

Politechnika Poznańska
Wydział Informatyki
Instytut Informatyki

Praca dyplomowa magisterska

**Bioserwer: administracja systemem i
wizualizacja danych**

Tomasz Tokarek 84892

Poznań, 2013

Promotor

prof. dr hab. inż. Jacek Błażewicz
dr inż. Marta Szachniuk

Karta pracy

Spis treści

1. Wprowadzenie	1
1.1. Multimedia w Bioinformatyce	1
1.2. Cel i zakres pracy	2
1.3. Układ pracy oraz uwagi dotyczące realizacji tematu	3
1.4. Charakterystyka źródeł	3
2. Wymagania i narzędzia projektowe	5
2.1. Technologie wykorzystane	5
2.1.1. Flex 4.0	5
2.1.2. PHP	6
2.1.3. XML	7
2.1.4. HTML	7
2.1.5. Java	7
2.1.6. Java Script	7
2.1.7. System Zarządzania Bazą Danych	8
2.2. Użyte narzędzia programistyczne	8
2.2.1. Google Analytics	8
2.2.2. Przeglądarka Lynx 2.8.7	9
2.2.3. Adobe Flash Builder 4.0	10
2.2.4. Apache Open Meetings 2.0.0	11
2.2.5. Notepad++ 5.9	12
2.2.6. WinSCP 4.2.9	13
2.2.7. Git / GitHub 1.1.1	13
2.2.8. VLC Media Player 2.0.3	14
2.2.9. SWFAddress 2.4	15
2.2.10. IpAdapter	15
2.3. Opis istniejącego systemu	15
2.4. Specyfikacja Wymagań	16
2.4.1. Przypadki użycia	17
2.4.2. Wymagania Pozafunkcjonalne	18
2.5. Schemat bazy danych	19
3. Część I: Zarządzanie Bioserwerem	21
3.1. Problematyka na styku Google / Bioserwer	21
3.1.1. Treść strony	21
3.1.2. Nawigacja	22

3.1.3. <i>Google - PageRank</i>	23
3.2. Administracja serwera - role użytkowników	24
3.2.1. Drzewo edycji - algorytm działania	25
3.3. Edytor stopki	26
4. Część II: Teleseminaria	27
4.1. Wprowadzenie	27
4.2. Wybór kamery	28
4.3. Wybór oprogramowania	29
4.3.1. <i>Adobe Flash Server</i>	29
4.3.2. <i>RED5</i>	29
4.3.3. <i>ZoneMinder</i>	30
4.4. Praca z OpenMeetings	31
4.5. Część Bioserwerowa - Video w czasie rzeczywistym	33
4.6. Część Bioserwerowa - Archiwum Teleseminariów	34
4.7. Wyniki końcowe	36
5. Podsumowanie	37
A. Załączniki	39
A.1. Słownik	39
A.2. Zestawienie kamer USB	40
Bibliografia	47

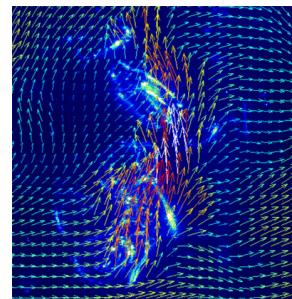
1. Wprowadzenie

1.1. Multimedia w Bioinformatyce

Komunikacja w Internecie pozwala na sprawną wymianę informacji pomiędzy różnymi miejscami na świecie niemal natychmiastowo. Zmienia to sposób postrzegania rzeczywistości przez ludzi, sprawiając że szybkość i jakość udostępniania wiadomości ma krytyczne znaczenie dla wizerunku firmy, organizacji, czy nawet pojedynczej osoby. Strona internetowa jest dziś wizytówką, bramą wizerunkową, od zawartości której zależy postrzeganie jej właściciela. Dlatego jej wygląd, treść, sposób obsługi mają ogromne znaczenie.

Szeroko rozumiane multimedia są dziś najlepszą formą dotarcia do odbiorcy. Połączenie dźwięku, obrazu i tekstu pozwala o wiele skuteczniej przekazać wiedzę niż każdy z tych elementów osobno, co wiemy m.in. dzięki badaniom mózgu i psychiki ludzkiej. Z racji tego że człowiek przyswaja wiedzę najlepiej odbierając informacje wszystkimi zmysłami, suchy tekst nie uczy tak dobrze jak ilustracja, zaś ilustracja jest gorszym nauczycielem niż animacja. Dlatego w ciągu ostatnich 50 lat w biotechnologii i fizyce biomolekularnej możemy obserwować trend zintensyfikowania użycia technologii multimedialnych celem odwzorowania zjawisk zachodzących w świecie rzeczywistym. Używa się ich aby przedstawić mikroskopię, obrazowanie rezonansu magnetycznego, dynamikę molekularną. Pomagają one również w analizie zależności w drzewach filogenetycznych, ewolucji genomu i jego asemblacji. Wykorzystuje się je również po to by ułatwić odbiór i analizę informacji zgromadzonych w trakcie eksperymentów, dzięki czemu proces edukacji przebiega sprawniej. Obserwowanie zmian, które w laboratorium trwałyby kilka dni i wymagałyby sprzętu wartego miliony można urzeczywistnić o wiele mniejszym nakładem środków.

Dlatego najlepszym sposobem na przekazanie najnowszych informacji z dziedziny bioinformatyki jest wykład w trakcie którego można wyjaśnić skomplikowane zagadnienia zgromadzonemu audytorium, posługując się przy tym multimedialnymi pomo-

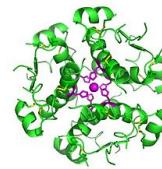


Rys. 1.1: Pole wektorów w płynie - przykład użycia wizualizacji danych (źródło [MAT13])

cami. Jednak co zrobić w momencie gdy zainteresowane osoby nie mogą zjawić się na miejscu konferencji? Tutaj z pomocą przychodzi technologia i możliwość transmisjonowania wydarzenia w Internecie. Wykorzystując fakt tworzenia nowej strony (*Bioserwer*), Grupa Bioinformatyczna Politechniki Poznańskiej postanowiła wyjść naprzeciw oczekiwaniom swoich słuchaczy i ułatwić im odbiór informacji, umieszczając dodatkowy moduł: *Teleseminariów* i pozwalając uczestniczyć w spotkaniach także za pośrednictwem Internetu.

Strona jest jednak jedynie na tyle przydatna, na ile można do niej trafić z zewnątrz. Z tego powodu stronę *Bioserwera*, stworzonej za pomocą technologii *Flex*, należało uczynić przyjazną dla wyszukiwarek, dbając o ranking strony i możliwość jej katalogowania przez *crawlers* firm zewnętrznych. Dodano również szereg zmian mających zadbać o komfort pracy zarówno po stronie administracyjnej jak i użytkownika końcowego.

Realizacja tego projektu pozwoliła na polepszenie znajomości osadzania multimediów w stronach internetowych jak również technik administracji i konserwacji aplikacji internetowych. Biorąc pod uwagę trendy rozwoju oprogramowania, umiejętności te z pewnością zostaną wykorzystywane w przyszłych pracach.



Rys. 1.2: Wizualizacja cząsteczki insuliny (źródło [WIK11])

1.2. Cel i zakres pracy

Celem niniejszej pracy jest wzbogacenie istniejącej portalu internetowego Grupy Bioinformatycznej o elementy multimedialne oraz dodanie nowej funkcjonalności do modułu administrowania serwisem. Praca została podzielona na dwie części: pierwsza odpowiadająca za bezpośrednie zmiany na stronie, druga za stworzenie dodatkowego modułu *Teleseminariów*.

Pierwsza część pracy odpowiada za opisanie zmian w obrębie strony internetowej. Zmiany te rozszerzają funkcjonalność strony poprzez wprowadzenie systemu ról w panelu administracyjnego oraz edytowalną stopkę. Strona została również wyposażona w dodatkowe elementy ułatwiające jej katalogowanie przez światowe wyszukiwarki.

Część druga, *Teleseminaria* jest poświęcona modułowi odpowiadającemu za tworzenie i zarządzanie transmisjami video z seminariów dyplomowych Grupy Bioinformatycznej. System jest w stanie działać zarówno w trybie online jak i odtwarzać archiwalne nagrania. Praca opisuje sposób tworzenia systemu od momentu wyboru odpowiedniego sprzętu do tego typu działalności, przez napisanie/wybór odpowiedniego oprogramowania wspomagającego działanie systemu, aż po uruchomienie systemu na Politechnice Poznańskiej.

1.3. Układ pracy oraz uwagi dotyczące realizacji tematu

W trakcie realizacji pracy zostały wyznaczone następujące zadania:

- Porównanie kamer internetowych - budynek CW w którym odbywają się zajęcia seminaryjne nie jest wyposażony w odpowiedni sprzęt video, zatem należało dokonać wyboru stosownego urządzenia do zakupu
- *Teleseminaria* - stworzenie systemu do telekonferencji, zarówno części widocznej dla użytkownika, jak i administracji oraz oprogramowania nagrywającego (napisania skryptów i wybranie gotowych narzędzi)
- Ułatwienia dla wyszukiwarek - stworzenie podstrony która jest widoczna dla crawlerów wyszukiwarek internetowych, która umożliwia katalogowanie *Bioserwera*
- System ról w administracji - dodanie możliwości edycji praw dostępu do artykułów dla poszczególnych użytkowników w panelu administracyjnym systemu

1.4. Charakterystyka źródeł

W trakcie realizacji projektu korzystano z zewnętrznych źródeł opisujących użyte technologie, są to między innymi:

Adobe Flex

Podstawy technologii zostały przedstawione w: [JS10], zaś pomoc w pisaniu kodu źródłowego i rozwiązywaniu możliwych błędów zapewniła baza danych funkcji firmy *Adobe* dostępna pod adresem [ADO13].

PHP

Internetowa baza danych języka *PHP*, dostępna pod adresem [PHP13]

MySQL

MySQL - składnia i procedury ze strony [SQL13]

Technologie Multimedialne

Przy wprowadzaniu elementów multimedialnych w systemie posługowano się książką [Dom10]

2. Wymagania i narzędzia projektowe

2.1. Technologie wykorzystane

2.1.1. Flex 4.0

Stworzony przez *Adobe System*: *Adobe Flex* to zestaw technologii bazujących na *Adobe Flash*, które służą do tworzenia multiplatformowych programów typu *RIA* (*Rich Internet Application*). Termin *RIA* odnosi się do aplikacji internetowej która pobiera większość potrzebnych danych w momencie rozpoczęcia działania, a w trakcie pracy składa jedynie żądania o dane uzupełniające (w przeciwieństwie do czystych aplikacji *HTML*, pobierających cały zestaw danych przy każdym żądaniu).

Kod źródłowy aplikacji *Flex* (składający się z kodu w językach: *MXML*, *ActionScript* i *CSS*) jest komplikowany do postaci pliku bitowego *Flash* (plik *SWF*), który jest odpalany po stronie klienta przez *ActionScript Virtual Machine* znajdującej się w oprogramowaniu *Flash Player* lub *Adobe AIR*. W trakcie połączenia z bazami danych i innymi systemami, *Flex* stosuje różne sieciowe interfejsy, od *XML*, *JSON* i *SOAP* przez *Action Message Format* (*AMF*), aż po *Real Time Messaging Protocol* (*RTMP*). [JS10]

Graficzna część aplikacji *Flex* jest tworzona w języku *MXML*. Język ten został stworzony w 2004 roku przez firmę *Macromedia*, która obecnie jest częścią *Adobe Systems*. *MXML* oparty jest na językiem opartym na *XML*, przez co jest łatwo modyfikowalny. Oprócz możliwości podania położenia, język ten pozwala również określać zachowania elementów strony na zdarzenia, wyjątki i zmiany stanów aplikacji. Przykład kodu napisanego w *MXML* w Algorytmie 2.1

Algorytm 2.1 Przykład kodu *MXML*

```
<s:BorderContainer width="25%" height="100%" borderVisible="true"
    borderWeight="3" borderStyle="solid">
    <s:VGroup width="100%">
        <s:CheckBox id="chkbxPerson" label="Add/Edit Person" />
        <s:CheckBox id="chkbxPosts" label="Edit Posts" />
        <s:CheckBox id="chkbxLogos" label="Edit Logos" />
        <s:CheckBox id="chkbxAdm" label="Administration" />
    </s:VGroup>
</s:BorderContainer>
```

Kolejnym językiem używanym przez *Flex* jest *Action Script*. *Action Script* również został stworzony przez firmę *Macromedia*, na podstawie tzw. *ECMAScript* (standardu języka skryptowego z którego wywodzi się między innymi *Javascript*). Oznacza to, że *Action Script* ma składnię i semantykę podobną właśnie do *Javascriptu* oraz jest w pełni obiektowym językiem oprogramowania. Algorytm 2.2 przedstawia przykład kodu źródłowego *Action Script* w wersji 3.0 użytej w aplikacji:

Algorytm 2.2 Przykład kodu Action Script

```
private function treeLabel(item :XML) :String
{
    var node :XML = XML(item);
    if( node.localName() == "node" )
        return node.@nazwa;
    if( node.localName() == "podstrona" )
        return node.@id;
    if( node.localName() == "user" )
        return item.@login;

    return "error:unknown_name_of_the_node--"
        + node.localName().toString();
}
```

Ponieważ środowisko uruchomieniowe (*Adobe AIR* i *Flash Player*) programu napisanego we *Flex* może zostać uruchomione praktycznie w każdym systemie i przeglądarce, to aplikacje napisane w tym języku zyskują na interoperacyjności, co przekłada się na mniejsze koszty utrzymywania kodu źródłowego.

2.1.2. PHP

PHP jest obiektowym i skryptowym językiem programowania, uruchamianym po stronie serwerowej, dzięki czemu umożliwia przeprowadzenie bezpiecznych operacji w Internecie bez potrzeby przesyłania wrażliwych informacji przez sieć. Język powstał z inicjatywy Rasmusa Lerdorfa, a jego skrót rozwijany był jako: *Personal Home Page* (w domyśle *Tools*), później zdecydowano się na rekursywne rozwinięcie skrótu *PHP: Hypertext Preprocessor*.

Serwer interpretujący język *PHP* może zostać uruchomiony na prawie każdym systemie operacyjnym (*Windows*, *Linux*, *Mac OS*), zaś polecenia *PHP* można użyć zarówno w języku *HTML* jak i w osobnych plikach. *PHP* może również być używany jako warstwa pośrednicząca w trakcie:

- połączeń z systemem zarządzania bazą danych

2.1 Technologie wykorzystane

- wczytywania plików *XML*
- wgrywania plików na serwer

W pracy użyto wersji 5.3.15 języka *PHP*.

2.1.3. XML

XML (Extensible Markup Language) to prosty i bardzo elastyczny sposób formowania tekstu otrzymany z języka *SGML*. Składa się z zestawu reguł służących do kodowania dokumentów w formie czytelnej dla programów komputerowych. Głównymi założeniami *XML* jest prostota, uniwersalność i użytkowniczość. Obecnie odgrywa istotną rolę w procesie wymiany informacji drogą internetową.

2.1.4. HTML

HTML to język hipertekstowego znakowania (*Hypertext Markup Language*), który stanowi ustanowioną przez konsorcjum *World Wide Web Consortium (W3C)* specyfikację nakreślającą postać dokumentów prezentowanych w Internecie. Twórcą języka *HTML* jest Tim Berners-Lee. *HTML* pozwala publikować w Internecie dokumenty zawierające nagłówki, tekst, tabele, listy, zdjęcia, formularze, pobierać za pośrednictwem „połączeń hipertekstowych” informacje z Internetu, projektować formularze oraz umieszczać w dokumentach arkusze kalkulacyjne, klipy wideo oraz inne aplikacje. [Sok02]

2.1.5. Java

Java jest obiektowym językiem programowania stworzonym w 1995 roku przez *Sun Microsystem* (w grupie kierowanej przez Jamesa Goslinga). Język ten został przewidziany jako język multiplatformowy. Aplikacja *Java* jest kompilowana do kodu bajtowego, który następnie może zostać uruchomiony na każdej *Java Virtual Machine (JVM)* zainstalowanej na komputerze użytkownika końcowego.

Obecnie *Java* jest rozwijana przez *Oracle Corporation*

2.1.6. Java Script

Java Script to język skryptowy służący do rozwinięcia funkcjonaności języka *HTML* o bardziej dynamiczne elementy (prace z plikami, ruch obiektów, video i audio). *Java Script* powstał jako implementacja standardów *ECMA Script* (dostępnych na stronie: <http://www.ecmascript.org/>).

Cechą *JavaScript* jest to, że musi on działać wewnętrz środowiska. Najpopularniejszym środowiskiem jest przeglądarka, jednak istnieją inne możliwości. *JavaScript*

może działać na serwerze, na pulpicie lub wewnątrz tzw. *rich media*. Obecnie *JavaScript* pozwala tworzyć:

- Duże aplikacje sieciowe o bogatych interfejsach (aplikacje działające w środowisku sieciowym, takie jak *Gmail*).
- Kod po stronie serwera. Może to być kod przypominający skrypty *ASP* lub uruchamiany przy użyciu narzędzi takich jak *Rhino* (silnik *JavaScript* napisany w *Javie*).
- Aplikacje rich media (*Flash*, *Flex*). Tworzy się je przy użyciu *ActionScriptu*, który jest oparty o *ECMAScript*.
- Skrypty, które automatyzują zadania administracyjne na pulpicie w *Windows*. Wykorzystuje się do tego *Windows Scripting Host*.
- Rozszerzenia i wtyczki dla wielu samodzielnych aplikacji, takich jak *Firefox*, *Dreamweaver* czy *Fiddler*. [Ste10]

2.1.7. System Zarządzania Bazą Danych

System Zarządzania Bazą Danych (SZBD) jest zorganizowanym zbiorem narzędzi umożliwiającym dostęp do baz danych i zarządzanie nimi. Dzięki *SZBD* dostępne są takie operacje, jak: przechowywanie danych, tworzenie i utrzymywanie struktur danych, umożliwienie równoczesnego dostępu wielu użytkownikom, wprowadzanie mechanizmu bezpieczeństwa i prywatności, odzyskiwanie danych i operowanie na przechowywanych danych, wprowadzanie i ładowanie danych, udostępnienie wydajnych mechanizmów indeksowania pozwalających na szybkie znalezienie wybranych danych, zapewnienie spójności różnych rekordów, ochrona przechowywanych danych przed utratą za pomocą kopii bezpieczeństwa i procedur odtwarzania. Zaś sama baza danych to zbiór danych zorganizowanych w pewien logiczny i zestrukturalizowany sposób[Cha01]

W projekcie użyto bazy danych *MySQL* w wersji 5.5.25.

2.2. Użyte narzędzia programistyczne

Do wykonania pracy będącej tematem niniejszej pracy skorzystano z szeregu gotowych programów komputerowych używanych w procesie produkcji oprogramowania. Wykorzystano nie tylko zintegrowane środowiska programistyczne, serwery video, ale także narzędzia używane w inżynierii oprogramowania.

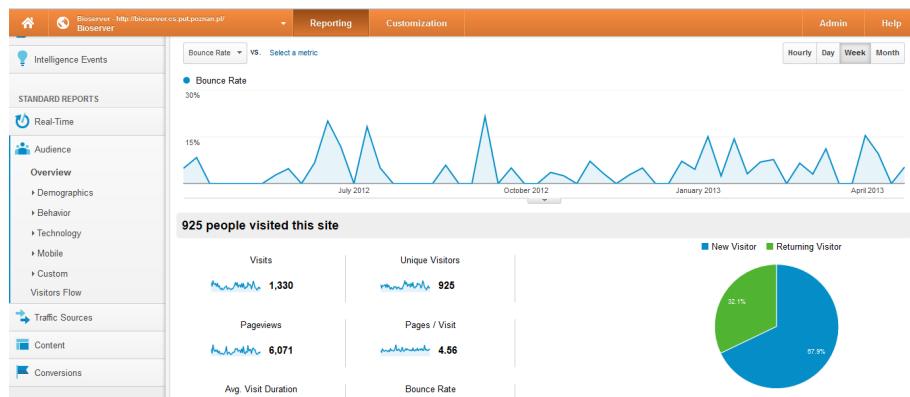
2.2.1. Google Analytics

Google Analytics jest internetowym narzędziem udostępnianym przez firmę *Google*, które umożliwia monitorowanie ruchu na stronie internetowej. Oprócz tego *GA*

2.2 Użyte narzędzia programistyczne

pozwala monitorować ranking strony w wynikach wyszukiwarki *Google*, integrować reklamy wyświetlane z usługą *AdWords*, badać profil użytkowników badanego serwisu (sposób odnalezienia strony, czas wizyty, kraj pochodzenia). *Google Analytics* zbiera informacje w oparciu o plik *JavaScript* który należy umieścić w plikach badanej strony internetowej.

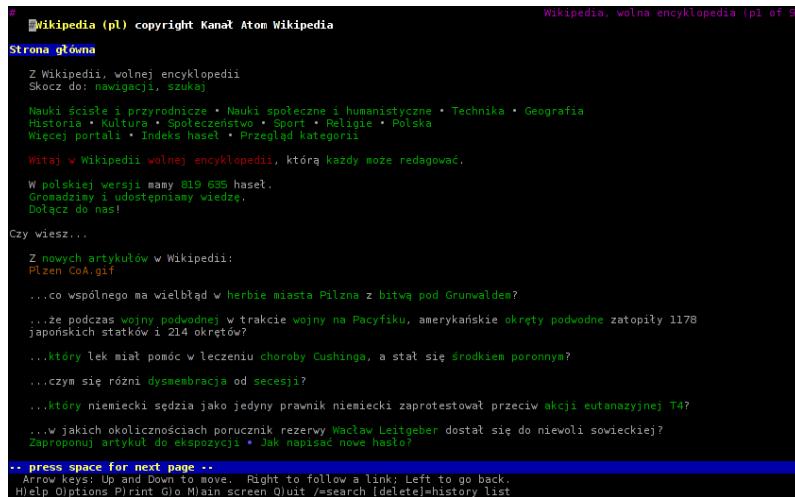
Usługa dostępna jest pod adresem <http://analytics.google.com/> i jest darmowe dla każdego użytkownika poczty *Google*.



Rys. 2.1: Panel główny aplikacji *Google Analytics* i strony *Bioserwer*

2.2.2. Przeglądarka Lynx 2.8.7

Przeglądarka *Lynx* (ang. Ryś) jest przeglądarką tekstową, której pierwsza wersja została stworzona w 1992 roku. Wyświetla ona stronę jedynie w wersji tekstowej - bez obrazków, elementów generowanych za pomocą Java Scriptu lub Adobe Flash. Dzięki temu jest wykorzystywana zarówno przez osoby niewidome do odczytywania treści dowolnej strony jak i przez specjalistów od *SEO* (*Search Engine Optimization*) pragnących obserwować jak ich strona jest widziana przez *crawery Google*.

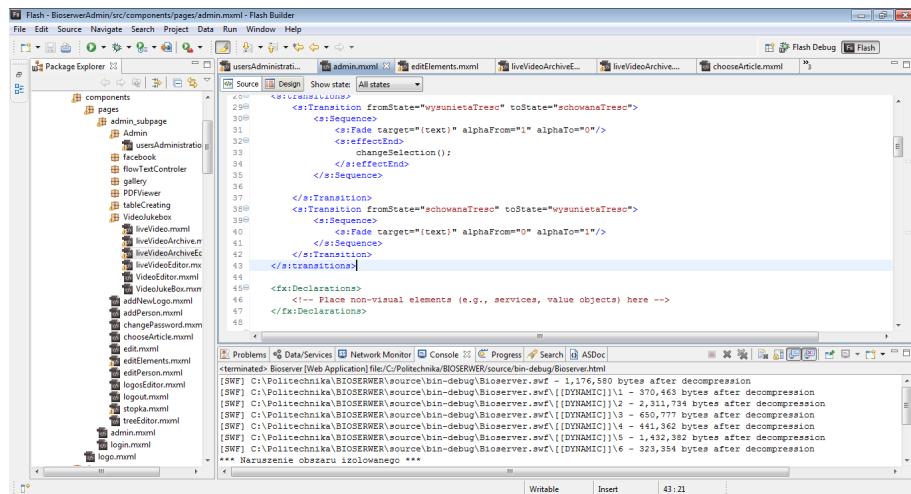


Rys. 2.2: Wygląd głównej strony Wikipedii w przeglądarce *Lynx*

2.2.3. Adobe Flash Builder 4.0

Adobe Flash Builder 4.0, znany również jako *Adobe Flex Builder* w poprzednich wersjach framework'u *Flex*, jest zintegrowanym środowiskiem programistycznym dla aplikacji typu *RIA* opartych na platformie *Flash*. *Flash Builder* to w zasadzie nakładka na *Eclipse'a*, który jest bardzo znanim środowiskiem w świecie programistów przede wszystkim *Javy*. *Flash Builder* wspiera programowanie w trybie kodu jak i wyglądu – *WYSIWYG* (ang. *What you see is what you get*, “To co widzisz jest tym co dostajesz”). Oprócz wsparcia dla technik kodowania w języku MXML i Action Script w podstawowej wersji nie brakuje: debuggera, podpowiadania i kolorowania składni oraz kreatora klas. *Flash Builder* w wersji 4.0 (użyty w projekcie) był dostępny za darmo do użytkownia w celach edukacyjnych. Obecnie w wersji 4.7 nie ma takiej możliwości.

2.2 Użyte narzędzia programistyczne

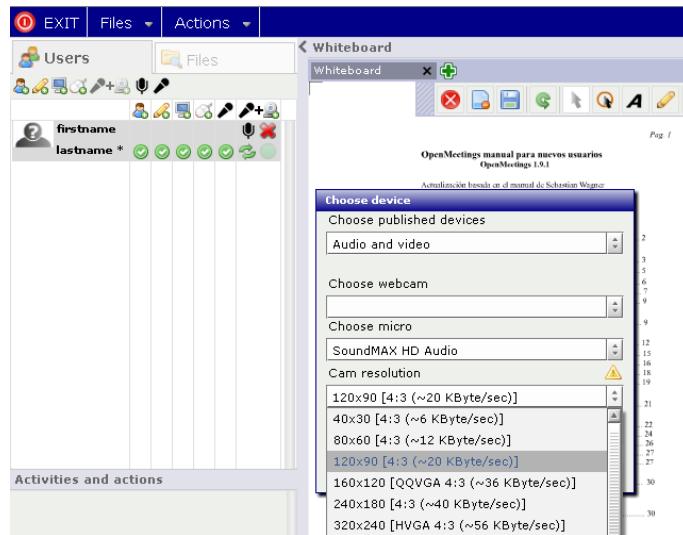


Rys. 2.3: Zrzut ekranu przedstawiający *Flash Builder* w trybie edycji kodu

2.2.4. Apache Open Meetings 2.0.0

Open Meetings jest programem służącym do strumieniowania video, współdzialeń pulpitu, wielosobowej edycji plików i tablic internetowych. Aby osiągnąć swój efekt *Open Meetings* efektywnie spina wiele mniejszych programów open source - *red5* (strumienianie video), *open laszlo* (wygląd aplikacji), baza danych *MySQL* (przechowywanie danych), *FFMPEG* (kodowanie plików multimedialnych) i wiele innych. Program może zostać uruchomiony zarówno w wersji desktopowej (dla jednego użytkownika) jak i serwerowej (wielu). Projekt został rozpoczęty w celu stworzenia alternatywy dla płatnego serwera Video firmy *Adobe* przez Sebastian Wagnera w 2006 roku, aby po paru latach przejść do modelu otwartego i wspieranego przez grupę międzynarodowych programistów.

Open Meetings rozpowszechniany jest na licencji *Apache 2.0* i dostępny jest na stronie: <http://openmeetings.apache.org/>



Rys. 2.4: Wybór rozdzielczości kamery w programie *Open Meetings*

2.2.5. Notepad++ 5.9

Notepad++ to rozbudowany edytor tekstowy, służący jako zamiennik dla systemowego notatnika na platformie *Windows*. Program ten jest napisany w *C++*. Ma wiele zalet, którymi zdecydowanie przewyższa systemowy notatnik. Przede wszystkim obsługuje wiele kart w jednym oknie co zdecydowanie ułatwia pracę z wieloma plikami. Dodatkowo koloruje składnie kilkunastu języków skryptowych i programowania. Dzięki obsłudze dodatkowych pluginów program można łatwo dostosować do wymagań programisty każdego języka za pomocą obsługi autouzupełnianie tekstu, wyszukiwania par nawiasów, dzielenia ekranu i innych przydatnych funkcji. Produkt ten jest darmowy.

```

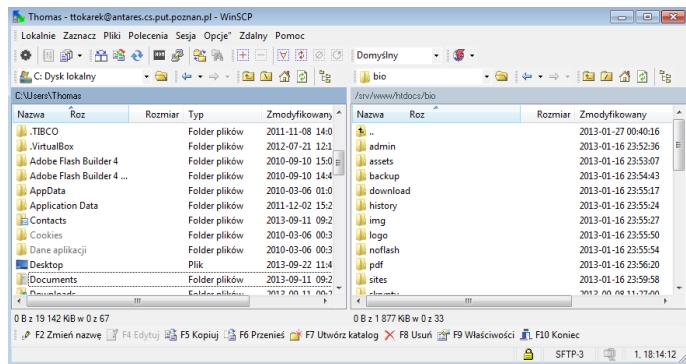
9 | exit();
10 |
11 |
12 | $query = "SELECT * from logo ORDER BY position;";
13 | $result = mysqli_query($link,$query);
14 |
15 | $beg=<logos nazwa="root">';
16 | $end=</logos>';
17 | echo $beg;
18 | while($row = mysqli_fetch_assoc($result))
19 | {
20 |     echo '<Logo ';
21 |     echo 'id="';
22 |     echo $row["id"];
23 |     echo '"';
24 |     echo 'location="';
25 |     echo $row["location"];
26 |     echo '"';
27 |     echo ' URL="';
28 |     echo $row["URL"];
29 |     echo '"';
30 |     echo 'position="';
31 |     echo $row["position"];
32 |     echo '"';
33 |

```

Rys. 2.5: Okno programu *Notepad++*

2.2.6. WinSCP 4.2.9

Program służący do bezpiecznego przenoszenia danych z systemu *Windows* przy pomocy *SSH (Secure Shell)* na zdalny komputer i odwrotnie. Narzędzie to zawiera graficzny interfejs, wspiera operacje drag and drop, a także obsługuje protokoły *SFTP (SSH File Transfer Protocol, Safe File Transfer Protocol)*, *SCP (Secure Copy Protocol)* i *FTP (File Transfer Protocol)*. Produkt ten jest darmowy.



Rys. 2.6: Program WinSCP, połączenie z *Bioserwerem*

2.2.7. Git / GitHub 1.1.1

System kontroli wersji stworzony przez Junio Hamano, Linusa Torvaldsa w 2005 roku, jako alternatywę do systemu *SVN (Subversion)*. Projekt powstał głównie z myślą o szybkości działania i pracy w środowisku linuksowym, jednak istnieją także programy dedykowane dla środowiska *Windows*. W projekcie *Bioserwera* użyto klienta strony *GitHub* (<https://github.com/>) do składowania kodu źródłowego aplikacji. Posiada on przejrzysty interfejs oraz intuicyjną obsługę. Program i repozytorium są dostępne darmowo, pod warunkiem udostępnienia kodów źródłowych innym osobom (istnieje możliwość wykupienia wersji zamkniętej).

```

diff --git a/src/components/pages/admin_subpage/Video/VideoLukeBox.mxml b/src/components/pages/admin_subpage/Video/VideoLukeBox.mxml
index 630bd98660a89c6897e25e51927c7d0c272582e5..630bd98660a89c6897e25e51927c7d0c272582e5
--- a/src/components/pages/admin_subpage/Video/VideoLukeBox.mxml
+++ b/src/components/pages/admin_subpage/Video/VideoLukeBox.mxml
@@ -219,13 +219,19 @@
 219     case playButton :
 220         // If the play button was clicked,
 221         resume the video playback.
 222         // If the video was already playing,
 223         this has no effect.
 224         ns.resume();
 225         if(ns!=null)
 226         {
 227             break;
 228         }
 229         case pauseButton :
 230             // If the pause button was clicked,
 231             the video will be paused.
 232             // If the video was already paused,
 233             the video will be resumed.
 234             ns.togglePause();
 235             if(ns!=null)
 236             {
 237                 ns.togglePause();
 238             }
 239         case stopButton :
 240             // If the stop button was clicked,
 241             the video will be stopped.
 242             ns.stop();
 243             if(ns!=null)
 244             {
 245                 ns.stop();
 246             }
 247         break;
 248     }
 249 
```

Rys. 2.7: Wygląd aplikacji GitHub

2.2.8. VLC Media Player 2.0.3

VLC Media Player (Video Lan Media Player), to odtwarzacz video który oprócz klasycznych możliwości odtwarzania plików, posiada możliwość odtwarzania strumieni danych *RTMB*, *TS* (*Transport Stream* - standard przesyłu danych telewizji cyfrowej w *MPEG*) oraz danych przechwytywanych przez dedykowane urządzenia zewnętrzne (tunery telewizji satelitarnej i naziemnej). Dzięki tak szerokiej funkcjonalności *VLC* jest idealnym programem do badania jakości połączenia i jego parametrów. Oryginalnie projekt został stworzony jako *Video LAN Client* w roku 1996 przez grupę francuskich studentów, jednak obecnie odtwarzacz jest rozwijany przez międzynarodową grupę: *VideoLAN Project*. *VLC* jest dostępny na licencji *LGPL (Lesser GNU Public Licence)*



Rys. 2.8: VLC Player odgrywający plik

2.2.9. SWFAddress 2.4

SWFAddress jest darmową grupą skryptów stworzoną w języku *JavaScript* przez bułgarską grupę programistyczną *Asual*. Skrypty te pozwalają na manipulację adresem i historią przeglądarki, używaniem wyszukiwarek do przeszukiwania treści strony do której są podłączone, a także tworzeniem zakładek stron w serwisach społecznościowych. Są one dostępne pod adresem: <http://www.asual.com/swfaddress/>

2.2.10. IpAdapter

Darmowy program rozprowadzany na licencji *Apache*. Pozwala on udostępniać publiczną kamerę IP, jako prywatne urządzenie multimedialne. Program został użyty w ostatniej fazie pracy, gdy zaistniała potrzeba udostępnienia kamery IP do serwera *OpenMeetings*. Aplikacja dostępna jest pod adresem: <http://ip-webcam.appspot.com/>

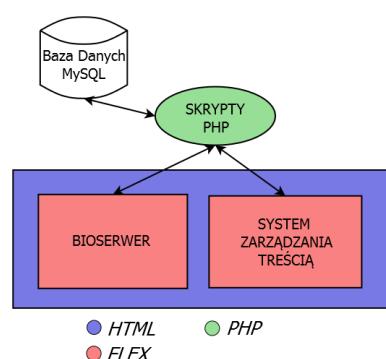
2.3. Opis istniejącego systemu

Istniejąca strona Bioserwera, dostępna była pod adresem <http://bio.cs.put.poznan.pl/> (z powodu restrukturyzacji strony internetowej została ona zmieniona w lipcu 2013 roku, obecnie jej kopia znajduje się pod adresem: <http://antares.cs.put.poznan.pl/bio/>), została stworzona w 2011 roku w ramach pracy inżynierskiej przez Michała Jackowskiego, Szymona Szałatę, Marka Szewczyka i Tomasza Tokarka pod nadzorem prof. Jacka Błażewicza i pani dr Marty Szachniuk. Strona główna Bioserwera jest zaprezentowana na rysunku 2.10. Została ona napisana w języku *Adobe Flex*, który komunikuje się z bazą danych (*MySQL*) przy pomocy skryptów *PHP* (patrz rysunek 2.9).

Pod względem treści system można podzielić na trzy części:

- Strony dostępnej dla użytkownika końcowego
- System Zarządzania Treścią (*SZT*)
- Baza Danych (do jej przechowywania)

W trakcie edycji, na każdej stronie - zwanej dalej też artykułem, można umieścić dowolną ilość elementów. Elementem może być tekst, tabela, galeria zdjęć lub video. Artykuły ułożone są w strukturze drzewiastej, zaś każdy artykuł może posiadać *potomków*, będąc



Rys. 2.9: Architektura Bioserwera

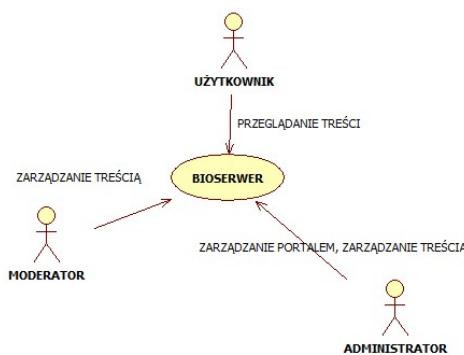
wtedy *rodzicem*. Treść artykułów można edytować z poziomu *SZT* napisanego na potrzeby pracy inżynierskiej (zwykle dostępnego jako podstrona *admin*: [http://\[STRONA_BIOSERWERA\]/admin/](http://[STRONA_BIOSERWERA]/admin/)). Dostęp do *SZT* jest chroniony loginem i hasłem, zaś hasło jest przechowywane na serwerze w formie zaszyfrowanej (*SHA-1*).

Największym problemem, który został nierozerwiany w oryginalnej implementacji jest brak podziału ról w administracji systemu oraz brak kopii zapasowej strony dla użytkowników nie posiadających wtyczki *Adobe Player* (w tym automatycznych botów indeksujących strony). Niniejsza praca magisterska opisuje sposób implementacji rozwiązań dla obu problemów.



Rys. 2.10: Wygląd strony Bioserwera z rozwiniętym menu

2.4. Specyfikacja Wymagań



Rys. 2.11: Aktorzy systemu

Wymagania funkcjonalne opisują dla programisty zakres pracy jaka musi zostać wykonana nad projektem, by mógł on zostać uznany przez klienta za spełniający oczekiwania. Wymagania zostały podzielone na funkcjonalne (przedstawione w postaci przypadków użycia w punkcie 2.4.1) i pozafunkcjonalne (patrz punkt 2.4.2). System posiada trzech aktorów używających systemu, są oni przedstawieni na rysunku 2.11.

2.4.1. Przypadki użycia

ID:	UC1
Główny aktor:	Administrator
Opis:	Administrator zarządza moderatorami w systemie
Scenariusz główny:	<ol style="list-style-type: none">1. Administrator loguje się do systemu2. Administrator zarządza moderatorami3. Administrator zarządza prawami dostępu moderatora do treści strony4. System zapisuje zmiany w systemie
Scenariusz rozszerzony:	<ol style="list-style-type: none">1.a. Złe dane wejściowe.<ol style="list-style-type: none">1.a.1. System wyświetla błąd. System wraca do punktu 1.

Tablica 2.1.: UC1 - Zarządzanie Rolami

ID:	UC2
Główny aktor:	Moderator
Opis:	Moderator decyduje o ustawieniach połączenia z <i>OpenMeetings</i>
Scenariusz główny:	<ol style="list-style-type: none">1. Moderator dodaje nowe połączenie video czasu rzeczywistego2. Moderator edytuje parametry połączenia3. System zapisuje zmiany

Tablica 2.2.: UC2 - Ustawienie parametrów połączenia *Teleseminariów*

ID:	UC3
Główny aktor:	
Moderator	
Opis:	
Moderator edytuje nagranie archiwalne połączenia <i>Teleseminariów</i>	
Scenariusz główny:	
1. Moderator dodaje archiwum do artykułu 2. Moderator ustawia video i terminarz zakładek nagrania 3. System zapisuje zmiany	

Tablica 2.3.: UC3 - Ustawienie parametrów archiwum *Teleseminariów*

ID:	UC4
Główny aktor:	
Użytkownik	
Opis:	
Użytkownik przegląda nagranie archiwalne	
Scenariusz Główny:	
1. Użytkownik otwiera stronę z nagraniem 2. Użytkownik ma możliwość oglądania treści i zmiany położenia nagrania pomocniczego 3. Użytkownik ma możliwość przejścia w tryb pełnoekranowy 4. Użytkownik ma możliwość nawigacji po nagraniu za pomocą zakładek	
Scenariusz rozszerzony:	
2.a Błąd połączenia do danych multimedialnych 2.a.1 System wyświetla odpowiedni komunikat błędu	

Tablica 2.4.: UC4 - Odtwarzacz archiwum *Teleseminariów* - użytkownik

2.4.2. Wymagania Pozafunkcjonalne

W celu zapewnienia komfortowej pracy z aplikacją powstał szereg wymagań pozafunkcjonalnych:

- Funkcjonalność (ang. *functionality*) System powinien zapewnić chroniony dostęp do edytowania danych;
- Niezawodność (ang. *reliability*) System powinien udostępnić łatwe tworzenie kopii zapasowej danych;
- Użyteczność (ang. *usability*) Edytory programu powinny być przejrzyste i łatwe w obsłudze;

- Przenośność (ang. *portability*) Możliwość uruchomienia strony na każdym systemie operacyjnym;

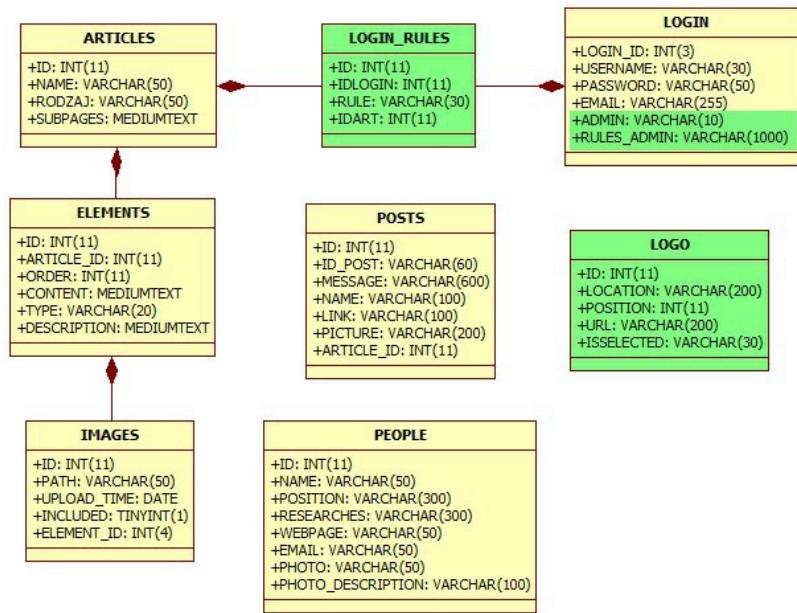
2.5. Schemat bazy danych

Rysunek 2.12 przedstawia schemat bazy danych która została użyta w oryginalnym projekcie *Bioserwera*. Elementy zaznaczone na zielono zostały stworzone obecnie, na potrzeby bieżącej pracy magisterskiej. Poniżej opisano, jak zmiany w bazie danych wpływają na wprowadzenie planowanych komponentów.

Budowa bazy danych umożliwia podpięcie dowolnej ilości elementów(*ELEMENTS*) pod dany artykuł, aby wprowadzić funkcjonalność live video, należało jedynie dodać nowy rodzaj logicznego komponentu *Element*, który odpowiadałby za dwa nowe elementy w podsystemie *Teleseminarium*. Jest to zmiana której nie wpływa na ogólną architekturę bazy danych, jest przeprowadzona po stronie aplikacji końcowej.

Wprowadzenie systemu ról w warstwie administracyjnego Bioserwera wiązało się z dodaniem tablicy *LOGIN_RULES* i dwóch dodatkowych kolumn służących do przechowywania praw edycji systemu. Każde konto moderatora (*LOGIN*) ma przypisaną flagę admina (informującą czy dany użytkownik jest administratorem systemu), czyli czy posiada uprawnienia do nadawania praw innym moderatorom (kolumna *ADMIN*). Konto opisuje również prawa do edycji stałych części portalu takich jak posty czy też stopka (*RULES_ADMIN*). Użytkownik może również mieć nadane prawo do edycji danego artykułu (zezwolenie lub zabronienie takiej edycji) w tablicy *LOGIN_RULES*. Dokładniejszy opis wykorzystania danych do przyznania moderatorowi praw edycji w artykułach w punkcie 3.2 .

Informacje o logach wyświetlane w stopce są przechowywane w tablicy *LOGO*, *LOCATION* - wskazuje grafikę do wyświetlenia, *URL* - adres strony docelowej, *POSITION* - numer pozycji we wszystkich logach, zaś *ISSELECTED* - informacje czy dane logo jest domyślnie wybrane w momencie załadowania strony.



Rys. 2.12: Baza danych Bioserwera

3. Część I: Zarządzanie Bioserwerem

3.1. Problematyka na styku Google / Bioserwer

Podstawowym kryterium skuteczności strony internetowej jest zdolność dotarcia z jej treścią do użytkownika końcowego. Informacja nie używana, niedostępna, jest informacją martwą. Dlatego wszelkiego rodzaju wyszukiwarki internetowe, w tym największa z nich - wyszukiwarka *Google*, sprzyjają twórcom stron internetowych w procesie dostarczania zawartości stron dla użytkownika końcowego. Robią to - katalogując strony internetowe za pomocą zautomatyzowanych skryptów zwanych *crawlerami*. *Crawlersy* przeglądają zawartość strony zapisanej w języku *HTML* i katalogują wszelkie linki do stron powiązanych z aktualnie wyświetlaną stroną. Niestety tutaj pojawia się główny problem języka *Flex* (bazującego na języku skryptowym *Adobe Flash*), który został użyty do stworzenia pierwotnej wersji strony dla grupy Bioinformatycznej (www.bioserwer.cs.put.poznan.pl). Treść strony stworzonej w języku *Flex* jest przechowywana w zamkniętym środowisku - interpretowanym przez wtyczkę *Flash Adobe Player*. Z perspektywy strony *HTML* podstrona napisana we *Flexie* jest czarną skrzynką (ang. *black box*), której wewnętrzna nawigacja i zawartość są dla niej nieznane.

Dlatego w projekcie należało rozwiązać dwa problemy:

- udostępnić treść strony z *Flex* w *HTML*
- ujednolicenie nawigacji na stronie (*HTML/Flex*)

oraz cel funkcjonalny:

- utrzymanie się strony na pierwszej stronie wyników dla hasła *bioserver* w wyszukiwarce *Google* (przed rozpoczęciem prac - drugie miejsce)

3.1.1. Treść strony

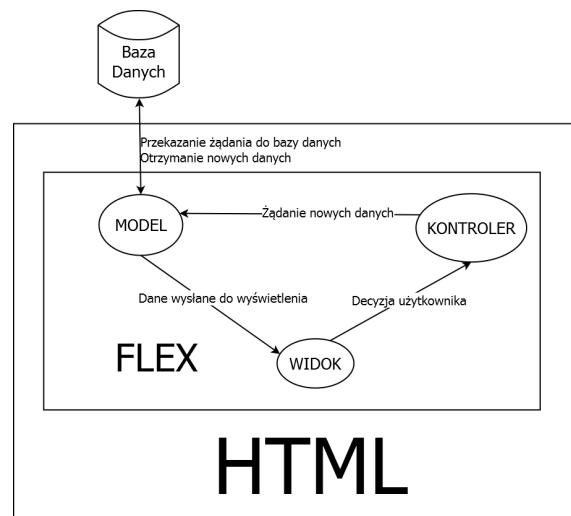
Najlepszym sposobem na wyświetlenie strony dla użytkowników (w tym *crawlerów*) nie posiadających wtyczki *Adobe Player* jest po prostu ściągnięcie jej treści z serwera i wyświetlenie w normalnym języku *HTML*. *Flex* jest oparty na architekturze *MVC* (ang. *Model-View-Controller*: Model-Widok-Kontroler) to znaczy że kod źródłowy aplikacji jest podzielony na części które odpowiadają kolejno za: przetwarzanie

danych, wyświetlanie danych i pozyskanie danych, zaś sama treść strony jest składowana w zewnętrznej bazie danych (patrz rysunek 3.1). Dlatego wyświetlenie zawartości strony ogranicza się jedynie do nawiązania połączenia z bazą danych i wyświetlenia treści, która jest już składowana w formacie *HTML* (dokonanie konwersji nie jest wymagane). Ponieważ jednak czysty język *HTML* nie obsługuje połączeń z bazą danych, w projekcie została użyta strona generowana przez skrypt *PHP*. Pobiera on informacje ze skryptu pomocniczego *SWFAddress* (patrz punkt 3.1.2) o *ID* aktualnie obserwowanego artykułu, a następnie pobiera treść tego artykułu z serwera. Skrypt działa jedynie wtedy gdy strona we *FLEX* nie załaduje się poprawnie - po to by nie dublować obciążenia w trakcie normalnej pracy strony internetowej.

3.1.2. Nawigacja

Użytkownik, który pragnie szybko dotrzeć do interesujących go treści często zapisuje adres podstrony interesującej go strony w przeglądarce. Jednak w przypadku czystego języka *FLEX* jest to niemożliwe, bo adres strony widziany przez przeglądarkę nie zmienia się. Sytuacja taka nazywa się brakiem *Deep Linking* (ang. Głębokie Linkowanie). Jest to również problem dla *crawlerów*, które nie są w stanie odpowiednio skategoryzować treści znalezionych stron. Dlatego w projekcie dodano możliwość zmiany adresu strony przy pomocy zewnętrznej biblioteki *SWFAddress*. Biblioteka ta umożliwia zmianę adresu strony automatycznie z poziomu języka *FLEX*. Przesyła ona również adres w drugą stronę, umożliwiając przeparsowanie w poszukiwaniu adresu *ID* artykułu, w celu znalezienia odpowiedniej treści do wyświetlenia. W algorytmie 3.1 widzimy dwa rodzaje adresów używanych przez aplikacje:

1. jest adresem ustawianym w momencie gdy strona pracuje za pomocą aplikacji *FLEX* - tutaj jedynym ważnym elementem jest przypisanie parametru *a* czyli *ID* wyświetlany elementu
2. ten adres jest ustawiany przez stronę *PHP*, z racji tego że artykuły na stronie mają strukturę drzewa można wyświetlić całą ścieżkę do artykułu, ma to znaczenie w przypadku katalogowania strony przez wyszukiwarkę *Google* (patrz punkt 3.1.3)



Rys. 3.1: Stara architektura Bioserwera

Algorytm 3.1 Formaty adresu internetowego

- 1) `http://[ADRES_STRONY]/?a=[ID_ARTYKULU]`
- 2) `http://[ADRES_STRONY]/[ROOT]/.../[RODZIC_N]/?a=[ID_ARTYKULU]`

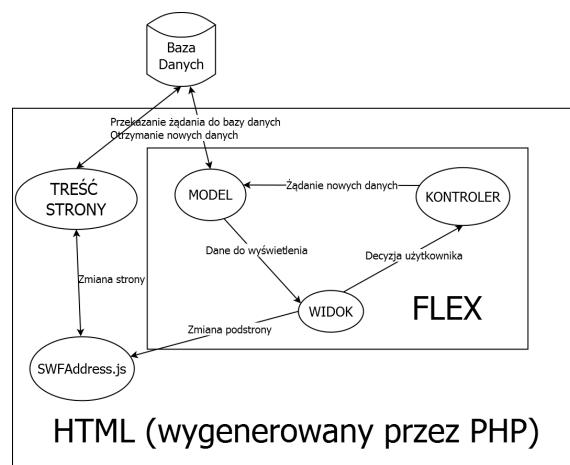
3.1.3. Google - PageRank

PageRank jest algorytmem używanym przez *Google* do weryfikacji jakości treści danej strony internetowej. W skrócie: opiera on się na założeniu że strona na którą powołują się inne strony o wysokiej wartości, też jest stroną o wysokiej wartości. Dodatkowo algorytm ten został rozbudowany o wiele dodatkowych, tajnych kryteriów które mają zapobiegać jego nadużywaniu i gwarantować prawidłowość wyniku. Dlatego w trakcie budowania strony dla *crawlerów* ważne było aby:

- strona miała normalną budowę - menu, treść, stopka. Strony z czystym tekstem jako treścią otrzymują mniejszy wynik
- strona posiadała nawigację opartą na systemie drzewa aby ułatwić katalogowanie zebranych informacji
- linki do treści w stronie wewnętrznej działały prawidłowo

Punkty 3.1.1 i 3.1.2 opisały w jaki sposób strona została przygotowana na przyjęcie automatycznych botów *Google*. Jednak by ten proces mógł się zakończyć stronę należy jeszcze zarejestrować w usługach *Google*. Służy ku temu narzędzie *Google Analytics*. Po zarejestrowaniu się w serwisie, należy stworzyć nowy profile dla adresu który ma być obserwowana, a następnie przekopiować wygenerowany plik *JavaScript* ze stworzonym unikalnym *TrackID* na nasz serwer. Od tego momentu administrator serwera może nie tylko obserwować aktualny ranking swojej strony, ale także każda wizyta osoby odwiedzającej serwis zostanie zapisana w systemie *Google* (czas wizyty, odwiedzone podstrony, kraj pochodzenia osoby wizytującej).

Ostatecznie, zmiana strony spowodowała spadek jedynie o dwa miejsca w rankingu *Google* dla słowa “*Bioserver*” (z drugiego na czwarte), co oznacza że wymaganie



Rys. 3.2: Nowa architektura

pozafunkcjonalne zostało osiągnięte. Spadek mógł być spowodowany niedoskonałościami metody: brakiem wyświetlania metadanych o rysunkach na stronie, nie wyświetlaniem informacji zawartych w tablicach oraz brakiem stopki z linkami do pokrewnych stron. To znaczy że metoda może zostać udoskonalona w przypadku użycia w przyszłych projektach.

3.2. Administracja serwera - role użytkowników

Bioserwer jest serwerem grupy Bioinformatycznej na Politechnice Poznańskiej, z założenia służy on prezentowaniu przedmiotów badań tej grupy w Internecie. W starej wersji strony możliwość edytowania treści miał jedynie administrator strony, który otrzymywał treść wiadomości od reszty członków grupy. Aby usprawnić ten proces postanowiono dać możliwość edytowania treści wszystkim użytkownikom poprzez System Zarządzania Treścią (*SZT*). Jednakże by niedopuszcić do zmniejszenia bezpieczeństwa na stronie (poprzez utworzenie wielu kont administracyjnych), w ramach niniejszej pracy dodano możliwość zakładania ograniczeń na utworzone konta.

Panel administracyjny znajduje się na podstronie *admin* głównej strony ([http://\[ADRES_SERWERA\]/admin](http://[ADRES_SERWERA]/admin)). W nowej wersji użytkownik z prawami administracyjnymi zauważa dodatkową zakładkę w której będzie mógł tworzyć konta i przydzielać im prawa do edytowania zawartości.



Rys. 3.3: Wygląd panelu administracyjnego

Na rysunku 3.3 widać standardowy widok panelu administracyjnego. Jest on podzielony na trzy części, licząc od lewej:

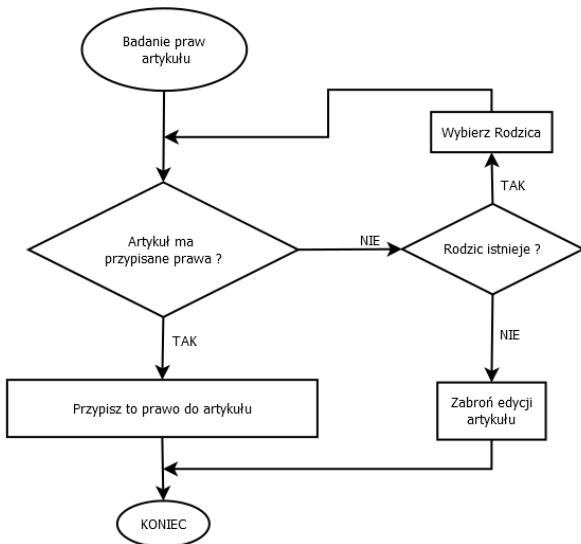
- pierwsza kolumna odpowiedzialna jest za administrację użytkownikami portalu: zmianę hasła, adresu *email*, oraz loginu, a także dodanie lub usunięcie użytkownika

- druga kolumna odpowiedzialna jest za przyznawanie praw do edycji artykułów w serwisie. Zasada działania tych opcji jest dokładniej opisana w podpunkcie 3.2.1
- trzecia kolumna to prawa administratorowskie - możliwość dostępu manipulacji listą osób, postów, linkami stopki, a także powyżej strony administracyjnej

3.2.1. Drzewo edycji - algorytm działania

Drzewo edycji służy do wyboru które artykuły użytkownik ma prawo edytować. Domyślnie użytkownik nie może edytować żadnego artykułu, dopiero kliknięcie na dany artykuł i przyznanie praw do edycji za pomocą przycisku *Grant rights* (ang. Przyznaj prawa) daje prawo do edycji w wybranym artykule i we wszystkich jego potomkach. Zaś wybranie przycisku *Remove rights* (ang. Odbierz prawa) je odbiera. Przycisk *Clear rights* usuwa prawa zapisane dla danego artykułu. Zapisanie praw dla całego drzewa artykułów, dla każdego z użytkowników osobno byłoby problematyczne z dwóch względów: zajmowałoby więcej miejsca na serwerze i wymagałoby dodatkowych wyliczeń/komunikacji z serwerem w przypadku edycji drzewa artykułów (np. przesunięcia elementu strony). Dlatego w projekcie wybrane zostało inne rozwiązanie (algorytm zaprezentowany na rysunku 3.4 i opisany poniżej).

System zapisuje jedynie prawa do pojedynczego artykułu (zezwolenie lub zakaz edycji), zaś algorytm w przypadku potrzeby określenia prawa dla innego artykułu, wywołuje rekurencyjnie sprawdzenie praw dla rodziców, dopóki nie natrafi na zezwolenie przydzielone przez administratora. W przypadku braku przydzielenia prawa - użytkownik nie ma zezwolenia na edycję artykułu. Na rysunku 3.3 artykuły które można edytować określone są prefixem *[YES]*, a te które nie można *[NO]*. Dzięki zastosowaniu takiego algorytmu na serwerze jest przechowywanych niewiele danych (pojedyńcze linie komend) i nie występuje problem przeliczania praw wszystkich użytkowników w przypadku zmiany położenia pojedynczego elementu drzewa. Jedynym mankamentem tej metody jest narzut obliczeniowy wymagany w przypadku próby użycia danego artykułu. Jednakże wynosi on $O(m)$ dla pojedynczego elementu i $O(m * n)$ dla wyświetlenia całego drzewa. Gdzie



Rys. 3.4: Algorytm wyboru praw edycji

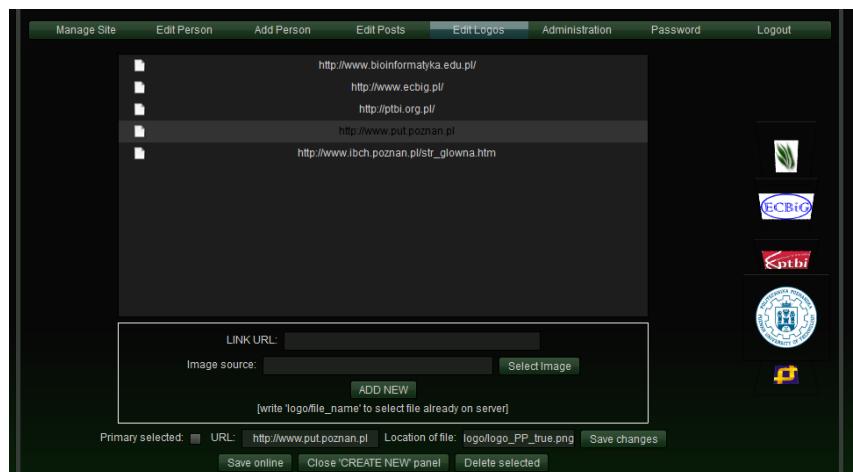
$O(m)$ dla pojedynczego elementu i $O(m * n)$ dla wyświetlenia całego drzewa. Gdzie

- m - średnie zagłębienie w drzewie - maksymalne zagłębienie w drzewie wynosi obecnie 5 lub 6 - tak więc nie jest to spora wartość
- n - liczba elementów (artykułów) w drzewie - liczba ta wynosi około 100

Ostateczne obciążenie procesora użytkownika końcowego nie jest więc problemem i to rozwiązanie może zostać użyte celem stworzenia szybkiej selektora praw edycji.

3.3. Edytor stopki

Strona nie powinna być przebudowywana z kodu źródłowego przy każdej zmianie, dlatego w trakcie pracy magisterskiej skorygowano błąd przeniesiony z poprzedniej wersji - brak możliwości edycji wyświetlanych logo w stopce strony. Teraz w menu administracyjnym znajduje się dodatkowa opcja która odpowiada za edycję elementów, a także za ich modyfikacje i tworzenie nowych podelementów (patrz rysunek 3.5). Aby uatrakcyjnić wygląd stopki, użyto komponentu *VCoverFlow-Container* stworzonego przez Douga McCune (komponent dostępny pod adresem: <http://dougmccune.com/blog/2007/11/03/overflow-flex-component/>), dzięki czemu stopka zyskuje bardziej dynamiczny charakter (pół przezroczyste tła i przesunięcie w momencie dotknięcia myszką).



Rys. 3.5: Edycja logo

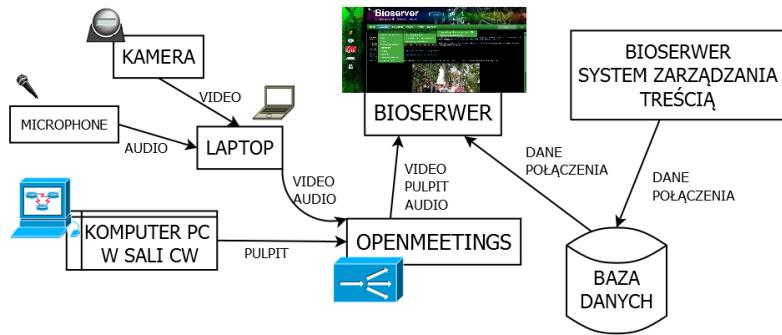
4. Część II: Teleseminaria

4.1. Wprowadzenie

Teleseminaria to nazwa kodowa elementu multimedialnego potrzebnego w pracy Grupy Bioinformatycznej Politechniki Poznańskiej. Grupa Bioinformatyczna organizuje regularne spotkania - Seminaria uczelniane - na których pracownicy przedstawiają efekty swoich badań. Aby uczynić te spotkania efektywniejszymi, postanowiono nagrywać je i transmitować przez Internet. Implementacja tej właśnie funkcjonalności jest celem drugiej połowy części pracy magisterskiej. W ramach tego zadania przygotowano:

- wybór potrzebnego sprzętu (kamery video)
- wybór i instalacja oprogramowania pośredniczącego
- dokonanie koniecznych zmian w Bioserwerze

Ostatecznie system został zbudowany w oparciu o architekturę zaprezentowaną na rysunku 4.1. Obraz przedstawiającego dane spotkanie jest rejestrowany w kamerze, a następnie przesyłany do laptopa podłączonego w sali, który dodaje do nagrania dźwięk z mikrofonu. Obraz przesyłany jest na serwer OpenMeetings (serwer zarządzający połączonymi audio/video). Komputer PC w sali przesyła na ten sam serwer zawartość ekranu (ang. *desktop*). Informacja o strumieniu jest przechowywana na serwerze i na nim również jest nagrywana. W przypadku gdy użytkownik końcowy żąda dostępu do strony Bioswerwera, otwiera ona nowy strumień danych z serwera Openmeetings na parametrach podanych z bazy danych. Parametry te mogą zostać zmienione przez administratora systemu w Systemie Zarządzania Treścią.



Rys. 4.1: Architektura systemu *Teleseminariów*

4.2. Wybór kamery

Wszystkie sale oprócz sali *Magna Aula* w Centrum Wykładowym Politechniki Poznańskiej są pozbawione sprzętu przeznaczonego do transmisji video i audio, dlatego pierwszym etapem pracy nad systemem *Teleseminariów* był wybór odpowiedniego sprzętu do użytkownika w trakcie realizacji projektu. Kamera została użyta w projekcie do przekazywania obrazu osoby prezentującej treść wystąpienia. Z racji pomocniczej wartości nagrania (kamera rejestruje jedynie dane drugorzędne - wygląd prowadzącego), obsługiwana przez kamerę rozdzielczość nie była kryterium podstawowym.



Rys. 4.2: Wybrana kamera - *TRENDnet TV-IP501P* [TRE13]

Pierwotnie planowano zakup kamery USB podłączanej bezpośrednio pod komputer (patrz załącznik A.2 - stan marzec 2012). Jednakże ze względu lepszą możliwość wykorzystania urządzenia w trybie przenośnym (wpinanie bezpośrednio w Sieć Uczelnianą), zdecydowano się zmienić model na kamerę IP.

Przeanalizowano ofertę rynkową, biorąc pod uwagę atrybuty:

- rodzaj kodowania sygnału (np.: *MJPEG*, *FLV*)
- cena
- jakość obrazu

Ostatecznie wybrany został model *TRENDnet TV-IP501P*. Do tej decyzji przyzyniły się bezpośrednie testy przeprowadzone w okresie marzec/kwiecień 2012 na

modelu *IP501*, udostępnionym przez Pana Iwo Lewandowskiego, oraz pozytywne wyniki prób transmisji obrazu.

4.3. Wybór oprogramowania

Element przekazujący i nagrywający audio/video jest głównym programem zapewniającym funkcjonowanie całego systemu *Teleseminariów*. Dlatego wymagał on dokładnego przygotowania i przetestowania wspieranej funkcjonalności. Od programu oczekiwano spełnienia następujących kryteriów:

- restrumieniowania danych audiowizualnych
- nagrywanie otrzymanych danych
- udostępniany na podstawie darmowej licencji

W fazie pierwszej rozważano stworzenie przykładowego projektu w ramach bieżącej pracy magisterskiej - jednak po inwestygacji, zadanie to zostało uznane za zbyt pracochnonne w porównaniu z zakresem projektu. Następnie rozpoczęto testy różnych systemów streamingowych, które zostały uznane za kompatybilne z ustalonimi wymaganiami. Systemy te objemują:

4.3.1. *Adobe Flash Server*

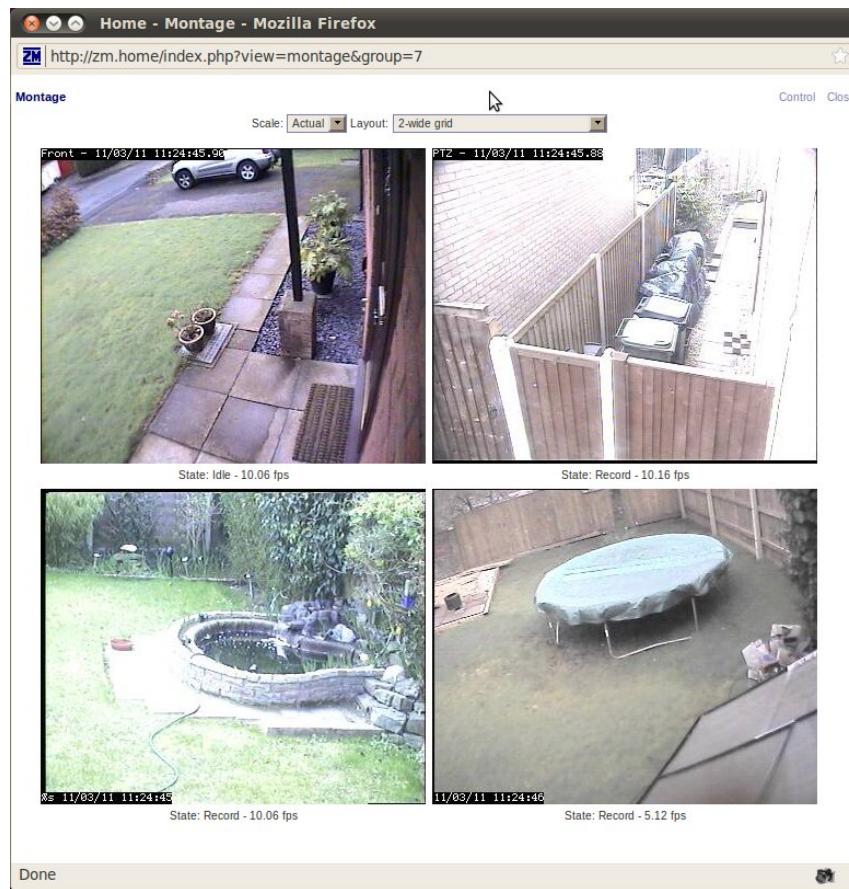
Adobe Flash Server jest serwerem usług multimedialnych stworzonym przez firmę *Adobe*. Pozwala on w łatwy sposób restrumieniować obraz z kamer i pulpitu, odbierając i wysyłając filmy w formacie *flv* i *MPEG*, również w protokole *RTMP*. Jednakże program ma sporą wadę - jest płatny. Jedyna wersja darmowa jaką istnieje - *Development* posiada ograniczenia do 10 minut nagranego materiału i 10 jednoczesnych połączeń. Jest to zbyt mało, dlatego program nie spełnia kryteriów i nie mógł być zaakceptowany do dalszej pracy.

4.3.2. *RED5*

Red5 jest darmowym projektem na licencji *Apache2*. Dostępnym pod adresem: <http://code.google.com/p/red5/>. Program ten znakomicie dekoduje i restrumieniuje różne rodzaje strumieni danych audiowizualnych (*flv*, *MP4*, *AAC*) i protokołów *RTMP*. Jednakże program nie ma wbudowanej możliwości współdzielenia obrazu pulpitu. Teoretycznie istnieje możliwość użycia w tym celu programu firmy trzeciej - *Adobe Flash Media Encoder* - zakodowanie przy jego pomocy obrazu pulpitu i wysłanie go na serwer Red5. Jednakże *ToS* (ang. licencja użytkowania) tego programu stwierdza, że można go użyć jedynie do współpracy z programem *Adobe Flash Server*, inne użycie byłoby nielegalne. Postanowiono znaleźć lepsze rozwiązanie.

4.3.3. ZoneMinder

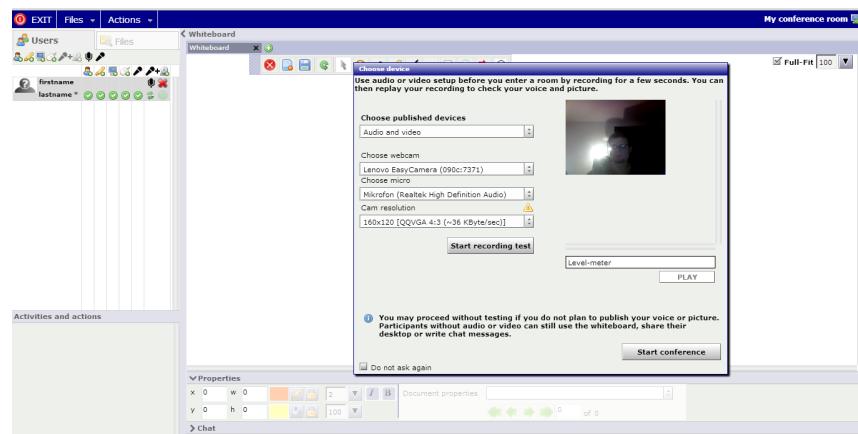
Program został stworzony jako serwer zbiorczy dla telewizji przemysłowej (wyświetlania i nagrywania obrazu kamer ochrony). Napisany został w języku *PHP*, *Perl*, *C++* i jest dostępny pod adresem: <http://www.zoneminder.com/>. Aplikacja, oprócz standardowego wsparcia dla protokołów *RTMP* i danych *flv*, oferuje również łatwiejszą możliwość połączenia się z sygnałem z perspektywy języka *Flex*. Program nawiązuje również bezpośrednie połączenie z kamerą przy pomocy kodowania *MJPEG*. Jednakże także ten program nie posiada wbudowanej możliwości współdzielenia pulpitu. Przez okres dwóch, trzech miesięcy prowadzono prace integracyjne z systemem *ZoneMinder*, jednak odkrycie programu obiecującego lepsze wyniki - spowodowały przerzucenia wysiłku na rzecz wykorzystanie aplikacji *OpenMeetings* (patrz punkt 4.4).



Rys. 4.3: Przykładowe okno programu pokazujące 4 kamery (źródło [ZON13])

4.4. Praca z OpenMeetings

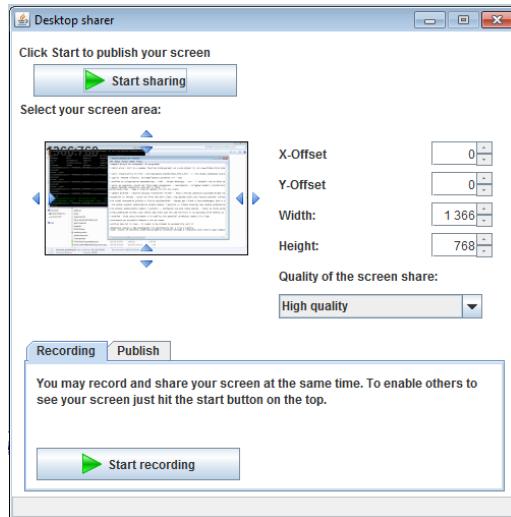
OpenMeetings jest oprogramowaniem, które ostatecznie spełniło wszystkie kryteria wymagane dla serwera pośredniczącego w wymianie strumieni multimedialnych. Oprogramowanie to jest zbiorem narzędzi internetowych (w jego skład wchodzi m.in.: podserwer *Red5*, przedstawiony w punkcie 4.3.2), które po połączeniu tworzą wirtualną platformę edukacyjną - umożliwiającą organizowanie spotkań, używanie wirtualnych tablic i przekazywanie/nagrywanie obrazów kamer i pulpitów. Po niewielkich zmianach w kodzie źródłowym, transmisje stały się również dostępne również po stronie *Bioserwera*.



Rys. 4.4: Wygląd pokoju

W pracy z programem *OpenMeetings* podstawowym pojęciem jest *pokój*. *Pokój* to specjalnie wydzielony obszar wirtualny dla użytkowników w ramach którego mogą oni uczestniczyć w wydzielonej konferencji. W ramach tej konferencji użytkownik może współdzielić swój pulpit i nagranie lokalnej kamery ze współużytkownikami (przykład pokoju na rysunku 4.4). Standardowa praca z *OpenMeetings* obejmuje trzy fazy:

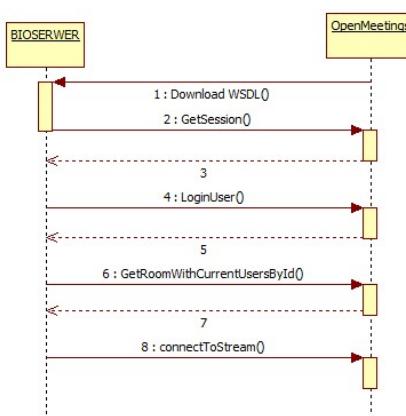
1. Użytkownik loguje się do systemu, ustawia opcje systemu, parametry *pokoju*;
2. Użytkownik wchodzi do *pokoju*, wybiera źródło sygnału audio/video



Rys. 4.5: Panel współdzielenia pulpitu

- wysyłanego na serwer, decyduje czy spotkanie ma być nagrywane;
3. Użytkownik kończy pracę w *pokoju*, System generuje i kataloguje pliki dodatkowe (np. nagrania);

Komunikacja z serwerem *OpenMeetings* odbywa się za pośrednictwem dokumentów *WSDL*. Dokumenty te określają jakie funkcjonalności udostępniane są przez serwer: nazwy funkcji, argumenty, wartości zwracane. Po pobraniu dokumentu, *Flex* posiada wbudowane opcje wykorzystania przetworzenia zawartości pobranego pliku *WSDL* na funkcje, które po wywołaniu powodują ściągnięcie wymaganych danych z serwera (patrz rysunek 4.6). Wpierw pobierany jest numer sesji użytkownika (2) i przeprowadzane logowanie do systemu (4). Następnie pobierana jest lista użytkowników danego *pokoju* (6). Jeśli Bioserwer zauważa w pokoju współdzielenie pulpitu lub kamery (są to dwa osobne strumienie danych), to zareaguje odpowiednio i podłączy się pod strumień danych ((8) z adresem w formacie: `rtmp://[OPENMEETINGS_URL]:[OPENMEETINGS_PORT]/[OPENMEETINGS_FOLDER]/[ID_POKOJU]`).



Rys. 4.6: Sekwencja stanów

Program OpenMeetings nie był jednak przygotowany na takie wykorzystanie. Funkcja zwracała jedynie pierwszy strumień na liście, a nie wszystkie dostępne w pokoju (np. z strumienia kamery trzech użytkowników + jeden strumień pulpitu, program zwróciłby jedynie jeden strumień o najmniejszym ID). Dlatego należało ściągać kod źródłowy programu, udostępniony na licencji Apache (można z niego korzystać pod warunkiem nie łamania praw patentowych oryginalnego właściciela) i przekompilować go zmieniając funkcje - `getClientListByRoom` w pliku `org/apache/openmeetings/session/SessionManager.java`, poprzez wykomentowanie ignorowania klientów AV i współdzielących pulpit. Po tej operacji Bioserwer miał pełen dostęp do zasobów OM.

W trakcie tworzenia serwera, należy pamiętać że program odbiera i wysyła dane na kilku portach, o odblokowaniu których trzeba pamiętać. Są to porty:

- port 1935 - port udostępniający połączenia *RTMP*
- port 5080 - port połączenia z główną aplikacją
- port 8088 - zapasowy port połączenia z *RTMP*

Program *OpenMeetings* nie był jednak przygotowany na takie wykorzystanie. Funkcja zwracała jedynie pierwszy strumień na liście, a nie wszystkie dostępne w pokoju (np. z strumienia kamery trzech użytkowników + jeden strumień pulpitu, program zwróciłby jedynie jeden strumień o najmniejszym ID). Dlatego należało ściągać kod źródłowy programu, udostępniony na licencji Apache (można z niego korzystać pod warunkiem nie łamania praw patentowych oryginalnego właściciela) i przekompilować go zmieniając funkcje - `getClientListByRoom` w pliku `org/apache/openmeetings/session/SessionManager.java`, poprzez wykomentowanie ignorowania klientów AV i współdzielących pulpit. Po tej operacji Bioserwer miał pełen dostęp do zasobów OM.

Numery portów można zmienić w pliku konfiguracyjnym *conf/red5.properties* (serwer) i *webapps/openmeetings/conf.xml* (klient). Obecnie Openmeetings został zainstalowany na serwerze dostępnym pod adresem: antares.cs.put.poznan.pl:5080.

4.5. Część Bioserwerowa - Video w czasie rzeczywistym

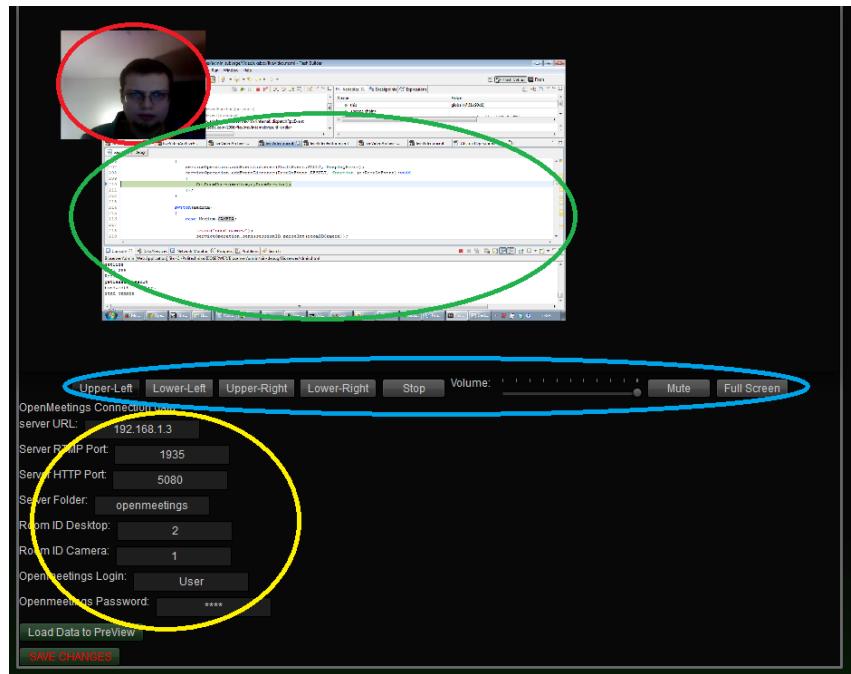
Aby wykorzystać możliwości jakie daje program *OpenMeeting*, należało wyposażyć Bioserwer w elementy, które będą dostosowane do odbioru danych wysyłanych przez *OpenMeetings*. Artykuły w Bioserwerze tworzą strukturę drzewa, gdzie liście to elementy końcowe widoczne dla użytkownika w danym artykule. Są one zapisane jako zbiór znaczników *XML* z opisem parametrów, to oznacza że dodanie nowego elementu (*LiveVideo* - ang. *Video w czasie rzeczywistym*) ograniczyło się do stworzenia nowej klasy i parsowania jej parametrów do i z bazy danych.

Na rysunku 4.7 zaprezentowany jest edytor elementu *LiveVideo*. Sam element składa się z trzech części, które ostatecznie są w takiej formie widoczne dla użytkownika końcowego w Bioserwerze:

- strumienianego obrazu z kamery (czerwona obwódka na obrazie)
- strumienianego obrazu z pulpitu komputera prowadzącego transmisję (obwódka zielona)
- panel sterujący (obwódka niebieska)

Sposób połączenia Bioserwera i *OpenMeetings* został już przedstawiony na rysunku 4.6. Element będzie domyślnie szukał ścieżki audio w połączeniu kamery. Zaś obraz uzyskiwany jest z dwóch różnych *pokoi* (obraz pulpitu z *pokoju konferencyjnego* (*Conference Room*), obraz kamery z *pokoju rozmów kwalifikacyjnych* (*Interview Room*)).

Panel sterujący oprócz możliwości zatrzymania/wznowienia połączenia (*Stop/Play*) i zmiany głośności (pasek głośności i przycisk *Mute*), posiada również możliwość przełączenia domyślnego położenia kamery (z lewego-górnego ang. *Upper-left* na dowolne) w każdym momencie odtwarzania połączenia. Możliwość ta została dodana, by obraz kamery nie przysyłał ważnego fragmentu prezentacji w trakcie konferencji. Przycisk *Fullscreen* umożliwia przejście do trybu pełnoekranowego, który powoduje usunięcie stopki po lewej i loga u góry strony (patrz rysunek 2.10) przez co zwiększa się obszar roboczy aplikacji. Jest to przydatne gdy w trakcie konferencji prezentowane są teksty drobnym drukiem, które w innym przypadku są trudno widoczne.



Rys. 4.7: System Zarządzania Treścią - połączenie czasu rzeczywistego

Elementy z żółtej obwódki, umożliwiają edycję parametrów połączenia. Parametry te opisują (od góry):

- adres server *OpenMeetings* (*serverURL*),
- port serwera *RTMP* (*RTMP server port*) - port po którym przesyłane są strumienie multimedialne,
- port *HTTP* (*HTTP*) - port na którym możliwe jest połączenie z programem *Openmeetings* - w celu pobrania pliku *WSDL*,
- ścieżka na serwerze do zawartości *OpenMeetings* (*Server folder*),
- ID pokoju udostępniającego pulpit (*Room ID Desktop*),
- ID pokoju udostępniającego widok kamery (*Room ID Camera*),
- Login do *OpenMeetings* (*OpenMeetings Login*),
- Hasło do *OpenMeetings* (*OpenMeetings Password*)

Dodatkowo dwa przyciski poniżej służą do załadowania ustawień do podglądu (*Load Data to PreView*) i zapisania ich na serwerze (*SAVE CHANGES*).

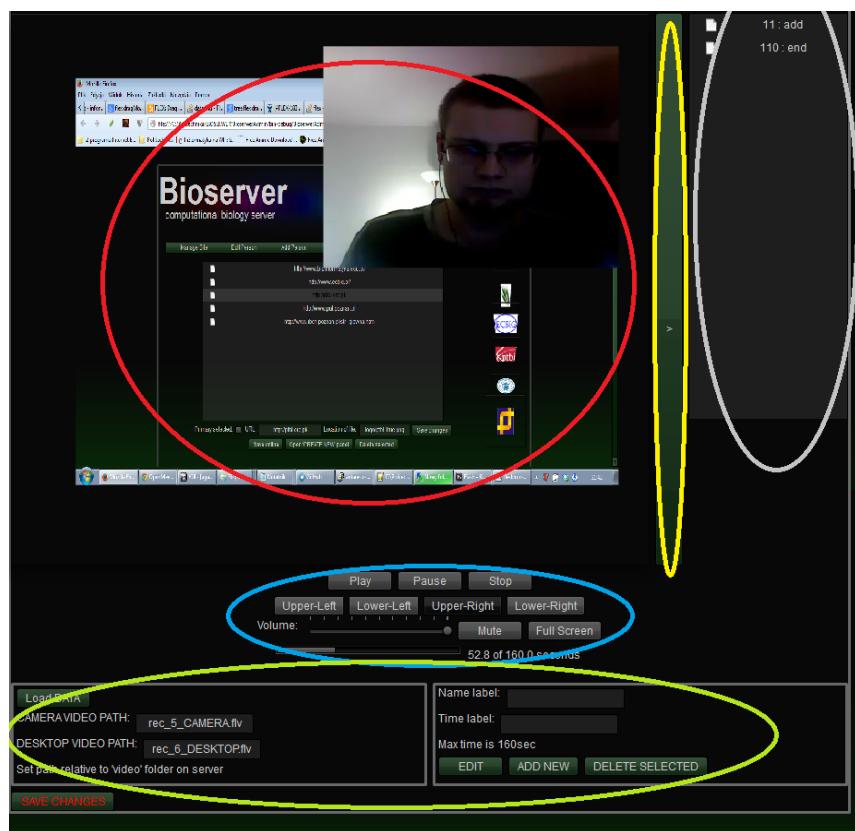
4.6. Część Bioserwerowa - Archiwum Teleseminariów

Archiwum *Teleseminariów*, zostało stworzone z myślą o przechowywaniu nagrań stworzonych przez *OpenMeetings* w trakcie transmisji obrazu. Na rysunku 4.8 za-

prezentowano wygląda edytora elementu archiwalnego w panelu administratorskim. Składa się on z:

- nagranie kamery i pulpitu (kolor czerwony),
- po prawej stronie (kolor biały) stworzono listę zakładek dla danego archiwum. Zakładki służą szybkiej nawigacji po odtwarzanym nagraniu. Składnia zakładki wygląda następująco: [CZAS_W_SEKUNDACH]:[ETYKIETA]. Naciśnięcie na zakładkę powoduje przesunięcie nagrania do danego momentu. Zakładki są sortowane rosnąco po atrybutie czasu,
- przycisk pokazaniu/ukryciu paska zakładek (kolor żółty),
- panel sterujący (kolor niebieski),

Na panelu sterującym znajdują się przyciski pozwalające tymczasowo zatrzymać (*Pause*), całkowicie zatrzymać (*Stop*) lub wznowić (*Play*) odtwarzanie. Poniżej są kontrolery położenia kamery na obszarze czerwonym (na rysunku 4.8 w pozycji prawy-górny róg - ang. *Upper-right*). Przyciski manipulacji głosem i trybem pełnoekranowym zostały już opisane w punkcie 4.5. Ostatnim elementem sterującym jest pasek postępu. Naciśnięcie na pasek powoduje przeskoczenie do danej sekundy nagrania.



Rys. 4.8: System Zarządzania Treścią - nagranie archiwalne

Panel administracyjny - w sekcji zielonej, składa się z dwóch części.

- Po lewej, należy podać ścieżki plików nagrań - powinny one się znajdować w serwerowym folderze *video*. Przyciskiem *Load DATA* system przeładowuje się aktualnie odtwarzane nagranie danymi z lewej kolumny.
- W kolumnie po prawej można edytować zakładki. Wybranie danej zakładki powoduje załadowanie jej danych do odpowiednich pól edytujących. Pole *Name label* odpowiada za nazwę etykiety, a *Time label* za czas w sekundach. Wyedytowanie wybranej zakładki następuje po naciśnięciu przycisku *EDIT*. Przycisk *DELETE SELECTED* usunie wybraną zakładkę, a *ADD* doda nową z danymi aktualnie wpisanymi w pola edytujące.

4.7. Wyniki końcowe

W ramach projektu udało się stworzyć stabilną bazę pod działanie systemu *Teleseminariów*. Przygotowano *Bioserwer* na przyjęcie nagrań video i administrowanie nimi. Zainstalowano również program *OpenMeetings* na serwerze Uczelnianym. Odblokowano porty: 80, 5080, 8088 w salach 1.21, 1.23 i 13 w budynku CW. Niestety w sali 13 brakuje odpowiedniego sprzętu do przekazywania dźwięku z mikrofonów do urządzeń zewnętrznych, dlatego pełna praca z systemem *Teleseminariów* jest możliwa jedynie w dwóch innych salach. System został przetestowany i jest gotów do użycia.

5. Podsumowanie

Dbanie o funkcjonalność i wygląd strony internetowej, oprócz tego że realizuje podstawowe wymagania klienta końcowego i poprawia jej użyteczność, jest także wyrazem szacunku dla niego. Akualizowana strona, nie jest stroną martwą, wskazuje na rozwój instytucji którą dana strona reprezentuje. Dlatego też wprowadzanie nowych funkcjonalności jest swoistym manifestem celów do jakich dąży ta organizacja. Wprowadzenie nagrań ze spotkań wskazuje na otwartość i chęć dzielenia się wiedzą.

Aby można było tą wiedzą podzielić się skutecznie zintegrowano *Bioserwer* z systemami wyszukiwania *Google*, dzięki czemu strona jest dostępna dla użytkownika końcowego z każdego miejsca na świecie. Dla skuteczniejszej manipulacji treścią znajdującą się na stronie wprowadzono system ról, który ułatwia zarządzanie i sprawia że wiedzę na dany temat dostarcza osoba najlepiej do tego zadania dostosowana. Przygotowano również narzędzia do strumieniowania danych multimedialnych przez Internet, które można wykorzystać nie tylko z *Bioserwerem*, ale z każdym systemem wspierającym odbiór danych multimedialnych

Ostatecznie wreszcie w ramach niniejszej pracy mogłem nauczyć się wielu rzeczy z zakresu nowoczesnych technologii internetowych, które z pewnością przydadzą się w dalszej pracy zawodowej.

A. Załączniki

A.1. Słownik

Pojęcie/Skrót	Definicja
<i>AAC</i>	ang. <i>Advanced Audio Coding</i> - zaawansowany koder audio - algorytm kodowania dźwięku
<i>Adobe Player</i>	Darmowa wtyczka do przeglądarek internetowych umożliwiająca wyświetlanie treści stworzonych w języku <i>Adobe Flash</i>
<i>Bioserwer</i>	Strona internetowa grupy Bioinformatycznej Politechniki Poznańskiej - temat niniejszej pracy magisterskiej
<i>Crawler</i>	Robot indeksujący, program komputerowy który automatycznie wizytuje i indeksuje parametry stron internetowych
<i>CSS</i>	ang. <i>Cascading Style Sheets</i> - Kaskadowe Arkusze Stylów - język pomocniczy używany do masowego opisywania elementów (np. strony internetowej <i>HTML</i>)
<i>CW</i>	Centrum Wykładowe - budynek Politechniki Poznańskiej
<i>Google</i>	Amerykańska międzynarodowa korporacja specjalizująca się w usługach internetowych, a także nazwa wyszukiwarki internetowej dostępnej na stronie www.google.com
<i>IP</i>	ang. <i>Internet Protocol</i> - Protokół Internetowy: zbiór reguł określających sposób wymiany danych w modelu OSI w wątku sieciowym
<i>JSON</i>	ang. <i>JavaScript Object Notation</i> - standard opisujący struktury danych
<i>JVM</i>	ang. <i>Java Virtual Machine</i> - Maszyna Wirtualna Java która pozwala na wykonywanie kodu bajtowego Javy i programów do niego kompilowanych
Kamera <i>IP</i>	Kamera IP to cyfrowe urządzenie do przekazywania obrazu i dźwięku za pomocą Internetu lub sieci wewnętrznej
<i>Live Streaming / Live Video</i>	ang. Media Strumieniowe - treść multimedialna (video i audio) nadawana przez połączenie internetowe w czasie rzeczywistym
<i>MP4 / MPEG4</i>	ang. <i>Moving Picture Group Layer 4</i> - grupa standardów opisująca kodowanie audio i video

<i>OM</i>	<i>Open Meetings</i> patrz punkt 2.2.4
<i>RIA</i>	ang. <i>Rich Internet Application</i> - Bogate Aplikacje Internetowe - termin opisujący aplikacje oferującą jednoekranowy, dynamiczny interfejs
<i>RTMP</i>	ang. <i>Real Time Messaging Protocol</i> - protokół przesyłania danych w Internecie stworzony przez <i>Adobe Systems</i>
<i>SGML</i>	ang. <i>Standard Generalized Markup Language</i> - Standardowy Uogólniony Język Znaczników służący do ujednoliciania struktury i formatu różnego typu informacji
<i>SOAP</i>	ang. <i>Simple Access Object Protocol</i> - protokół wywoływania zdalnego dostępu do obiektów
<i>SZT</i>	System Zarządzania Treścią
<i>ToS</i>	ang. <i>Terms of Service</i> - licencja użytkownia
<i>WSDL</i>	ang. <i>Web Services Description Language</i> - oparty na <i>XML</i> język definiowania usług sieciowych
<i>XML</i>	ang. <i>Extensible Markup Language</i> - Rozszerzalny Język Znaczników - język służący do przedstawiania danych w ustrukturalizowany sposób

A.2. Zestawienie kamer USB

Web-Kamery – wybrane modele

2.03.2012

Wstęp

Spis treści

- 1** Wstęp
 - 2** Tracer Rocket HD
 - 2** Genius FaceCam 2000
 - 3** Logitech C270 HD
 - 3** Logitech C525 HD
 - 4** Logitech C615 HD
 - 4** Logitech C910 HD
 - 5** Podsumowanie + dodatkowy zakup
- Poniżej przedstawiam zestawienie proponowanych przez mnie kamer HD, które mogłyby posłużyć w realizacji projektu Tele-seminariów.
- Cena danego modelu została podana z serwisu Skapiec. Cena obejmuje zakres od ceny minimalnej do średniej.
- Należy zwrócić że web kamery tutaj przedstawione są raczej małe w swoim rozmiarze – przybliżone porównanie do rozmiaru laptopa widać w pierwszym prezentowanym produkcie – po szczegółowe porównania należy udać się do linków z recenzjami.



Tracer Rocket HD

Najtańsza wśród znalezionych kamer.

Dane:

- długość przewodu: 1.4 m;
- przetwornik: 2 Mpx;

Plusy:

- cena
- możliwość dowolnej manipulacji kamery (kierunki horyzontalne i wertykalne);

Minusy:

- praktycznie 'no-name' nie wiadomo jak będzie wyglądała kwestia sterowników i przechwytywania obrazu;
- brak recenzji w Internecie;

Dodatkowe uwagi:

Cena: 67 zł – 77 zł

Genius FaceCam 2000

Drugi model charakteryzuje się o wiele lepszymi możliwościami postawienia go na powierzchniach płaskich, jednak brak informacji o długości kabla może być kłopotliwy(zawsze można zastosować jakiś przedłużacz USB).

Dane:

- Długość kabla: ? m;
- Przetwornik: 2 Mpx;

Plusy:

- Duża 'mobilność' kamery;
- łatwe mocowanie i stabilne ;
- cena;

Minusy:

- 'no-name' – brak informacji o zewnętrznym przechwytywaniu obrazu;
- krótki kabel USB (wg komentarzy);
- brak recenzji w internecie;
- brak opcji – Auto-focus

Dodatkowe uwagi:

Cena: 76 zł – 92 zł



Logitech C270 HD



Jeden z pierwszych modeli z którym nie powinno być problemów przy pracy, recenzje wyrażają się pochlebnie o Nim (ze względu na stosunek jakość/cena).

Dane:

- Długość kabla: 1.5 m;
- Przetwornik: 3 Mpx;

Plusy:

- cena;
- Technologia Right-Light – dobra praca przy słabym oświetleniu;
- Auto-Focus;

Minusy:

- Kamera porusza się jedynie w 'pionowym' wymiarze (nie na boki);

Dodatkowe uwagi/recenzje:

<http://www.youtube.com/watch?v=6ToDM2PWtQE&feature=related>

<http://www.youtube.com/watch?v=XHlUxSkuqTs&feature=fvwrel>

Cena: 80zł – 110 zł

Logitech C525 HD

Rozwinięcie modelu podanego powyżej, umożliwia pracę na pełne 360 stopni + lepszą jakość obrazu.

Dane:

- Długość kabla: 1.5 m;
- Przetwornik: 8 Mpx;

Plusy:

- Technologia Right-Light 2 – do pracy w środowisku pół-ciemnym;
- pełna 'mobilność' kamery;
- Auto-focus;

Minusy:

- cena

Dodatkowe uwagi:

<http://www.youtube.com/watch?v=r5odrTJGxdA> – nagranie w piwnicy

Cena: 165zł – 190zł



Logitech C615 HD

Prawdopodobnie najlepszy model, ograniczenie długości kabla, można obejść stosując przedłużacz, jednakże w porównaniu z poprzednim modelem można zaobserwować znaczny skok cenowy.

Dane:

- Długość kabla: 1 m;
- Przetwornik: 8 Mpx;

Plusy:

- Technologia Right-Light 2 – do pracy w środowisku pół-ciemnym;
- pełna 'mobilność' kamery;
- posiada podstawę pod statyw (brak statyw w zestawie);
- Auto-focus;

Minusy:

- cena;
- trocę krótki kabel, w zależności od sali może być potrzebny przedłużacz USB;

Dodatkowe uwagi:

<http://www.youtube.com/watch?v=95BUIpTmxUw&feature=related>
<http://www.youtube.com/watch?v=K36u9lh9TlI&feature=related>

Cena: 294zł – 329zł



Logitech C910 HD

Model ten prezentuję najlepszą jakość nagrywania (której pewnie ze względu na ograniczenia przesyłu sieci nie wykorzystamy), jednakże jest ograniczony pod względem ergonomicznym i trudniej by się go używało niż kilku wyżej wymienionych tańszych modelu.

Dane:

- Długość kabla: 1.5 m;
- Przetwornik: 10 Mpx;

Plusy:

- Technologia Right-Light 2 – do pracy w środowisku pół-ciemnym;
- Auto-focus;

Minusy:

- cena;
- Brakuje możliwości poruszania kamery horyzontalnie (lewo-prawo);
- Nóżka jest trochę mała, co może oznaczać, że kamera nie będzie samodzielnie stała;

Dodatkowe uwagi:

<http://www.notebookcheck.net/Review-Microsoft-HD-6000-and-Logitech-C910-Professional-HD-Webcam.42710.0.html> - porównanie do kamery Microsoftu HD-6000 (kamera Microsoftu ma bardzo krótki kabel – 50 cm, co może nie stać się na nasze potrzeby);

Cena: 347 – 398zł

Podsumowanie + dodatkowy zakup

Pod adresem: <http://www.digitalversus.com/webcam/webcam-reviews-a287.html> znajduje się dodatkowy test większości modeli prezentowanych w tym sprawozdaniu.

Alternatywą dla Web-kamer byłby zakup kamery cyfrowej, jednakże należałoby się wtedy upewnić, że taka kamera ma możliwość przekazywania obrazów do komputera za pomocą kabla USB (aby umożliwić tworzenie nagrań LIVE), cena najtańszego modelu który zapewniał w opisie taką usługę wynosiła: 500 zł (być może istnieją tańsze odpowiedniki, ale zdobycie większej ilości informacji wymagałoby czasu).

Uważam, że przy podjęciu decyzji należy się skupić nad modelami C270 / C525 / C615. Dodatkowo należy jednakże dokonać zakupu statywów na których można by umieścić zakupioną kamerę.

Uniezależniłby to nas od dostępności stolów/krzesel w danej sali w której odbywałoby się seminarium. Koszt zakupu statywów to około: 20 zł, poniżej umieszczam kilka linków allegro:

- <http://allegro.pl/statyw-aluminiowy-114cm-3d-tr130-futeral-poziomica-i2163499954.html>
- <http://allegro.pl/3d-zlozony-tylko-35cm-wysoki-statyw-gratis-i2173780230.html>



Bibliografia

- [ADO13] Adobe. Internet:
http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/index.html, ostatnia aktualizacja: 2013.
- [Cha01] Maria Chałon. *Systemy Baz Danych - Wprowadzenie*. Oficyna wydawnicza Politechniki Wrocławskiej, drugie edition, 2001.
- [Dom10] Marek Domański. *Obraz cyfrowy. Reprezentacja, kompresja, podstawy przetwarzania. Standardy JPEG i MPEG*. Wydawnictwo Komunikacji i Łączności, 2010.
- [JS10] Vince Genovese Jeanette Stallon, Andres Shorten. *Getting Started with Flex 4*. Adobe Developer Library, first edition, 2010.
- [MAT13] Vector fields. Internet:
<http://mathforum.org/mathimages/index.php/VectorFields>, ostatnia aktualizacja: 2013.
- [PHP13] Php foundation. Internet:
<http://www.php.net/manual/en/>, ostatnia aktualizacja: 2013.
- [Sok02] Maria Sokół. *ABC języka HTML*. Wydawnictwo Helion, 2002.
- [SQL13] Mysql developers. Internet:
<http://dev.mysql.com/doc/>, ostatnia aktualizacja: 2013.
- [Ste10] Stoyan Stefanov. *JavaScript Programowanie obiektowe*. Wydawnictwo Helion, 2010.
- [TRE13] Kamery-ip. Internet:
<http://www.kamery-ip.com/TRENDSnet-TV-IP501P-6605.html>, ostatnia aktualizacja: 2013.
- [WIK11] Wikipedia. Internet:
<http://en.wikipedia.org/wiki/File:InsulinHexamer.jpg>, ostatnia aktualizacja: 2011.
- [ZON13] Zoneminder. Internet:
<http://www.zoneminder.com/sites/zoneminder.com/files/Screenshot-Homeontage>