

Thesis Checker

System sprawdzający plagiaty

autorzy:

Kuś Małgorzata

Morek Tomasz

Paszkiewicz Paweł

Rak Łukasz

Sadkiewicz Bartłomiej

Sikora Magdalena

Szczygieł Michał

Śmierciak Aleksander

Wilczyński Andrzej

Witek Krzysztof

Spis treści:

1. Wstęp
 - a. Temat projektu
 - b. Istniejące rozwiązania
 - c. Metodyka pracy zespołu i użyte narzędzia
2. Specyfikacja wymagań
 - a. Funkcjonalne
 - b. Niefunkcjonalne
3. Diagramy i schematy
 - a. Schematy bazy danych
 - b. Diagram przypadków użycia
 - c. Diagram funkcjonalny
 - d. Diagramy przepływu
 - e. Schemat komunikacji z bazą danych
4. Środowisko pracy
 - a. Java EE6
 - b. Maven 3.0
 - c. Baza danych: MariaDB
 - d. Spring
 - e. Spring MVC
 - f. Biblioteki
5. Testy
6. Wizualizacja aplikacji
7. Możliwości rozwoju
8. Wnioski o pracy i o projekcie
9. Bibliografia

1. Wstęp

a. Temat projektu

Tematem projektu było stworzenie systemu umożliwiającego promotorom sprawdzanie prac studentów pod kątem plagiatów. Każdy zarejestrowany użytkownik może wczytać do systemu pracę studenta i rozpocząć analizę, która wykryje czy i w jakim stopniu jest to plagiat i wyświetli prace, z których zaczerpnięte zostały fragmenty tekstu.

b. Istniejące rozwiązania

Przed przystąpieniem do implementacji programu została przeprowadzona analiza teoretyczna istniejących rozwiązań. Poniżej znajdują się opisy kilku wybranych przykładów:

1. [Sherlock](#) - znajduje podobieństwa w dwóch plikach tekstowych. Używa podpisów cyfrowych, aby znaleźć podobny fragment tekstu. Program w C, pod Unix/Linux.
2. [WCopyFind](#) - jest darmowym programem opartym na Windows, który porównuje dokumenty i sprawdza podobieństwo. Można nim sprawdzać pliki doc, docx, txt, htm, html, pdf.
3. [CopyScape](#) - wyszukuje podobieństwa na stronach online. Program jest darmowy, konto premium jest płatne.

c. Metodyka pracy zespołu i użyte narzędzia

Praca zespołu była oparta na metodyce Scrum, która jest zaliczana do metodyk zwinnych, zgodnych z Agile. Scrum jest metodyką iteracyjno-inkrementacyjną. Rozwój całego projektu jest podzielony na mniejsze części. Co tydzień lub dwa odbywają się sprint review podczas którego analizowany jest postęp prac.

Oprogramowanie usprawniające i ułatwiające pracę w zespole:

1. [GitHub](#) - hostingowy serwis internetowy przeznaczony dla projektów programistycznych. Umożliwia wykorzystywanie systemu kontroli wersji GIT.
2. [HuBoard](#) - platforma do zarządzania zadaniami zintegrowana z GitHubem
3. [Doodle](#) - serwis pozwalający wyznaczyć wspólny termin spotkań.

2. Specyfikacja wymagań

a. Wymagania niefunkcjonalne

1. Bezpieczeństwo - system powinien zagwarantować odpowiednie bezpieczeństwo. Hasła użytkowników przechowywane w bazie danych powinny być szyfrowane.
2. Wymagania do aplikacji - uruchomienie aplikacji jest możliwe w każdej dostępnej przeglądarce. Istotnym czynnikiem przy uruchamianiu aplikacji jest również dostęp do internetu.
3. System operacyjny - aplikacja zapewnia możliwość uruchamiania na wszystkich dostępnych systemach operacyjnych z poziomu przeglądarki internetowej.
4. Logowanie - aplikacja może być używana w tym samym czasie przez wielu różnych użytkowników.

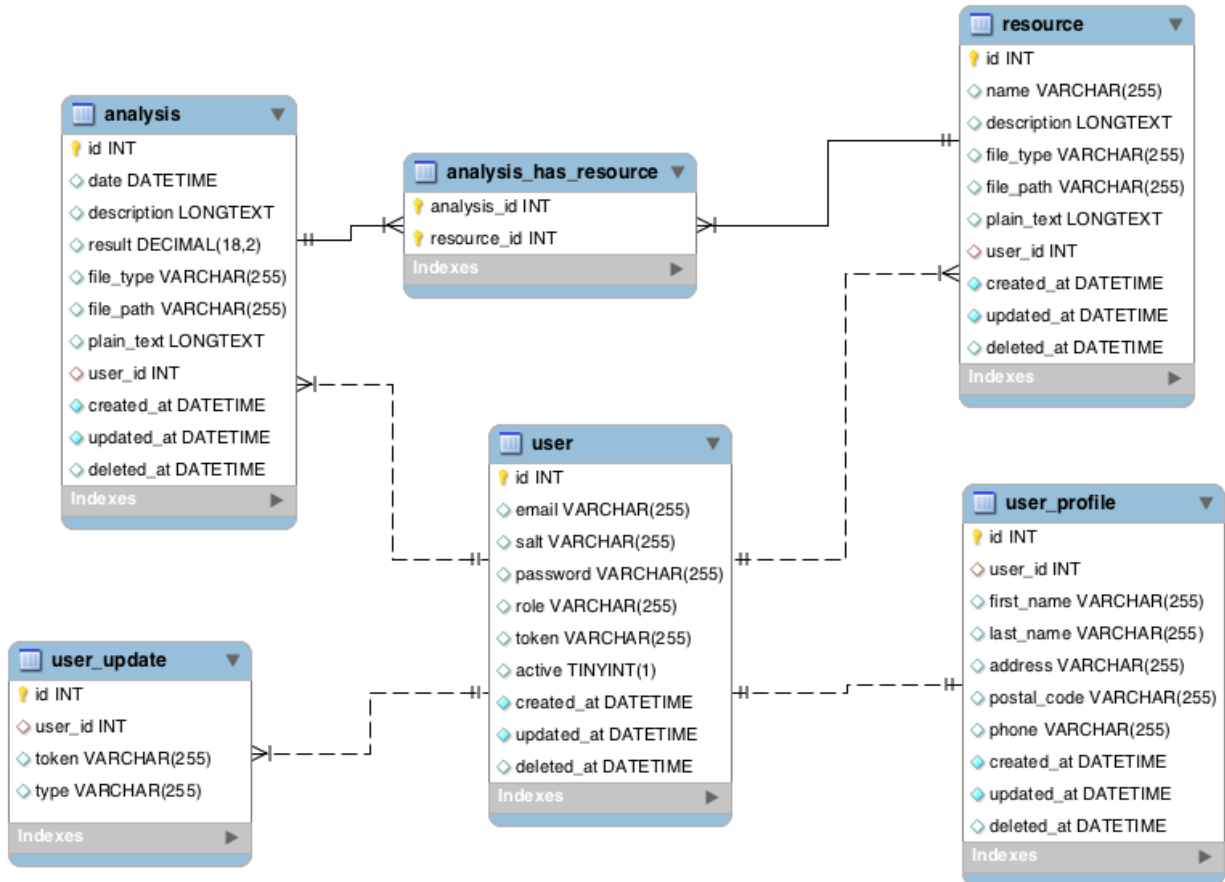
b. Wymagania funkcjonalne

1. Typy użytkowników:
 - a. gość
 - b. promotor
 - c. administrator
2. Interfejsy:
 - a. Interfejs dla gościa, który powinien zawierać formularz do zarejestrowania się.
 - b. Panel sterowania dla promotora i administratora - interfejs, który powinien zawierać widoki do realizacji działań przeznaczonych odpowiednio dla promotora i administratora.

3. Rejestracja promotora - promotor powinien zarejestrować się w systemie za pomocą formularza znajdującego się na stronie internetowej.
4. Logowanie - po dokonaniu rejestracji i aktywacji konta, promotor może zalogować się do systemu za pomocą formularza znajdującego się na stronie internetowej.
5. Wylogowanie - po zalogowaniu promotor może wylogować się z systemu.
6. Usuwanie konta - promotor może usunąć konto z systemu.
7. Wysyłanie wiadomości do administratora - promotor może wysłać wiadomość do administratora z pytaniem dotyczącym funkcjonowania systemu.
8. Sprawdzanie prac pod kątem plagiatu - promotor może sprawdzić pracę czy jest plagiatem.
9. Przeglądanie ręczne skopiowanych fragmentów w pracy - promotor może porównać skopiowany tekst w dwóch okienkach w czasie rzeczywistym .
10. Dodawanie pracy i przeglądanie listy prac w bazie danych - administrator i promotor może dodawać prace do bazy danych i może przeglądać listę prac za pomocą formularza i tabeli.
11. Dodawanie literatury do bazy danych - administrator i promotor może dodawać do bazy danych literaturę związaną z pracą.
12. Profil edycji danych - administrator i użytkownik może edytować dane w profilu przy użyciu formularza w panelu.
13. Dodawanie promotora - administrator może dodać konto promotora.
14. Aktywacja promotora - po zarejestrowaniu się promotora, administrator musi aktywować konto, aby promotor mógł się zalogować.
15. Przeglądanie historii porównań - administrator może przejrzeć historię porównań wszystkich prac wykonanych przez promotorów.

3. Diagramy i schematy

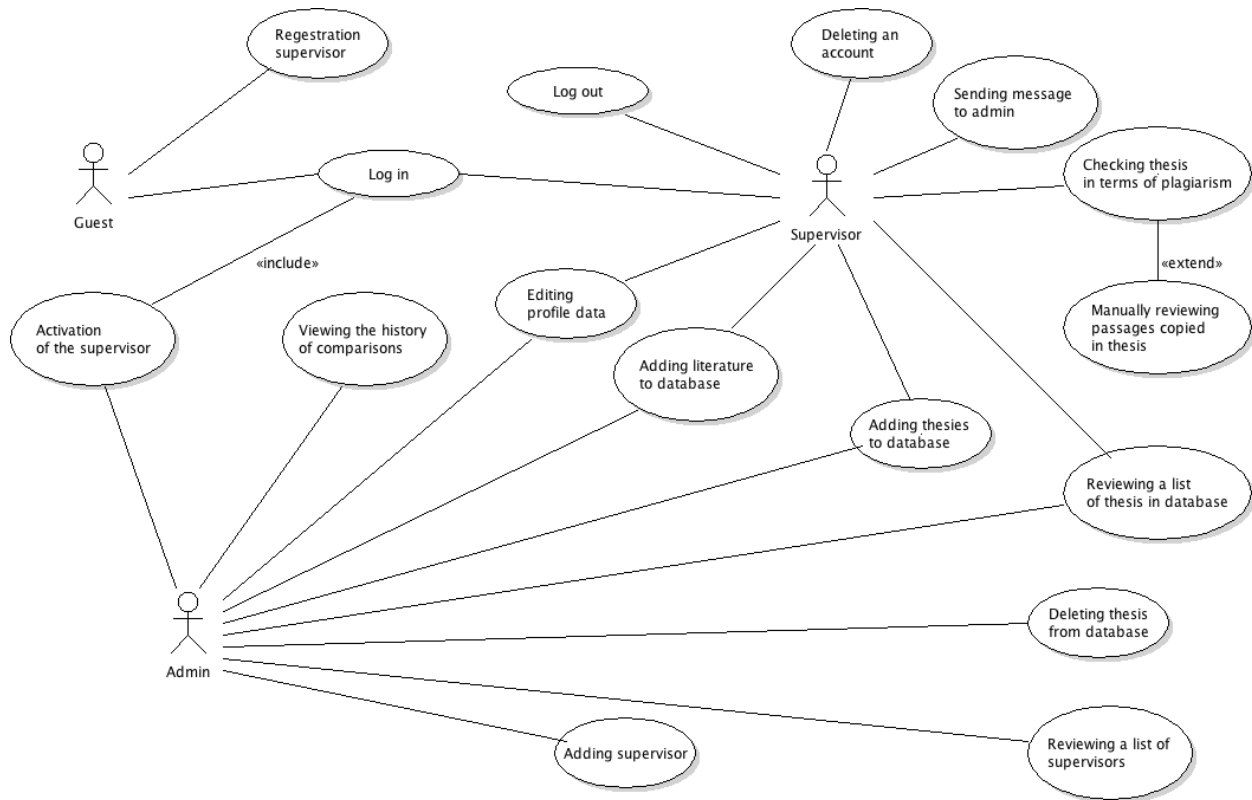
a. Schematy bazy danych



Baza danych spełnia następujące wymagania:

- możliwość przechowywania dokumentów PDF (Portable Document Format)
- Office Open XML (Docx dokument)
- plik powiązane z promotorami
- przechowywanie analizy podobieństwa sprawdzonych prac
- Wsparcie dla zalogowanego promotora:
 - zamieszczenie pracy w systemie
 - sprawdzanie analizy porównania
 - sprawdzanie pracy podobnej do sprawdzanej

b. Diagram przypadków użycia



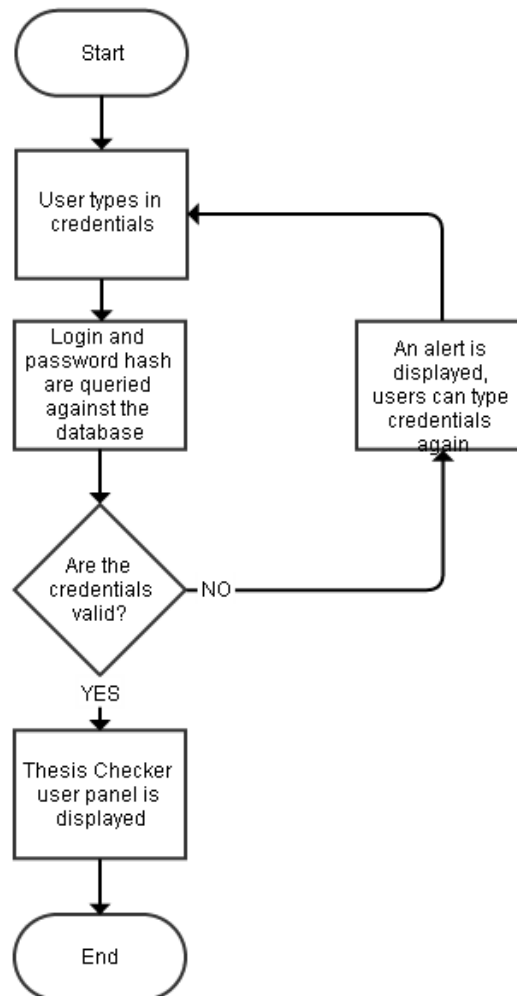
Lista przypadków użycia, gdzie użytkownik oznacza promotora:

- Rejestracja użytkownika - Użytkownik, który chce się zarejestrować wysła e-mail do administratora z prośbą o rejestrację podając wymagane dane takie jak imię, nazwisko, e-mail uczelniany i inne. Administrator systemu tworzy konto z loginem i hasłem oraz pozostałymi danymi i wysła informacje o utworzeniu konta na podany adres e-mail. Użytkownik, który otrzyma potwierdzenie, zmienia hasło przy pierwszym logowaniu.
- Logowanie użytkownika - Użytkownik wpisuje swój login i hasło do formularza logowania.
- Wylogowanie użytkownika - Użytkownik klika w przycisk "wyloguj".
- Edycja danych przez użytkownika - Użytkownik może zmienić swoje dane takie jak imię, tytuł akademicki oraz pozostałe informacje przez przejście do profilu i zmianę odpowiednich danych.
- Wczytanie pracy studenta przez promotora - Użytkownik jest zalogowany i za pomocą odpowiedniego widoku wczytuje plik w formacie PDF lub DOCX.

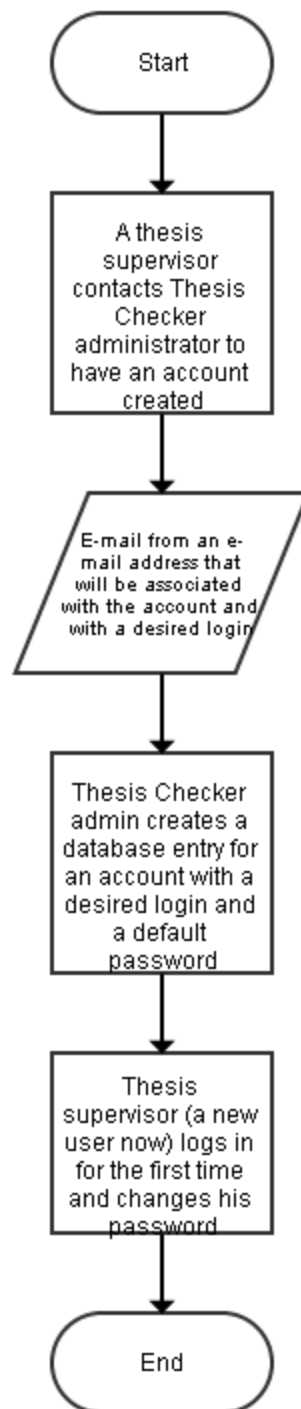
- Sprawdzanie podobieństwa prac przez promotora - Po wczytaniu pracy, promotor wybiera analizę pracy, gdzie wyświetla się wynik porównania.
- Sprawdzanie identycznych fragmentów innych prac - Użytkownik może sprawdzić z którą pracą jest podobna praca studenta. W czasie analizy wyświetlane są dwa dokumenty obok siebie: jedno z aktualnie sprawdzaną pracą i drugi z pracą, w której znajdują się podobne fragmenty.
- Kontakt użytkownika z administratorem - Użytkownik może skontaktować się z administratorem przez kliknięcie w łącze.

c. Diagram przepływu

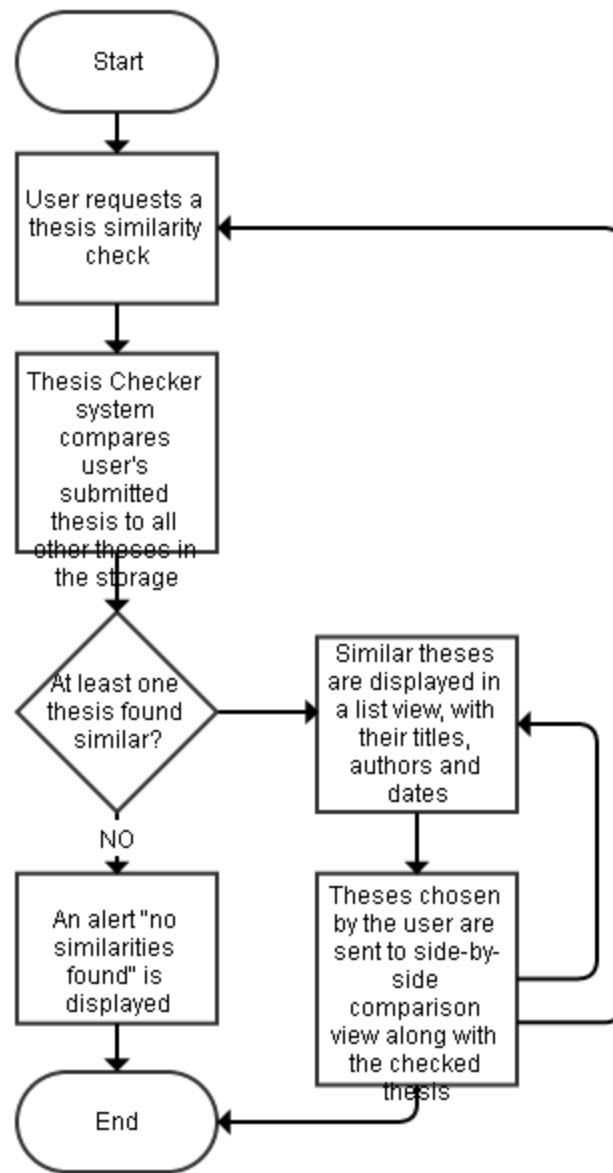
i. Autentykacja użytkownika



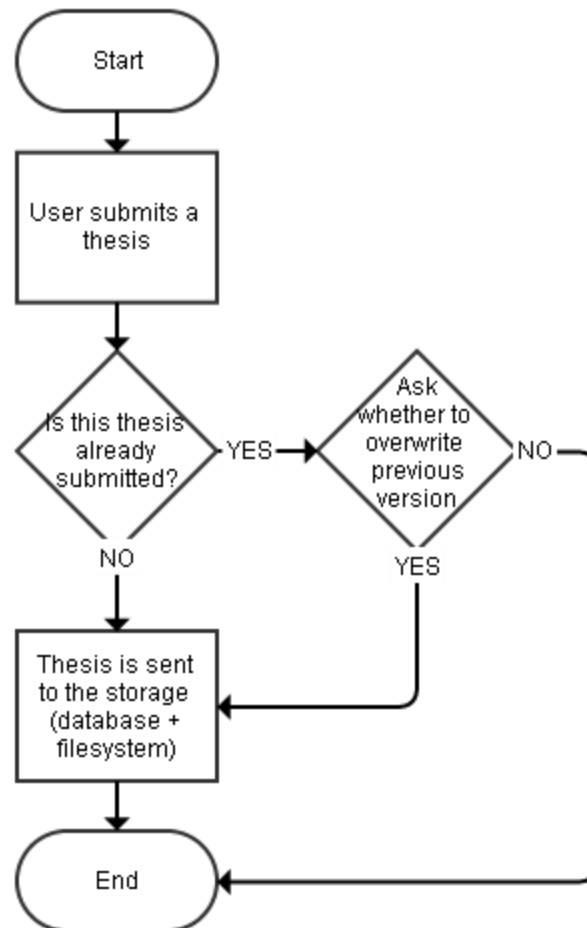
ii. Rejestracja



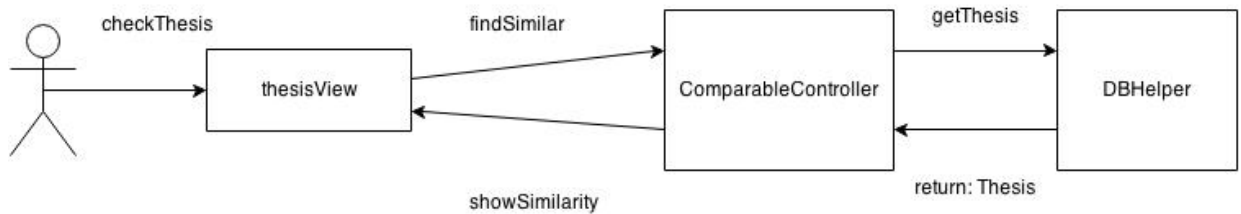
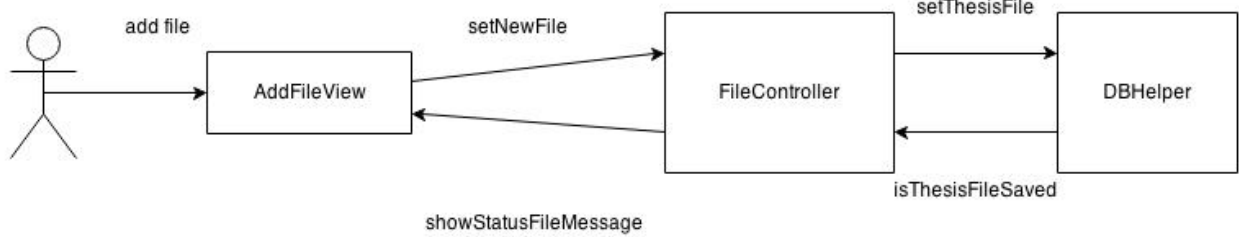
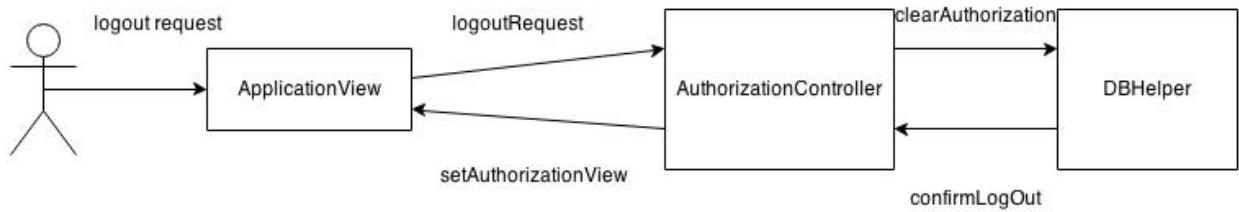
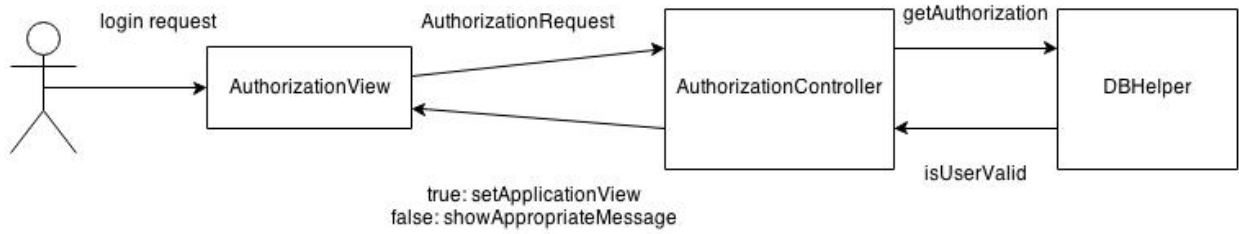
iii. Porównywanie prac



iv. Wysyłanie prac na serwer

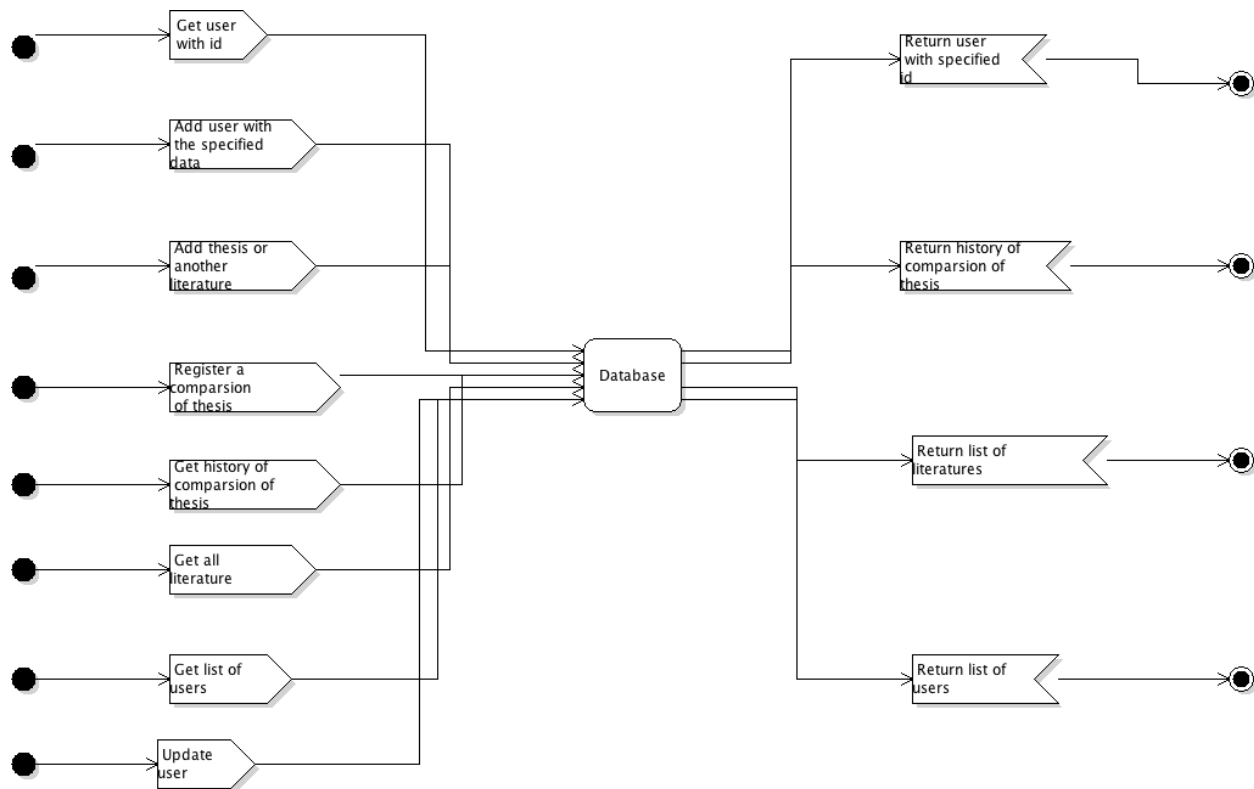


d. Diagram funkcjonalny



Schematy prezentują interakcje elementów systemu.

e. Schemat komunikacji z baza danych



Schematy te przedstawiają wszystkie operacje wejścia/wyjścia występujące w bazie danych.

4. Środowisko pracy - programy i biblioteki

a. Java EE6

Java jest obiektowym językiem programowania. Java jest wykorzystywana do tworzenia aplikacji. Jest niezależna od systemu operacyjnego i procesora. Program jest wykonywany na wirtualnej maszynie Javy.

b. Maven 3.0

Maven jest narzędziem, które pozwala automatyzować budowę oprogramowania na platformie Java. Może być używane do budowania i zarządzania projektem. Wybrane funkcjonalności są realizowane przez wtyczki pobierane automatycznie przy pierwszym wykorzystaniu. Każda wtyczka Maven posiada wyspecyfikowane cele.

c. Baza danych - MariaDB

MariaDB jest to realacyjna baza danych stworzona na wcześniejszej wersji MySQL. Jest udostępniana na wolnej licencji i jest kompatybilna z wcześniejszymi wersjami MySQL.

Baza danych MariaDB została wybrana ze względu na darmowy dostęp i duże możliwości. Można ją wykorzystać do projektów, w których jest wielu użytkowników oraz tam, gdzie wymagane są różne typy uprawnień do różnych tabel.

d. Framework aplikacji - Spring

Spring jest to framework ułatwiający tworzenia aplikacji w języku Java dla platformy Java EE/J2EE. Jest alternatywą dla programowania aplikacji z użyciem Enterprise JavaBeans. Bazowe komponenty Springa mogą być używane w każdej aplikacji.

e. Biblioteki

- i. [Apache POI](#) - odczyt i przetwarzanie dokumentów MS Word.
- ii. [PDFBox](#) - odczyt i przetwarzanie dokumentów pdf to czystego tekstu.
- iii. [Google Diff Match Patch](#) - algorytmy do porównywania dwóch bloków tekstu i wykazywania listy różnic.

5. Testy

Testy jednostkowe

Większość funkcjonalności zostało pokryte testami jednostkowymi pisanymi przez programistów, wykorzystując metodykę TDD (Test Driven Development). Wymaga aby najpierw zostały napisane testy, a dopiero później zaimplementowane funkcjonalności. Po przejściu testów, następuje etap refaktoryzacji kodu.

Testy manualne

Interfejs graficzny aplikacji został przetestowany manualnie.

6. Wizualizacja aplikacji

Zrzuty ekranu z działania aplikacji dostępne w folderze: **/ThesisChecker/sceenshots/**

7. Możliwości rozwoju

Możliwości rozwoju aplikacji:

- Sprawdzanie plagiatów w kodach źródłowych programów.
- Wyszukiwanie plagiatów nieograniczone do dokumentów w bazie, ale otwarte na wyszukiwania w internecie.
- Możliwość sprawdzania odporności na plagiat własnej pracy przez studenta zanim trafi do promotora.
- Połączenie bazy danych z uczelnianym archiwum prac oraz innymi uczelniami.

8. Wnioski

Praca z metodyką SCRUM pozwoliła na dobrą komunikację w zespole. Podział prac i regularne spotkania zmotywował członków zespołu do pracy przez co zaangażowanie w wytwarzanie oprogramowania było bardziej efektywne jednak brak możliwości organizacji codziennych spotkań i codziennej pracy nad projektem nie wpłynęły pozytywnie na jego implementację.

9. Bibliografia

Artykuły:

- <http://www.e-mentor.edu.pl/artykul/index/numer/12/id/213>
- http://www.forumakad.pl/archiwum/2003/01/artykuly/23-on-automatyczne_wykrywanie_plagiatow.htm
- <http://filip-lakomik.blogspot.com/2010/05/antyplagiat-co-potrafi.html>

Algorytmy:

- algorytm Levenshteina - <http://bjoniec.prv.pl/document/Praca%20licencjacka%20-%20Bartosz%20Joniec.pdf>
- Algorytm Karpa-Rabina - http://pl.wikipedia.org/wiki/Algorytm_Karpa-Rabina
- Algorytm LCS (*longest common subsequence*) - http://pl.wikipedia.org/wiki/Najd%C5%82u%C5%BCszy_wsp%C3%B3lny_podci%C4%85g