



Zadanie 6. Zastosowanie technologii informatycznych w medycynie

Sprawozdanie

z wykonania pracy w ramach umowy o dzieło p.t.:

Przygotowanie środowiska deweloperskiego i opracowanie konfiguracji serwera aplikacyjnego JBoss w ramach repozytorium obrazów i sygnałów w projekcie Biocentrum.

Opracowanie mechanizmów autentykacji i autoryzacji oraz projekt ekranów logowania i głównego ekranu aplikacji w ramach repozytorium obrazów i sygnałów w projekcie Biocentrum.

Implementacja modułu testowego platformy wraz ze wstępną wizualizacją wirtualnych preparatów w ramach repozytorium obrazów i sygnałów w projekcie Biocentrum.

z dnia

02.11.2011

termin wykonania
dzieła:

20.12.2011

Wykonawca dzieła:

mgr Grzegorz Barański
mgr Przemysław Chmielewski
mgr Radosław Chmielewski

Przygotowanie środowiska deweloperskiego i opracowanie konfiguracji serwera aplikacyjnego JBoss w ramach repozytorium obrazów i sygnałów w projekcie Biocentrum.

1. Środowisko deweloperskie

1.1 Narzędzia developerskie

Na serwerze zostało zainstalowane następujące oprogramowanie:

1. Visual SVN
2. Trac
3. Apache Archiva

Repozytorium Visual SVN dla projektu Biocentrum znajduje się w katalogu D:\svn\svnRepositories\BioCentrum i można z niego korzystać poprzez adres <https://10.0.10.253/svn/BioCentrum>.

Trac jest systemem do zarządzania projektami i zgłaszania błędów. Lokalizacja plików instalacyjnych i konfiguracyjnych dla Trac znajduje się odpowiednio w katalogach

D:\BioC\BitNamiTracStack i D:\BioC\dev\). Aby umożliwić zarządzanie użytkownikami dla systemu Trac został zainstalowany plugin Trac Account Manager w wersji 0.3.2, za pomocą którego można zarządzać użytkownikami, grupami a także uprawnieniami dla nich.

Do konfiguracji projektu biorepozytorium w systemie Trac zostało dodane repozytorium kodów z Visual SVN, aby umożliwić przeglądanie plików z poziomu systemu Trac. Adres, pod którym dostępny jest system Trac to: <http://10.0.10.253:8380/trac/BioRepozytorium/>

Apache Archiva to repozytorium lokalne dla systemu budowania Apache Maven. Jest ono dostępne pod adresem <http://10.0.10.253:8180/archiva/>. Użytkownik dla systemu Archiwa to: admin i hasło: biorepo11.

1.2 Narzędzia deweloperskie dla programisty

Aby skonfigurować środowisko deweloperskie należy pobrać następujące oprogramowanie:

1. Oracle Java SDK w wersji 1.6_29
2. Klient SVN np. subversion
3. Eclipse IDE for Java Developers
4. Apache Maven w wersji 3.0
5. PostgreSQL w wersji 9.1
6. serwer JBoss AS 7.0.2.GA

Po zainstalowaniu Java SDK należy dodać zmienną systemową JAVA_HOME zgodnie z wybranym katalogiem instalacji, a następnie dopisać zmienną do ścieżki systemowej.

Do korzystania z repozytorium Visual SVN wymagany jest klient SVN np. w postaci programu Subversion.

Instalacja Eclipse IDE ogranicza się do rozpakowania pobranego archiwum zip/tar. Następnie należy zainstalować następujące pluginy:

1. Subclipse lub Subversive
2. Google Plugin for Eclipse

Pluginy można zainstalować z poziomu Eclipse Marketplace lub poprzez dodanie lokalizacji pluginu w Available Software Site.

Należy się upewnić czy wybrany plugin do korzystania z repozytorium subversion jest zgodny z zainstalowanym klientem SVN. Jeżeli jako klient SVN korzystamy z subversion

1.6.x to zgodnym pluginem jest Subclipse w wersji 1.6.x. Dla subversion 1.7.x należy zainstalować plugin Subclipse w wersji 1.8.x.

Instalacja Apache Maven polega na rozpakowaniu pobranego archiwum, a następnie dodaniu zmiennej systemowej M2_HOME i dopisanie jej do ścieżki systemowej w postaci M2_HOME/bin.

Po zainstalowaniu serwera bazy danych PostgreSQL należy utworzyć bazę danych biorepo oraz rolę biorepo. Skrypty tworzące schemat bazy danych znajdują się w kodach źródłowych w module biorepo-sql.

Instalacja serwera JBoss AS 7 sprowadza się do rozpakowania pobranego archiwum ze strony jboss.org. Następnie należy wykonać poniższe kroki w celu konfiguracji serwera:

1. Moduł PostgreSQL JDBC Driver

W katalogu modules, który znajduje się w katalogu instalacyjnym JBoss'a należy założyć strukturę org/postgresql/main, a następnie skopiować zawartość katalogu postgresql z lokalizacji trunk/env/jboss/modules w repozytorium SVN.

2. Moduł sun.jdk

Należy skopiować zawartość katalogu sun.jdk z lokalizacji trunk/env/jboss/modules/sun.jdk do modules/sun/sdk/main w katalogu instalacyjnym JBoss AS 7

3. Pozostałe pliki konfiguracyjne

Należy skopiować plik standalone.xml lokalizacji trunk/env/jboss/ do katalogu instalacyjnego JBoss'a w standalone oraz plik standalone.conf z trunk/env/jboss/env/ do katalogu bin w katalogu instalacji serwera Jboss.

W pliku standalone.xml zostały skonfigurowane parametry dla serwera JBoss AS 7. Są to:

1. Źródło danych (datasource)

Konfiguracja parametrów połączenia do bazy danych polega na modyfikacji parametrów w tagach xml dotyczących źródła danych biorepoDS.

Domyślna konfiguracja:

- ⑩ host: localhost
- ⑩ port: 5432
- ⑩ nazwa bazy danych: biorepo
- ⑩ użytkownik bazy danych: biorepo
- ⑩ hasło użytkownika bazy danych: biorepo

2. Security domain

Została utworzona w pliku konfiguracyjnym standalone.xml.

security-domain: biorepo.

Konfiguracja zawarta w powyższym pliku jest oparta o strukturę uprawnień zawartej w bazie danych.

4. Środowisko djabatoka

Konfiguracja skryptu startowego standalone.conf umożliwiającą korzystanie z silnika djabatoka zamieszczona została w lokalizacji trunk/env/pliki/bin. Należy podmienić plik w katalogu bin w lokalizacji instalacji serwera aplikacyjnego jboss.

Skrypt startowy start.sh umożliwia uruchomienie serwera aplikacyjnego jbossa z odpowiednimi parametrami runtime'owymi. Zawiera on konfiguracje zmiennych środowiskowych, które należy odpowiednio ustawić:

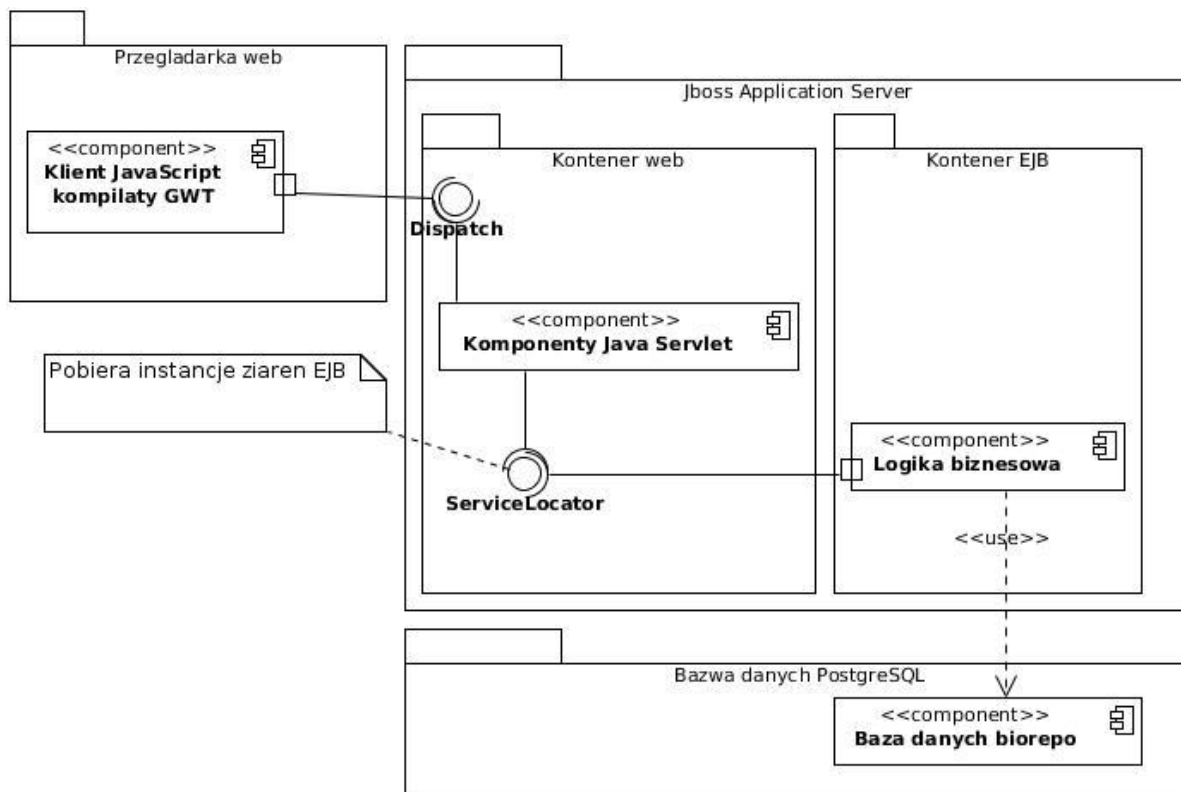
- JAVA_HOME - katalog domowy Java
- JBOSS_HOME - katalog domowy serwera JBoss
- DJATOKA_HOME - katalog domowy djabatoka

W lokalizacji trunk/env/djabatoka/bin znajduje się poprawiony i dostosowany skrypt env.sh. Należy go podmienić w lokalizacji DJATOKA_HOME/bin.

Opracowanie mechanizmów autentykacji i autoryzacji oraz projekt ekranów logowania i głównego ekranu aplikacji w ramach repozytorium obrazów i sygnałów w projekcie Biocentrum.

Wstęp do architektury.

Poniższym diagram prezentuje poglądowo architekturę systemu w ujęciu rozlokowania komponentów.



1. Warstwa klienta.

Interfejs użytkownika został opracowany z zachowaniem dzisiejszych standardów budowy interfejsu użytkownika aplikacji typu web. Do implementacji aplikacji użyty został framework Google Web Toolkit (GWT), co pozwala zapewnić monolityczne środowisko deweloperskie.

Aby zapewnić elastyczną budowę systemu zostały wykorzystane wzorce projektowe używane do budowy dużych systemów. Komponenty systemu budowane są w oparciu o wzorec Model-View-Preznter (MVP). Komunikacja między komponentami systemu realizowana jest przez wzorec EventBus zapewniając tym samym przejrzystą implementację poszczególnych elementów.

Komunikacja klienta (klienta Javascriptowego) z częścią serwerową (Java Servlet) zrealizowana jest używając wzorca Dispatch ułatwiająca komunikację RPC. Zapewniło to przejrzystą implementację i wysoką kohezję klas realizujących akcje po stronie serwera webowego.

Architektura części klienckiej, zarówno Javascriptowa oraz Java Servlet, wykorzystuje paradygmat Inversion of control realizowany przez Dependency Injection.

Do implementacji części klienckiej wykorzystana została biblioteka GWT Platform zapewniająca zręby implementacji powyższych aspektów architektury.

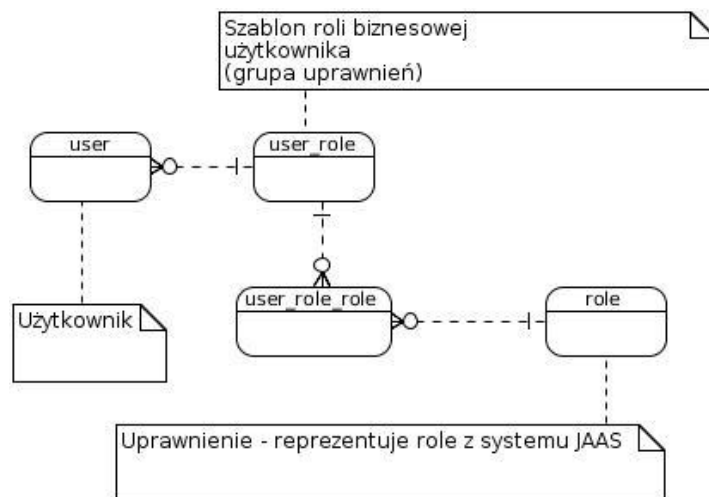
2. Warstwa logiki biznesowej.

Logika biznesowa realizowana jest przy wykorzystaniu Java Enterprise Edition 6. W celu zapewnienia komunikacji komponentów klienta web z warstwą logiki biznesowej opracowany został komponent ServiceLocator pozwalający pobrać instancję ziarna obiektu biznesowego zapewniając przy tym propagację sesji użytkownika na warstwę komponentów biznesowych.

3. Autentykacja i autoryzacja.

Autentykacja i autoryzacja zrealizowana została przy wykorzystaniu mechanizmu Java Authentication and Authorization Service.

Schemat ról został rozszerzony przez dodanie szablonu użytkownika – agregatu dla podstawowego pojęcia roli w nomenklaturze JAAS. Prezentacja na poniższym diagramie. W bazie danych poniższe tabele zostały umieszczone w schemacie security.



Po stronie klienta webowego wykorzystany został mechanizm Gatekeeper dostarczony przez framework GWT Platform. Autentykacja użytkownika jest sprawdzana przez klasę LoggedInGatekeeper. W przypadku negatywnej próby autentykacji prezentowana jest strona logowania. Po poprawnym zalogowaniu po stronie klienta tworzona jest sesja – implementacja w klasie Session.

Autoryzacja chronionych komponentów systemu, servletów oraz metod logiki biznesowej, realizowana jest przez standardowe mechanizmy platformy Java EE 6 sprawdzenia roli użytkownika. Odpowiednio dla servletów – strefa security constraint w deskrytorze aplikacji webowej web.xml, dla metod logiki biznesowej anotacja @RolesAllowed na metodzie. Servlety wymagające weryfikacji dostępu umieszczone są w ścieżce /secured/.*.

4. Modularyzacja.

Modularyzacja zrealizowana została przez użycie kontenerów Dependency Injection, Gin – dla warstwy klienckiej wykonywanej po stronie przeglądarki, Guice – dla części web serwera. Zostały wyróżnione moduły: kartoteka, repozytorium, auth jako moduł agregujący funkcjonalność autentykacji użytkownika.

Klasy definiujące moduł np. dla modułu kartoteka to:

- KartotekaGinjector – opisująca interfejs modułu
- KartotekaClientModule – opisująca komponenty modułu dla części klienckiej
- KartotekaServerModule – opisująca komponenty modułu dla części serwerowej

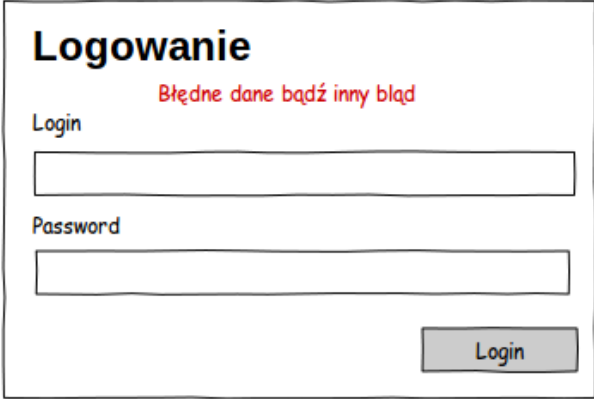
Moduły wpinane są do systemu przez odpowiednie deklaracje.

Cześć kliencką deklaruje się w interfejsie BiorepoGinjector rozszerzając go o interfejs modułu np. KartotekaGinjector oraz deklarując klasę np. KartotekaClientModule w anotacji @GinModules.

Cześć serwerową deklaruje się w klasie BiorepoGuiceServletContextListener dopisując instancję klasy np. KartotekaServerModule w metodzie getModules().

5. Interfejs użytkownika.

5.1 Ekran logowania.



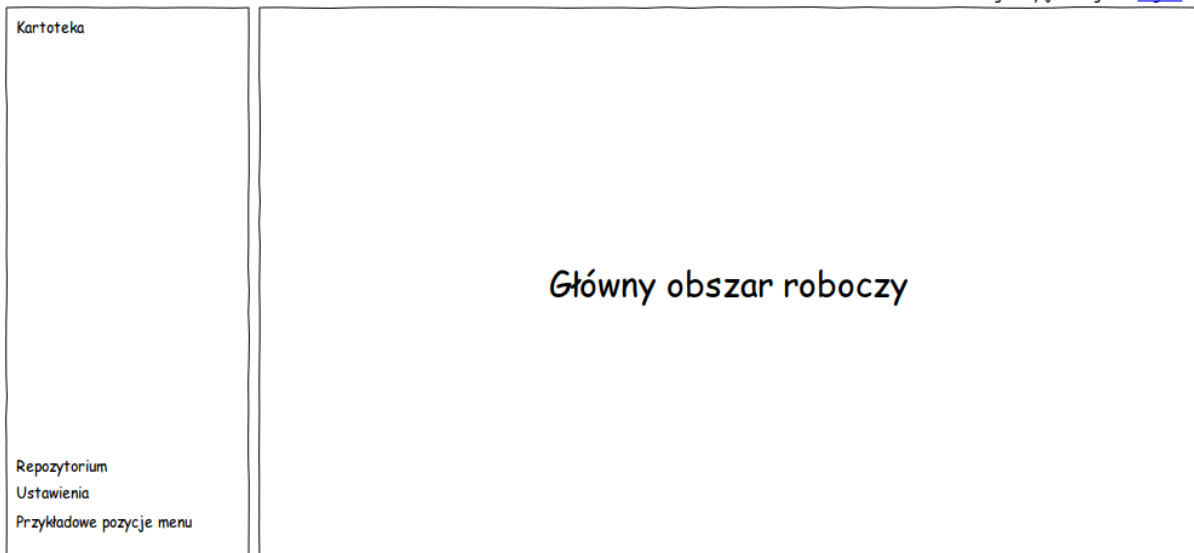
The image shows a login form with the following elements:

- Title: **Logowanie**
- Error message: **Błędne dane bądź inny błąd** (in red)
- Input field: **Login**
- Input field: **Password**
- Button: **Login**

Projekt ekranu logowania został przedstawiony poniżej.

5.2 Główny ekran systemu.

Główny ekran został zaprojektowany tak, aby zmaksymalizować wielkość obszaru roboczego. Menu główne aplikacji posiada możliwość ukrycia do lewej strony. Projekt wg poniższej grafiki.



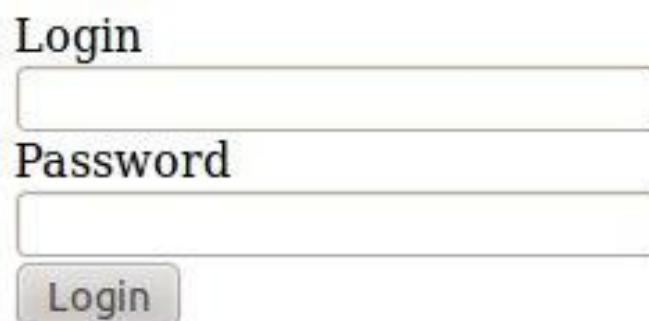
Implementacja modułu testowego platformy wraz ze wstępną wizualizacją wirtualnych preparatów w ramach repozytorium obrazów i sygnałów w projekcie Biocentrum.

1. Wstęp do implementacji modułu testowego platformy

Celem pracy jest wykonanie modułu testowego do wstępnej wizualizacji wirtualnych preparatów, wykorzystując poszczególne elementów platformy opisane powyżej. Kody źródłowe modułu zostały umieszczone na serwerze svn <https://10.0.10.253/svn/BioCentrum>. Wykonaną aplikację uruchomiono <http://10.0.10.253:8080/biorepo>.

2. Ekran logowania

Okno logowania do systemu. Umożliwia wpisanie loginu i hasła, następnie uruchomienie akcji autoryzacji wybierając przycisk **Login**



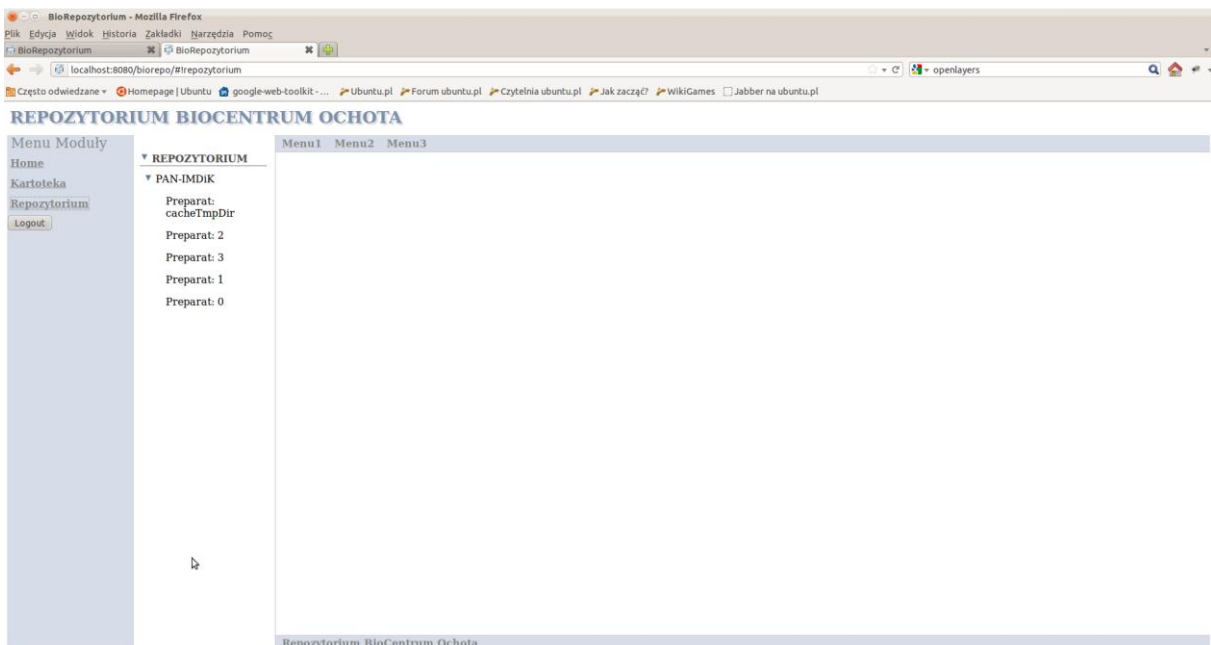
3. Okno główne

Główne okno aplikacji przedstawione poniżej. Zawiera po lewej stronie listę modułów dodanych do systemu.



4. Okno repozytorium

Poniższe okno jest głównym ekranem modułu testowego. Prezentującym w formie drzewa listę wirtualnych preparatów dostępnych w repozytorium.



5. Wizualizacja preparatu

Poniższy ekran przedstawia wstępną wizualizację wirtualnego składu się on z następujących elementów:

1. Obszar prezentacji preparatu
2. Pasek narzędzi powiększania i nawigacji preparatem
3. Pasek dostępnych warstw
4. Pasek narzędziowy rysowania powiązany z warstwą wektorową (**Grafika**)
5. Oznaczenie punktu na wirtualnym preparacie powiązany z warstwą (**Markers**)
6. Pasek informacyjny prezentuje informacje o pozycji kursora.



6. Podsumowanie

Wykonane prace związane z utworzenie środowiska deweloperskim i implementacją platformy, jak również modułu testowego. Pozwoliły na połączenie poszczególnych komponentów softwarowych i stworzenie środowiska do projektowania i implementacji modułów funkcyjnych w ramach repozytorium obrazów i sygnałów w projekcie Biocentrum.

Podpis Wykonawcy dzieła

Stwierdzam wykonanie pracy zgodnie z umową

data

Podpis przyjmującego pracę