



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

KATEDRA INFORMATYKI STOSOWANEJ

Praca dyplomowa inżynierska

*Serwis internetowy dla studentów zaimplementowany w architekturze
SOA*

Internet service for students implemented in SOA architecture

Autor:

Marek Ryznar

Kierunek studiów:

Informatyka

Opiekun pracy:

dr inż. Mirosław Gajer

Kraków, 2015

Oświadczam, świadomy(-a) odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Serdecznie dziękuję ... tu ciąg dalszych podziękowań np. dla promotora, żony, sąsiada itp.

Spis treści

1. Wprowadzenie	7
1.1. Przedmowa - Przerobic na ladny tekst	7
1.2. Cel pracy	7
2. Podstawy teoretyczne	9
2.1. SOA	9
2.2. Java	10
2.3. Przechowywanie danych	10
2.4. Serwer Aplikacyjny	11
3. Projekt systemu	13
3.1. Opis systemu	13
3.1.1. Cel projektu	13
3.1.2. Uzytkownicy	13
3.1.3. Granice systemu	14
3.1.4. Lista możliwości	14
3.2. Specyfikacja i analiza wymagań	15
3.2.1. Wymagania oprogramowania	15
3.2.2. Ryzyka operacyjne	16
3.2.3. Przypadki użycia	16
3.3. Model systemu	25
3.3.1. Architektura systemu	25
3.3.2. Model bazy danych	25
3.3.3. Model klas	25
3.3.4. Diagramy sekwencji	25
3.4. Projekt interfejsu?	26
4. Implementacja	29
4.1. Architektura	29
4.1.1. Dostęp do bazy danych	29

4.1.2. Widok aplikacji	29
4.2. Zabezpieczenia	29
4.3. Testy	29
5. Wyniki i podręcznik użytkownika	31
5.1. No idea	31
6. Podsumowanie	33

1. Wprowadzenie

1.1. Przedmowa - Przerobic na ladny tekst

- Cos o tym ze dawnej studenci musieli zadupcac kserowac notatki, spisywac od znajomych. No ale zaznaczyc to ze musieli zadupcac po to wszystko

- nagle buch - internet sie stal - wszystkie wymienione wyzej problemy mozna rozwiazac nie ruszajac sie z fotela.

- Gdy sie nie wie kiedy kolos - piszesz na portalu społecznościowym, forum itd. (Tutaj moze jak sie da to ladnie wstawic to przyklady portali fejs,twitter,fora itd)

-A gdy potrzebujesz notatki to albo wysyla Ci ktos mailem, albo udostepnia na dysku internetowym, co tez ulatwia zdecydowanie zycie (No i tutaj opisac dyski internetowe? ze google wiezie prym bo ma maila i google dysk, dodatkowo napisac o innych istniejacych rozwiiazaniach (dropbox, oneDrive itd)).

- Tutaj trzeba bedzie napisac o tym ze to jest wazna rzecz w zyciu studenta, ze bez tego to jak bez reki.

- Ze wszystkie opisane powyzej rozwiiazania sa eksta itd, (ale wg mojego researchu?) nie ma rozwiiazania dedykowanego stricte dla studentow tj portalu na ktorym mogli by wymieniac swobodnie pliki, zapisywac terminy w kalendarzu itd.

- I tutaj BUM - ja chce cos takiego zrobic

• Dlaczego autor chcial zajac sie danym tematem?

Przykladowe odniesienie do bibliografii

1.2. Cel pracy

- Celem pracy jest stworzenie serwisu internetowego opartego na architekturze SOA.

- Gdzies tu trzeba opisac co bedzie sie zawieralo w pracy, ze strona www i jaka funkcjonalnosc na tej stronie

- jaki serwer

- No i tutaj w sumie wspomiec, ze wybralem taki temat bo chcialem poznac technologie SOA? albo wspomiec o tym gdzie indziej

- NO i tutaj po kolei opisac co robie w pracy, z grubsza z czego korzystam

- Byc moze co w ktorym rozdziale tak z grubsza

- To wszystko razem powinno zająć wystarczająco miejsca

- Co autor chciał osiągnąć, czego się spodziewał w efekcie realizacji pracy?
- Czego czytelnik dowie się po przeczytaniu pracy?

2. Podstawy teoretyczne

Rozdział ten przedstawia podstawy teoretyczne projektu. W pierwszym poroździale znajduje się krótkie wprowadzenie do architektury SOA, na której oparty jest serwis. W następnym opisany jest język implementacji. Trzeci podrozdział prezentuje formę przechowywania danych w serwisie. Ostatni punkt zaznajamia nas z przeznaczeniem serwerów aplikacyjnych i konkretnym zastosowaniu w implementowanym projekcie.

2.1. SOA

Rozwinięciem skrótu SOA jest *Service Oriented Architecture* czyli architektura zorientowana na usługi. Według książki *SOA Design Principles for Dummies* SOA składa się z trzech podstawowych aspektów:

- **Usługa** - czyli powtarzalne zadanie biznesowe np. Dodawanie plików na serwer lub tworzenie nowego konta.
- **Orientacja na usługi** - Sposób postrzegania produktu jako zbiór usług połączonych w całość.
- **SOA** - Podejście architektoniczne bazujące na zasadach orientacji na usługi.

Architektura zorientowana na usługi posiada następujące właściwości:

- **Luźne powiązanie usług**
- **Abstrakcyjne usługi**
- **Autonomiczność usług**
- **Możliwość wielokrotnego użycia usług**
- **Bezstanowość usług**
- **Dzielenie usług na komponenty**

[1] SOA jest architekturą wspierającą modularność więc produkty pisane w tej architekturze łatwo jest rozwijać poprzez dodawanie nowych usług.

- a tutaj może ze może na projekt studencki to nie jest najlepsze rozwiązanie, ale jeżeli miało by to się rozwinąć to to jest super ekstra pomysł i dlatego tak jest! (Tutaj zapewne wspomnieć, że SOA jest modularne i bez problemu można dodawać kolejne moduły).

2.2. Java

Język programowania Java został stworzony na początku lat dziewięćdziesiątych przez grupę inżynierów z firmy Sun zwaną "Green Team". Ideą tworzonego języka była praca w rozwijającym się środowisku internetu [2]. Java została uformowana na podstawie języka C++ przy czym wymusza ona programowanie obiektowe. Jest jednym z najbardziej popularnych języków programowania i każdy kto miał jakkolwiek styczność z programowaniem przynajmniej o nim słyszał (Tutaj mam nadzieję, że Paulinka mi zrefakturuje to zdanie).

Java dzięki swojej platformie programistycznej Java EE (Java Enterprise Edition) pozwala pisać multiplatformowe, wielowarstwowe niezawodne i bezpieczne aplikacje sieciowe. Dzięki wielu darmowo udostępnianym bibliotekom i frameworkom wielu programistów decyduje się na wybranie tej właśnie platformy [3]. W swojej pracy główne frameworki platformy Java EE pomagające w tworzeniu opisywanej aplikacji to:

- **JSF** - Java Server Faces, który upraszcza tworzenie internetowego interfejsu użytkownika, dzięki udostępnianym tagom, z których można korzystać tworząc strony WWW. Posiada także Managed Beans czyli klasy odpowiedzialne za połączenie logiki aplikacji z widokiem czyli ze stronami internetowymi.
- **Hibernate** - jest to biblioteka dla języka Java dostarczająca framework do realizacji warstwy dostępu do danych. Zapewnia translację danych z relacyjnej bazy danych na klasy Java i vice versa (Object/Relational Mapping). Zastępuje bezpośrednie metody dostępu do danych za pomocą wysokopoziomowych metod operujących na zmapowanych obiektach.

2.3. Przechowywanie danych

Wirtualne dane można przechowywać na wiele sposobów, najprostszym z nich jest przechowywanie danych lokalnie czyli na dysku twardym stacji roboczej, aczkolwiek w tym wypadku istnieje zagrożenie utraty tych danych w razie zepsucia się komputera. W obawie przed utratą danych użytkownicy często szukają sposobu na umieszczenie ich zapasowej kopii w internecie. Istnieje wiele sposobów na przechowywanie naszych plików w internecie:

- **Dyski sieciowe** - jest to typ urządzenia przechowującego dane, które zapewnia dostęp do nich w lokalnej sieci LAN.

- **Dyski internetowe** - umożliwiają przechowywanie plików w "internecie", tzn. korzystają między innymi z chmur internetowych do zapisu danych.
- **Bazy danych** - z tego rozwiązania korzysta się raczej w przypadku ustrukturyzowanych danych np. Informacje o użytkownikach (login użytkownika serwisu w postaci krótkiego pola tekstowego lub rok jego urodzenia w postaci liczby naturalnej). Pomimo tego niektóre bazy danych udostępniają możliwość przechowywania danych w postaci binarnej czyli bez podania ich typu co pozwala na zapisywanie danych bez wymuszonego ich typu.

Do implementacji aplikacji użyto bazy danych. Wybór był podyktowany tym, że platforma Java EE udostępnia framework Hibernate, który pozwala na prosty dostęp do bazy danych. Wybrany system do zarządzania relacyjną bazą danych to PostgreSQL, system ten jest wolnodostępny. Do przechowywania danych o użytkownikach i ich kalendarzach wystarczą podstawowe typy danych dostępne we większości systemów zarządzających bazą danych. Do przechowywania większych plików takich jak całe książki w dowolnym formacie służy typ danych BLOB czyli Binary Large Objects co jak sama nazwa wskazuje służy do przechowywania większych plików binarnych.

2.4. Serwer Aplikacyjny

Serwer aplikacyjny jest kontenerem webowym, który odpowiada za przechowywanie, zdalne uruchamianie i użytkowanie aplikacji Java EE. Ponadto pomaga programiście tworzyć aplikacje oraz umożliwia oddzielenie logiki biznesowej od usług dostarczanych przez dany serwer aplikacyjny np. zabezpieczenia aplikacji [4]. Najbardziej znane serwery aplikacyjne to JBoss, IBM WebSphere, Apache Tomcat, GlassFish. Serwer używany do implementacji opisywanej aplikacji to JBoss AS 7.1.1. Serwer ten został wybrany ponieważ jest darmowy (udostępniany na licencji *GNU Lesser General Public License*), ponadto implementuje pełen zestaw usług JEE. Dodatkowym argumentem przemawiającym za wyborem właśnie tego serwera była jego integracja z Eclipse (środowiskiem programistycznym w którym wykonany został projekt) za pomocą wtyczki JBossTools co znacznie ułatwiło pracę.

3. Projekt systemu

Rozdział ten w 3 podrozdziałach przedstawia koncepcję implementowanej aplikacji. W podrozdziale pierwszym opisany jest system, w drugim znajduje się specyfikacja wymagań, a w trzecim model systemu. Dokumentacja projektu została częściowo oprata na szablonie dokumentacji projektu stworzonym przez dr Piotra Szweda [5].

3.1. Opis systemu

W tym podrozdziale opisany jest cel systemu, jego granice, lista możliwości systemu oraz jego użytkownicy. Ponadto za pomocą diagramów aktywności przedstawione jest działanie aplikacji.

3.1.1. Cel projektu

Celem projektu jest implementacja serwisu internetowego dedykowanego dla studentów w architekturze SOA. Głównym zadaniem serwisu ma być możliwość przechowywania plików oraz udostępnianie ich innym użytkownikom. Ponadto każdy z użytkowników będzie mógł korzystać z dedykowanego dla niego kalendarza, w którym może zapisywać ważne wydarzenia. Oprócz opisanych wcześniej funkcji użytkownik za pomocą strony powiadomień będzie mógł się dowiedzieć o nadchodzących wydarzeniach oraz o udostępnionych dla niego plikach.

3.1.2. Użytkownicy

Grupę użytkowników stanowią wszyscy korzystający z serwisu, z wyszczególnieniem:

- **Użytkownik** - Docelowym użytkownikiem serwisu jest student potrzebujący dodać lub uzyskać dostęp do pliku poprzez udostępnienie od innych użytkowników serwisu. Nie jest warunkiem koniecznym, aby użytkownik był studentem.
- **Administrator** - Użytkownik posiadający wszystkie uprawnienia, takie jak usuwanie użytkowników oraz wgląd do wszystkich plików w systemie.

3.1.3. Granice systemu

Granicą systemu dla użytkownika jak i dla administratora będzie strona internetowa aplikacji. Zawarty będzie tam dostęp do wszystkich funkcjonalności systemu. Administrator dodatkowo będzie miał dostęp do specjalnego panelu za pomocą którego będzie mógł zarządzać kontami wszystkich użytkowników.

Wejścia w systemie:

- formularz nowego konta – użytkownik wypełnia dane: login, hasło, imię, nazwisko, adres e-mail,
- formularz logowania – zalogowanie się do systemu za pomocą loginu i hasła użytkownika,
- formularz edycji konta – zmiana danych użytkownika,
- formularz nowego pliku – załącznik oraz opcjonalnie tytuł i opis pliku,
- formularz edycji pliku - zmiana załącznika, tytułu lub opisu,
- formularz udostępnienia pliku - login użytkownika, któremu ma być udostępniony wybrany plik,
- formularz nowego wydarzenia - wybrana data, nazwa wydarzenia oraz opcjonalny opis,
- formularz edycji wydarzenia - nowe dane do wybranego wydarzenia,

Wyjścia w systemie:

- Strona powiadomień - na tej stronie znajdować się będą informacje o udostępnionych plikach oraz o przypomnieniu o nadchodzących wydarzeniach.
- Strona przeglądania własnych plików - użytkownik na tej stronie może przeglądać dodane przez siebie pliki,
- Strona przeglądania udostępnionych plików - tutaj użytkownik będzie mógł przeglądać pliki mu udostępnione,
- Strona kalendarza - Strona odpowiedzialna za wyświetlanie kalendarza.

3.1.4. Lista możliwości

Lista wymagań funkcjonalnych:

- Logowanie,
- Wylogowanie,
- Rejestracja,
- Zarządzanie kontem,

- Zarządzanie materiałami,
- Pobieranie pliku,
- Udostępnianie pliku,
- Zarządzanie kalendarzem,
- Otrzymywanie powiadomień,

Dodatkowe funkcje administratora:

- Zarządzanie kontami użytkowników.

Na kolejnych stronach zostały przedstawione diagramy czynności typowych akcji.

- Rejestracja

3.2. Specyfikacja i analiza wymagań

Dzięki analizie wymagań funkcjonalnych możemy zidentyfikować oczekiwane zachowania budowanego systemu. Wymaganie funkcjonalne jest to „stwierdzenie, jakie usługi ma oferować system, jak ma reagować na określone dane wejściowe oraz jak ma się zachowywać w określonych sytuacjach. W niektórych wypadkach wymagania funkcjonalne określają, czego system nie powinien robić” [6]. W tym podrozdziale opisane zostaną lista wymagań funkcjonalnych oraz niefunkcjonalnych oraz przedstawiony zostanie diagram przypadków użycia z dołączonymi scenariuszami.

3.2.1. Wymagania oprogramowania

Wymagania funkcjonalne zostały wypisane w podpunkcie 3.1.4. Wymagania niefunkcjonalne:

1. Wymaganie organizacyjne:

- Implementacji : językiem programowania jest Java,
- Użyteczności : Interfejs graficzny użytkownika jest przejrzysty i łatwy do przyswojenia,

2. Wymagania zewnętrzne:

- Poufności - Projekt przestrzega wymagań prawnych ustanowionych w Ustawie o Ochronie Danych Osobowych.
- Bezpieczeństwa : System nie pozwala użytkownikom na nieautoryzowany dostęp do bazy danych

3.2.2. Ryzyka operacyjne

Ryzykiem w implementowanym systemie może być niedostateczne zabezpieczenie bazy danych lub pojawienie się nieobsłużonych błędów. //TODO: Poszukać i dodać tu jeszcze coś

3.2.3. Przypadki użycia

Dzięki przypadkom użycia możemy określić jak użytkownik będzie korzystał z systemu. Dzięki diagramowi możemy zobaczyć jakie czynności może wykonywać dany użytkownik [7]. Opisy czynności zawartych na diagramie znajdują się w scenariuszach przypadków użycia.

Poniższe diagramy prezentują możliwe interakcje użytkowników z systemem. W poniższym diagramie pominięto przypadki użycia, które posiada zwykły użytkownik, lecz nie należy zapominać, że administrator również je posiada.

Scenariusze przypadków użycia:

(I) Wyloguj

Tabela 3.1. Wyloguj

Aktorzy:	Użytkownik, Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Użytkownik chce się wylogować.
Zdarzenie wyzwalające	Użytkownik wciska przycisk “Wyloguj”.
Warunki wstępne:	Użytkownik jest zalogowany.
Warunki końcowe dla sukcesu:	Użytkownik jest wylogowany.
Warunki końcowe dla niepowodzenia:	Użytkownik jest zalogowany.

Scenariusz główny:

1. Użytkownik wciska przycisk wyloguj widoczny w prawym górnym rogu strony.
2. Pojawia się okno z napisem: “Czy na pewno chcesz się wylogować?”
3. Użytkownik wciska przycisk “tak”.
4. Użytkownik jest wylogowany i przekierowany na stronę główną systemu.

Scenariusz alternatywny:

- 3.a. Użytkownik wciska “nie”.
- 3.a.1. Okno z napisem znika, użytkownik dalej jest zalogowany.

(II) Zarządzaj kontem

Scenariusz główny:

1. Użytkownik wciska przycisk: “Zarządzaj kontem”.

Tabela 3.2. Zarządzaj kontem

Aktorzy:	Użytkownik, Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Użytkownik chce zmienić login lub hasło.
Zdarzenie wyzwalające	Użytkownik wciska przycisk “Zarządzaj kontem”.
Warunki wstępne:	Użytkownik jest zalogowany.
Warunki końcowe dla sukcesu:	Login lub hasło użytkownika są zmienione.
Warunki końcowe dla niepowodzenia:	Login lub hasło użytkownika nie są zmienione.

2. System przenosi użytkownika do strony zarządzania kontem, na której ma do wyboru dwie opcje: “Zmień login” i “Zmień hasło”
3. Użytkownik wybiera opcje “Zmień login”.
4. System przenosi użytkownika do strony zmiany loginu, na której znajduje się formularz z polem tekstowym na nowy login.
5. Użytkownik wpisuje nowy login i wciska “Zatwierdź”.
6. System sprawdza czy nowy login nie występuje w bazie danych.
7. System wyświetla okno o pomyślnej zmianie loginu i przekierowuje użytkownika do strony zarządzania kontem.

Scenariusz alternatywny:

- 3.a. Użytkownik wybiera opcje “Zmień hasło”.
 - 3.a.1. System przenosi użytkownika do strony zmiany hasła, na której znajduje się formularz z polem tekstowym na nowe hasło.
 - 3.a.2. Użytkownik wpisuje nowe hasło w pola “hasło” oraz “potwierdź hasło” i wciska “Zatwierdź”.
 - 3.a.3. System sprawdza czy wartości z pola hasło i potwierdź hasło są takie same.
 - 3.a.4. System sprawdza czy hasło ma odpowiednią długość i zawiera duże znaki.
 - 3.a.5. System wyświetla komunikat o poprawnej zmianie hasła i przenosi użytkownika do strony zarządzania kontem.

Scenariusz alternatywny:

- 3.a.3.a Wartość z pola hasło i potwierdź hasło nie są takie same.
 - 3.a.3.a.1 System wyświetla odpowiedni komunikat błędu **Scenariusz alternatywny:**
- 7.b. System wyświetla komunikat błędu: “Podany login już istnieje w systemie”.
 - 7.b.1. Powrót do pkt.4 scenariusza głównego.

(III) Zarządzaj kalendarzem:

Tabela 3.3. Zarządzaj kalendarzem

Aktorzy:	Użytkownik, Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Użytkownik chce dodać lub usunąć wydarzenie w kalendarzu.
Zdarzenie wyzwalające	Użytkownik wciska przycisk “Zarządzaj kalendarzem”.
Warunki wstępne:	Użytkownik jest zalogowany.
Warunki końcowe dla sukcesu:	Użytkownik dodał, edytował lub usunął wydarzenie.
Warunki końcowe dla niepowodzenia:	Użytkownik nie dodał, nie edytował lub nie usunął wydarzenia

Scenariusz główny:

1. Użytkownik wciska przycisk “Zarządzaj kalendarzem”.
2. System przekierowuje użytkownika na stronę zarządzania kalendarzem.
3. Użytkownik wybiera datę w kalendarzu.
4. System podświetla wybrany przez użytkownika dzień i wyświetla przycisk “dodaj wydarzenie”, “usuń wydarzenie” oraz “edytuj wydarzenie”, jeżeli w danym dniu jest jakieś zapisane.
5. Użytkownik wciska przycisk “dodaj wydarzenie”.
6. System przenosi użytkownika na stronę dodawania wydarzenia, na której znajduje się data, pole tekstowe na nazwę wydarzenia oraz na godzinę wydarzenia i czas trwania.
7. Użytkownik wypełnia formularz i wciska przycisk “zatwierdź”.
8. System sprawdza poprawność wpisanych danych.
9. Wydarzenie jest dodane do kalendarza

Scenariusz alternatywny:

- 1.b. System wyświetla komunikat błędu: “Błąd serwera”
- 1.b.1. Powrót do pkt.1 scenariusza głównego.

Scenariusz alternatywny:

- 8.a. System wyświetla komunikat błędu: “Pole “czas trwania” musi być wypełnione liczbą.”
- 8.a.1. Powrót do pkt. 4 scenariusza głównego.

Scenariusz alternatywny:

- 4.a Użytkownik wciska przycisk “usuń wydarzenie”.
- 4.a.1 System przenosi użytkownika na stronę usuwania wydarzenia, na której znajduje się lista wydarzeń zaplanowanych na wybrany dzień oraz przyciski “usuń” obok każdego z nich.

- 4.a.2 Użytkownik wciska przycisk “usuń” obok jednego z wydarzeń.
- 4.a.3 System wyświetla okno z pytaniem czy na pewno usunąć wydarzenie.
- 4.a.4 Użytkownik wciska przycisk “tak”.
- 4.a.5 System usuwa wydarzenie i wyświetla komunikat o pomyślnym usunięciu wydarzenia.
- 4.a.6 System przekierowuje użytkownika na stronę zarządzania kalendarzem.

Scenariusz alternatywny:

- 4.a.3.a Użytkownik wciska przycisk “nie”.
- 4.a.3.a.1 Powrót do pkt. 4 scenariusza głównego.

Scenariusz alternatywny: 4.b Użytkownik wciska przycisk “edytuj wydarzenie”.

- 4.b.1 System przenosi użytkownika na stronę edycji wydarzenia, na której znajduje się lista wydarzeń zaplanowanych na wybrany dzień oraz przyciski “edytuj” obok każdego z nich.
- 4.a.2 Użytkownik wciska przycisk “edytuj” obok jednego z wydarzeń.
- 4.a.3 System wyświetla okno z pytaniem czy na pewno usunąć wydarzenie.
- 4.a.4 Użytkownik wciska przycisk “tak”.
- 4.a.5 System usuwa wydarzenie i wyświetla komunikat o pomyślnym usunięciu wydarzenia.
- 4.a.6 System przekierowuje użytkownika na stronę zarządzania kalendarzem.

(IV) Zarządzaj materiałami:

Tabela 3.4. Zarządzaj materiałami

Aktorzy:	Użytkownik, Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Użytkownik chce dodać, usunąć, pobrać, lub udostępnić plik.
Zdarzenie wyzwalające	Użytkownik wciska przycisk “Zarządzaj materiałami”.
Warunki wstępne:	Użytkownik jest zalogowany.
Warunki końcowe dla sukcesu:	Użytkownik dodał, usunął, pobrał, lub udostępnił plik.
Warunki końcowe dla niepowodzenia:	Użytkownik pozostał na dotychczasowej stronie.

Scenariusz główny:

- 1. Użytkownik wciska przycisk “zarządzaj materiałami”.
- 2. System przekierowuje użytkownika na stronę zarządzania materiałami.
- 3. Użytkownik wciska przycisk “dodaj plik”.
- 4. System wyświetla nowe okno z eksplorerm plików w celu wybrania pliku z lokalnego dysku

użytkownika.

5. Użytkownik wybiera plik i zatwierdza wybór.

6. System sprawdza czy plik nie jest zbyt duży.

7. System dodaje plik do bazy i wyświetla informacje o pomyślnym dodaniu pliku do bazy.

8. Nowy plik pojawia się w oknie zarządzania materiałami.

Scenariusz alternatywny:

6.a. System wyświetla komunikat błędu: "Plik zbyt duży."

Scenariusz alternatywny:

1.a. System wyświetla komunikat "Błąd serwera".

1.a.1. Użytkownik pozostaje na dotychczasowej stronie.

Scenariusz alternatywny:

2.a Użytkownik wybiera plik z listy plików użytkownika.

2.a.1 System otwiera stronę zarządzania plikiem z informacją o wybranym pliku, oraz trzema przyciskami: "Usuń", "Pobierz" oraz "Udostępnij".

2.a.2 Użytkownik wybiera opcję "Usuń".

2.a.3. System wyświetla okno "Czy na pewno chcesz usunąć plik?".

2.a.4 Użytkownik wciska przycisk "tak".

2.a.5 Plik zostaje usunięty z systemu.

2.a.6 System przekierowuje użytkownika do strony zarządzania materiałami.

Scenariusz alternatywny:

2.a.3.a Użytkownik wybiera opcję "nie".

2.a.3.a.1 Powrót do pkt 2 scenariusza głównego.

(V) Udostępnij plik:

Tabela 3.5. Udostępnij plik

Aktorzy:	Użytkownik, Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Użytkownik chce udostępnić plik.
Zdarzenie wyzwalające	Użytkownik wciska przycisk "udostępnij plik".
Warunki wstępne:	Użytkownik jest zalogowany. Użytkownik jest na stronie zarządzania materiałami.
Warunki końcowe dla sukcesu:	Plik został udostępniony.
Warunki końcowe dla niepowodzenia:	Plik nie został udostępniony.

Scenariusz główny:

1. Użytkownik wybiera plik z listy.
2. System otwiera stronę zarządzania plikiem z informacją o wybranym pliku, oraz trzema przyciskami: “Usuń”, “Pobierz” oraz “Udostępnij”.
3. Użytkownik wybiera opcję “Udostępnij”.
4. System otwiera stronę udostępniania pliku, na której znajduje się formularz, z polem na nazwę użytkownika oraz przycisk “zatwierdź”.
5. Użytkownik wpisuje nazwę użytkownika i wciska przycisk “zatwierdź”.
6. System po znalezieniu użytkownika w bazie udostępnia plik wybranemu użytkownikowi.
7. System wyświetla wiadomość o pomyślnym udostępnieniu pliku.

Scenariusz alternatywny:

- 5.a. System nie znajduje użytkownika w bazie.
- 5.a.1. System wyświetla komunikat o niepowodzeniu.

(VI) Pobierz plik:

Tabela 3.6. Pobierz plik

Aktorzy:	Użytkownik, Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Użytkownik chce pobrać plik.
Zdarzenie wyzwalające	Użytkownik wciska przycisk “pobierz plik”.
Warunki wstępne:	Użytkownik jest zalogowany. Użytkownik jest na stronie zarządzania materiałami.
Warunki końcowe dla sukcesu:	Plik został pobrany.
Warunki końcowe dla niepowodzenia:	Plik nie został pobrany.

Scenariusz główny:

1. Użytkownik wybiera plik z listy.
2. System otwiera stronę zarządzania plikiem z informacją o wybranym pliku, oraz trzema przyciskami: “Usuń”, “Pobierz” oraz “Udostępnij”.
3. Użytkownik wybiera opcję “Pobierz”.
4. System rozpoczyna pobieranie pliku do domyślnego folderu na lokalnej maszynie użytkownika.
5. System zakończył pobieranie pliku i wyświetla komunikat o powodzeniu.

Scenariusz alternatywny:

- 4.a. System nie może pobrać pliku, ponieważ nastąpiło rozłączenie z serwerem.
- 4.a.1. System wyświetla komunikat o niepowodzeniu.

(VII) Zobacz powiadomienia: **Scenariusz główny:**

Tabela 3.7. Zobacz powiadomienia

Aktorzy:	Użytkownik, Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Użytkownik chce zobaczyć powiadomienia.
Zdarzenie wyzwalające	Użytkownik wciska przycisk “Zobaczpowiadomienia”.
Warunki wstępne:	Użytkownik jest zalogowany.
Warunki końcowe dla sukcesu:	Użytkownik znajduje się na stronie powiadomień.
Warunki końcowe dla niepowodzenia:	Użytkownik pozostał na dotychczasowej stronie.

1. Użytkownik wciska przycisk “Zobacz powiadomienia”.
2. System otwiera stronę powiadomień.

Scenariusz alternatywny:

- 1.a. System wyświetla błąd.
- 1.a.1. Użytkownik pozostaje na dotychczasowej stronie.

(VIII) Zarejestruj:

Scenariusz główny:

1. System wyświetla formularz rejestracyjny.
2. Użytkownik wypełnia formularz.
3. Użytkownik wciska zatwierdź.
4. System sprawdza poprawność danych i unikalność loginu.
5. Konto użytkownika zostaje utworzone

Scenariusz alternatywny: 4.a System znajduje konto o podanym loginie.

- 4.a.1 System wyświetla ponownie formularz z informacją, że konto o podanym loginie już istnieje.
- 4.a.2 Powrót do punktu 2 scenariusza głównego.

Scenariusz alternatywny:

- 4.b System wykrył polskie znaki w hasle.
- 4.b.1 System wyświetla ponownie formularz z informacją, że nie może być polskich znaków w hasle
- 4.b.2 Powrót do punktu 2 scenariusza głównego.

Tabela 3.8. Zarejestruj

Aktorzy:	Użytkownik, Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Niezarejestrowany Użytkownik chce się zarejestrować.
Zdarzenie wyzwajające	Niezarejestrowany Użytkownik wciska przycisk "Zarejestruj".
Warunki wstępne:	Brak.
Warunki końcowe dla sukcesu:	Niezarejestrowany Użytkownik znajduje się na stronie głównej.
Warunki końcowe dla niepowodzenia:	Konto nie zostaje stworzone , a użytkownik zostaje poinformowany o niepowodzeniu oraz podane zostają przyczyny błędu np. adres email jest już wykorzystany lub login jest już zajęty.

Scenariusz alternatywny: 2.a Użytkownik odświeża stronę.

2.a.1 Użytkownik przekierowany jest na stronę główną serwisu.

(IX) Zaloguj:

Tabela 3.9. Zaloguj

Aktorzy:	Użytkownik, Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Użytkownik chce się zalogować.
Zdarzenie wyzwajające	Użytkownik wciska przycisk zaloguj.
Warunki wstępne:	Użytkownik musi posiadać konto w systemie.
Warunki końcowe dla sukcesu:	Użytkownik znajduje się na stronie głównej.
Warunki końcowe dla niepowodzenia:	Użytkownik jest niezalogowany.

Scenariusz główny:

1. System wyświetla formularz logowania.
2. Użytkownik wypełnia formularz.
3. Użytkownik wciska zatwierdź.
4. System sprawdza poprawność danych.

5. Użytkownik zostaje zalogowany.

Scenariusz alternatywny:

4.a System odrzuca wprowadzone dane, ponieważ login lub hasło są nieprawidłowe.

4.a.1 System wyświetla ponownie formularz z informacją o błędnym loginie lub hasle.

4.a.2 Powrót do punktu 2 scenariusza głównego.

(X) Zarządzaj kontami użytkowników:

Tabela 3.10. Zarządzaj kontami użytkowników

Aktorzy:	Administrator
Zakres:	System
Poziom:	Sytemowowy
Udziałowcy i ich cele:	Administrator chce usunąć, lub edytować konto użytkownika.
Zdarzenie wyzwajające	Administrator wciska przycisk “zarządzaj kontami”.
Warunki wstępne:	Administrator jest zalogowany.
Warunki końcowe dla sukcesu:	Konto użytkownika zostało usunięte lub edytowane.
Warunki końcowe dla niepowodzenia:	Konto użytkownika nie zostało usunięte lub edytowane.

Scenariusz główny:

1. Administrator wciska przycisk “zarządzaj kontami”.

2. System przekierowuje na stronę zarządzania kontami użytkowników, gdzie wylistowane są loginy wszystkich użytkowników, a obok nazwy są dwa przyciski: “usuń” oraz “edytuj” oraz bad listą pole na wpisanie loginu i przycisk do zatwierdzania.

3. Administrator klika “usuń” przy wybranym loginie. 4. System wyświetla komunikat o treści “Czy na pewno chcesz usunąć użytkownika z bazy danych?” oraz dwie opcje odpowiedzi: “tak” i “nie”.

5. Administrator wciska “tak”

6. System usuwa użytkownika z bazy danych.

7. System wyświetla komunikat o udanej operacji.

Scenariusz alternatywny:

2.a Administrator wciska “edytuj”.

2.a.1 System przekierowuje administratora na stronę edycji użytkownika z formularzem do edycji.

2.a.2 Administrator wprowadza nowe dane i potwierdza zmianę przyciskiem “zatwierdź”.

2.a.3 System zmienia dane użytkownika w bazie.

2.a.4 System wyświetla komunikat o udanej operacji.

2.a.5 Powrót do pkt 2 scenariusza głównego.

Scenariusz alternatywny:

2.b Administrator wpisuje login w odpowiednie pole i wciska przycisk szukaj

2.b.1 System wyświetla tylko użytkownika o podanym loginie.

3.3. Model systemu

nad tym tytułem pomyslec!

3.3.1. Architektura systemu

- najpierw napisac ze soa
- zastanowic sie czy tutaj opis soa czy we wstepie, ale chyba we wstepie, tutaj jedynie można wspomiec ze wcześniej opisany w punkcie X.x.
- Diagram
- byc moze opis

3.3.2. Model bazy danych

- tutaj jakis ladny wstep.
- MODEL ERD - ostateczny

3.3.3. Model klas

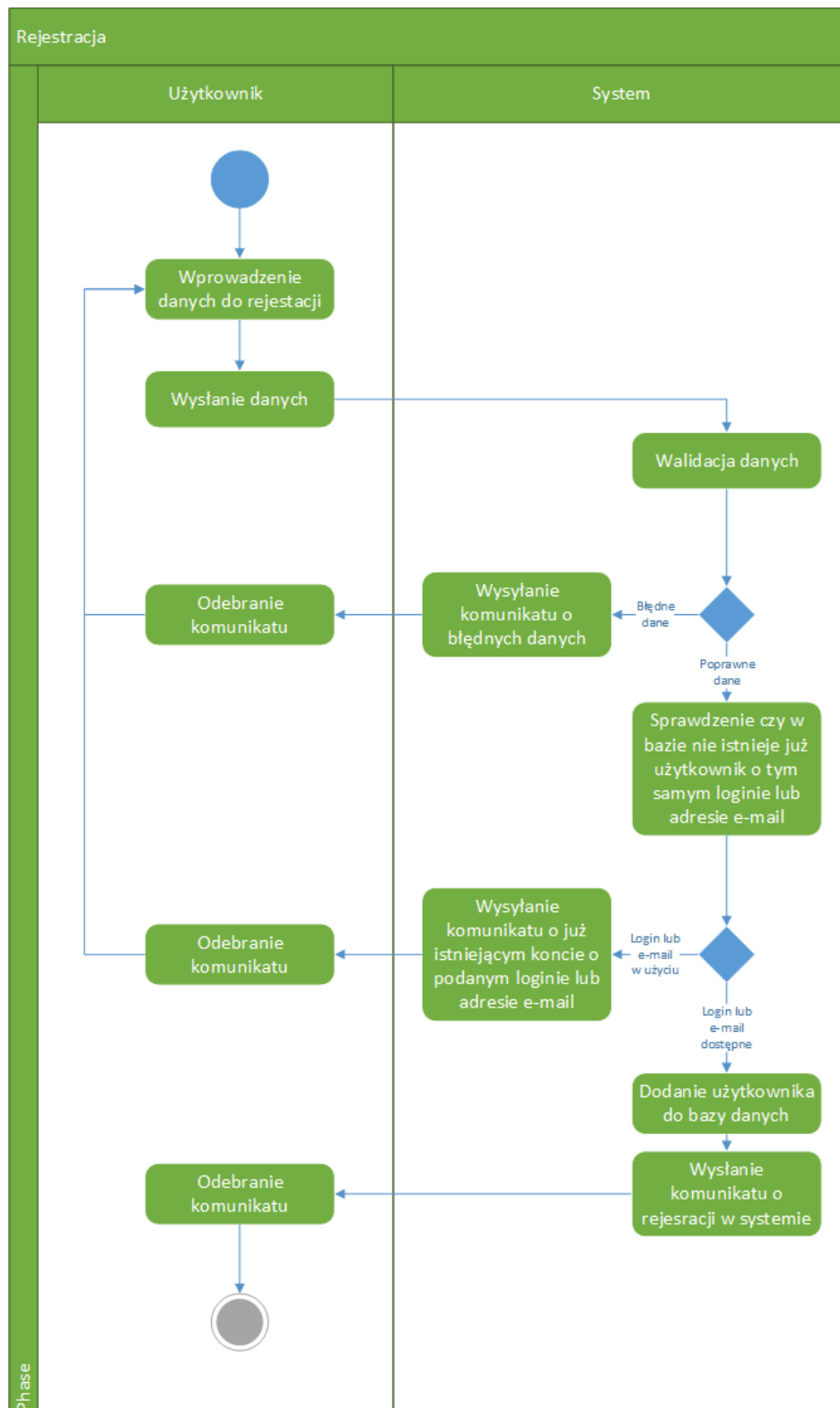
- rowniez jakis ladny wstep
- No i diagram klas tutaj ładnie sklepać trzeba będzie

3.3.4. Diagramy sekwencji

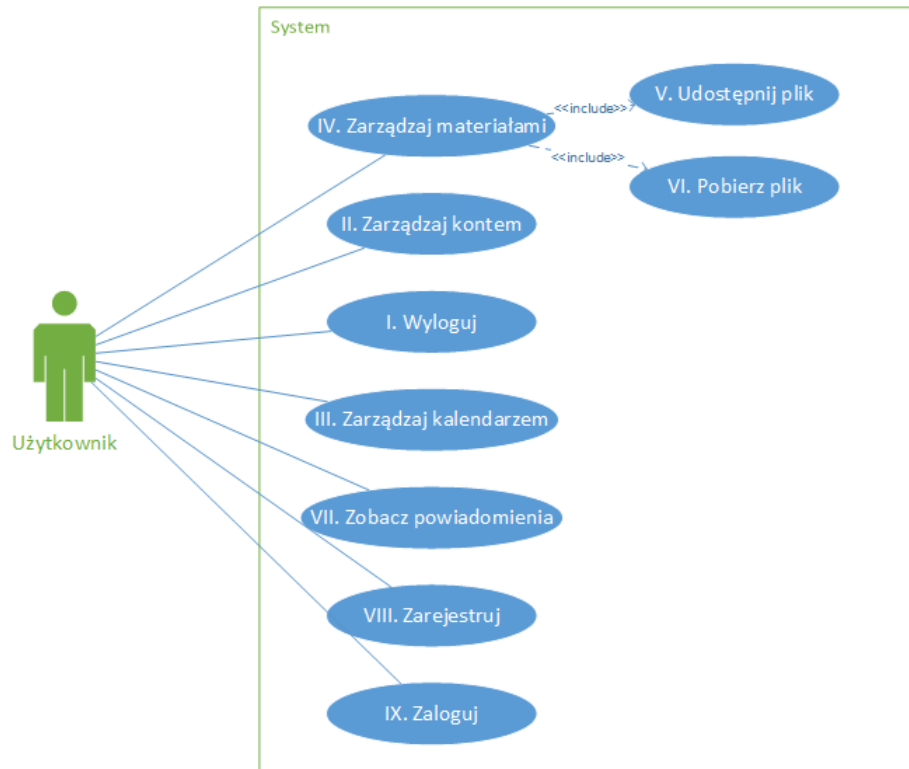
- Wstep, ze to wynika z diagramu klas, cos ładnego o diagramach sekwencji - tutaj moze sie wykładami szweda posłużyć
- No i wklepac diagramy sekwencji - moze nie bedzie tak zle jak już beda diagramy klas

3.4. Projekt interfejsu?

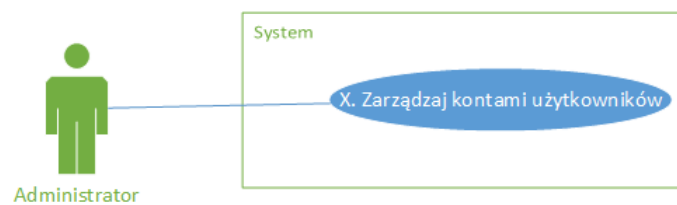
Nad tym sie zastanowic czy to robic czy dac dupie siana



Rys. 3.1. Rejestracja



Rys. 3.2. Diagram przypadków użycia dla użytkownika



Rys. 3.3. Diagram przypadków użycia dla administratora

4. Implementacja

Kilka zdań dotyczących zawartości poszczególnych podrozdziałów

4.1. Architektura

Opis warstw systemu, rozmieszczenie geograficzne oprogramowania
tutaj pewnie trzeba będzie napisać że to na mvc
no i że podzielone przez ejb na war, ejb no ten końcowy

4.1.1. Dostęp do bazy danych

tutaj subsekcje w jednej dostęp do danych dao i hibernate

4.1.2. Widok aplikacji

te dwa poniżej może w dodatkowych podsekcjach JSF
przykładowe formularze

4.2. Zabezpieczenia

tutaj coś napisać o tych zabezpieczeniach co u rogiusa bazgrałem

4.3. Testy

To chyba można tutaj bo tego nie będzie dużo i nie ma co osobnego rozdziału
Może napisze jakieś junity to będzie coś
można napisać o planach testów manualnych itd

5. Wyniki i podręcznik użytkownika

Albo to upchnąć w implementacji albo jak się wyrobie to uda się jakiś osobny rozdział

5.1. No idea

6. Podsumowanie

W podsumowaniu pracy należy zebrać wnioski z jej realizacji. Odpowiedzieć na pytania: czy cel pracy został osiągnięty i w jakim stopniu. Co inaczej byłoby realizowane, gdyby autor od nowa zaczął tę pracę? Można tu podać trochę ciekawostek z jej realizacji. Kto i jaki pożytek może mieć z tej pracy? Czy warto kontynuować tą pracę i w jaki sposób? Jakie nowe problemy zostały zidentyfikowane i które z nich mogą być przedmiotem kolejnych prac dyplomowych? Podsumowanie nie ma na celu wykazywać, że stworzony produkt jest idealny i bezbłędny, lecz to, że autor jest rzetelnym i kompetentnym projektantem i analitykiem. Do błędów popełnionych w pracy należy się uczciwie przyznać.

Tutaj bez podrozdziałów.

Bibliografia

- [1] Claus T. Jensen. *SOA Design Principles for Dummies*. New Jersey: John Wiley and Sons, Inc., 2013.
- [2] *The History of Java Technology*.
<http://www.oracle.com/technetwork/java/javase/overview/javahistory-index198355.html>. Oracle.
- [3] *JEE - wprowadzenie do Javy Enterprise Edition*.
<http://javastart.pl/static/jee/jeewprowadzeniedojavaenterprise-edition/>.
- [4] R. Matusik. *Serwery aplikacji*.
http://math.uni.lodz.pl/radmat/serwery/Wyklad_d_01.pdf. Uniwersytet Łódzki.
- [5] P. Szwed. *Szablon dokumentacji projektowej*.
http://home.agh.edu.pl/pszwed/wiki/doku.php?id=amo:rup_tailored. Akademia GórniczoHutnicza. 2015.
- [6] K. Zmitrowicz. *Analiza wymagań funkcjonalnych*.
<http://wymagania.net/bazawiedzy/36bazawiedzy/88analizawymagan-funkcjonalnych>.
- [7] H. Wesołowska. *Jak opisywać przypadki użycia?*
<http://analizait.pl/2012/jakopisywacprzypadkiuzycia/>. 2012.