Walidacja przebiega pozytywnie, kurs zostaje zmodyfikowany, a zmiany zostają zapisane w bazie danych. System przekierowuje użytkownika do sekcji wyświetlającej wszystkie kursy.
- Ścieżki alternatywne**
- 13.6.a Walidacja przebiega negatywnie, użytkownik zostaje poinformowany o Błędach, następuje powrót do punktu 13.3.
14. Tworzenie nowej lekcji w ramach stworzonego kursu
- Ścieżka główna**
- 14.1. Będąc w sekcji wyświetlającej konkretny kurs, jego twórca naciska przycisk tworzenia nowej lekcji.
- 14.2. System wyświetla użytkownikowi formularz tworzenia nowej lekcji.
- 14.3. Użytkownik wypełnia formularz tworzenia nowej lekcji.
- 14.4. System waliduje unikalność danych wprowadzonych przez użytkownika.
- 14.5. Walidacja przebiega pozytywnie, lekcja zostaje utworzona i zapisana w bazie danych. System przekierowuje użytkownika do sekcji wyświetlającej nowo powstały kurs.
- Ścieżki alternatywne**
- 14.5.a Walidacja przebiega negatywnie, użytkownik zostaje poinformowany o błędach, następuje powrót do punktu 14.2.
15. Edycja istniejącej lekcji
- Ścieżka główna**
- 15.1. Będąc w sekcji wyświetlającej konkretną lekcję, jej twórca naciska przycisk edycji lekcji.
- 15.2. System wyświetla formularz edycji lekcji.
- 15.3. Użytkownik wypełnia formularz edycji lekcji.
- 15.4. System waliduje poprawność danych wprowadzonych przez użytkownika.
- 15.5. Walidacja przebiega pozytywnie, lekcja zostaje zmodyfikowana, a zmiany zostają zapisane w bazie danych. System przekierowuje użytkownika do sekcji wyświetlającej konkretną lekcję.
- Ścieżki alternatywne**

- 15.5.a Walidacja przebiega negatywnie, użytkownik zostaje poinformowany o błędach, następuje powrót do punktu 15.2.
16. Wgrywanie materiałów w ramach stworzonej lekcji
- Ścieżka główna**
- 16.1. Będąc w sekcji wyświetlającej konkretną lekcję, jej twórca naciska przycisk wgrywania nowego materiału.
 - 16.2. System wyświetla formularz wyboru pliku.
 - 16.3. Użytkownik wybiera plik z pamięci komputera.
 - 16.4. Plik zostaje wgrany na serwer i przypisany do konkretnej lekcji.
- Ścieżki alternatywne**
- 16.4.a System odrzuca plik wybrany przez użytkownika ze względu na zbyt duży rozmiar lub niedozwolony format. Użytkownik zostaje poinformowany o błędach i przekierowany do sekcji wyświetlającej konkretną lekcję.
17. Nagrywanie wykładu w ramach stworzonej lekcji
- Ścieżka główna**
- 17.1. Będąc w sekcji wyświetlającej konkretną lekcję, jej twórca naciska przycisk nagrywania wykładu.
 - 17.2. System wyświetla formularz wyboru pliku prezentacji.
 - 17.3. Użytkownik wybiera plik prezentacji z pamięci komputera.
 - 17.4. Plik zostaje wgrany na serwer i przetworzony na graficzne reprezentacje slajdów, które zostają wysłane z powrotem do klienta.
 - 17.5. System wyświetla pierwszy slajd wraz z przyciskami kontrolującymi transzycje slajdów oraz ekran pokazujący obraz z kamery podłączonej do komputera użytkownika.
 - 17.6. Użytkownik naciska przycisk nagrywania znajdujący się na podglądzie widoku z kamery.
 - 17.7. System zaczyna proces nagrywania.
 - 17.8. Użytkownik naciska przycisk transzycji slajdów.
 - 17.9. System rejestruje transzycję slajdów i zapisuje ją do tymczasowego obiektu synchronizującego.
 - 17.10. Użytkownik naciska przycisk zakończenia nagrywania znajdujący się na podglądzie widoku nagrywanego obrazu.
 - 17.11. System wysyła zarejestrowany w formie audiowizualnej wykład na stronę serwera wraz z obiektem synchronizującym.
 - 17.12. System zapisuje odebrane pliki w pamięci i przekierowuje użytkownika do sekcji wyświetlającej konkretną lekcję
- Ścieżki alternatywne**
- 17.4.a System odrzuca plik wybrany przez użytkownika ze względu na zbyt duży rozmiar lub niedozwolony format. Użytkownik zostaje poinformowany o błędach i przekierowany do sekcji wyświetlającej konkretną lekcję.

- 17.4.b System nie jest w stanie przetworzyć wybranego pliku. Użytkownik zostaje poinformowany o błędach i przekierowany do sekcji wyświetlającej konkretną lekcję.
 - 17.6.a System nie jest w stanie uzyskać dostępu do urządzeń peryferyjnych. Użytkownik zostaje poinformowany o błędach i przekierowany do sekcji wyświetlającej konkretną lekcję.
 - 17.12.a System nie jest w stanie zapisać plik utworzonego przez użytkownika. Użytkownik zostaje poinformowany o błędach i przekierowany do sekcji wyświetlającej konkretną lekcję.
18. Przechodzenie do nowo powstałej lekcji poprzez link podany w notyfikacji
- Ścieżka główna**
- 18.1. System wyświetla użytkownikowi notyfikację o nowo utworzonej lekcji w ramach kursu, do którego użytkownik ma dostęp.
 - 18.2. Użytkownik naciska przycisk przejścia do sekcji lekcji umieszczony na notyfikacji.
 - 18.3. System przekierowuje użytkownika do sekcji lekcji.

5.3 Interfejs

5.3.1 Omówienie

Interfejs aplikacji powinien opierać się na jednostronicowej aplikacji. Dzięki temu strona sprawia wrażenie zbliżone do programu pulpituowego. Powinno to być osiągnięte przez zastosowanie odpowiednich narzędzi - na przykład frameworku Angular.js [5].

Ważnym aspektem aplikacji jest jej intuicyjna nawigacja między sekcjami, co powinno być zapewnione poprzez pasek nawigacyjny, który zgodnie z przyjętymi standardami znajduje się zazwyczaj na górze lub po lewej stronie wyświetlanego widoku.

Główny zasób aktualnie wyświetlanego widoku nie powinien nachodzić na elementy nawigujące i być możliwie wyeksponowany w centrum interfejsu.

Kolejnym ze stałych i ważnych elementów strony jest przycisk służący wylogowywaniu, który powinien znajdować się w prawy górnym rogu interfejsu.

Kolejnym bardzo istotnym aspektem wizualnym jest dobranie odpowiedniej kolorystyki strony, która powinna korespondować z edukacyjnym charakterem serwisu. Zaleca się więc, aby dobrać kolory stonowane i spokojne.

5.3.2 Makiety

Podczas implementacji interfejsu należy sugerować się poniższymi makietami widoków. Wszystkie z nich zostały zaprojektowane przy użyciu narzędzia Balsamiq [35].

A Web Page

http://

Rejestracja

Pseudonim

E-Mail

Hasło

Zarejestruj

The image shows a wireframe of a web browser window titled "A Web Page". The address bar contains "http://". The main content area features a registration form titled "Rejestracja". The form includes three input fields labeled "Pseudonim", "E-Mail", and "Hasło", and a "Zarejestruj" button at the bottom.

Created with Balsamiq - www.balsamiq.com

Rysunek 5.2: Makieta widoku rejestracji użytkownika

A Web Page

http://

Logowanie

E-Mail

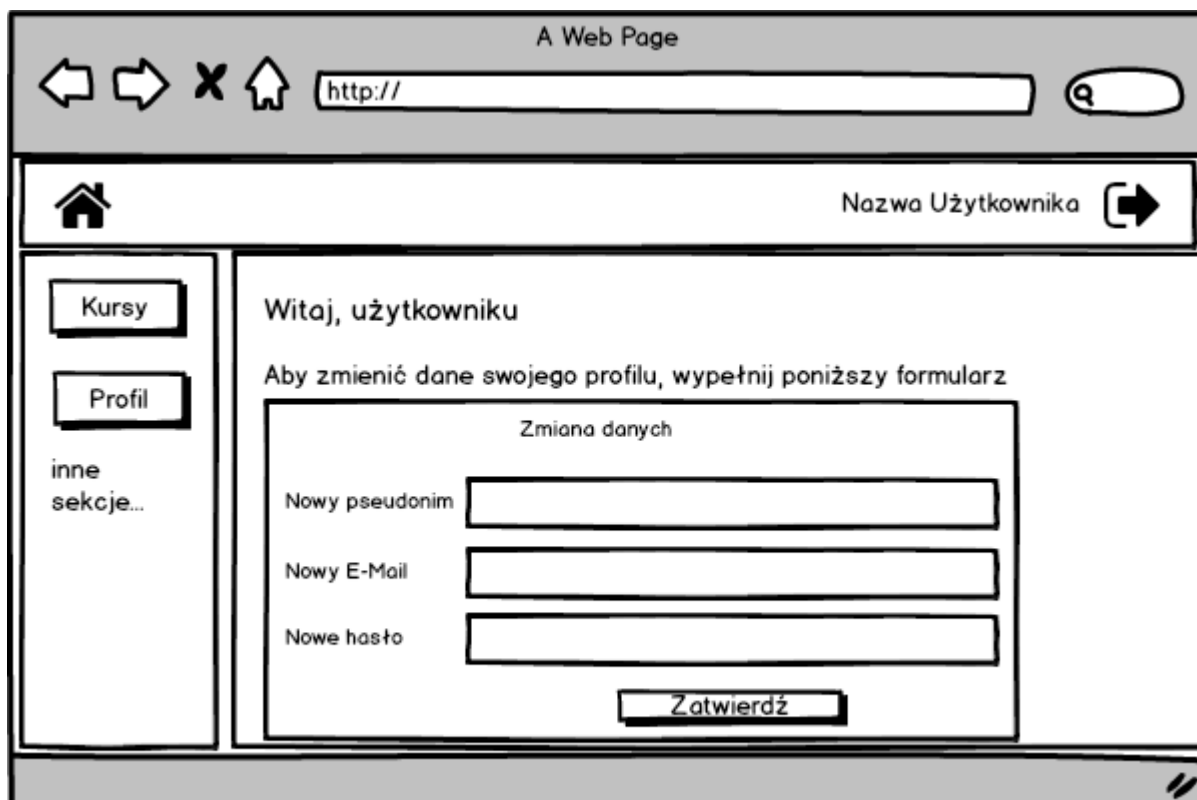
Hasło

Zaloguj

The image shows a wireframe of a web browser window titled "A Web Page". The address bar contains "http://". The main content area features a login form titled "Logowanie". The form includes two input fields labeled "E-Mail" and "Hasło", and a "Zaloguj" button at the bottom.

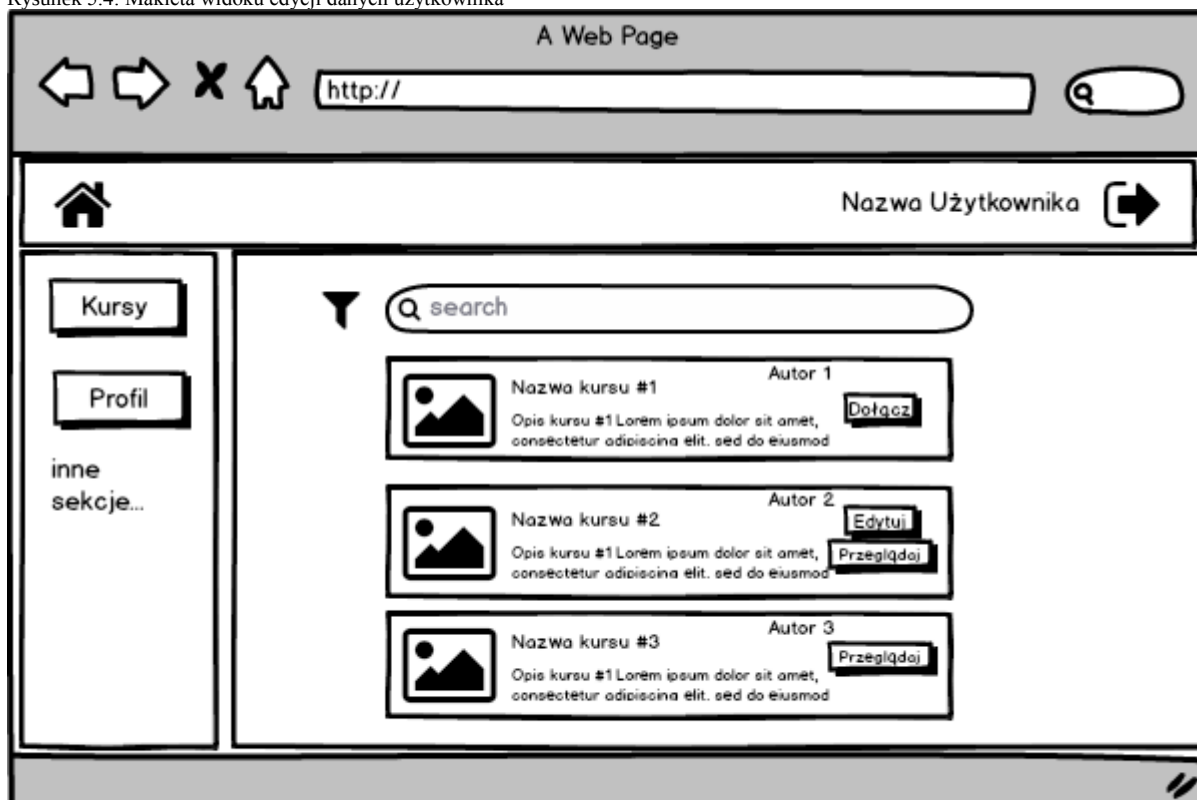
Created with Balsamiq - www.balsamiq.com

Rysunek 5.3: Makieta widoku logowania użytkownika



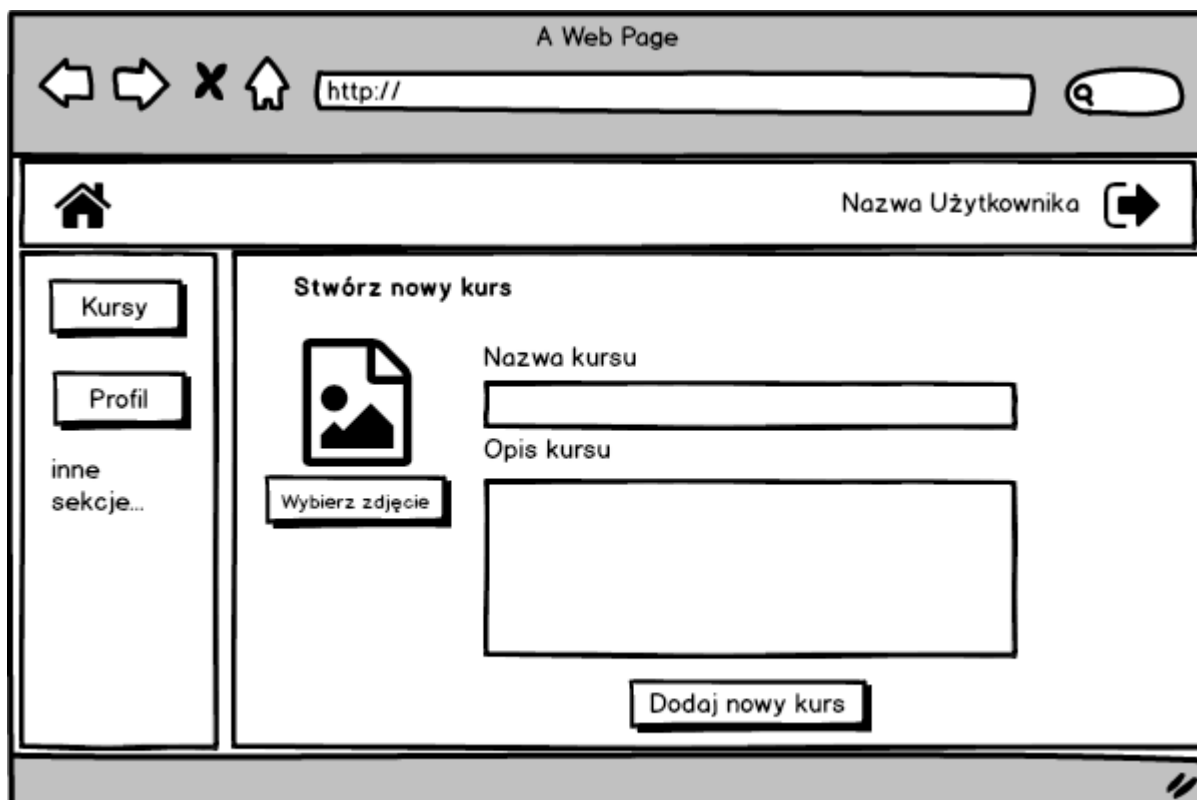
Created with Balsamiq - www.balsamiq.com

Rysunek 5.4: Makieta widoku edycji danych użytkownika



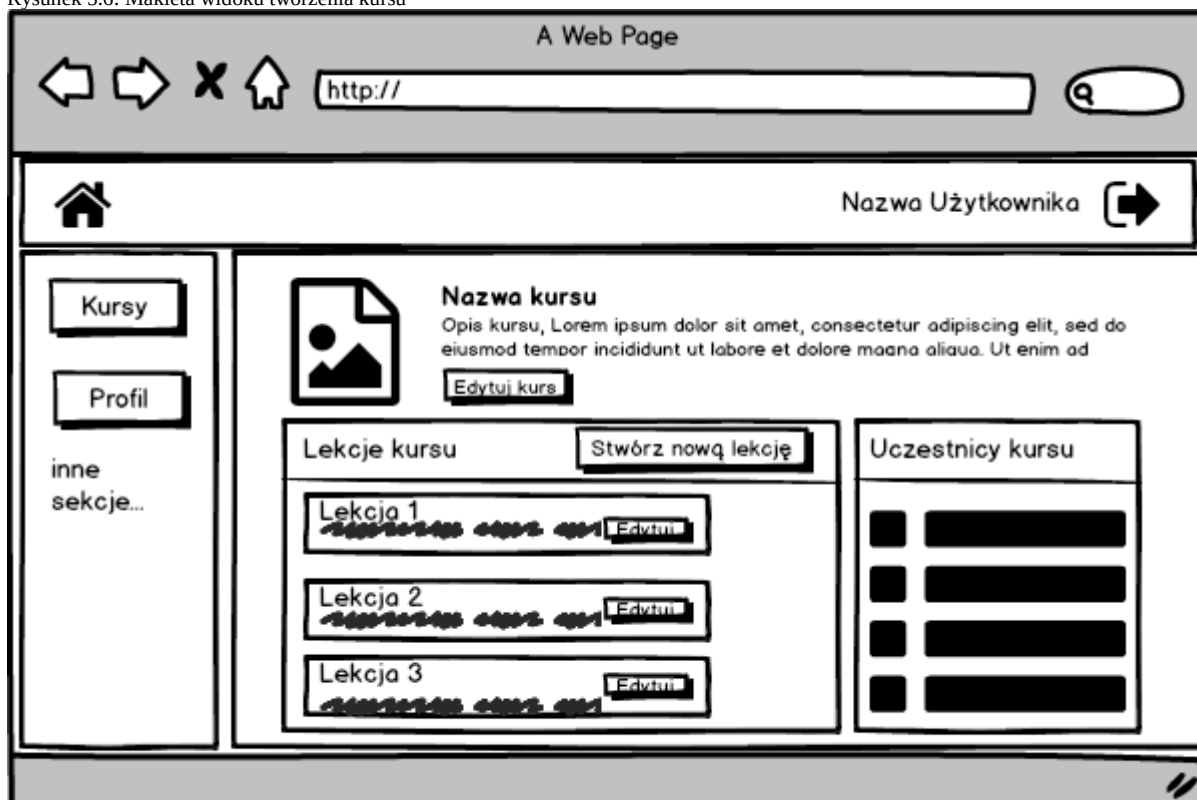
Created with Balsamiq - www.balsamiq.com

Rysunek 5.5: Makieta widoku listy kursów



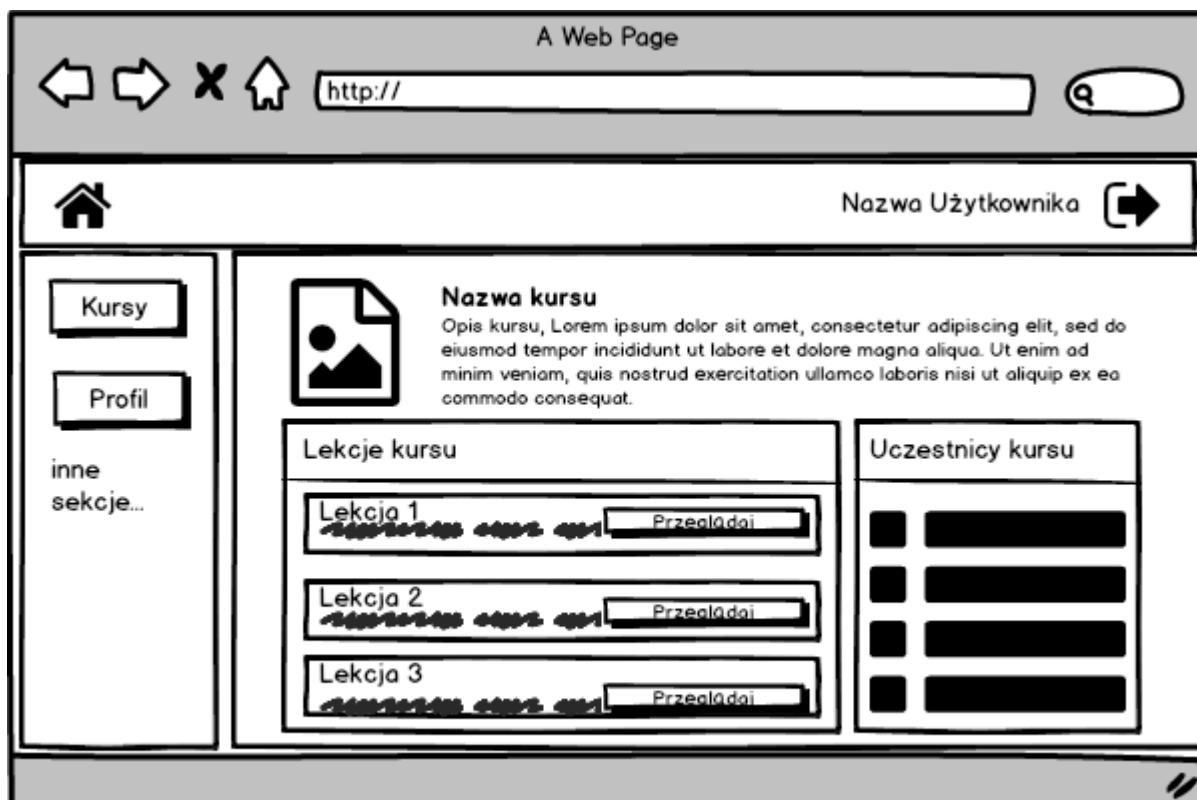
Created with Balsamiq - www.balsamiq.com

Rysunek 5.6: Makieta widoku tworzenia kursu



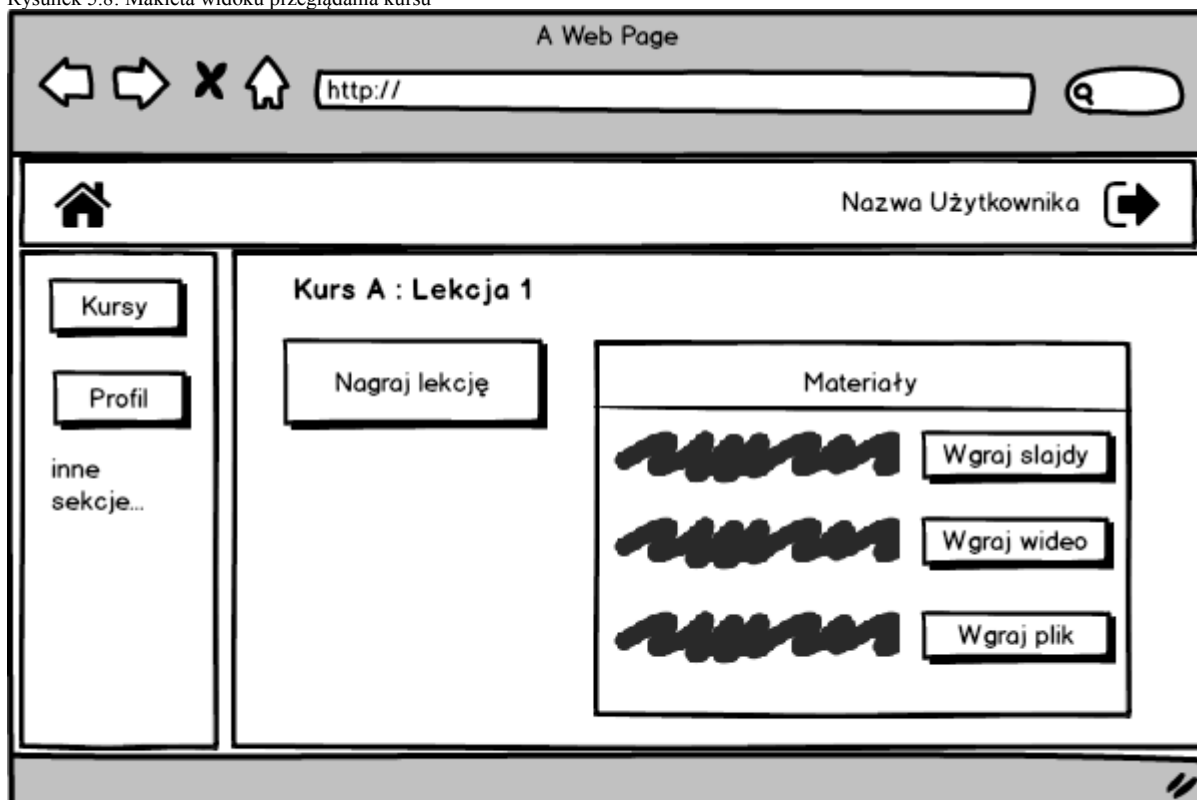
Created with Balsamiq - www.balsamiq.com

Rysunek 5.7: Makieta widoku edycji kursu



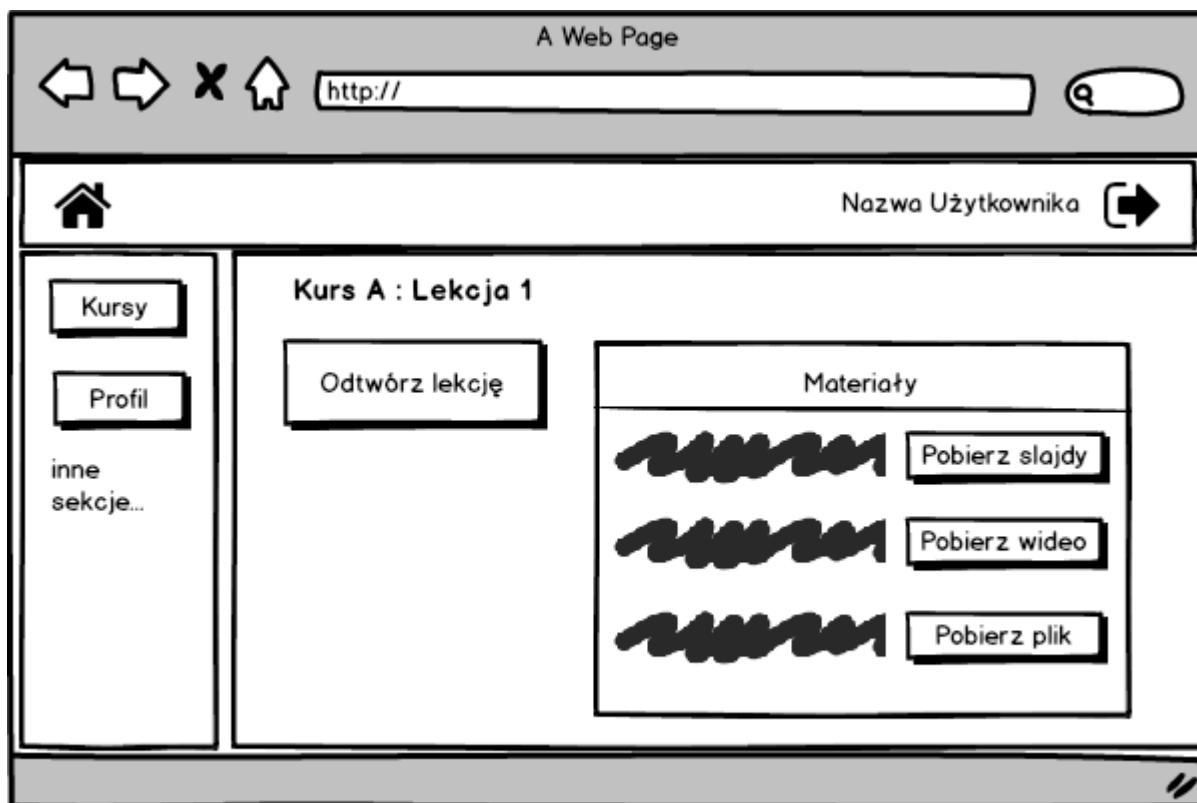
Created with Balsamiq - www.balsamiq.com

Rysunek 5.8: Makieta widoku przeglądania kursu



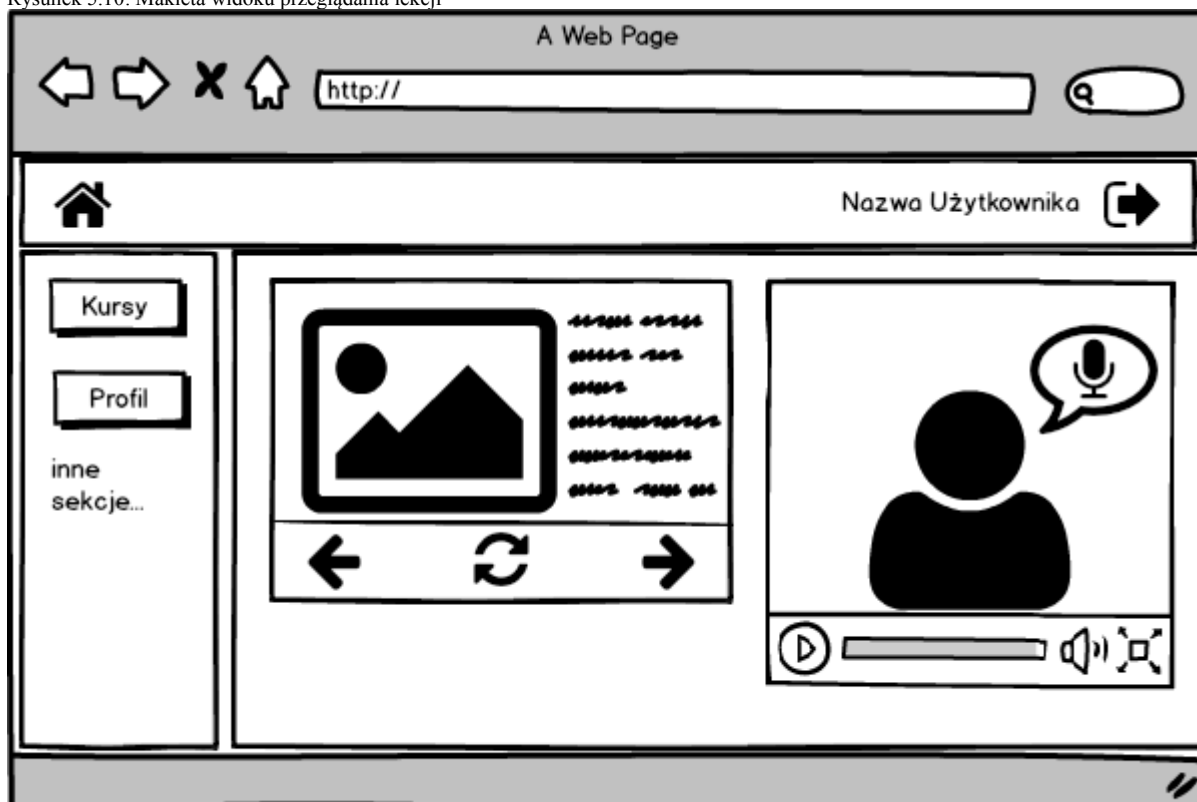
Created with Balsamiq - www.balsamiq.com

Rysunek 5.9: Makieta widoku edycji lekcji



Created with Balsamiq - www.balsamiq.com

Rysunek 5.10: Makieta widoku przeglądania lekcji



Created with Balsamiq - www.balsamiq.com

Rysunek 5.11: Makieta widoku nagrania lekcji

5.4 Baza danych

Do poprawnego funkcjonowania aplikacja konieczna jest implementacja działającej bazy danych. Z racji użycia w projekcie nie relacyjnej bazy danych, MongoDB, są one reprezentowane w formie dokumentów, które faktycznie przyjmują format JSON [10].

W bazie danych aplikacji przechowywane są typy obiektów przedstawione poniżej.

1. “User” - encja realizująca logiczną reprezentację użytkownika zarejestrowanego w ramach systemu.

W modelu obiektu użytkownika definiujemy następujące pola:

- “email” - reprezentacja adresu mailowego użytkownika, przechowywana w postaci danych tekstowych, będąca wartością konieczną i unikalną w ramach zbioru zarejestrowanych użytkowników,
- “username” - reprezentacja pseudonimu użytkownika, przechowywana w postaci danych tekstowych, będąca wartością konieczną i unikalną w ramach zbioru zarejestrowanych użytkowników,
- “hash” - reprezentacja wartości funkcji skrótu dla hasła podanego przez użytkownika, przechowywana w postaci danych tekstowych,
- “salt” - reprezentacja jednorazowo wygenerowanej losowo wartości używanej podczas aplikowania funkcji skrótu na hasło podanym przez użytkownika, przechowywana w postaci danych tekstowych.

2. “Course” - encja realizująca logiczną reprezentację kursu stworzonego w ramach systemu.

W modelu obiektu kursu definiujemy następujące pola:

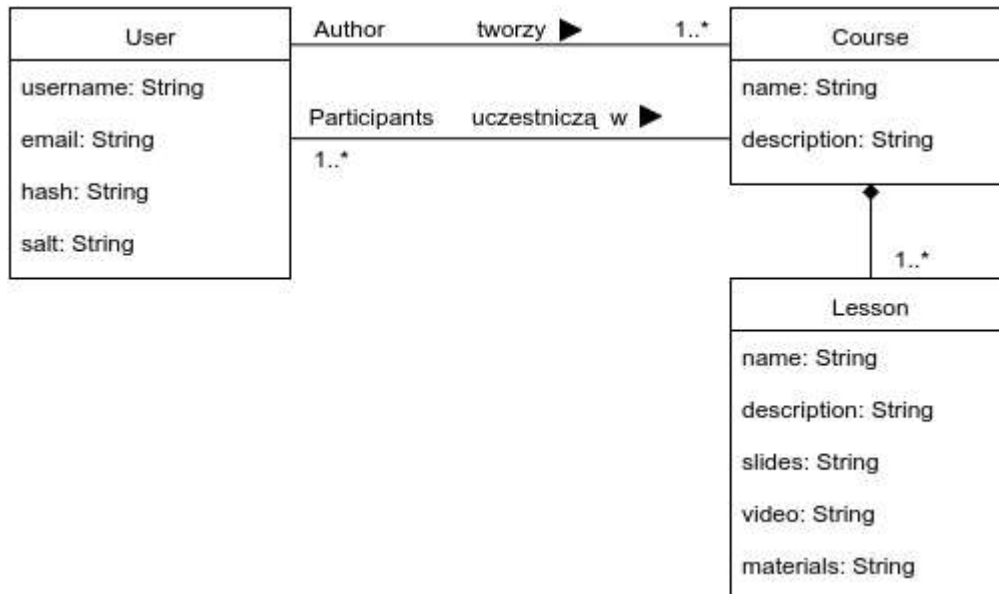
- “name” - reprezentacja nazwy kursu, przechowywana w postaci danych tekstowych, będąca wartością konieczną i unikalną w ramach zbioru utworzonych w systemie kursów,
- “author” - referencja do obiektu użytkownika, który utworzył dany kurs,
- “description” - reprezentacja opisu kursu, przechowywana w postaci danych tekstowych,
- “participants” - tablica referencji do obiektów użytkowników, którzy dołączyli do danego kursu
- “lessons” - tablica obiektów realizujących logiczną reprezentację lekcji danego kursu, które zostały opisane poniżej.

W ramach obiektu kursu jedno z pól odpowiada tablicy obiektów lekcji, dla których definiujemy następujące pola:

- “name” - reprezentacja nazwy lekcji stworzonej w ramach danego kursu, przechowywana w postaci danych tekstowych,
- “description” - reprezentacja opisu lekcji stworzonej w ramach danego kursu, przechowywana w postaci danych tekstowych,
- “slides” - reprezentacja lokalizacji pliku prezentacji wgranego w ramach danej lekcji,
- “video” - reprezentacja lokalizacji pliku wideo wgranego w ramach danej lekcji,

- “materials” - reprezentacja lokalizacji plików wgranych jako dodatkowe materiały danej lekcji.

Relacje pomiędzy modelami w bazie danych odpowiadają relacjom w całej aplikacji, które są zobrazowane przez diagram przedstawiony poniżej.



Rysunek 5.12: Diagram klas

5.5 Bezpieczeństwo

W projekcie aplikacji zakłada się również dużą rolę bezpieczeństwa. Do poprawnie działającego programu potrzebna jest implementacja mechanizmów uwierzytelniania i autoryzacji, co pozwoli na efektywne wdrożenie procesów rejestracji oraz logowania.

Konieczna będzie też odpowiednia ochrona zasobów, a więc udostępnianie plików związanych z konkretnym kursem tylko użytkownikom mającym do niego dostęp - autorowi oraz uczestnikom.

Wszystkie powyższe kwestie można zrealizować opierając się na modelu autoryzacji tokenem - unikalnym, zakodowanym i rozszyfrowywalnym jedynie przez aplikację ciągiem znaków przekazywanym przez klienta przy każdym żądaniu w ramach nagłówka HTTP. Z powodu użycia architektury typu REST, którą cechuje między innymi bezstanowość, nie jesteśmy w stanie przechowywać żadnych informacji na temat połączenia w pamięci aplikacji - dlatego też ciągłe przekazywanie tokenu w ramach żądania HTTP jest jak najbardziej uzasadnionym rozwiązaniem.

W ramach opisanego tokenu, przechowywane będą musiały być dane użytkownika poddane wcześniejszej enkrypcji.

Za każdym razem, kiedy aplikacja na serwerze będzie chciała uwierzytelnić i autoryzować klienta, konieczne będzie pobranie tokena z nagłówku odebranego żądania i odkodowanie go. Z tak przetworzonej informacji potrzeba będzie następnie wyekstraktować dane identyfikujące użytkownika (email oraz pseudonim), na podstawie których nastąpi jego identyfikacja w bazie danych. Mając już dostęp do konkretnego obiektu, aplikujemy na otrzymanym w tokenie hasle algorytm haszujący z wykorzystaniem soli przypisanej danemu użytkownikowi i porównujemy z oczekiwaną wartością funkcji skrótu zapisaną przy

rejestracji. Na podstawie tego porównania aplikacja jest w stanie stwierdzić czy użytkownik faktycznie jest tym za kogo się podaje.

5.6 Przetwarzanie mediów

Jak wspomniane zostało już w punkcie 4.5., kwestie przetwarzania multimediów w przypadku tego projektu dotyczą wyłącznie wykonania zapisu materiału audiowizualnego za pomocą odpowiednich urządzeń peryferyjnych będących częścią lub podłączonych do komputera użytkownika.

W tym celu wykorzystany zostanie projekt WebRTC, będący częścią standardu HTML5, który za pomocą dostępnych w nowych przeglądarkach rozwiązań uzyska dostęp do wspomnianych urządzeń peryferyjnych oraz możliwość zapisu strumieni danych, w tym przypadku typu audio i wideo, z nich płynących [13]. Konkretnie, wykorzystana będzie tu metoda **`getUserMedia()`**, która zwraca udekorowane i spakowane strumienie danych z urządzeń, które nas interesują pod postacią interfejsu **MediaStream**.

5.7 Podsumowanie

Zdefiniowane powyżej założenia projektowe stanowią odpowiednik dokumentacji, zgodnie z którą można przejść do fazy implementacji całego systemu. Dzięki szczegółowym i konkretnym wytycznym wypisanym w tym rozdziale proces realizacji produktu powinien przebiec sprawnie i bez niejasności.

6. Implementacja

6.1 Wprowadzenie

Implementowana aplikacja składa z dwóch głównych modułów - strony klienta, obsługiwanej przez przeglądarkę internetową, która wyświetla interfejs i odpowiada za prezentację danych użytkownikowi oraz strony serwera, na którym dane są przechowywane oraz odbywa się całość, tak zwanej, logiki biznesowej aplikacji.

W poniższym rozdziale omówione zostaną niektóre aspekty implementacji projektu. Uwzględnione będą aspekty związane z realizacją wymagań bezpieczeństwa oraz przetwarzaniem mediów, które są z technicznego punktu widzenia najciekawszymi elementami programu.

6.2 Wykorzystane środowiska i narzędzia programistyczne

Całość implementacji odbyła się przy użyciu systemu operacyjnego Ubuntu 14.04 64-bit. Środowiskiem programistycznym, w którym napisana została aplikacja, jest WebStorm autorstwa firmy JetBrains.

Podczas procesu implementacji wykorzystany został serwer Apache Tomcat 8. Warstwa danych obsługiwana była przez system zarządzania bazą danych MongoDB.

Funkcjonalność strony klienta była testowana przy pomocy następujących przeglądarek:

- Mozilla Firefox 49.0,
- Google Chrome 54.0,
- Opera 39.0.

Podczas implementacji całość kodu była zapisywana i zarządzana przy pomocy systemu kontroli wersji Git, co niejednokrotnie zapobiegło utracie danych.

6.3 Wykorzystane biblioteki i frameworki

Podczas procesu implementacji wykorzystane zostały, zgodnie z wymogami licencyjnymi, gotowe rozwiązania wymienione w Tabeli 1 oraz Tabeli 2.

Nazwa	Opis	Autor	Licencja
Strona serwera			
Body-parser [15]	Narzędzie parsujące ciała żądań HTTP	Doug Wilson	MIT

Cookie-parser [16]	Narzędzie parsujące ciasteczka HTTP	Doug Wilson	MIT
Express.js [6]	Framework wspomagający tworzenie serwera HTTP przy pomocy Node.js	Node.js Foundation	Creative Commons Attribution-ShareAlike 3.0 United States License
Express-jwt [20]	Narzędzie obsługujące tokeny JWT na serwerach opartych o Express.js	Auth0	MIT
Mongoose [9]	Framework zapewniający prostą konfigurację bazy danych MongoDB w ramach serwera opartego na Node.js	Automattic	MIT
Multer [17]	Narzędzie wspomagające proces wgrywania plików na serwer oparty na Node.js	Linus Unnebäck	MIT
Node-uuid [18]	Narzędzie generujące identyfikatory UUID	Alvin Oneal	MIT
Passport.js [14]	Narzędzie ułatwiające proces autoryzacji i uwierzytelniania w ramach serwera opartego na Node.js	Jared Hanson	MIT
Pdf2img [19]	Narzędzie konwertujące pliki pdf do plików PNG/JPEG	Filtra Aditya	MIT
Socket.IO [7]	Framework wspierający zarządzanie technologią WebSocket w ramach serwera opartego na Node.js	Guillermo Rauch	MIT
Socketio-jwt [21]	Narzędzie łączące ze sobą technologie Socket.IO i JWT	Auth0	MIT
Winstonjs [22]	Framework pozwalający na sprawne tworzenie logów serwera opartego na Node.js	Charlie Robbins	MIT

Tabela 1: Wykorzystane biblioteki i frameworki po stronie klienta

Nazwa	Opis	Autor	Licencja
Strona klienta			
Angular.js [5]	Framework wspomagający tworzenie aplikacji webowych oparty na jednej stronie	Google	MIT
Angular File Saver [23]	Narzędzie wspomagające zapisywanie plików po stronie klienta przy pomocy Angular.js	Philipp Alferov	MIT
Bootstrap 3.0 [8]	Framework ułatwiający implementację	Twitter	MIT

	responsywnych interfejsów webowych		
jQuery [11]	Biblioteka pozwalająca na efektywną i efektowną manipulację dokumentem html	jQuery Foundation	jQuery
Ng-file-upload [24]	Narzędzie wspomagające wgrywanie plików na serwer przy pomocy Angular.js	Danial Farid	MIT
Record RTC [26]	Narzędzie pozwalające na zapis strumieni danych otrzymanych z urządzeń peryferyjnych przy pomocy WebRTC	Muaz Khan	MIT
Videojs-record [25]	Narzędzie wspomagające obsługę urządzeń peryferyjnych przez przeglądarkę	Collab	MIT

Tabela 2: Wykorzystane biblioteki i frameworki po stronie klienta.

6.4 Interfejs

Poniżej przedstawione zostały zrzuty ekranu wykonane podczas używania aplikacji.

Rysunek 6.1: Formularz logowania

Rysunek 6.2: Formularz rejestracji z prawidłowymi danymi

Please register

Username can not be empty

This email is invalid

Password has to be at least 6 characters long

Register

Already registered? [Sign in!](#)

Rysunek 6.3: Formularz rejestracji z nieprawidłowymi danymi

Hello, Jakub Amanowicz!

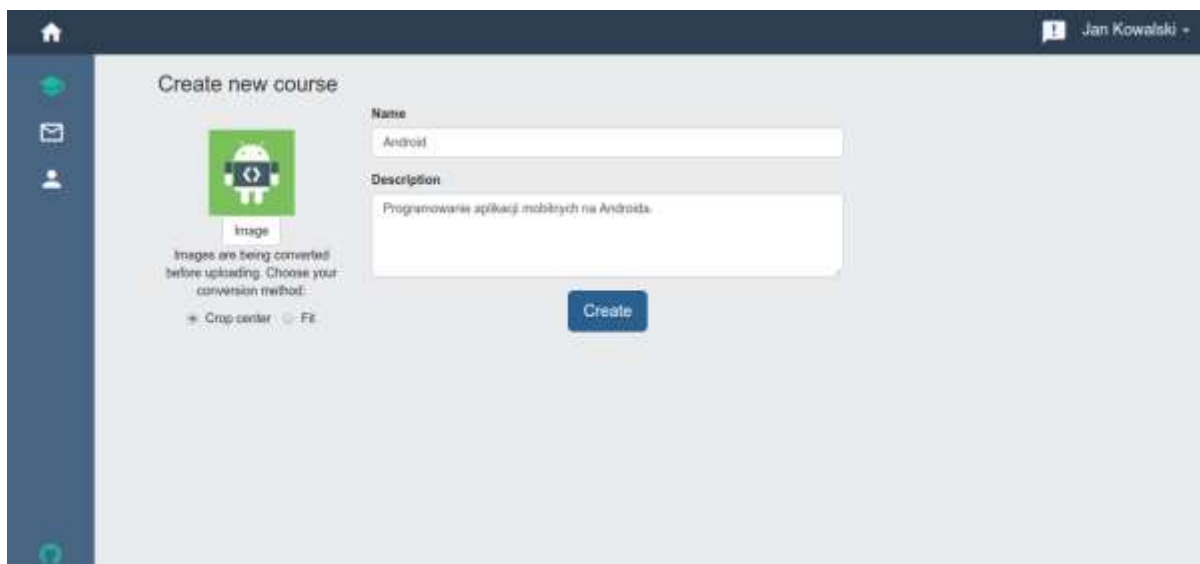
Change your settings

Would you like to update your profile?

Change only the fields you would like to update. The rest will remain untouched.

Update

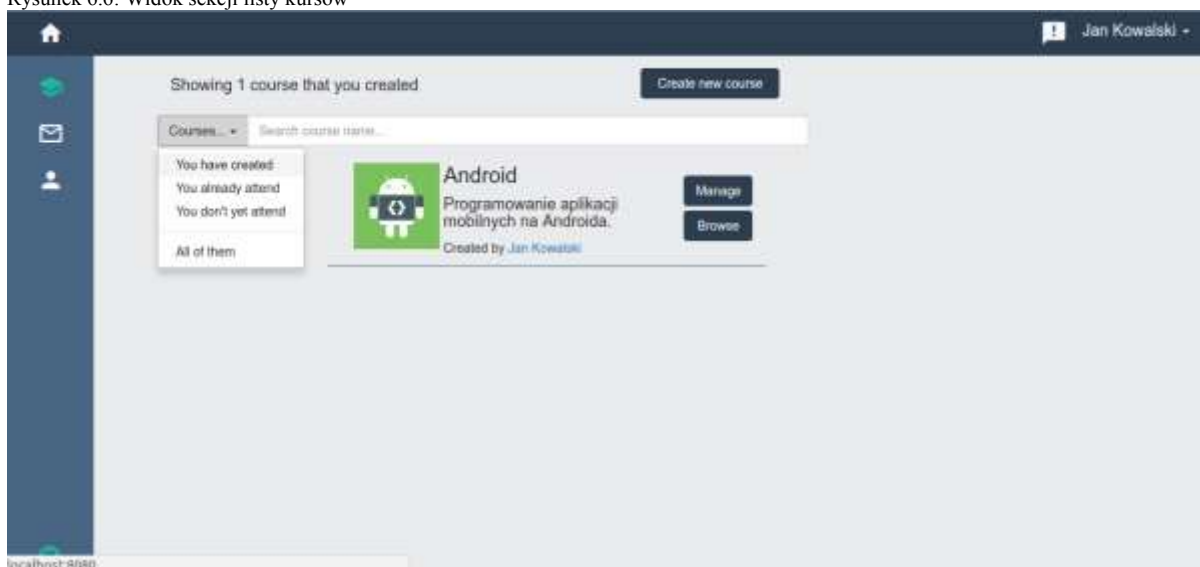
Rysunek 6.4: Widok sekcji profilu użytkownika z formularzem edycji danych



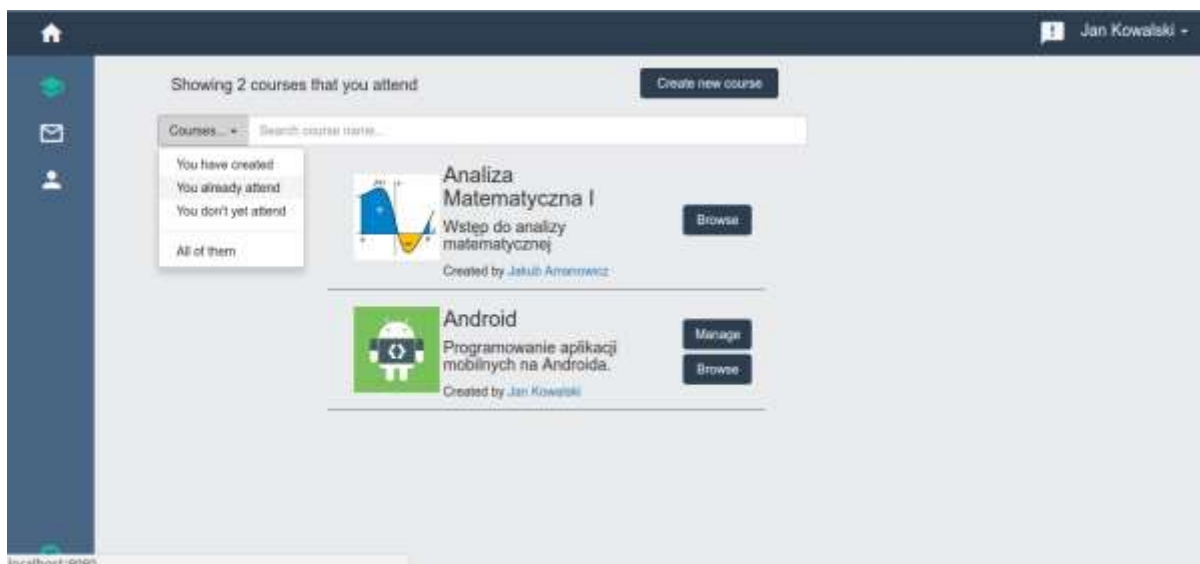
Rysunek 6.5: Widok sekcji tworzenia kursu z formularzem tworzenia kursu



Rysunek 6.6: Widok sekcji listy kursów



Rysunek 6.7: Widok sekcji listy kursów po zastosowaniu filtra wyszukiwania - wyświetlane tylko kursy autorskie



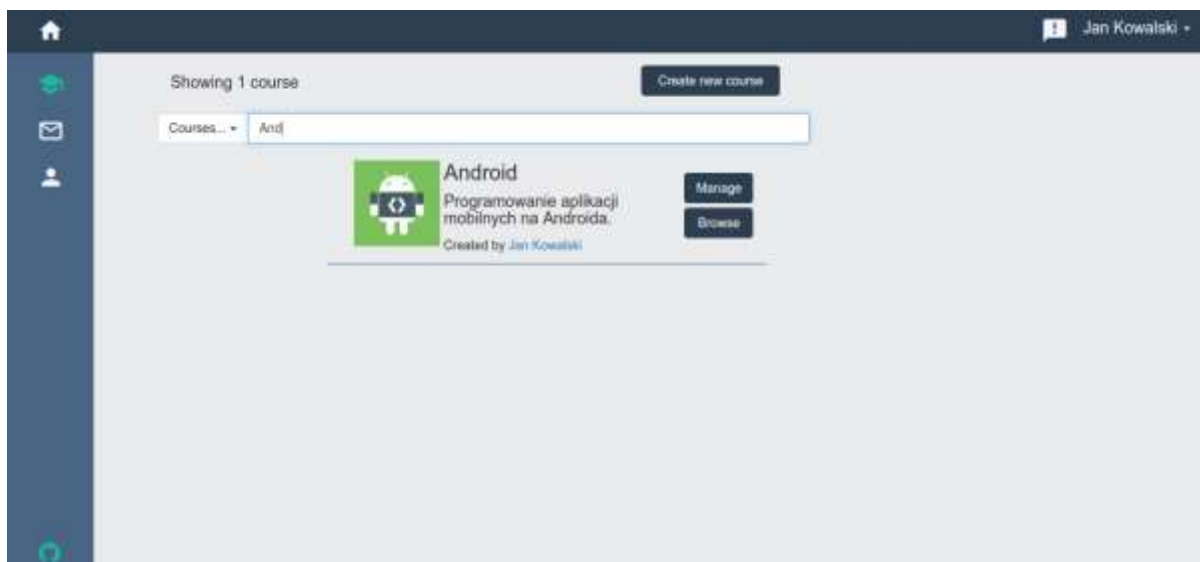
Rysunek 6.8: Widok sekcji listy kursów po zastosowaniu filtra wyszukiwania - wyświetlane tylko kursy, do których ma się dostęp



Rysunek 6.9: Widok sekcji listy kursów po zastosowaniu filtra wyszukiwania - wyświetlane tylko kursy, do których nie ma się dostępu



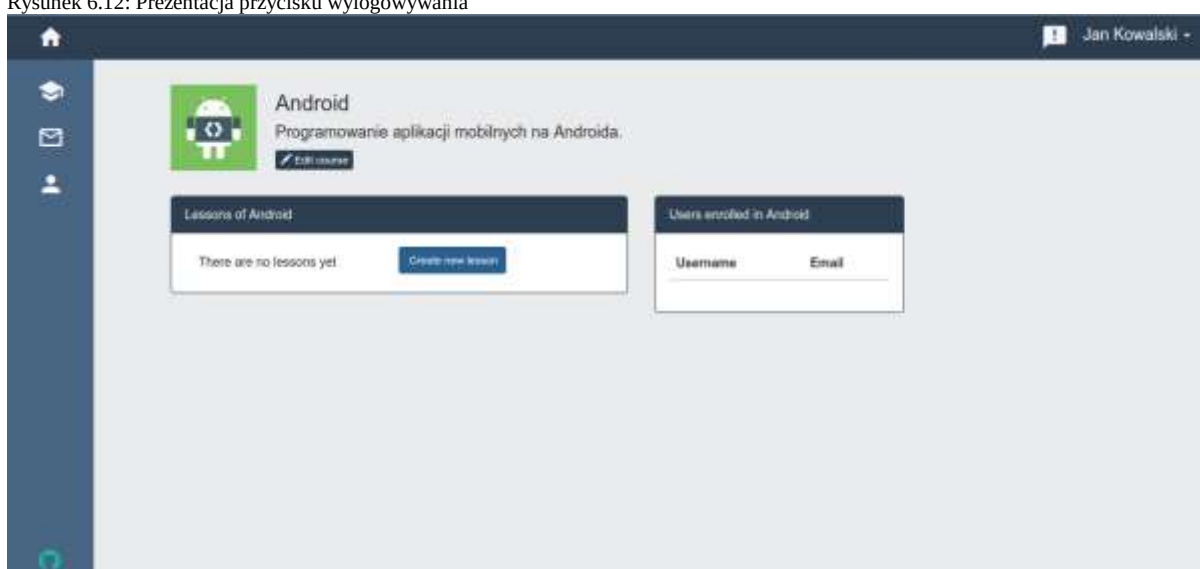
Rysunek 6.10: Widok sekcji listy kursów po zastosowaniu filtra wyszukiwania - wyświetlane wszystkie kursy



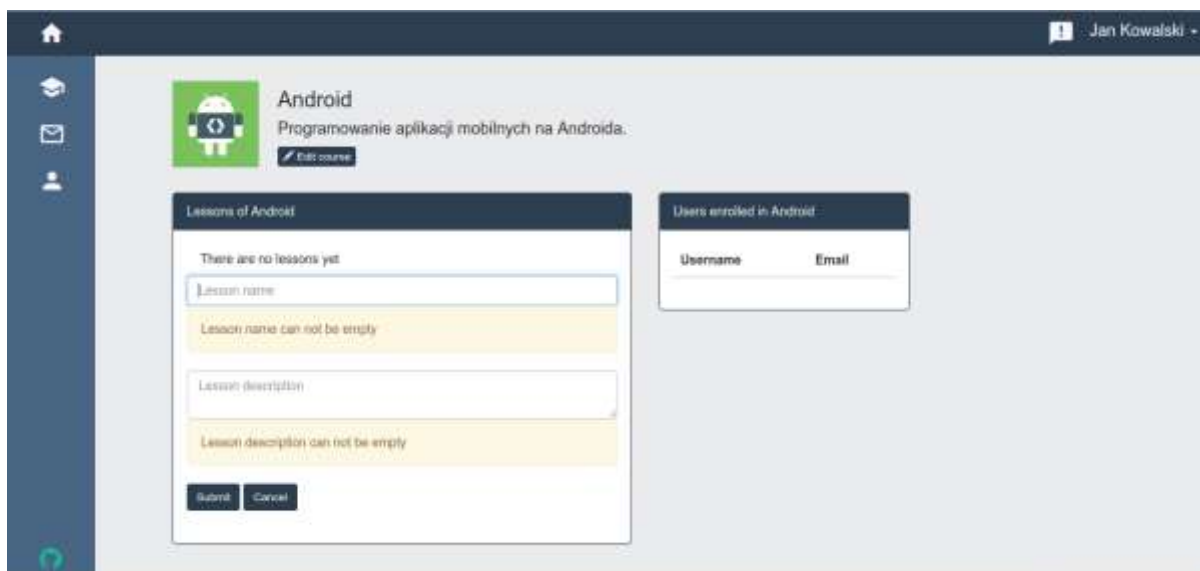
Rysunek 6.11: Widok sekcji listy kursów po zastosowaniu wyszukiwania po nazwie



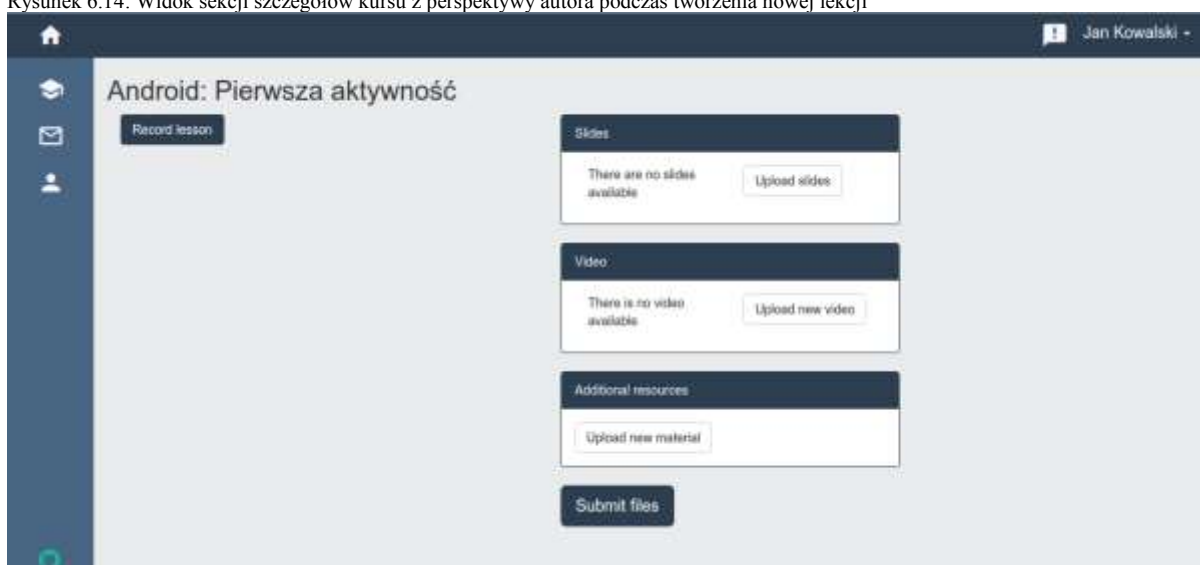
Rysunek 6.12: Prezentacja przycisku wylogowywania



Rysunek 6.13: Widok sekcji szczegółów kursu z perspektywy autora



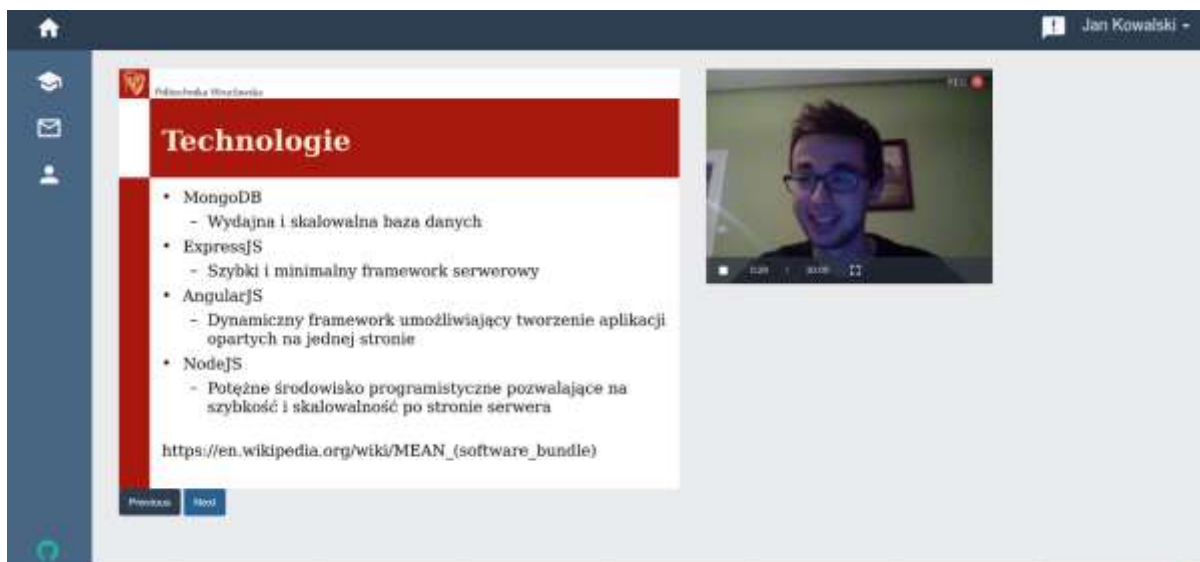
Rysunek 6.14: Widok sekcji szczegółów kursu z perspektywy autora podczas tworzenia nowej lekcji



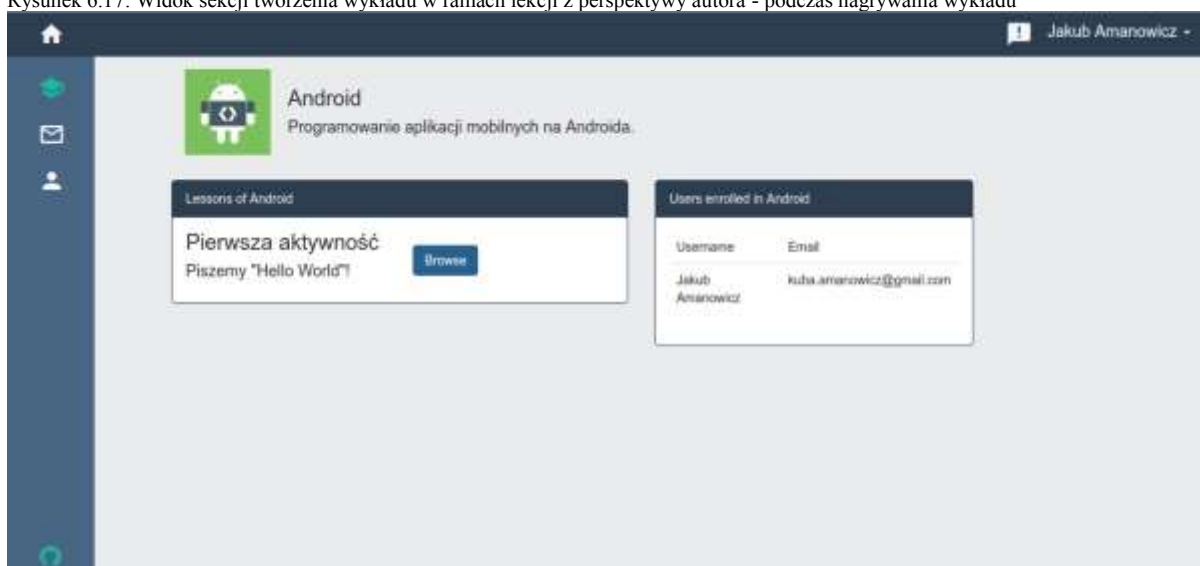
Rysunek 6.15: Widok sekcji szczegółów lekcji z perspektywy autora



Rysunek 6.16: Widok sekcji tworzenia wykładu w ramach lekcji z perspektywy autora - przed włączeniem urządzeń peryferyjnych



Rysunek 6.17: Widok sekcji tworzenia wykładu w ramach lekcji z perspektywy autora - podczas nagrywania wykładu



Rysunek 6.18: Widok sekcji szczegółów kursu z perspektywy uczestnika



Rysunek 6.19: Widok sekcji odtwarzania wykładu w ramach lekcji

6.5 Bezpieczeństwo

Jednym z większych problemów podczas implementowania aplikacji było opracowanie rozwiązania, które spełniałoby wymogi bezpieczeństwa opisane w podpunkcie 5.5. Ostatecznie kwestie autoryzacji i uwierzytelniania zostały rozwiązane poprzez zastosowanie JSON Web Token (JWT) [12].

JWT jest standardem bezpieczeństwa komunikacji między podmiotami opartym na tokenie zapisanym w formacie JSON. Dane przekazywane za pomocą tego rozwiązania pozwalają na bezpieczny proces autoryzacji i uwierzytelniania dzięki zastosowaniu algorytmu haszującego HMAC. Technologia JWT została opisana przez stowarzyszenie IETF, w ramach którego przyznany został jej standard RFC 7519.

Wspomniane wcześniej rozwiązanie zostało wykorzystane przy procesach rejestracji, logowania oraz uwierzytelniania podczas prób uzyskania przez użytkowników dostępu do materiałów przypisanych do kursów.

6.6 Przetwarzanie mediów

Istotnym elementem aplikacji, jest opisane w punkcie 5.6 podejście do mediów w ramach systemu.

Multimediami, które program powinien obsługiwać są pliki audiowizualne nagrane przez użytkownika oraz pliki prezentacji - zazwyczaj posiadające rozszerzenia .pdf, .ppt i .pptx.

Najbardziej problematycznym zagadnieniem była synchronizacja slajdów z nagrywanym przez twórcę lekcji wykładem. Wymagało to odpowiedniej obróbki plików prezentacji oraz zastosowania obiektu konfiguracyjnego, dzięki któremu można było przypisać odpowiedni slajd do przedziału czasowego nagranych wideo.

Również wyświetlanie oraz programistyczna manipulacja aktualnie pokazywanym slajdem stanowiła problem, który nie był rozwiązywalny przy użyciu narzędzi oferowanych przez HTML5 - głównie wspomniany wcześniej WebRTC [13].

Z powodów opisanych powyżej konieczne było zastosowanie obróbki plików prezentacji polegającej na ekstrakcji slajdów do plików graficznych typu PNG. Pozwoliło to na reprezentację prezentacji przy pomocy oferowanego przez framework Bootstrap rozwiązania Carousel odpowiadającego na przystępne wyświetlanie zbioru obrazów.

Podczas procesu nagrywania wykładu, który jest opisany przez przypadek użycia o numerze 17, gdy użytkownik zmienia wyświetlany slajd, system rozpoznaje aktualny czas nagrywanego pliku wideo i zapisuje go do obiektu konfiguracyjnego odpowiedzialnego za synchronizację. Obiekt ten, będący faktycznie mapowaniem czasu tranzycji do docelowego slajdu, jest po zakończeniu nagrywania wysyłany razem z plikiem audiowizualnym do serwera aplikacji. Później, gdy użytkownik ogląda utworzony już w ramach lekcji wykład, aplikacja po stronie klienta zmienia slajdy poprzez monitorowanie czasu odtwarzania i porównywanie go z obiektem synchronizującym.

6.7 Instalacja oprogramowania

Aby zainstalować oprogramowanie należy wykonać listę kroków wymienionych poniżej.

1. Instalacja systemu zarządzania bazą danych - MongoDB.
2. Konfiguracja bazy danych w ramach aplikacji - wpisanie URI używanej bazy do pliku konfiguracyjnego.
3. Instalacja w systemie środowiska uruchomieniowego Node.js.
4. Będąc w folderze aplikacji, wywołanie z wiersza poleceń następujących komend:
 - 4.1. `npm install` - instalacja aplikacji,
 - 4.2. `node server.js` - uruchomienie serwera aplikacji.

Cały powyższy proces powinien odbyć się na komputerze mającym dostępem do internetu.

6.8 Instrukcja użytkownika

Aby móc korzystać z oprogramowania należy mieć dostęp do internetu oraz mieć pewność, że aplikacja została zainstalowana na dostępnym w sieci serwerze.

Na początku konieczne jest połączenie się ze stroną aplikacji poprzez wpisanie odpowiedniego adresu URL na zainstalowanej przeglądarce internetowej.

Po przekierowaniu na stronę aplikacji wyświetlony zostanie formularz logowanie i rejestracji - w zależności od tego czy użytkownik istnieje już w systemie należy wypełnić jeden z nich, aby móc korzystać dalej z serwisu. W wypadku nieprawidłowego wypełnienia formularza, na przykład w rejestracji istniejących już w systemie danych, wyświetlony zostanie odpowiedni komunikat błędu, który pozwoli zidentyfikować przyczynę problemu. W wypadku pozytywnie zakończonej rejestracji użytkownik zostaje automatycznie zalogowany.

Po pomyślnie zakończonym procesie logowania, użytkownikowi zostaje wyświetlony ekran startowy aplikacji, który jest jednocześnie widokiem dostępnych w ramach systemu kursów. Każdy kurs posiada odpowiednie przyciski akcji, zależne od relacji użytkownika i danego kursu. Możliwe akcje to dołączenie, edycja i przeglądanie.

W ramach widoku przedstawiającego wspomnianą wyżej listę kursów, użytkownik jest w stanie filtrować i wyszukiwać wyniki za pomocą narzędzia wyszukiującego umieszczonego w górnej sekcji strony.

Po lewej stronie wyświetlanego interfejsu umieszczone są przyciski nawigujące między sekcjami aplikacji. Po kliknięciu na daną sekcję użytkownik jest w stanie wyświetlić listę kursów oraz ustawienia profilu.

W widoku ustawień profilowych wyświetlany jest formularz, dzięki któremu użytkownik jest w stanie zmienić swoje dane takie jak pseudonim, email lub hasło. W przypadku błędnie wprowadzonych danych, przykładowo - naruszających unikalność w ramach aplikacji, wyświetlony zostanie odpowiedni komunikat błędu, który pozwoli zidentyfikować przyczynę problemu. W razie pozytywnego zakończenia procesu użytkownik również zostanie o tym poinformowany.

Jak już zostało wspomniane wcześniej, z widoku listy kursów można przejść do przeglądania danego kursu lub, jeśli jest się jego autorem, również do sekcji edycji.

W przypadku przeglądania kursu, wyświetlane są jego podstawowe informacje jak nazwa i opis oraz lista zdefiniowanych w jego ramach lekcji. Każda lekcja, podobnie jak kurs, posiada akcje, które można na niej wykonać w zależności od tego czy jest się tylko uczestnikiem kursu czy autorem. Te akcje to przeglądanie i edycja lekcji.

Podczas edycji kursu, wyświetlony zostaje widok prezentujący listę zdefiniowanych lekcji oraz formularze edycji danych kursu oraz tworzenia nowej jednostki lekcyjnej. Wypełnienie odpowiedniego z nich skutkuje utworzeniem nowej lekcji lub modyfikacją

danych kursu. W wypadku błędnego wypełnienia formularza, wyświetlony zostanie odpowiedni komunikat błędu, który pozwoli zidentyfikować przyczynę problemu.

Przy przeglądaniu jednostki lekcyjnej wyświetlany jest widok prezentujący listę plików, materiałów, wchodzących w skład lekcji. Po kliknięciu na dany materiał następuje proces ściągnięcia go z serwera na komputer użytkownika. Kolejnym przyciskiem dostępnym w ramach tego widoku jest prezentacja przejścia do wykładu.

Będąc w sekcji wykładu zdefiniowanego w ramach lekcji wyświetlone zostają dwa ekrany - pierwszy prezentujący aktualny slajd oraz drugi będący wyświetlaczem wideo. Przy wyświetlanych slajdach znajdują się również przyciski odpowiadające za transycję między slajdami. Użytkownik sam wybiera czy chce samemu zmieniać aktualnie wyświetlaną zawartość czy slajdy mają zmieniać się automatycznie z biegiem wyświetlanego materiału wideo. Wyświetlacz wideo pozwala użytkownikowi na manipulację czasem oraz głośnością materiału.

Przy edycji jednostki lekcyjnej, która dostępna jest tylko dla jej autora, możliwe jest wgranie nowych materiałów za pomocą odpowiednich formularzy dostępnych na stronie. Widok oferuje też przycisk przenoszący autora do sekcji nagrywania wykładu.

Podczas nagrywania wykładu użytkownik musi najpierw wgrać na serwer odpowiedni plik prezentacji za pomocą wyświetlanego przez aplikację formularza. Po wybraniu pliku, serwer musi go przetworzyć i przekonwertować do serii plików PNG, co może zająć trochę czasu zależnie od możliwości serwera. Podczas tego procesu użytkownik informowany jest o postępach za pomocą animacji. Po pozytywnym zakończeniu tego procesu użytkownikowi wyświetlane są dwa ekrany - pierwszy prezentujący slajdy wgrane przez użytkownika oraz drugi pokazujący obraz z kamery użytkownika. Drugi z wspomnianych ekranów jest wyposażona również w przycisk nagrywania, po kliknięciu którego rozpoczyna się zapis obrazu z kamery, dźwięku z mikrofonu oraz czasów transycji slajdów. Po kliknięciu przycisku zakończenia nagrywania, który wyświetlany jest pod wyświetlanym obrazem z kamery, następują wysłanie plików na serwer.

Po prawej stronie na samej górze interfejsu wyświetlany jest pseudonim użytkownika, na który można kliknąć, aby ukazać przycisk wylogowania, po kliknięciu na który następuje proces wylogowania użytkownika z systemu.

6.9 Archiwizacja danych

Wszystkie dane nie będące obrazami, plikami wideo ani plikami prezentacji zapisywane są w zainstalowanej przez użytkownika bazie danych MondoDB.

Wszystkie dane, które nie są zapisywane we wspomnianej bazie, zapisują się w ustrukturyzowanym podfolderze aplikacji.

6.10 Podsumowanie

Dzięki dobrze sporządzonemu projektowi oraz zastosowaniu starannie wyselekcjonowanych bibliotek i rozwiązań proces implementacji przeszedł relatywnie sprawnie i bez dodatkowych, niespodziewanych komplikacji. Aplikacja działa sprawnie i spełnia kryteria oraz założenia przedstawione we wcześniejszych rozdziałach.

7. Podsumowanie

7.1 Ewaluacja i wnioski

Realizacja projektu przebiegła bardzo sprawnie dzięki odpowiednio poczynionym przygotowaniom - zwłaszcza w postaci jasno zdefiniowanej dokumentacji. Pozwoliła ona również na sprawdzenie umiejętności informatycznych nabytych przez okres studiów inżynierskich. Duży nacisk położono, między innymi, na aspekty związane z projektowaniem oprogramowania, planowania interfejsu oraz implementacją systemów webowych.

Całość procesu tworzenia wspomnianego systemu dobrze oddaje niektóre mechanizmy, z którymi można się spotkać na płaszczyźnie zawodowej. Mowa tu na przykład o takich standardach jak iteracyjny rozwój produktu, podążanie za wcześniej przygotowanym harmonogramem prac przy jednoczesnej zwinności implementacyjnej czy używanie systemu kontroli wersji.

Ponadto, podczas pracy nad projektem, widoczne były minusy pracy indywidualnej nad rozwiązaniem tej skali. Zwiększenie zespołu programistycznego o osoby realizujące rolę testera i projektanta interfejsu, nad którym w tym przypadku spędzono zdecydowanie za dużo czasu, przyczyniłoby się znacznie do poprawy jakości i szybkości implementacji rozwiązania. Tego typu podejście widoczne jest zresztą w sferze zawodowej, gdzie poważne firmy informatyczne opierają swoje zespoły na grupach liczących czasem nawet dziesięć osób. Jest to naturalnie zależne od złożoności projektu oraz realiów czasowych. Optymalizacja ilości pracujących w zespole osób jest zresztą tematem wielu rozważań naukowych, które bardzo często bazują na, tak zwanej, liczbie Dunbara.

Znaczna część zaimplementowanego serwisu opiera się na rozwiązaniach już gotowych, oferowanych zgodnie z myślą wolnego oprogramowania, które dostępne są w ramach takich repozytoriów jak GitHub czy npm. Dzięki temu zaoszczędzone zostało wiele czasu przy jednoczesnym zachowaniu jakości projektu.

7.2 Możliwe ścieżki rozwoju

Naturalnie, istnieje wiele płaszczyzn i funkcjonalności, których rozwój zapewniłby większą jakość końcowego produktu. Z perspektywy czasu można tu przykładowo wspomnieć o większej roli testów w całym procesie realizacji projektu. Gdyby były one realizowane wcześniej oraz w większej ilości, a więc stosując technikę tworzenia oprogramowania opartą na testach, proces projektowania i implementacji przeszedłby sprawniej z mniejszą szansą błędu.

7.2.1 Strumieniowanie i komunikacja

Serwis o takim potencjale rynkowym, którego odbiorcą jest środowisko akademickie, pełne młodych ludzi śledzących na bieżąco aktualne trendy i rozwiązania technologiczne, jest niemal zobligowany do tego, aby stale się rozwijać i wdrażać innowacyjne rozwiązania.

Jednym z nich mogłaby być funkcjonalność oferująca prowadzącym realizowanie wykładu w czasie rzeczywistym za pomocą strumieniowania na żywo, tak jak ma to miejsce w popularnych serwisach tego typu jak YouTube czy Twitch. Takie rozwiązanie byłoby bazą do rozbudowy dużej ilości funkcjonalności opierających się na interakcji między prowadzącym i studentami w czasie rzeczywistym. Twórca kursu miałby przykładowo możliwość prowadzenia swego rodzaju listy obecności poprzez monitorowanie listy połączonych użytkowników i ich aktywności oraz tworzenia testów online. Studenci z kolei mieliby możliwość zadawania pytań do aktualnie omawianych treści wykładowych, co również przełożyłoby się na zwiększenie efektywności dydaktycznej.

Opisywaną powyżej funkcjonalność z punktu widzenia programistycznego można zrealizować przy odrobinę bardziej zaawansowanym wykorzystaniu technologii już uwzględnionych w projekcie. Jednym z możliwych podejść do implementacji jest użycie projektu WebRTC do stworzenia sieci połączeń typu P2P między klientami biorącymi udział w danej jednostce lekcyjnej, a więc prowadzącym i studentami. Dzięki użyciu tej technologii oraz architektury możliwa byłaby obsługa zarówno strumieniowania zsynchronizowanego obrazu i dźwięku jak i wymiany wiadomości w czasie rzeczywistym. Konieczne w tym rozwiązaniu byłoby również zaimplementowanie modułu sygnalizującego, bez którego WebRTC nie jest w stanie nawiązać połączenia P2P między dwoma klientami. Upraszczając aspekty techniczne, ów moduł byłby odpowiedzialny za proces koordynowania komunikacji, który opiera się na

- kontrolowaniu aktualnej sesji poprzez wymianę odpowiednich wiadomości sygnalizujących przykładowo rozpoczęcie oraz zakończenie połączenia,
- wymianie wiadomości na temat błędów,
- wymianie wiadomości zawierających metadane, opisujące przykładowo zestaw potrzebnych kodeków,
- wymianie wiadomości zawierających tokeny bezpieczeństwa,
- wymianie wiadomości zawierających szczegóły konfiguracji sieciowej, a więc adresy IP klientów i ich porty.

Opisaną wyżej usługę można zaimplementować przy użyciu technologii WebSocket, co wpłynie pozytywnie na szybkość sygnalizacji. W ramach zaimplementowanego w aplikacji serwera można użyć wykorzystywanego już frameworka - Socket.IO.

7.2.2 Interfejs

Aspekty wizualne aplikacji również mogłyby zostać poddane pewnym poprawkom. Jedną z nich, która drastycznie wpłynęłaby na szybkość, i tak płynnego już, interfejsu jest zmiana narzędzia odpowiedzialnego za dynamiczną obsługę obiektowego modelu dokumentu HTML. Aktualnie używany framework, Angular 1, zostaje aktualnie wypierany przez swojego następcę - Angulara 2, który podobnie jak inne konkurencyjne rozwiązania, jest znacznie bardziej optymalny.

Kolejnym obszarem, który mógłby ewentualnie zostać zmieniony jest sam styl wizualny aplikacji, aktualnie bazujący na gotowych arkuszach CSS oraz manipulatorach obiektowym modelem dokumentu, napisanych przy pomocy jQuery, oferowanych przez framework Bootstrap 3. Alternatywą byłoby napisanie od podstaw własnego zestawu tego typu plików lub użycie konkurencyjnych rozwiązań tego typu jak Material Design czy Bootstrap 4.

Bibliografia

1. Strona serwisu Coursera, <https://www.coursera.org/>, dostęp: 13.11.2016.
2. Strona serwisu Udacity, <https://www.udacity.com/>, dostęp: 13.11.2016.
3. Strona serwisu edX, <https://www.edx.org/>, dostęp: 13.11.2016.
4. Dokumentacja Node.js, <https://nodejs.org/api/>, dostęp: 13.11.2016.
5. Dokumentacja Angular.js, <https://docs.angularjs.org/api>, dostęp: 13.11.2016.
6. Dokumentacja Express.js, <https://expressjs.com/en/4x/api.html>, dostęp: 13.11.2016.
7. Dokumentacja Socket.IO, <http://socket.io/docs/>, dostęp: 13.11.2016.
8. Spis komponentów frameworku Bootstrap, <http://getbootstrap.com/components/>, dostęp: 13.11.2016.
9. Dokumentacja Mongoose, <http://mongoosejs.com/docs/guide.html>, dostęp: 13.11.2016.
10. Dokumentacja MongoDB, <https://docs.mongodb.com/manual/>, dostęp: 13.11.2016.
11. Dokumentacja jQuery, <http://api.jquery.com/>, dostęp: 13.11.2016.
12. Strona rozwiązania JWT, <https://jwt.io/>, dostęp: 16.11.2016.
13. Strona projektu WebRTC, <https://webrtc.org/>, dostęp: 16.11.2016.
14. Strona projektu Passport.js, <http://passportjs.org/>, dostęp: 16.11.2016.
15. Strona modułu “body-parser” w serwisie npm, <https://www.npmjs.com/package/body-parser>, dostęp: 13.11.2016.
16. Strona modułu “cookie-parser” w serwisie npm, <https://www.npmjs.com/package/cookie-parser>, dostęp: 13.11.2016.
17. Strona modułu “multer” w serwisie npm, <https://www.npmjs.com/package/multer>,
18. Strona modułu “node-uuid” w serwisie npm, <https://www.npmjs.com/package/node-uuid>, dostęp: 13.11.2016.
19. Strona modułu “pdf2img” w serwisie npm, <https://www.npmjs.com/package/pdf2img>, dostęp: 13.11.2016.
20. Repozytorium projektu “express-jwt” w serwisie GitHub, <https://github.com/auth0/express-jwt>, dostęp: 13.11.2016.
21. Repozytorium projektu “socketio-jwt” w serwisie GitHub, <https://github.com/auth0/socketio-jwt>, dostęp: 13.11.2016.
22. Repozytorium projektu “winstonjs” w serwisie GitHub, <https://github.com/winstonjs/winston>, dostęp: 13.11.2016.
23. Repozytorium projektu “angular-file-save” w serwisie GitHub, <https://github.com/alferov/angular-file-saver>, dostęp: 13.11.2016.
24. Repozytorium projektu “ng-file-upload” w serwisie GitHub, <https://github.com/danialfarid/ng-file-upload>, dostęp: 13.11.2016.
25. Repozytorium projektu “videojs-record” w serwisie GitHub, <https://github.com/collab-project/videojs-record/>, dostęp: 16.11.2016.
26. Strona projektu RecordRTC, <https://www.webrtc-experiment.com/RecordRTC/>, dostęp: 16.11.2016.
27. Mike Cantelon, Marc Harter, T.J. Holowaychuk, Nathan Rajlich, *Node.js in Action*, Wyd. 2, Shelter Island, Manning Publications Co., 2013, ISBN 978-1-617290-57-2.

28. Manfred Steyer, Vildan Softic, *AngularJS*, O'Reilly, 2015, ISBN 978-3-955619-50-3.
29. Jason Krol, *Web Development with MongoDB and Node.js*, O'Reilly, 2014, ISBN 978-1-78398-731-3.
30. Christopher Allen, *The Dunbar Number as the Limit To Group Sizes*, 2004, http://www.lifewithalacrity.com/2004/03/the_dunbar_numb.html, dostęp: 16.11.2016.
31. Jeremy Knox, *Digital culture clash: "massive" education in the E-learning and Digital Cultures MOOC*, Distance Education Journal, 2014.
32. Strona serwisu YouTube, <https://www.youtube.com/>, dostęp: 22.11.2016.
33. Strona serwisu Twitch, <https://www.twitch.tv/>, dostęp: 22.11.2016.
34. Christopher Pappas, *eLearning statistics and facts for 2015*, eLearning Industry, 2015, <https://elearningindustry.com/elearning-statistics-and-facts-for-2015>, dostęp: 26.11.2016.
35. Strona narzędzia Balsamiq, <https://balsamiq.com/>, dostęp: 26.11.2016.

Spis ilustracji

Rysunek 3.1: Zrzut ekranu przedstawiający serwis webowy Coursera	11
Rysunek 3.2: Zrzut ekranu przedstawiający serwis webowy Udacity	12
Rysunek 3.3: Zrzut ekranu przedstawiający serwis webowy edX	13
Rysunek 5.1: Diagram przypadków użycia	22
Rysunek 5.2: Makieta widoku rejestracji użytkownika	29
Rysunek 5.3: Makieta widoku logowania użytkownika	29
Rysunek 5.4: Makieta widoku edycji danych użytkownika	30
Rysunek 5.5: Makieta widoku listy kursów	30
Rysunek 5.6: Makieta widoku tworzenia kursu	31
Rysunek 5.7: Makieta widoku edycji kursu	31
Rysunek 5.8: Makieta widoku przeglądania kursu	32
Rysunek 5.9: Makieta widoku edycji lekcji	32
Rysunek 5.10: Makieta widoku przeglądania lekcji	33
Rysunek 5.11: Makieta widoku nagrania lekcji	33
Rysunek 5.12: Diagram klas	35
Rysunek 6.1: Formularz logowania	39
Rysunek 6.2: Formularz rejestracji z prawidłowymi danymi	39
Rysunek 6.3: Formularz rejestracji z nieprawidłowymi danymi	40
Rysunek 6.4: Widok sekcji profilu użytkownika z formularzem edycji danych	40
Rysunek 6.5: Widok sekcji tworzenia kursu z formularzem tworzenia kursu	41
Rysunek 6.6: Widok sekcji listy kursów	41
Rysunek 6.7: Widok sekcji listy kursów po zastosowaniu filtra wyszukiwania - wyświetlane tylko kursy autorskie	41
Rysunek 6.8: Widok sekcji listy kursów po zastosowaniu filtra wyszukiwania - wyświetlane tylko kursy, do których ma się dostęp	42

Rysunek 6.9: Widok sekcji listy kursów po zastosowaniu filtra wyszukiwania - wyświetlane tylko kursy, do których nie ma się dostępu	42
Rysunek 6.10: Widok sekcji listy kursów po zastosowaniu filtra wyszukiwania - wyświetlane wszystkie kursy	42
Rysunek 6.11: Widok sekcji listy kursów po zastosowaniu wyszukiwania po nazwie	43
Rysunek 6.12: Prezentacja przycisku wylogowywania	43
Rysunek 6.13: Widok sekcji szczegółów kursu z perspektywy autora	43
Rysunek 6.14: Widok sekcji szczegółów kursu z perspektywy autora podczas tworzenia nowej lekcji	44
Rysunek 6.15: Widok sekcji szczegółów lekcji z perspektywy autora	44
Rysunek 6.16: Widok sekcji tworzenia wykładu w ramach lekcji z perspektywy autora - przed włączeniem urządzeń peryferyjnych	44
Rysunek 6.17: Widok sekcji tworzenia wykładu w ramach lekcji z perspektywy autora - podczas nagrywania wykładu	45
Rysunek 6.18: Widok sekcji szczegółów kursu z perspektywy uczestnika	45
Rysunek 6.19: Widok sekcji odtwarzania wykładu w ramach lekcji	45

Spis tabel

Tabela 1: Wykorzystane biblioteki i frameworki po stronie serwera.	37-38
Tabela 2: Wykorzystane biblioteki i frameworki po stronie klienta.	38-39