|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | |  | Wydział Informatyki i Zarządzania kierunek studiów: Informatyka Praca dyplomowa - inżynierskaSystem przeprowadzania testów z wielowyboremMaciej Struś słowa kluczowe:  1 linia  2 linia  3 linia  krótkie streszczenie:  1 linia  2 linia  3 linia  4 linia  5 linia   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | opiekun pracy  dyplomowej | .................................................. | | ....................... | | ....................... | | | *Tytuł/stopień naukowy/imię i nazwisko* | | *ocena* | | *podpis* | | | Ostateczna ocena za pracę dyplomową | | | | | | | | | Przewodniczący Komisji egzaminu dyplomowego | | | ..................................................  *Tytuł/stopień naukowy/imię i nazwisko* | | ....................... | | ....................... | | *ocena* | | *podpis* |   *Do celów archiwalnych pracę dyplomową zakwalifikowano do:\**   1. *kategorii A (akta wieczyste)* 2. *kategorii BE 50 (po 50 latach podlegające ekspertyzie)*   *\* niepotrzebne skreślić*   |  | | --- | | pieczątka wydziałowa | |
|  |  |  | Wrocław 2018 |

**Streszczenie**

**Spis treści**

1. Wstęp3

* 1. Wprowadzenie do tematyki, motywacja3
  2. Cel i zakres pracy4

2. Przegląd literatury w zakresie tematyki pracy 5

2.1. Przegląd istniejących rozwiązań5

2.1.1. Moodle5

2.1.2. Google Forms6

2.2. Przegląd technologii7

3. Specyfikacja wymagań9

3.1. Opis problemu, wizja systemu

3.1.1. Użytkownicy9

3.1.2. Historyjki użytkowników9

3.2. Wymagania funkcjonalne i niefunkcjonalne3

3.3. Przypadki użycia2

3.3.1. Diagram przypadków użycia2

3.3.2. Opis przypadków użycia2

3.4. Prototypy interfejsu użytkownika3

3.5. Model domenowy, reguły biznesowe i ograniczenia dziedzinowe3

4. Projekt aplikacji 4

4.1. Architektura aplikacji6

4.1.1. Diagram pakietów2

4.1.2. Diagram klas2

4.2. Baza danych6

4.2.1. Model fizyczny2

5. Implementacja1

5.1. Rozwiązania programistyczne2

5.2. Interfejs użytkownika3

5.3. Narzędzia programistyczne2

6. Testowanie produktu1

7. Podsumowanie1

**1. Wstęp**

**1.1 Wprowadzenie do tematyki, motywacja**

Obecnie coraz częściej osoby prowadzące kursy na uczelniach wyższych oraz nauczyciele w szkołach rezygnują z tradycyjnego przeprowadzania sprawdzianów na papierze i decydują się na testy w formie elektronicznej. Dotyczy to nie tylko kursów prowadzonych w postaci e-learningów. Może to być spowodowane postępem technologicznym, rozwojem technologii webowych, który doprowadził do ułatwienia dostępu do interaktywnych, nieskomplikowanych w obsłudze aplikacji webowych.

Dostępność takich systemów i możliwość przeprowadzania testów w formie elektronicznej jest dogodnym ułatwieniem dla dydaktyków. „Sprawdzanie stanu wiedzy pochłania często połowę czasu przeznaczonego na proces kształcenia. Skrócenie czasu potrzebnego na sprawdzenie wiedzy, pozwalałoby zwiększyć efektywność procesu kształcenia.” [1] Umożliwia również prostsze konstruowanie testów składających się z pytań wielokrotnego wyboru.

Warto odpowiedzieć sobie na pytanie czym jest pytanie testowe wielokrotnego wyboru. Mianowicie jest to zestaw składający się z treści pytania, niepustego zbioru opcji będących odpowiedziami na pytanie, z czego co najmniej jedna odpowiedź jest prawidłowa. Osoba rozwiązująca test ma za zadanie przy każdym pytaniu wybrać te odpowiedzi, które jego zdaniem są prawidłowe.

Aby ocenić test składający się z takich pytań musimy ocenić każde pytanie z osobna. „O ile w klasycznych testach „kartkowych” ich twórcy dążą do maksymalnej prostoty systemu oceniania albowiem ułatwia to później sprawdzanie i wystawianie ocen, o tyle w testach komputerowych stopień komplikacji systemu oceniania nie ma znaczenia z punktu widzenia czasochłonności procesu wystawiania ocen.” [11] Ma znaczenie natomiast z punktu widzenia skomplikowania oraz możliwości realizacji żądanego sposobu oceniania. Z tego powodu (Przyłuski, 2014) [11] proponuje trzy systemy oceniania:

1. Dwuwartościowa ocena (DO)
2. Odpowiedzi częściowo prawidłowe bez żadnej nieprawidłowej odpowiedzi (CO)
3. Odpowiedzi częściowo poprawne i częściowo niepoprawne (TO) (totalna ocena)

Dwuwartościowa ocena polega na pozytywnym wyniku jedynie kiedy rozwiązujący test wybierze jedynie wszystkie prawidłowe odpowiedzi z danego pytania.

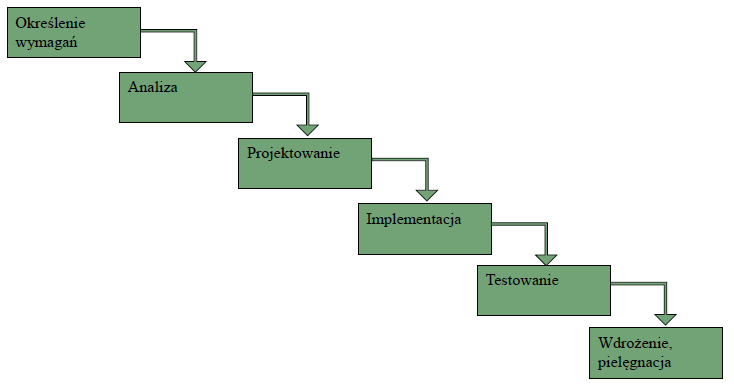
System oceniania (CO) pozwala oceniać pozytywnie również rozwiązania prawidłowe, natomiast niepełne. Za każdą prawidłową odpowiedź rozwiązujący test dostaje ocenę cząstkową za pytanie zależną od wagi danej opcji.

Totalna ocena daje możliwość indywidualnej oceny każdego rozwiązania pytania. Odpowiedzi prawidłowe są liczone pozytywnie do oceny za pytanie, natomiast odpowiedzi nieprawidłowe negatywnie. W tym systemie również istotne są wagi nadawane odpowiedziom.

Motywacją do podjęcia tematu pracy inżynierskiej umożliwienie przeprowadzanie testów z pytaniami wielokrotnego wyboru z możliwością wyboru systemu oceniania pytań z systemów DO, CO oraz TO. Oprócz tego umożliwienie wyboru czy przy korzystaniu z systemu TO punkty negatywne będą liczone do puli punktów z całego testu, czy będą traktowane jako rozwiązane źle i punkty za odpowiedź będą zerowane.

**1.2 Cel i zakres pracy**

Celem pracy jest dostarczenie oprogramowania spełniającego określone w procesie specyfikacji wymagań wymagania. Oprogramowanie ma umożliwiać tworzenie oraz przeprowadzanie testów z pytaniami wielokrotnego wyboru oraz wybieranie systemu oceniania.  
 Zakres pracy obejmuje przegląd literatury zawierający przegląd rozwiązań dostępnych obecnie na rynku oraz przegląd literatury dotyczącej wykorzystanych w pracy technologii.  
Następnie w pracy znajdzie się opis procesu wytwarzania oprogramowania zgodny z klasycznym modelem kaskadowym z wykluczeniem etapu „Wdrożenie, pielęgnacja”.

  
Rys 1.1. Model kaskadowy cyklu życia oprogramowania   
Staszak, U. *Projektowanie oprogramowania. Wykład 1. Politechnika Wrocławska, 2017*   
Tak jak zostało to przedstawione na Rys 1.1. „Jest to model, który został zaproponowany poprzez analogię do sposobu prowadzenia przedsięwzięć w innych dziedzinach inżynierii, na przykład budownictwie. Budowa mostu rozpoczyna się od określenia głównych zadań jakie ma on spełniać, a następnie szczegółowego określenia wymagań, które zapewnią realizację tych zadań. Kolejnym krokiem jest wykonanie szczegółowego projektu konstrukcji. Dalej następują faza budowy oraz testowania wybudowanego mostu. Ostatnim etapem jest faza konserwacji.” [4] W ramach określania wymagań zostanie przygotowany opis problemu, na podstawie którego opracowana zostanie wizja systemu. Następnie określone zostaną wymagania na system. W etapie analizy wymagań opracowane zostaną: diagram przypadków użycia wraz z opisami przypadków użycia, prototypy interfejsu. Opracowany również zostanie diagram domenowy wraz z określonymi regułami biznesowymi i ograniczeniami dziedzinowymi.  
Podczas etapu projektowania aplikacji opracowana zostanie architektura aplikacji, w skład której wchodzić będzie diagram pakietów oraz diagram klas, oraz baza danych z jej modelem fizycznym. Etap implementacji składać będzie się z opisu rozwiązań programistycznych, opisu zaimplementowanego interfejsu użytkownika oraz wykorzystanych narzędzi programistycznych.**2. Przegląd literatury w zakresie tematyki pracy**

**2.1 Przegląd istniejących rozwiązań**

Obecnie są już dostępne na rynku systemu umożliwiające przeprowadzanie testów wielokrotnego wyboru. Są one szeroko wykorzystywane zarówno w celach edukacyjnych jak również w życiu codziennym. Nic dziwnego, ponieważ problem, który pozwalają rozwiązać nie jest zagadnieniem nowym w świecie technologii.   
 Przedstawione poniżej rozwiązania dotyczą tego samego problemu, jednak różnią się w wielu aspektach. Są kierowane do innej grupy użytkowników, proponują inne rozwiązania i funkcjonalności, a przede wszystkim tworzone są z myślą o innych zastosowaniach testów wielokrotnego wyboru.

2.1.1 Moodle

Moodle [8] jest darmową platformą e-learningową rozwijaną jako otwarte oprogramowanie. Wymaga on do działania uruchomienia własnej strony internetowej, na której działało będzie środowisko Moodle. Skutkuje to tym, że do utrzymywania tego serwisu jest potrzeba posiadania własnej infrastruktury, serwerów oraz administratorów. W związku z tym jeżeli nie posiadamy skonfigurowanej strony Moodle stworzenie prostego testu wielokrotnego wyboru może okazać się ciężkim i czasochłonnym wyzwaniem.   
 Moodle jednak oferuje znacznie większą liczbę funkcjonalności niż potrzebna jest do przeprowadzania testów wiedzy. Pozwala on na tworzenie kursów, dzielenie się materiałami, tworzenie forów aktualności, przesyłanie plików, przechowywanie pytań, dzielenie się pytaniami oraz wiele innych [9]. Również podczas tworzenia sprawdzianów Moodle oferuje znaczną ilość możliwych typów pytań.   
 Jednak najważniejszą funkcjonalnością Moodle w kontekście tematu pracy są pytania wielokrotnego wyboru:   
„The teacher can select "multiple answers are allowed" in a Multiple Choice question type. "Multiple answers" questions types in a quiz allow one or more answers to be chosen [...]. Each answer may carry a positive or negative grade, so that choosing ALL the options will not necessarily result in good grade. If the total grade is negative then the total grade for this question will be zero. [..] The All or nothing multiple choice question type add-on is adapted from the existing multichoice question.[…] If the chosen answers correspond exactly to the correct choices defined in the question, the respondent gets 100%. If he/she chooses any incorrect choices or does not select all of the correct choices, the grade is 0%. “  
*“Nauczyciel może wybrać “dozwolone jest wiele odpowiedzi” w rodzaju pytania Wielokrotnego Wyboru. Rodzaj pytania „wiele odpowiedzi” w quizie pozwalają na wybranie jednej lub więcej odpowiedzi [...]. Każda odpowiedź może mieć pozytywną lub negatywną ocenę aby wybranie wszystkich opcji nie koniecznie prowadziło do uzyskania dobrej oceny. Jeżeli sumaryczna ocena jest ujemna, wtedy ostateczna ocena za to pytanie będzie wynosić zero.[…] Dodatek umożliwiający pytania typu Wszystko albo nic jest przystosowany z istniejącego typu pytania wielokrotnego wyboru.[…] Jeżeli wybrane opcje odpowiadają dokładnie odpowiedziom zdefiniowanym jako poprawne, odpowiadający uzyskuje 100%. Jeżeli wybierze jakąkolwiek nieprawidłową opcję lub nie wybierze wszystkich poprawnych uzyskuje 0%”* [10]  
Oznacza to, że Moodle wykorzystuje system TO bez punktów negatywnych w ramach całego testu oraz po zainstalowaniu dodatku również system DO. Warto przypomnieć, że instalowanie dodatku musi zostać dokonane przez administratora, więc uzyskanie dodatkowego systemu oceniania nie jest łatwym procesem.

2.1.2 Formularze Google

Formularze Google [5] jest to system, którego główną funkcjonalnością jest tworzenie ankiet internetowych. Główną grupą użytkowników, do których skierowany jest system są ludzie, którzy potrzebują szybko i w prosty sposób utworzyć formularz do zbierania informacji. Jednak producent pomiędzy planowaniem wypadu pod namiot, rejestracją osób na szkolenie czy sondą wymienia również przeprowadzanie testu. Formularze Google umożliwiają wybór różnych typów pytań od jednokrotnego wyboru przez wielokrotny aż po menu czy krótkie pytania otwarte. Formularze umożliwiają również graficzna reprezentację zebranych wyników.  
 Podobnie jak w przypadku Moodle w kontekście tematu pracy interesujące są pytania wielokrotnego wyboru oraz systemy oceniania. Formularze Google oferują funkcjonalność automatycznego oceniania odpowiedzi, natomiast jedynie w systemie CO. Chcąc uzyskać inny system oceniania, odpowiedzi należy oceniać ręcznie. Co prawda pozwala ocenianie grup odpowiedzi natomiast nie dzieje się to bez ingerencji osoby sprawdzającej, co może być przeszkodą w stosowaniu bardziej skomplikowanych systemów oceniania. Jedyną możliwością konfiguracji automatycznego oceniania jest przypisywanie punktów poszczególnym zadaniom. Co więcej Google w swoim produkcie nie pozwala na przenoszenie pytań do innych testów ani na jakiekolwiek ich zapisywanie.

****Rys 2.1. Widok tworzenia testu w aplikacji Formularze Google   
Zrzut ekranu ze strony: https://docs.google.com/forms/d/e/1FAIpQLSdFIFB5c2ZphgFoJnsf69URQ0lmZuccaYaTHiBCOOODnbThcg/viewscore?viewscore=AE0zAgAlA8Du2XTy8qyrQ-0qoIAwWhd\_AIWjGhrQcfoJTSYCgAE7VQ6uk739JQ  
 Znacznym atutem aplikacji jest jej wygląd zaprezentowany na Rys 2.1. Jest on zgodny z zasadami projektowania zaproponowanymi przez Google [6]. Jest to zamysł projektowania interfejsów użytkownika, z którym każdy użytkownik produktów Google jest zaznajomiony. Zamysł ten jest szeroko stosowany nie tylko w produktach Google. Uważany jest za intuicyjny i przyjazny użytkownikowi.

**2.2 Przegląd technologii**

2.2.1. Vue.js

Vue.js [15] jest platformą programistyczną języka JavaScript umożliwiającą budowanie interfejsów użytkownika. Platforma przez firmę Vue Technology, której założycielem jest Evan You, były pracownik firmy Google. Vue.js rozwijane zgodnie z regułami otwartego oprogramowania, dzięki czemu posiada liczną i aktywną społeczność.   
 Jest on platformą, którą wykorzystać można zarówno w małych, statycznych stronach internetowych jak i w dużych, dynamicznych aplikacjach webowych. „[...] Wystarczy dodanie tagu <script> do strony i możesz tworzyć reaktywną aplikację JavaScript. [...] (Vue) **rośnie razem z Twoimi potrzebami**. Możesz pisać swój kod w najzwyklejszym JavaScripcie, ale również używać tzw. **single file components [...]** . Vue udostępnia też dedykowane rozwiązanie do **obsługi routingu**, dzięki któremu stworzysz pełnoprawną **SPA**. [...] Dzięki takiemu podejściu krzywa uczenia się jest mało stroma […].**” [14] Umożliwia ono również naukę tej technologii wraz z rozwijaniem własnej aplikacji webowej.  
 Wspomniane „single file components”[15], czyli jedno plikowe komponenty są mechanizmem platformy pozwalającej na grupowanie kodu źródłowego aplikacji zgodnie z przyjętą konwencją. Komponent posiada unikalną w kontekście aplikacji nazwę. Składa się z szablonu zawierającego strukturę zdefiniowaną w języku znaczników HTML, części skryptowej, w której umieszczone są skrypty w języku JavaScript pozwalające na dynamiczne zarządzanie komponentem oraz części stylu, w której można zdefiniować styl komponentu za pomocą języka CSS. Taki podział odpowiedzialności pozwala na ustrukturyzowanie kodu na stronie, co poprawia czytelność, przyśpiesza proces implementacji oraz ułatwia utrzymanie i pielęgnację.  
 Wspomniane przez autora określenie „SPA” jest akronimem angielskiego wyrażenia „single-page application” oznaczającego aplikację jednostronną. Jest to technologia, która rozwinęła się wraz z rozwojem nowoczesnych przeglądarek internetowych. W klasycznym podejściu do kreowania stron internetowych kolejne podstrony czy widoki były plikami wysyłanymi z serwera, podczas gdy klient zajmował się jedynie odpowiednim ich wyświetleniem. W podejściu aplikacji jednostronnej aplikacja webowa posiada jeden niezmienny szablon, do którego zaczytywane są nowe dane zależne od kontekstu. Wszystkie te działania wykonywane są właśnie po stronie klienta. Również nawigowanie po stronie, trasowanie, nazywane „routingiem”, realizowane jest przez klienta.**

2.2.2. Spring Boot

Spring Boot [3] jest rozwiązaniem typu “konwencja ponad konfiguracją” bazujące na platformie programistycznej Spring Framework. Oznacza to, że jest platformą, która upraszcza, jednocześnie nie ograniczając możliwości, konfigurację i uruchomienie projektu. Spring Boot jest rozwijany na zasadach otwartego oprogramowania przez firmę Pivotal Software, która odpowiedzialna jest również za sam Spring Framework. Spring Boot ułatwia również zarządzanie zależnościami w projekcie, dzięki czemu dodanie do projektu nowej biblioteki programistycznej jest proste i bezproblemowe.   
 Spring Framework natomiast jest platformą programistyczną języka Java również rozwijaną przez firmę Pivotal jako otwarte oprogramowanie. Powstał jako alternatywa i rozszerzenie J2EE.

Umożliwia on korzystanie z zaawansowanych programistycznych mechanizmów jak wstrzykiwanie zależności, które pozwala ograniczyć liczbę zależności w projekcie poprzez wstrzyknięcie instancji obiektu zamiast tworzenia obiektu. Dzięki wykorzystaniu tej platformy możliwe jest uporządkowanie kodu źródłowego oraz wykorzystanie zaawansowanych mechanizmów programistycznych w nieskomplikowany sposób.   
 Spring Framework wspomaga również zastosowanie stylu architektonicznego skierowanego na zasoby REST stosowanego w tworzeniu aplikacji webowych. REST jest akronimem od angielskiego wyrażenia „representational state transfer”. REST charakteryzuje się tym, że jest w łatwy sposób skalowalny oraz wysyłane żądania są czytelne. „Główne założenia REST to:

* wszystko jest zasobem (reprezentowanym przez jednoznaczny URI)
* jednolity interfejs (dużo rzeczowników reprezentujących zasoby i mało czasowników opisujących dozwolone operacje)
* bezstanowa komunikacja (serwer nie tworzy sesji = łatwa skalowalność)
* reprezentacja zasobów – zasoby są abstrakcje, ale można się do nich dostać za pomocą zdefiniowanej reprezentacji (ten sam zasób może być udostępniany w wielu reprezentacjach, np. w formie XML lub JSON)”[2]

Dozwolone operacje ograniczają się do interfejsu HTTP:

* GET – pobiera reprezentację zasobu, jest bezpieczną operacją (nie zmienia żadnego obiektu) i może być cache’owane
* HEAD – podobne do GET lecz zamiast zwracania reprezentacji zasobu zwraca tylko nagłówki (może służyć do sprawdzenia czy wersja zasobu w cache jest ciągle aktualna)
* POST – tworzy nowy zasób i w odpowiedzi w nagłówku Location zwraca URI nowego zasobu
* PUT – aktualizuje zasób, operacja idempotenta
* DELETE – kasuje zasób, operacja idempotenta”[2]

2.2.3 H2

H2 [7] jest relacyjnym systemem zarządzania bazą danych napisanym w języku Java, rozwijanym jako otwarte oprogramowanie od 2004 roku. Oryginalnym twórcą H2 jest Thomas Mueller. Dzięki temu, że ten SZBD jest wytwarzany w języku Java zintegrowanie go z aplikacją napisaną w tym samym języku jest ułatwione. H2 nie jest duże, zajmuje zaledwie 1,5 MB. Dużą zaletą H2 jest możliwość szybkiego skonfigurowania czy będzie działać w trybie wbudowanym w aplikację, działając na tym samym JVM, czy w trybie serwera. Jest to funkcjonalność przydatna na etapie implementacji, ponieważ uruchomienie bazy danych w trybie wbudowanym znacznie przyśpiesza działanie.

**3. Specyfikacja wymagań**

Specyfikacja wymagań [12] jest procesem zapisywania wymagań użytkownika oraz wymagań na system. Wymagania powinny być czytelne, jednoznaczne, łatwe do zrozumienia, kompletne oraz spójne. Powinny być zrozumiałe dla osoby, która nie posiada technicznego wykształcenia, więc nie powinny zawierać szczegółów projektu aplikacji, a opisywać jedynie zachowanie systemu. Dokument zawierający wymagań powinien korzystać z języka naturalnego jak i nieskomplikowanych diagramów.   
 Jednak przed przejściem do procesu specyfikacji wymagań potrzebne jest wcześniejsze zebranie wymagań.

**3.1. Opis problemu, wizja systemu**

Produkt skierowany jest do osób prowadzących zajęcia na uczelniach, w szkołach lub prowadzących szkolenia, dotyczy on również osób rozwiązujących testy, studentów. W celu ułatwienia dostępności, niezależnie od systemu operacyjnego, realizowany jest jako aplikacja webowa.

Problemem, który tworzony system ma na celu rozwiązanie, są trudności z tworzeniem testów wielokrotnego wyboru oraz brak możliwości konfiguracji sposobu oceniania zadań. Dotychczasowe rozwiązania udostępniają takie funkcjonalności, natomiast żadne z dostępnych nie jest całkowicie zorientowane na tworzeniu oraz przeprowadzaniu testów wielokrotnego wyboru. Są to rozwiązania, w których testy wielokrotnego wyboru są jedną z wielu proponowanych funkcjonalności, co nie jest zaletą dla osoby, której jedynym celem jest jedynie przeprowadzenie takiego testu. Dodatkowo rozwiązania te nie oferują wielu możliwości sposobów oceniania pytań wielokrotnego wyboru.

3.1.1. Użytkownicy

Użytkownik, czyli najważniejsza część systemu, jest osobą, do której skierowany jest nasz system. Z potrzeb, wymagań użytkownika wynikają wymagania na system. Z tego powodu ważnym punktem jest zdefiniowanie użytkowników systemu.

W tabeli 3.1. zdefiniowani zostali użytkownicy systemu.

Tabela 3.1. Użytkownicy systemu

| **Nazwa** | **Opis** | **Odpowiedzialności** |
| --- | --- | --- |
| Gość | Osoba nie posiadająca konta w systemie. | Zakładanie konta. |
| Niezalogowany użytkownik | Osoba posiadająca konto w systemie, nie będąca obecnie zalogowana. | Logowanie się do aplikacji. |
| Nauczyciel | Osoba, która chce przeprowadzić test wielokrotnego wyboru, posiadająca konto w systemie. | Tworzenie pytań testowych, grupowanie ich w przedmioty, działy. Tworzenie testów. Przeglądanie wyników testów. Edytowanie własnego profilu |
| Student | Osoba studiująca na uczelni, posiadająca konto w systemie. | Przeglądanie własnych testów, rozwiązywanie testów, edytowanie własnego profilu |

3.1.2. Historyjki użytkowników

Historyjka użytkownika [13] jest krótką historią opisującą potrzebę użytkownika, której celem jest zebranie wymagania od użytkownika. Nie zagłębia się ona w jaki sposób system ma realizować żądaną przez użytkownika funkcjonalność. Skupia się ona jedynie na potrzebie. Zakres historyjki powinien być ograniczony do jednej potrzeby, a jeżeli dana potrzeba jest zbyt rozległa, historyjka powinna zostać rozbita. Zaletą historyjek jest to, że są krótkie, zrozumiałe dla każdego niezależnie od wykształcenia czy ekspertyzy.

W tabeli 3.2. wypisane zostały historyjki użytkowników systemu.

Tabela 3.2. Historyjki użytkowników

|  |
| --- |
| **Historyjka użytkownika** |
| Jako Niezalogowany użytkownik potrzebuje możliwości zalogowania się na swoje konto aby móc korzystać z systemu. |
| Jako Gość potrzebuję możliwości zakładania konta. |
| Jako Student potrzebuję możliwości rozwiązywania testu. |
| Jako Student potrzebuję informacji o swoim wyniku z testu. |
| Jako Nauczyciel potrzebuję możliwości tworzenia zadań i przypisywania odpowiedziom punktów aby mieć wykorzystać możliwości systemów oceniania. |
| Jako Nauczyciel potrzebuję możliwości grupowania zadań aby ułatwić ich przeszukiwanie. |
| Jako Nauczyciel potrzebuję możliwości przeprowadzania testów aby sprawdzać wiedzę swoich kursantów. |
| Jako Nauczyciel potrzebuję możliwości konfigurowania sposobu oceniania testów, wyboru z systemu DO, CO, TO z punktami ujemnymi oraz TO bez punktów ujemnych aby stosować różne techniki dydaktyczne. |
| Jako Nauczyciel potrzebuje możliwości konfigurowania sposobu nawigowania po rozwiązywanym teście. |
| Jako Nauczyciel potrzebuje dostępu do wyników testów aby móc wystawić oceny kursantom. |

**3.2. Wymagania funkcjonalne i niefunkcjonalne**

Wymagania funkcjonalne są stwierdzeniami o żądanych możliwościach systemu. Wynikają one bezpośrednio z wymagań użytkowników, natomiast nie dotyczą tego co może zrobić użytkownik, ale jakie funkcjonalności udostępni system. [13]   
 Wymagania funkcjonalne zostały przedstawione w tabeli 3.3.

Tabela 3.3. Wymagania funkcjonalne

|  |
| --- |
| **Wymagania funkcjonalne** |
| Umożliwia rejestrację do systemu. |
| Umożliwia logowanie się do systemu. |
| Umożliwia odczytanie oraz modyfikację danych własnego profilu. |
| Umożliwia dodawanie, odczytanie, modyfikacje oraz usuwanie przedmiotów. |
| Umożliwia dodawanie, odczytanie, modyfikacje oraz usuwanie działów przedmiotów. |
| Umożliwia dodawanie, odczytanie, modyfikacje oraz usuwanie pytań testowych. |
| Umożliwia dodawanie, odczytanie, modyfikacje oraz usuwanie testów. |
| Umożliwia konfigurowanie sposobu oceniania sposób oceniania testów oraz sposobu nawigowania po rozwiązywanym teście. |
| Umożliwia dostęp do wyników testu. |
| Pozwala rozwiązywać testy, do których został przyznany dostęp. |

Wymagania niefunkcjonalne są ograniczeniami nakładanymi na funkcjonowanie systemu. Często dotyczą raczej całości systemu niż poszczególnych funkcjonalności systemu. [12] Wymagania niefunkcjonalne dla systemu zostały przedstawione w tabeli 3.4.

Tabela 3.4. Wymagania niefunkcjonalne

| **Wymagania niefunkcjonalne** |
| --- |
| System jest dostępny przez Internet. |
| Możliwe jest korzystanie z systemu na poniżej wymienionych przeglądarkach internetowych:   * Mozilla Firefox w wersji 63.0.1 lub wyższej * Google Chrome w wersji 70.0.3538.77 lub wyższej |
| Funkcjonowanie systemu nie powinno zależeć od systemu operacyjnego, na którym jest uruchamiany. |

**3.3. Przypadki użycia**

Przypadek użycia jest opisem interakcji pomiędzy aktorami. Aktorzy mogą być użytkownikami systemu lub częściami aplikacji. Przypadek użycia posiada szerszy zakres niż historyjki użytkownika. W przypadku użycia posiadamy już informacje o tym w jaki sposób system będzie się zachowywał, a nie jedynie co i z jakiego powodu oczekuje od systemu użytkownik. Przypadki użycia są również bardziej ustrukturyzowane niż historyjki użytkownika. Zazwyczaj opis scenariusza przypadku użycia składa się z tytułu przypadku użycia, głównego scenariusza, który jest scenariuszem najczęściej wykonywanym oraz rozszerzeniami, czyli scenariuszami alternatywnymi, które mogą być również innymi przypadkami użycia. [13]

3.3.1. Scenariusze przypadków użycia

Poniżej w tabelach 3.5. – 3.22. przedstawione zostały opisy przypadków użycia. Przedstawione zostały w postaci tabel w celu poprawienia czytelności. W pierwszym wierszu tabeli znajduje się tytuł przypadku użycia. W drugim wierszu znajduje się identyfikator postaci PUXXX, gdzie XXX reprezentuje kolejne numery. W trzecim wierszu znajduje się aktor inicjujący przypadek użycia. W kolejnym wierszu opisany jest główny scenariusz przypadku użycia. W ostatnim wierszu tabeli znajdują się scenariusze alternatywne, wraz z odnośnikami do rozszerzających przypadków użycia.

Tabela 3.5. PU001

|  |  |
| --- | --- |
| Rejestracja | |
| Identyfikator | PU001 |
| Aktor | Gość |
| Scenariusz:   1. Użytkownik chce się zarejestrować. 2. System wyświetla formularz rejestracji. Do wpisu adres e-mail, hasło oraz powtórzenie hasła. Do wybrania jest również żądana rola: Nauczyciel lub Student. 3. Użytkownik wprowadza dane. 4. System stwierdza poprawność danych i przekierowuje na stronę logowania. | |
| Scenariusz alternatywny – wejście punkt 4:   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych. | |

Tabela 3.6. PU002

|  |  |
| --- | --- |
| Logowanie | |
| Identyfikator | PU002 |
| Aktor | Niezalogowany użytkownik |
| Scenariusz:   1. Użytkownik chce się zalogować. 2. System wyświetla formularz logowania. Do wpisu adres e-mail oraz hasło. 3. Użytkownik wprowadza dane. 4. System stwierdza poprawność danych i przekierowuje na stronę główną aplikacji. | |
| Scenariusz alternatywny – wejście punkt 4:   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych. | |

Tabela 3.7. PU003

|  |  |
| --- | --- |
| Edytowanie profilu | |
| Identyfikator | PU003 |
| Aktor | Zalogowany użytkownik |
| Scenariusz:   1. Użytkownik chce edytować swój profil. 2. System wyświetla aktualne informacje o profilu użytkownika w formie edytowalnej. 3. Użytkownik zmienia dane, które chce zmodyfikować ( adres email, hasło ). 4. Użytkownik zatwierdza zmianę danych. 5. System stwierdza poprawność danych i wyświetla "Profil zaktualizowano". | |

Tabela 3.8. PU003

|  |  |
| --- | --- |
| Przeglądanie kursów. | |
| Identyfikator | PU004 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce przeglądać kursy. 2. System wyświetla listę kursów. 3. Nauczyciel chce zakończyć przeglądanie kursów. 4. System zamyka listę kursów. | |
| Scenariusze alternatywne   1. Wejście punkt 3: PU004 – Dodawanie kursu 2. Wejście punkt 3: PU005 – Modyfikowanie kursu 3. Wejście punkt 3: PU006 – Przeglądanie działów kursu 4. Wejście punkt 3: PU009 – Przeglądanie pytań | |

Tabela 3.9. PU005

|  |  |
| --- | --- |
| Dodawanie kursu | |
| Identyfikator | PU005 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce dodać kurs. 2. System wyświetla formularz dodawania kursu. Do uzupełnienia nazwa kursu. 3. Nauczyciel wprowadza dane i potwierdza operację. 4. System stwierdza poprawność danych i przekierowuję do listy kursów. | |
| Scenariusz alternatywny – wejście punkt 3   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych | |

Tabela 3.10. PU006

|  |  |
| --- | --- |
| Modyfikowanie kursu | |
| Identyfikator | PU006 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce zmodyfikować kurs. 2. System wyświetla formularz modyfikowania kursu. Do modyfikacji nazwa kursu. 3. Nauczyciel modyfikuje nazwę kursu i potwierdza operację. 4. System stwierdza poprawność danych i przekierowuje do listy kursów. | |
| Scenariusz alternatywny – wejście punkt 4   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych   Scenariusz alternatywny – wejście punkt 1   1. Nauczyciel chce usunąć kurs. 2. System wyświetla ekran żądający potwierdzenia operacji. 3. Nauczyciel potwierdza operację. 4. System usuwa kurs i przekierowuje do listy kursów.   Scenariusz alternatywny:  3a. Nauczyciel anuluje żądanie | |

Tabela 3.11. PU007

|  |  |
| --- | --- |
| Przeglądanie działów kursu. | |
| Identyfikator | PU007 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce przeglądać działy kursu. 2. System wyświetla listę działów kursu. 3. Nauczyciel chce zakończyć przeglądanie działów kursu. 4. System zamyka listę działów kursu. | |
| Scenariusze alternatywne   1. Wejście punkt 3: PU007 – Dodawanie kursu 2. Wejście punkt 3: PU008 – Modyfikowanie kursu 3. Wejście punkt 3: PU009 – Przeglądanie pytań 4. Wejście punkt 3: PU009 – Przeglądanie pytań – poprzez wybranie szczególnego działu kursu. | |

Tabela 3.12. PU008

|  |  |
| --- | --- |
| Dodawanie działu kursu | |
| Identyfikator | PU008 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce dodać dział kursu. 2. System wyświetla formularz dodawania działu kursu. Do uzupełnienia nazwa działu. 3. Nauczyciel wprowadza dane i potwierdza operację. 4. System stwierdza poprawność danych i przekierowuję do listy działów kursu. | |
| Scenariusz alternatywny – wejście punkt 3   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych | |

Tabela 3.13. PU009

|  |  |
| --- | --- |
| Modyfikowanie działu kursu | |
| Identyfikator | PU009 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce zmodyfikować dział kursu. 2. System wyświetla formularz modyfikowania działu kursu. Do modyfikacji nazwa działu. 3. Nauczyciel modyfikuje nazwę działu i potwierdza operację. 4. System stwierdza poprawność danych i przekierowuje do listy działów kursu. | |
| Scenariusz alternatywny – wejście punkt 4   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych   Scenariusz alternatywny – wejście punkt 1   1. Nauczyciel chce usunąć dział. 2. System wyświetla ekran żądający potwierdzenia operacji. 3. Nauczyciel potwierdza operację. 4. System usuwa dział i przekierowuje do listy działów.   Scenariusz alternatywny:  3a. Nauczyciel anuluje żądanie | |

Tabela 3.14. PU010

|  |  |
| --- | --- |
| Przeglądanie pytań. | |
| Identyfikator | PU010 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce przeglądać pytania. 2. System wyświetla listę pytań ( jeżeli wejście było z PU003 wyświetlane są wszystkie pytania, jeżeli z PU006 ze scenariusza alternatywnego 3 wyświetlane są pytania z wybranego kursu, jeżeli z PUU006 ze scenariusza alternatywnego 4 wyświetlane są pytania z wybranego działu kursu ). 3. Nauczyciel chce zakończyć przeglądanie pytań. 4. System zamyka listę pytań. | |
| Scenariusze alternatywne   1. Wejście punkt 3: PU010 – Dodawanie pytania 2. Wejście punkt 3: PU011 – Modyfikowanie pytania 3. Wejście punkt 3: 4. Nauczyciel chce usunąć pytanie wybrane z listy. 5. System wyświetla ekran żądający potwierdzenia operacji. 6. Nauczyciel potwierdza operację. 7. System usuwa pytanie.   Scenariusz alternatywny:  3a. Nauczyciel anuluje operacje. | |

Tabela 3.15. PU011

|  |  |
| --- | --- |
| Dodawanie pytania | |
| Identyfikator | PU011 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce dodać pytanie. 2. System wyświetla formularz dodawania pytania. Do uzupełnienia kurs, dział kursu, treść pytania, odpowiedzi na pytanie wraz z punktami. 3. Nauczyciel wprowadza dane i potwierdza operację. 4. System stwierdza poprawność danych i przekierowuję do listy pytań. | |
| Scenariusz alternatywny – wejście punkt 3   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych | |

Tabela 3.16. PU012

|  |  |
| --- | --- |
| Modyfikowanie pytania | |
| Identyfikator | PU012 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce zmodyfikować pytanie. 2. System wyświetla formularz modyfikowania pytania. Do modyfikacji kurs, dział kursu, treść pytania, odpowiedzi na pytanie wraz z punktami. 3. Nauczyciel modyfikuje dane i potwierdza operację. 4. System stwierdza poprawność danych i przekierowuję do listy pytań. | |
| Scenariusz alternatywny – wejście punkt 3   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych | |

Tabela 3.17. PU013

|  |  |
| --- | --- |
| Przeglądanie testów. | |
| Identyfikator | PU013 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce przeglądać testy. 2. System wyświetla listę testów. 3. Nauczyciel chce zakończyć przeglądanie testów. 4. System zamyka listę testów. | |
| Scenariusze alternatywne   1. Wejście punkt 3: PU013 – Dodawanie testu 2. Wejście punkt 3: PU014 – Modyfikowanie testu 3. Wejście punkt 3: PU015 – Przeglądanie wyników testu 4. Wejście punkt 3: 5. Nauczyciel chce usunąć test wybrany z listy. 6. System wyświetla ekran żądający potwierdzenia operacji. 7. Nauczyciel potwierdza operację 8. System usuwa pytanie.   Scenariusz alternatywny:  3a. Nauczyciel anuluje operację. | |

Tabela 3.18. PU014

|  |  |
| --- | --- |
| Dodawanie testu | |
| Identyfikator | PU014 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce dodać test. 2. System wyświetla formularz dodawania testu. Do uzupełnienia: kurs, data i czas początku testu, data i czas końca testu, metoda oceniania testu, nawigowalność testu, próg zaliczenia testu, hasło do testu. Do wybrania z listy również są pytania oraz studenci. 3. Nauczyciel wprowadza dane i potwierdza operację. 4. System stwierdza poprawność danych i przekierowuję do listy testów. | |
| Scenariusz alternatywny – wejście punkt 3   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych | |

Tabela 3.19. PU015

|  |  |
| --- | --- |
| Modyfikowanie testu | |
| Identyfikator | PU015 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce dodać test. 2. System wyświetla formularz modyfikowania testu. Do modyfikacji: kurs, data i czas początku testu, data i czas końca testu, metoda oceniania testu, nawigowalność testu, próg zaliczenia testu, hasło do testu. Do wybrania z listy również są pytania oraz studenci. 3. Nauczyciel wprowadza dane i potwierdza operację. 4. System stwierdza poprawność danych i przekierowuję do listy testów. | |
| Scenariusz alternatywny – wejście punkt 3   1. System nie stwierdza poprawności danych i wyświetla komunikat o błędnym wprowadzeniu danych | |

Tabela 3.20. PU016

|  |  |
| --- | --- |
| Przeglądanie wyników testu | |
| Identyfikator | PU016 |
| Aktor | Nauczyciel |
| Scenariusz:   1. Nauczyciel chce przeglądać wyniki testu. 2. System wyświetla ekran z wynikami testu. Wyświetlone są: średnia zdobyta liczba punktów, wszystkie przesłane rozwiązania. 3. Nauczyciel chce zakończyć przeglądanie wyników testu. 4. System zamyka widok wyników testu. | |
| Scenariusz alternatywny – wejście punkt 3   1. Nauczyciel chce zobaczyć rozwiązanie wybrane z listy. 2. System wyświetla żądane rozwiązanie: pytania, wybrane odpowiedzi oraz uzyskany wynik. 3. Nauczyciel chce zakończyć przeglądanie rozwiązania. 4. System zamyka widok i przekierowuje do widoku wyników testu. | |

Tabela 3.21. PU017

|  |  |
| --- | --- |
| Przeglądanie własnych testów. | |
| Identyfikator | PU017 |
| Aktor | Student |
| Scenariusz:   1. Student chce przeglądać testy, do których jest przypisany. 2. System wyświetla listę testów. 3. Student chce zakończyć przeglądanie testów. 4. System zamyka listę testów. | |
| Scenariusze alternatywne   1. Wejście punkt 3: PU016 – Rozwiązywanie testu | |

Tabela 3.22. PU018

|  |  |
| --- | --- |
| Rozwiązywanie testu | |
| Identyfikator | PU018 |
| Aktor | Student |
| Scenariusz:   1. Student chce rozwiązywać test, do którego jest przypisany. 2. System wyświetla ekran proszący o hasło do testu. 3. Student wprowadza hasło. 4. System stwierdza poprawność wprowadzonego hasła i przekierowuje na widok rozwiązywania testu. 5. Dla każdego dostępnego pytania Student wybiera i zapisuje odpowiedzi. 6. Student chce zakończyć test. 7. System przesyła i ocenia rozwiązanie i przekierowuje na stronę z wynikiem testu. | |

3.3.2. Diagram przypadków użycia



**3.4. Prototypy interfejsu użytkownika**

**3.5. Model domenowy, reguły biznesowe, ograniczenia dziedzinowe**

**Bibliografia**

[1] Bałaszow, B., Królikowski, T., Susłow, W. i Szatkiewicz, T. (brak daty). *Portal wspomagania nauczania Politechniki Koszalińskiej StudiaNET.pl*. Koszalin, Polska.

[2] Bielski, K. (2012). *Implementacja REST w Spring MVC 3.0*. Pobrano z lokalizacji Krzysztof Bielski Blog: https://kbielski.wordpress.com/tag/rest-web-service/

[3] *Building an Application with Spring Boot*. (2018). Pobrano z lokalizacji Spring: https://spring.io/guides/gs/spring-boot/

[4] Dąbrowski, W. (brak daty). *Materiały edukacyjne Polsko-Japońskiej Akademii Technik Komputerowych*. Pobrano z lokalizacji http://edu.pjwstk.edu.pl/wyklady/byt/scb/wyklady/2\_2.html

[5] *Edytory dokumentów - Pomoc*. (2018). Pobrano z lokalizacji Google: https://support.google.com/docs/answer/7032287?hl=pl&ref\_topic=6063584&fbclid=IwAR3n1tdG0EaZzcOWA4MPSGMdgVdWqVFILofZjpbxJpkmH\_YHHbn7\_QruRyY

[6] *Google Design*. (2018). Pobrano z lokalizacji Google: https://design.google/

[7]*H2*. (brak daty). Pobrano z lokalizacji H2 Database Engine: http://www.h2database.com/html/main.html

[8] *moodledocs - Features*. (2014). Pobrano z lokalizacji https://docs.moodle.org/2x/pl/Features

[9] *moodledocs - Multiple Choice question type*. (2014). Pobrano z lokalizacji https://docs.moodle.org/2x/pl/Multiple\_Choice\_question\_type#Multiple-answer\_questions

[10] *moodledocs*. (2014). Pobrano z lokalizacji https://docs.moodle.org/2x/pl/Strona\_g%C5%82%C3%B3wna

[11] Przyłuski, W. (2014). Wielokrotny wybór czyli rozważania o pewnych pytaniach testowych. *ELEKTRONIKA - KONSTRUKCJE, TECHNOLOGIE, ZASTOSOWANIA*.

[12] Sommerville, I. (2011). *Software Engineering, 9th Edition.* Boston, Massachusetts: Pearson.

[13] Stephens, R. (2015). *Beginning Software Engineering.* Indianapolis: John Wiley & Sons, Inc.

[14] Urbański, W. (2018). *Wprowadzenie do kursu Vue.js od podstaw*. Pobrano z lokalizacji type<of>web: https://typeofweb.com/2018/06/08/wprowadzenie-do-kursu-vue-js-od-podstaw/

[15] *Vue.js Guide*. (2018). Pobrano z lokalizacji Vue.js: https://vuejs.org/v2/guide/

**Załączniki**

Rys 1.1. Model kaskadowy cyklu życia oprogramowania; Staszak, U. Projektowanie oprogramowania. Wykład 1. Politechnika Wrocławska, 20174

Rys 2.1. Widok tworzenia testu w aplikacji Formularze Google; Zrzut ekranu ze strony: ttps://docs.google.com/forms/d/e/1FAIpQLSdFIFB5c2ZphgFoJnsf69URQ0lmZuccaYaTHiBCOOODnbThcg/viewscore?viewscore=AE0zAgAlA8Du2XTy8qyrQ-0qoIAwWhd\_AIWjGhrQcfoJTSYCgAE7VQ6uk739JQ 6

Tabela 3.1. Użytkownicy aplikacji 9

Tabela 3.2. Historyjki użytkowników 10

Tabela 3.3. Wymagania funkcjonalne10

Tabela 3.4. Wymagania niefunkcjonalne11

Tabela 3.5. PU001 12

Tabela 3.6. PU00212

Tabela 3.7. PU00312

Tabela 3.8. PU00413

Tabela 3.9. PU00513

Tabela 3.10. PU00613

Tabela 3.11. PU00714

Tabela 3.12. PU00814

Tabela 3.13. PU009 15

Tabela 3.14. PU01015

Tabela 3.15. PU01116

Tabela 3.16. PU01216

Tabela 3.17. PU01316

Tabela 3.18. PU01417

Tabela 3.19. PU01517

Tabela 3.20. PU01617

Tabela 3.21. PU017 18

Tabela 3.22. PU01818