



Złożenie pracy online:
2012-02-18 00:07:59
Kod pracy:
6883

Łukasz Sawaściuk
(nr albumu: 17954*INF/LIC)

Praca inżynierska

Edvir - Platforma wspomagająca przeprowadzanie zdalnych lekcji w oparciu o komunikację audiowizualną

Edvir - The platform supporting performing remote lessons based on audiovisual communication

Wydział: Informatyki

Kierunek: Informatyka

Specjalność: technologie multimedialne

Promotor: dr Franciszek Białas

SPIS TREŚCI

1. Wstęp.....	4
1.1 Wprowadzenie	4
1.2 Cele projektu	4
2. Dobór technologii.....	5
2.1 Środowisko uruchomieniowe	5
1.1 Narzędzia programistyczne	5
1.1.1 Adobe Flash Professional CS5.5	5
1.1.2 Eclipse Indigo IDE.....	6
3. Opis funkcjonalności.....	7
3.1 Koncepcja interfejsu	7
3.2 Logowanie	8
3.3 Uczeń.....	10
3.3.1 Ekran główny.....	10
3.3.2 Lekcja i klasa	14
3.3.3 Profil	17
3.4 Nauczyciel	17
3.4.1 Ustawienia lekcji.....	18
3.4.2 Przebieg lekcji.....	23
3.4.3 Aktywność uczniów.....	24
3.5 Praca grupowa	26
3.5.1 Komunikacja wewnątrz	26
3.5.2 Dokument grupowy	27
4. Realizacja projektu.....	29
4.1 Klient Programu	29
4.1.1 Aplikacja SWF.....	29

4.1.2	Wrapper WWW	30
4.2	Serwer multimedialny	31
4.2.1	Wybrana specyfikacja	32
4.2.2	Aplikacja serwerowa	33
4.2.3	Streaming audio-Video	35
4.2.4	Skalowalność	36
4.3	Komunikacja klient-serwer	37
4.3.1	Metody zdalne	37
4.3.2	Funkcje Broadcast'owe	38
4.3.3	Obiekty współdzielone	39
4.4	Baza danych	41
4.4.1	Struktura XML	41
4.4.2	Przechowywanie danych	41
4.4.3	Bezpieczeństwo	42
5.	Podsumowanie	42
5.1	Rezultat końcowy	42
5.2	Zastosowanie	43
	Bibliografia	44
	Opis ilustracji	44

1. WSTĘP

1.1 WPROWADZENIE

E-learning – jest zelektronizowaną formą nauczania, gdzie tradycyjne metody dydaktyki wspierane są przez nowoczesne urządzenia takie jak komputery osobiste, sprzęt mobilny, Internet oraz wszelkiego rodzaju elektroniczne nośniki danych. Systemy e-learning’owe najczęściej są jednym z elementów tradycyjnych kursów, jednak z racji swoich ogromnych możliwości stanowią trzon nauczania tuż obok typowych spotkań. Najnowsze platformy pozwalają na przeprowadzanie szkoleń i ukończenie kursów wraz z egzaminem w pełni zdalnie, zaletą takich kursów jest dostosowanie tempa nauczania do indywidualnych predyspozycji oraz nie jest uzależnione od miejsca zamieszkania. Szkolenia zdalne za pośrednictwem Internetu mają nieograniczony obszar działania, co zwiększa ich popularność.

Z ekonomicznego punktu widzenia usługi e-learning’owe są tańsze zarówno dla usługodawcy jak i usługobiorcy. Pozwalają na ograniczenie personelu nienaukowego uczelni, lokali do przeprowadzania zajęć, a ze strony kursanta nie generują kosztów podróży lub zmiany miejsca zamieszkania. Niskie koszty prowadzenia kursów sprzyjają zwiększeniu ich ilości i różnorodności.

Znakomita większość programów edukacyjnych, przy wszystkich swych zaletach związanych z wykorzystaniem nowoczesnych urządzeń i nieograniczonemu dostępowi do informacji elektronicznych, zatracą w swej konstrukcji czynniki ludzkie, takie jak bezpośrednia rozmowa z prowadzącym oraz rozmowy i kontakt audiowizualny pomiędzy uczestnikami. Zgubiono również poczucie jedności grupy, pracy zespołowej, a także konstruktywnej rywalizacji. Typowym systemom brakuje możliwości przekazywania zarówno komunikatów werbalnych jak i niewerbalnych, wyrażających zdziwienie, zaciekawienie, aprobatę itp.

1.2 CELE PROJEKTU

Celem projektu było stworzenie podstaw do systemu przeprowadzania zdalnych lekcji łączącego w sobie szeroką dostępność oraz swobodę w korzystaniu z kursów za pośrednictwem Internetu, z komunikacją audiowizualną aby nie utracić w swym przekazie elementów typowo ludzkich, tak mocno zakorzenionych w naszej naturze jako podstawowa forma przekazywania wiedzy. Następnymi celami było zapewnienie pełnej funkcjonalności dla przeprowadzania typowych wykładów, prac grupowych, zadawania pytań oraz dystrybucji materiałów do zajęć. Cały system powinien być łatwy w obsłudze i zarządzaniu, a ze strony użytkowników dostępny na różnych platformach i systemach operacyjnych.

2. DOBÓR TECHNOLOGII

2.1 ŚRODOWISKO URUCHOMIENIOWE

Przy doborze technologii do realizacji niniejszego projektu kluczowym kryterium wyboru była możliwość uruchomienia aplikacji klienckich na różnych platformach. Jednym z założeń projektu było stworzenie systemu dającego swobodę i wygodę korzystania, a więc priorytetem stała się możliwość uruchomienia aplikacji nie tylko na komputerach stacjonarnych ale także na tabletach, telefonach typu Smartphone i komputerach kieszonkowych PDA. Wsparcie dla urządzeń mobilnych jak również tabletów i komputerów przenośnych z różnym oprogramowaniem wymagało znalezienia optymalnego i uniwersalnego rozwiązania. Najlepszą platformą okazał się *Adobe Flash Player*. Według statystyk, nowo sprzedawane komputery osobiste wyposażone w urządzenia do połączenia z Internetem w 99% przypadków posiadają zainstalowane oprogramowanie *Adobe Flash Player* (badanie Millward Brow, przeprowadzone w lipcu 2011), podczas tego samego badania dowiedziono obecność *Flash Player'a* na 60% produkowanych obecnie telefonach typu Smartphone. *Adobe Flash Player* jest obecny na większości nowoczesnych Tabletów z różnymi systemami operacyjnymi. Penetrację tego standardu zwiększają także wbudowane autorskie wtyczki w popularnych przeglądarkach internetowych.

Technologia Flash daje ogromne możliwości pod względem efektownego wyglądu interfejsu użytkownika i łatwego budowania aplikacji w obiektowym języku *ActionScript 3.0*. *Adobe Flash* posiada wsparcie dla obsługi kamer, mikrofonów, a także sprzętowego kodowania i dekodowania materiałów wysokiej jakości H.264.

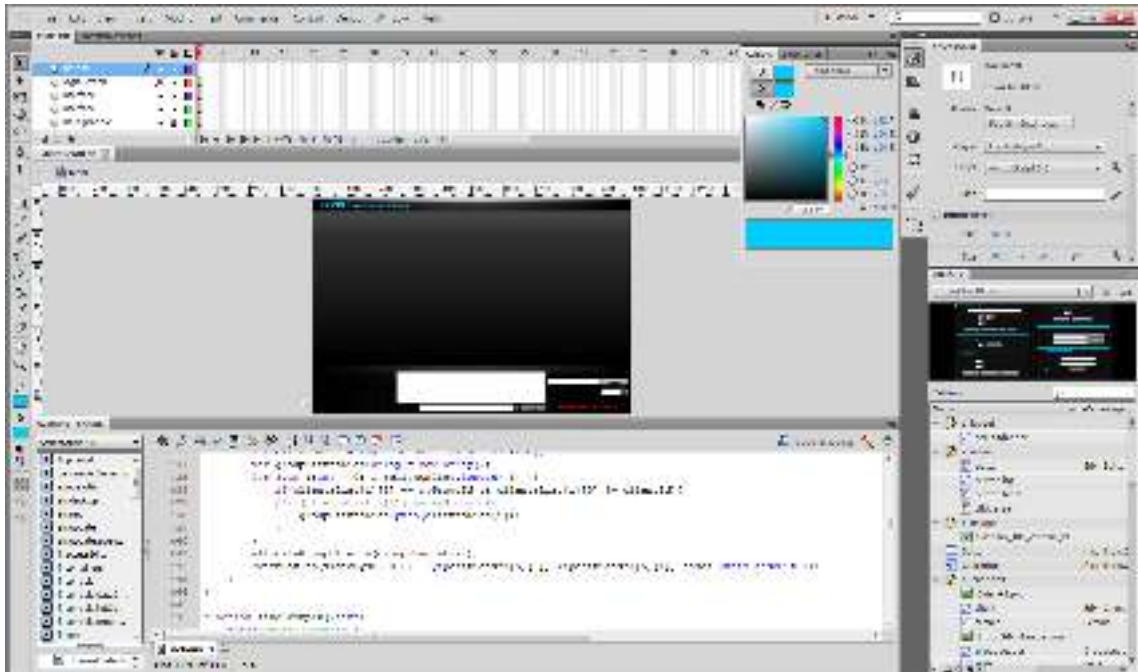
W przypadku serwera multimedialnego wybór padł na *Wowza Media Server*. Produkt jest rozwiązaniem komercyjnym jednak daje możliwość korzystania z wersji bezpłatnej dla osób chcących wypróbować to rozwiązanie. Jedynym ograniczeniem wersji próbnej jest możliwość jednoczesnego podłączenia do 10 użytkowników. Obecność płatnej wersji na rynku dostarcza firmie stałe przychody, a więc *Wowza Media Server* jest produktem bardzo dobrze dopracowanym i otrzymuje stałe aktualizacje. Pod względem obsługi formatów oraz łatwości pisania aplikacji serwerowych, jest bezkonkurencyjnie lepszy od promowanego przez firmę *Adobe, Media Interactive Server* oraz *Red5*, tworzonego na zasadach OpenSource.

1.1 NARZĘDZIA PROGRAMISTYCZNE

1.1.1 ADOBE FLASH PROFESSIONAL CS5.5

Flash Professional to kompleksowe narzędzie przeznaczone do tworzenia rozbudowanych i atrakcyjnych wizualnie aplikacji. Pełne multimedialnych rozwiązań programy są ciekawe dla użytkownika, przejrzyste i funkcjonalne. Pliki wyjściowe mogą być samodzielnymi programami, umieszczone na stronach WWW w formie

elementów takich jak banery reklamowe, animacje. Możliwości Flash Professional pozwalają także na stworzenie pełnego projektu strony internetowej. W aplikacjach Flash mogą być wykorzystywane m.in. formaty MP3, FLV, MP4. Flash od wersji 9 pozwala również na programowanie w języku obiektowym ActionScript 3.0



Rys. 2.1 Adobe Flash Professional CS5.5 edycja Klienta systemu.

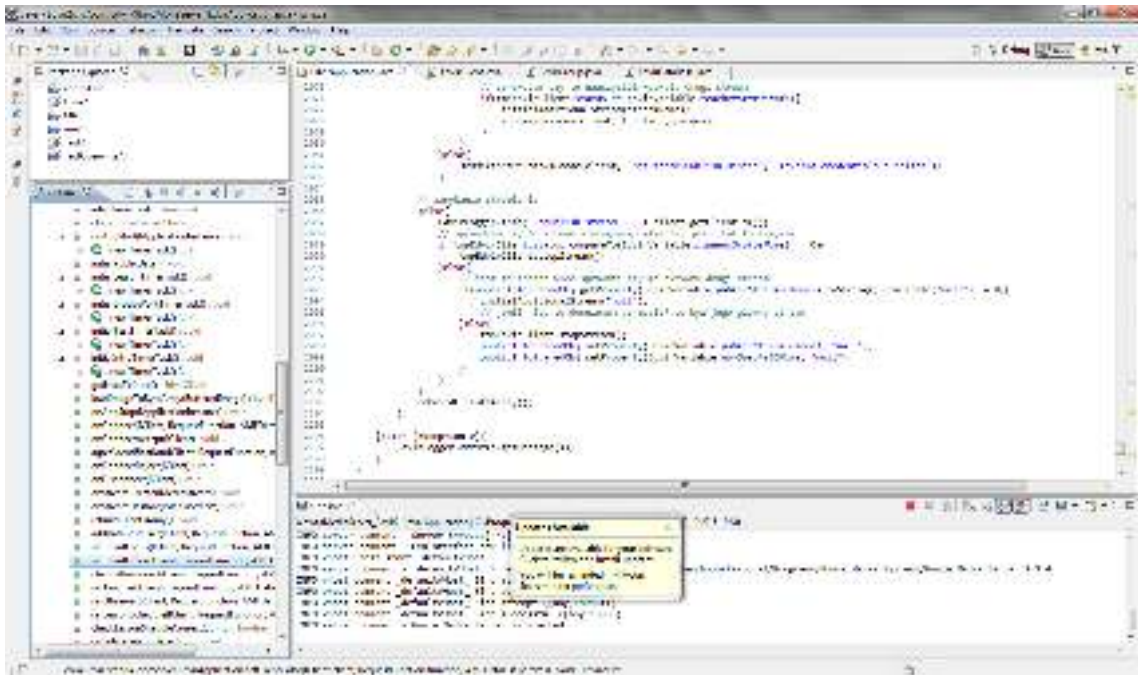
Interfejs użytkownika zbudowany jest z paneli funkcyjnych zawierających w sobie narzędzia z wspólnej dziedziny. Flash Professional posiada 5 predefiniowanych ustawień interfejsu dla zoptymalizowania pracy np. animatora, programisty, designera. Po wybraniu której z konfiguracji zostają wyważone rozstawienia kluczowych paneli takich jak: *oś czasu*, *biblioteka*, *akcje*, *właściwości*. Panele dają się łatwo przestawiać wg własnych upodobań i predyspozycji. Można np. rozdzielić pracę z programem na dwa monitory w trybie pełnoekranowym.

Program zapewnia pełną kompatybilność z innymi produktami firmy Adobe, które wspierają tworzenie grafiki wykorzystywanej w aplikacjach Flash. Istnieje sposobność importowania obrazów wektorowych z Adobe Ilustrator oraz rastrowych z Adobe Photoshop. Dzięki integracji Adobe Device Central możliwy się podgląd wyglądu tworzonej aplikacji na urządzeniach mobilnych, a po odpowiednim skonfigurowaniu nieodpłatnych dodatków ze strony Adobe, możliwe staje się tworzenie aplikacji na systemy iOS oraz Android.

1.1.2 ECLIPSE INDIGO IDE

Eclipse to zintegrowane środowisko programistyczne do tworzenia aplikacji. Platforma została napisana w języku Java i również dla tworzenia aplikacji w tym języku jest rozpowszechniana w podstawowej odmianie. Eclipse został tak

zaprojektowany, aby można było tworzyć dla niego wtyczki mocno ingerujące w platformę i znacznie rozszerzające jej funkcjonalność.



Rys. 2.2 Eclipse Indigo IDE 3.7.1 – otwarty projekt aplikacji, uruchomiony serwer Wowza.

Spółeczność wokół Eclipse, jak również sam wydawca, Eclipse Foundation, dostarcza wtyczki wzbogacające platformę o narzędzia służące do tworzenia kodu, budowania aplikacji, m.in. pisanych w językach Java, C/C++, tworzenie graficznych interfejsów, modelowanie aplikacji z użyciem UML. Platforma zapewnia współpracę z serwerami baz danych oraz serwerami aplikacji. W omawianym projekcie Eclipse posłużył do stworzenia aplikacji serwerowej. Dodatek udostępniany przez Wowza Media Systems (producenta Wowza Media Server), umożliwia połączenie Eclipse z lokalnym serwerem, sterowanie jego pracą, tworzenie nowych aplikacji i debugowanie.

3. OPIS FUNKCJONALNOŚCI

3.1 KONCEPCJA INTERFEJSU

Plany odnośnie interfejsu zakładały stworzenie prostego i przejrzystego *GUI* (*Graphical User Interface*) które będą mogły obsługiwać osoby o niskim poziomie wiedzy komputerowej. Z założenia komputer i jego obsługa jest tylko narzędziem w pozyskiwaniu wiedzy na konkretny temat, a więc duży nacisk położono na wyeliminowanie konieczności przeszkolenia w zakresie obsługi samego programu, zanim jeszcze rozpocznie się właściwe szkolenie w ramach kursu. Wszystkie aktywne obiekty oraz informacje które w danej chwili nie są potrzebną, są ukrywane. Interfejs staje się bardziej przejrzysty i pozwala skupić się na temacie zajęć. Wszystkie

informacje są w miarę możliwości, wyczerpująco opisane i jednoznaczne. Jasność obsługi programu pozwala również na obniżenie progu wiekowego i tym samym wykorzystanie go do wspomagania szkolnictwa nawet w najniższych klasach szkół podstawowych.

Interfejs nauczyciela z racji większych wymagań, jest również bardziej rozbudowany. Dodano dodatkowe zakładki w menu głównym które nie przeszkadzają w prowadzeniu lekcji i są wykorzystywane sporadycznie. Najczęściej używane funkcje do prowadzenia zajęć zostały wyciągnięte na główny ekran, aby można z nich było szybko korzystać bez potrzeby zagłębiania się w dodatkowe menu, zakładki itp.



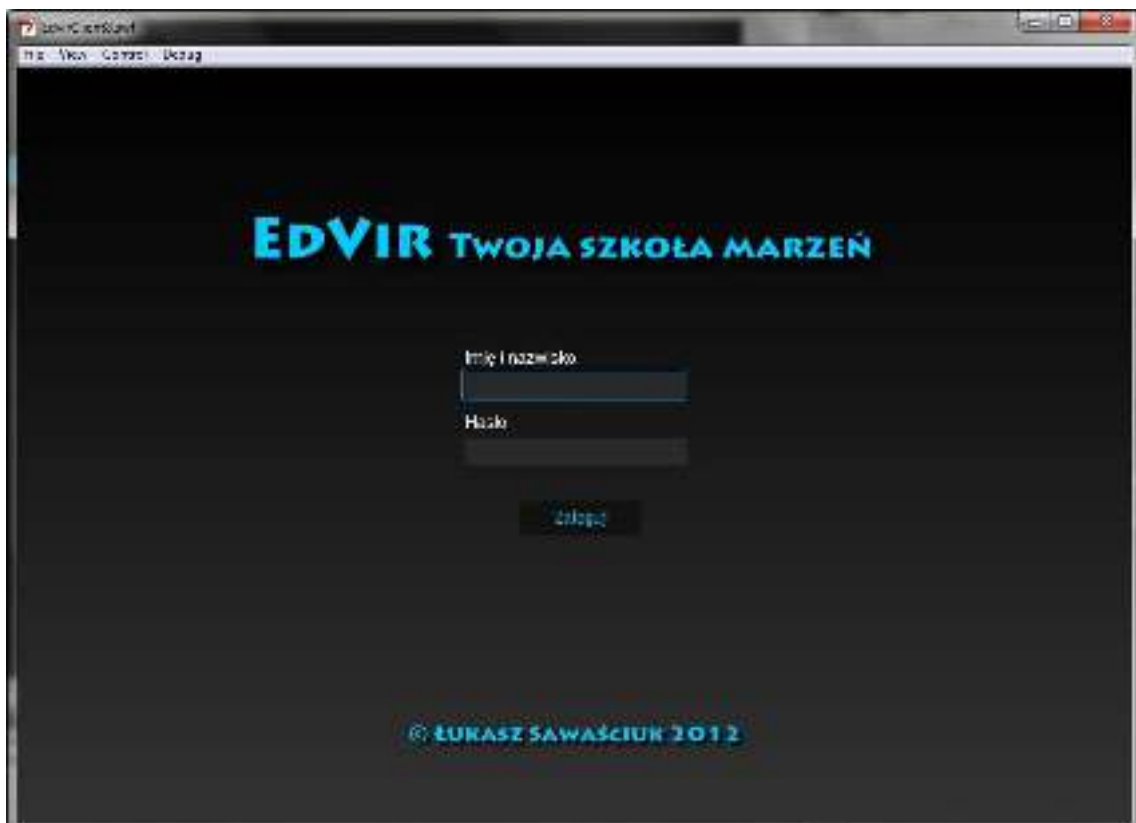
Rys. 3.1 Klient aplikacji w trakcie testów w środowisku sieciowym

Powyższa ilustracja została wykonana podczas testów sieciowych z wykorzystaniem trzech komputerów. Większość prac nad systemem przeprowadzana była lokalnie na pojedynczym komputerze dlatego na rysunkach opisujących poszczególne sekcje programu widoczna jest ta sama osoba. Faktycznie są to zupełnie inne obiekty wideo i mimo podobnego obrazu stanowią niezależne transmisje audiowizualne.

3.2 LOGOWANIE

Logowanie do systemu odbywa się przy pomocy uwierzytelnienia, w skład którego wchodzi login (najlepiej imię i nazwisko rozdzielone spacją) oraz hasło. Zapisanie loginów i haseł w formie kodowania UTF-8, umożliwiło rezygnację z tradycyjnych

loginów na rzecz pełnych imion i nazwisk z polskimi znakami. Uczeń nie musi pamiętać pary wyrazów: login + hasło, wystarczy jedynie że zna swoje hasło, do którego zapamiętania i tak zawsze przywiązuje się większą wagę niż loginu. Wiele dzisiejszych serwisów internetowych również stara się pomóc użytkownikom w zapamiętaniu loginu, prosząc o adres e-mail jako nazwę użytkownika. Biorąc pod uwagę domniemanie prywatności w Internecie jest to jednak pół środek pomiędzy anonimowością i łatwością zapamiętania. W omawianym systemie, który de facto jest szkołą i nigdzie indziej nie wykorzystuje danych użytkownika można było wykorzystać Imię i nazwisko, które każdy świadomy człowiek pamięta. Dopuszczalne są polskie litery a także także wszystkie inne tzw. znaki widzialne.



Rys. 3.2 Ekran logowania.

Aby przejść do głównego ekranu programu należy poprawnie się zalogować. Jeśli takowe nastąpi, ekran logowania automatycznie zniknie, odsłaniając ekran główny klienta programu. Wygląd ekranu jest także uzależniony od tego czy zalogowana osoba jest nauczycielem czy uczniem. Interfejs jest odpowiednio modyfikowany w obu przypadkach, aby zapewnić adekwatną funkcjonalność obu stronom. W sytuacji braku połączenia z serwerem lub podania błędnych danych do logowania pojawi się odpowiedni komunikat pod przyciskiem *Zaloguj*.

3.3 UCZEŃ

3.3.1 EKРАН GŁÓWNY

Po poprawnym zalogowaniu, ekran główny ucznia wygląda następująco, przeważające kolory to odcienie szarości, biały oraz cyjan. Biel na czerni to typowe kolory współczesnych interfejsów, odpowiednim cieniowaniem zostały podkreślone istotniejsze napisy, natomiast cyjan dla urozmaicenia, przełamuje monochromatyczność i pojawia się w różnych miejscach ekranu.



Rys. 3.3 Ekran główny.

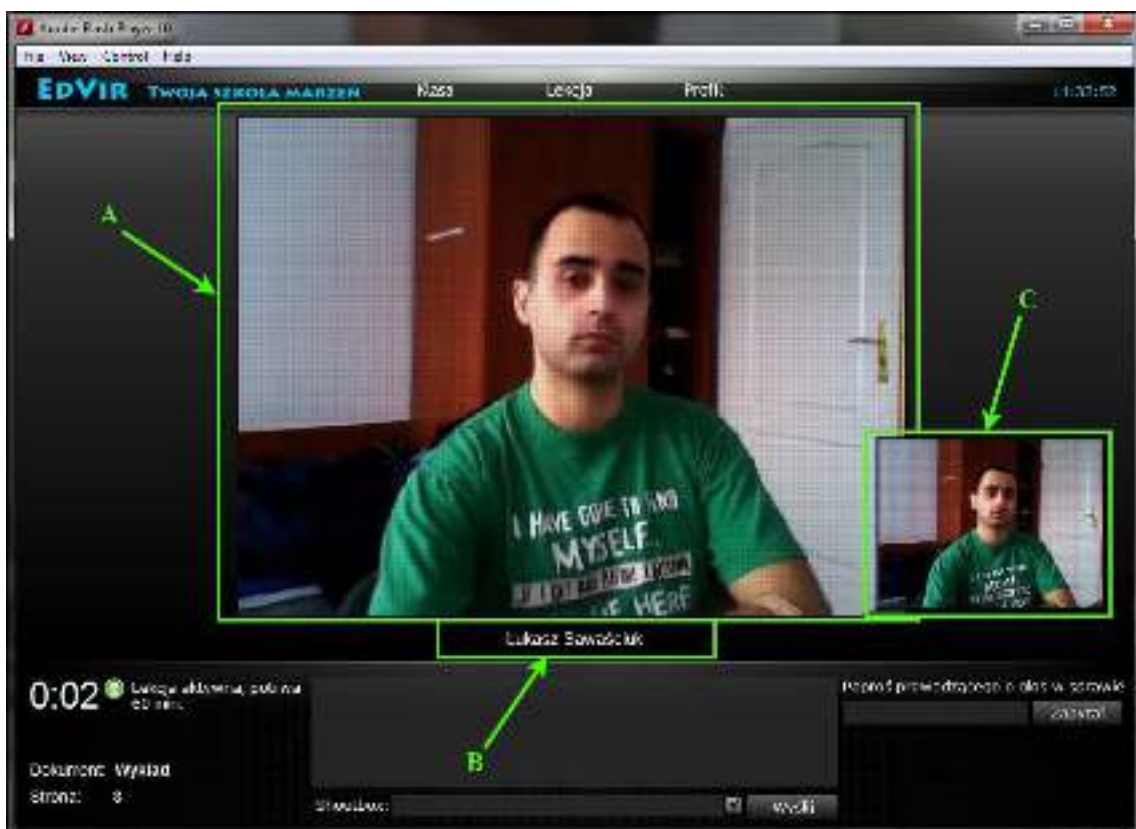
- A) Menu główne zawierający przyciski *Klasa*, *Lecja*, *Profil* (szczegółowo opisane w dalszej części), po prawej stronie znajduje się zegar z aktualnym czasem serwera.
- B) Główny obszar roboczy. Tutaj wyświetlane są obiekty wideo, zarówno własny, jak i publiczny kanał wideo, najczęściej z widokiem kamerki nauczyciela. Obszar roboczy zmienia swój wygląd i funkcjonalność w zależności od trybu pracy, np. dla pracy grupowej.
- C) Sekcja z podstawowymi informacjami dotyczącymi przebiegu lekcji. Zawiera między innymi informacje o aktywności lekcji, aktualnym czasie oraz bieżącym dokumencie do zajęć. Jeśli którykolwiek z elementów nie jest aktualnie istotny

dla przebiegu lekcji jest automatycznie ukrywany. (szerzej opisane w dalszej części).

- D) Czat globalny, zawiera elementy służące do aktywnej rozmowy ze wszystkimi uczestnikami danej lekcji.
- E) Obszar powiadomień oraz moduł zadawania pytań prowadzącemu zajęciu.

3.3.1.1 KOMPONENTY WIDEO

Podczas typowej lekcji najczęściej występującymi obiektami wideo na scenie jest własna kamera ucznia oraz Kanał główny w którym wyświetlany jest obraz osoby prowadzącej zajęcia. Jakość obrazu dla obu komponentów jest dynamicznie dostosowywana do możliwości łącza, jednak z racji mniejszego wykorzystania kamery uczniów, zakłada się cztery razy mniejszą jakość w stosunku do obrazu nauczyciela.



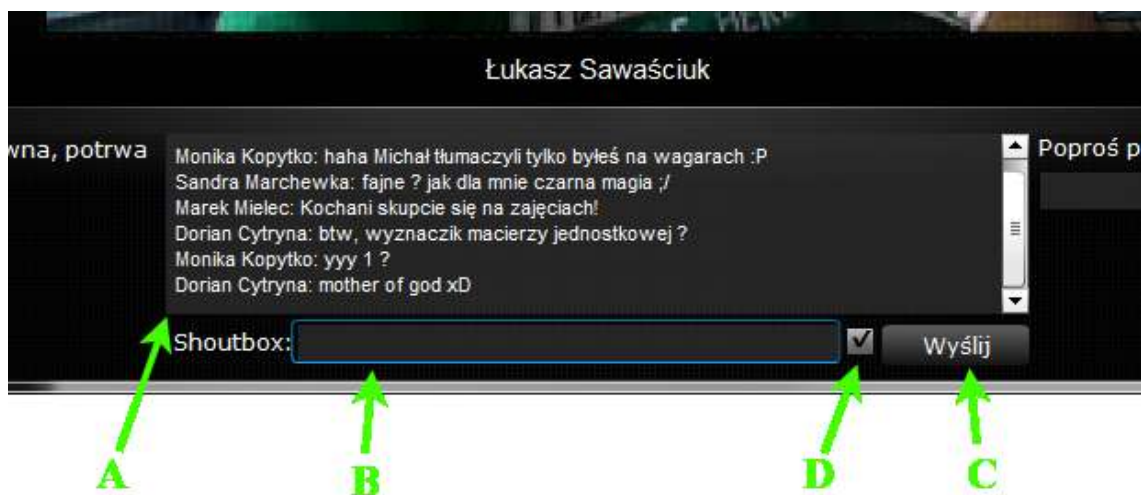
Rys. 3.4 Komponenty wideo.

- A) Obszar wideo w którym wyświetlany jest obraz z publicznego kanału głównego. Podczas trwania lekcji najczęściej jest to obraz z kamery nauczyciela, zdarza się że jest to inna osoba, np. zadająca pytanie. W oknie publicznego wideo podczas przedstawiania uczestników, wszyscy uczniowie, po kolei pojawiają się na kilkanaście sekund.
- B) Imię i nazwisko osoby która aktualnie jest pokazywana w oknie wideo kanału publicznego.

- C) Obraz z własnej kamery uczestnika, pozostaje tam niezmiennie przez cały czas trwania zajęć.

3.3.1.2 CZAT GLOBALNY

Czat globalny służy do mniej formalnej komunikacji podczas zajęć. Można tutaj dodawać własne spostrzeżenia dotyczące tematu zajęć i tego o czym mówi prowadzący. Uczestnicy mogą sobie pomagać, odpowiadając na drobne pytania lub przypomnieć mniej uważnemu uczestnikowi o czym prowadzący już mówił. Czat globalny pełni też ważną funkcję w komunikacji elektronicznej ponieważ można za jego pomocą łatwo dzielić się odnośnikami do stron www lub innymi treściami z elektronicznych materiałów np. cytatów z *e-book'ów* lub materiałów do zajęć.



Rys. 3.5 Czat globalny.

- A) Główne okno czatu, posiada mechanizm przewijania zawartości który automatycznie pojawia się po zapełnieniu domyślnego rozmiaru, oraz automatycznie przesuwają się na sam dół swojego obszaru aby wyszczególnić nowo otrzymaną wiadomość.
- B) Pole wpisywania wypowiedzi. Jeśli wiadomość została poprawnie rozesłana to pole zostaje wyczyszczone dla wpisania kolejnej wypowiedzi.
- C) Przycisk wysłania wiadomości
- D) *Checkbox* dla automatycznego wysłania wiadomości po naciśnięciu klawisza *Enter*.

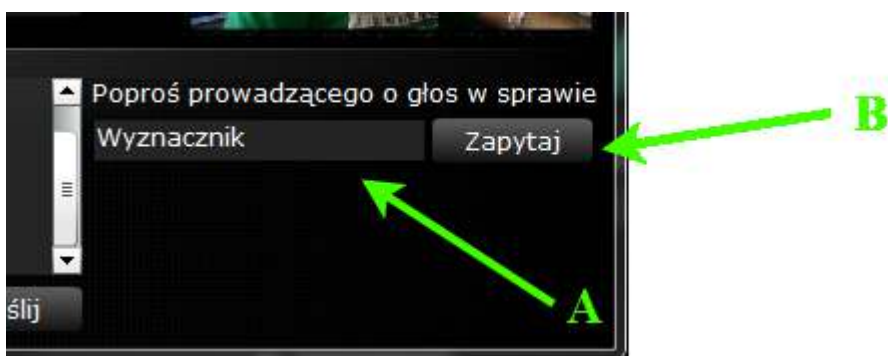
3.3.1.3 ZADAWANIE PYTAŃ

W trakcie trwania lekcji uczeń ma możliwość wysłania zapytania do prowadzącego na jakiś nurtujący go temat. W tym celu powinien określić w jakiej sprawie chce zadać pytanie, ta informacja będzie pomocna dla nauczyciela jeśli kilka osób będzie chciało zapytać o to samo. Wtedy już tylko od niego zależy które pytanie zostanie rozpatrzone,

a które będzie mógł usunąć bez dalszego drążenia tematu. Po wysłaniu pytania i zaakceptowaniu go przez nauczyciela, osoba pytająca zostaje wyświetlona w drugim publicznym kanale wideo. Na ekranie wszystkich uczestników pojawiają się dwa obrazy, jeden nauczyciela, drugi ucznia. Jest to klasyczny rodzaj rozmowy jaki zachodzi w tradycyjnej szkole. Inni uczestnicy słuchają pytania kolegi ponieważ może to pomóc w odpowiedzi na ich pytania lub też uświadomienia sobie nieścisłości w zajęciach, których sami nie byli świadomi. Po zakończeniu rozmowy, nauczyciel wyłącza obraz ucznia i kontynuuje zajęcia na głównym kanale publicznym .



Rys. 3.6 Uczeń podczas rozmowy z nauczycielem.



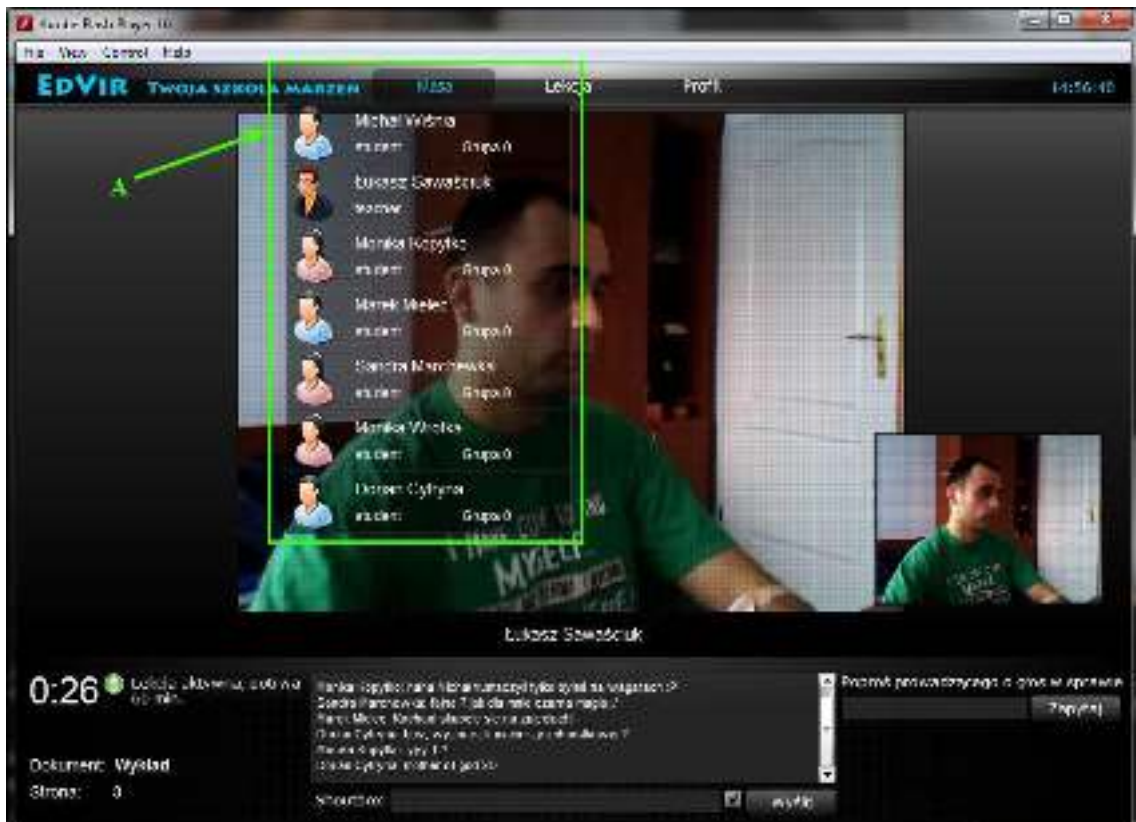
Rys. 3.7 Panel zadawania pytań.

A) Pole tekstowe do którego wpisuje się temat zapytania. Może to być kilka słów (np. „wyznacznik macierzy”), które pomogą nauczycielowi przygotować się do

odpowiedzi. Nauczyciel ma możliwość usunięcia podobnych pytań zgłoszonych przez innych uczestników.

- B) Przycisk wysyłający zapytanie do nauczyciela. Nauczyciel zostanie automatycznie powiadomiony o pytaniu, a uczniowi pozostaje czekać na możliwość wypowiedzi, o ile sam nauczyciel uprzednio nie rozwieje jego wątpliwości.

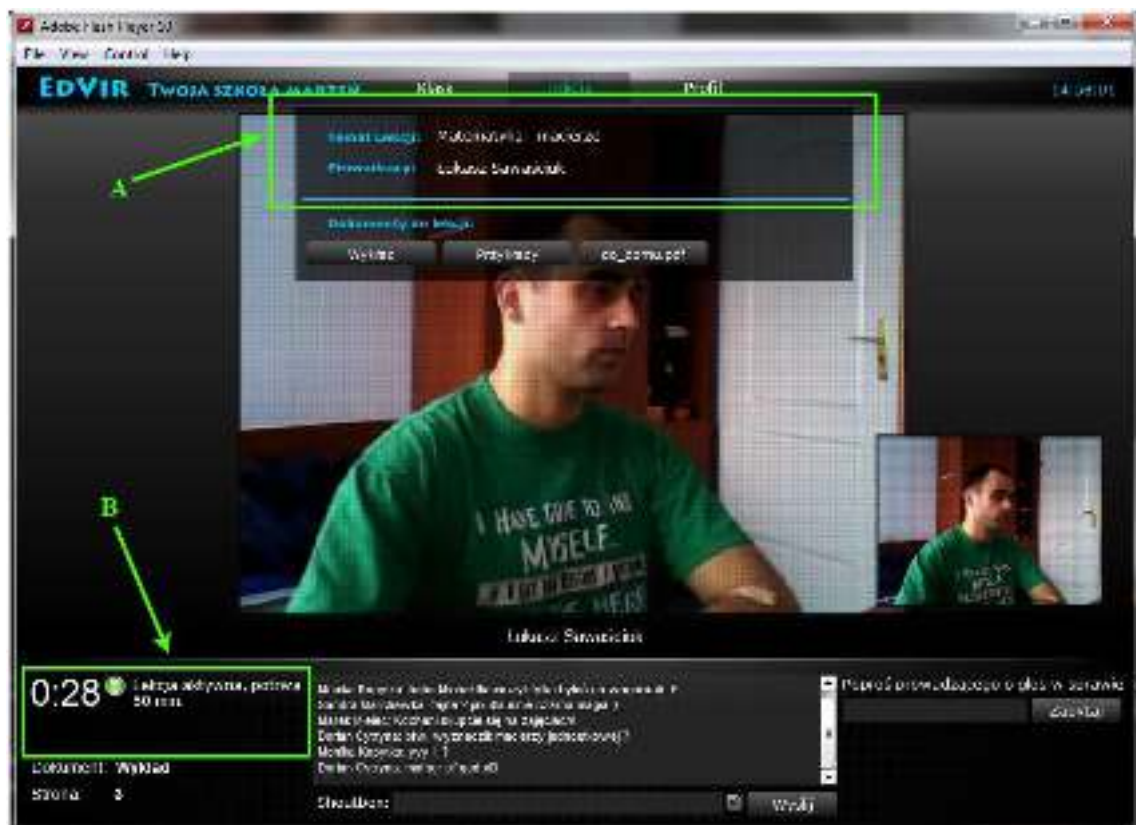
3.3.2 LEKCJA I KLASA



Rys. 3.8 Lekcja i klasa.

- A) Z pozycji *Menu głównego*, *Klasa* dostępna jest lista wszystkich zalogowanych uczestników lekcji. Prezentowane są takie informacje jak *status* mogący przybierać formę nauczyciela lub studenta, przynależność do danej grupy, a także przybliżona reprezentacja płci uczestnika w formie adekwatnego awatara (nie dotyczy nauczyciela). W przypadku rozpoczęcia pracy grupowej jeden z uczestników grupy zostaje wybrany liderem, bezpośrednio odpowiedzialnym za grupę i prezentację dokumentu grupowego. Jego Avatar przyjmuje postać generała z kilkoma osobami u swojego boku. W łatwy sposób uczestnik ma możliwość podglądu osób które uczestniczą w zajęciach oraz z kim jest w grupie i kto jest jej liderem.

3.3.2.1 PRZEBIEG LEKCJI



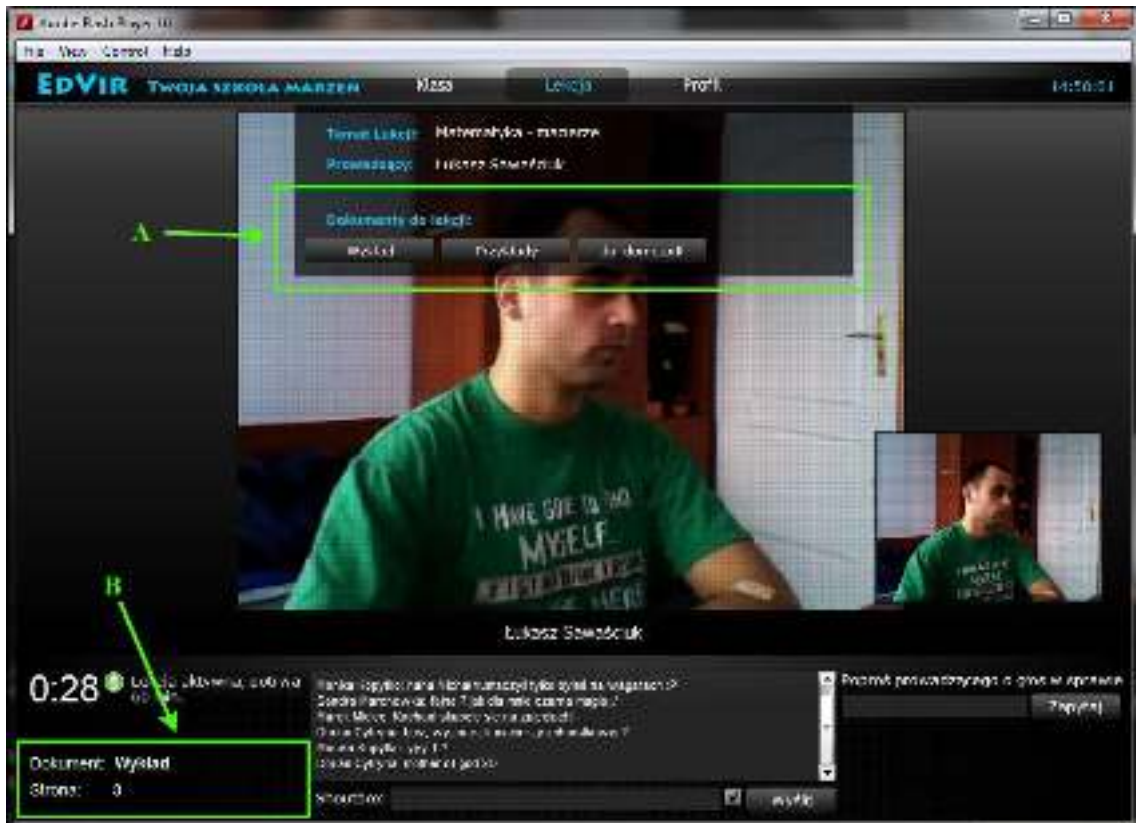
Rys. 3.9 Przebieg lekcji.

- A) Z pozycji *Menu głównego*, *Lekcja* uczestnik ma dostęp do podstawowych informacji, takich jak: temat lekcji oraz kto jest prowadzącym zajęcia. Ma to jedynie charakter formalny i zasadniczo uczeń tylko jeden raz w trakcie trwania lekcji pozyskuje tę informację, stąd też jest ona dostępna z *Menu głównego* aby niepotrzebnie nie rozpraszać ucznia podczas zajęć.
- B) Standardowy Zegar lekcji liczący do przodu, jeśli harmonogram lekcji nie jest ustawiony zegar również nie jest widoczny. Po prawej stronie widnieje ikonka sygnalizująca aktywność zegara, a dalej na prawo informacja tekstowa oraz dodatkowy opis. Zegar pracy grupowej liczy do tyłu ponieważ tutaj większa znaczenie ma szybkość pracy i informacja dla uczestników ile czasu im pozostało na ukończenie wyznaczonego zadania.

3.3.2.2 MATERIAŁY ZEWNĘTRZNE

Jednym z kluczowych aspektów prowadzenia współczesnych zajęć edukacyjnych jest możliwość wspólnego przeglądania materiałów. W szkołach odbywa się to za pośrednictwem rzutników ekranowych, tradycyjnej tablicy do pisania a także wspólnej pracy z książką lub innymi materiałami jak Atlas geograficzny, zeszyt ćwiczeń. W prezentowanym systemie praca z dodatkowymi materiałami została tak zorganizowana aby wszystkie dokumenty były dostępne z poziomu programu klienckiego, jednak

uruchamiane w programie zewnętrznym tj. przeglądarce stron internetowych. Przeglądarka internetowa otwiera ogromne perspektywy, ponieważ w bardzo intuicyjny i znany użytkownikowi sposób daje możliwość przeglądania dowolnych zawartości stron internetowych, przeglądania zdjęć, a także przy pomocy dodatkowych wtyczek, które i tak mają ogromną penetrację na runku daje sposobność przeglądania plików w formacie *pdf* i *swf*.



Rys. 3.10 Materiały dodatkowe.

- A) Z poziomu Menu głównego, lekcja mamy dostęp do obszaru *Dokumenty do lekcji*, do którego nauczyciel dodał materiały istotne dla bieżącej lekcji. Materiały te mogą być ściśle związane z lekcją i stanowić np. plik *pdf* z opracowaną prezentacją, materiałami mogą być zadania domowe które należy zapisać do późniejszego opracowania. Każdy przycisk jest powiązany z odnośnikiem do zajęć oraz odpowiednim mechanizmem wyzwalającym. Po naciśnięciu na któryś z przycisków następuje uruchomienie domyślnej przeglądarki internetowej oraz wywołanie adresu w nowej zakładce.
- B) W trakcie trwania zajęć, jeśli nauczyciel korzysta z dodatkowego pliku z materiałami może o tym poinformować uczniów, wtedy w obszarze pod zegarami lekcji pojawia się tytuł bieżącego dokumentu oraz numer strony której treść, prowadzący aktualnie omawia.

3.3.3 PROFIL



Rys. 3.11 Zakładka Profil

W zakładce *Profil*, *Menu głównego* dostępny jest podgląd własnego profilu zalogowanej osoby. Wyświetla się Avatar pod jakim użytkownik jest widziany przez innych uczestników, imię i nazwisko osoby, jej status oraz przynależność do danej grupy. Jeśli nie jest aktywna praca grupowa domyślnym numerem grupy jest 0. W tym miejscu mamy możliwość zmiany hasła do swojego profilu. W przypadku poprawnej zmiany, u dołu zakładki wyświetli się odpowiedni komunikat. Monitowane są również nieudane próby zmiany hasła oraz co jest ich przyczyną.

3.4 NAUCZYCIEL



Rys. 3.12 Ekran główny nauczyciela.

Po prawidłowym zalogowaniu się do systemu za pomocą konta z uprawnieniami nauczyciela ekran główny programu zostaje rozszerzony o dodatkową funkcjonalność, a część z opcji domyślnie wyświetlanych dla profilu ucznia, a istotnych dla przebiegu zajęć zostaje zmieniona na te adekwatne dla roli nauczyciela.

- A) W *Menu głównym* zostają dodane dwie zakładki dotyczące kolejno, *Aktywności* uczniów oraz *Ustawień* lekcji.
- B) W obszarze po prawej stronie od czatu głównego zostaje aktywowany panel do obsługi dodatkowych dokumentów. Z punktu widzenia nauczyciela, przełączanie stron jest najczęstszą aktywnością podczas typowego wykładu, stąd też obsługa znalazła się na głównym ekranie.

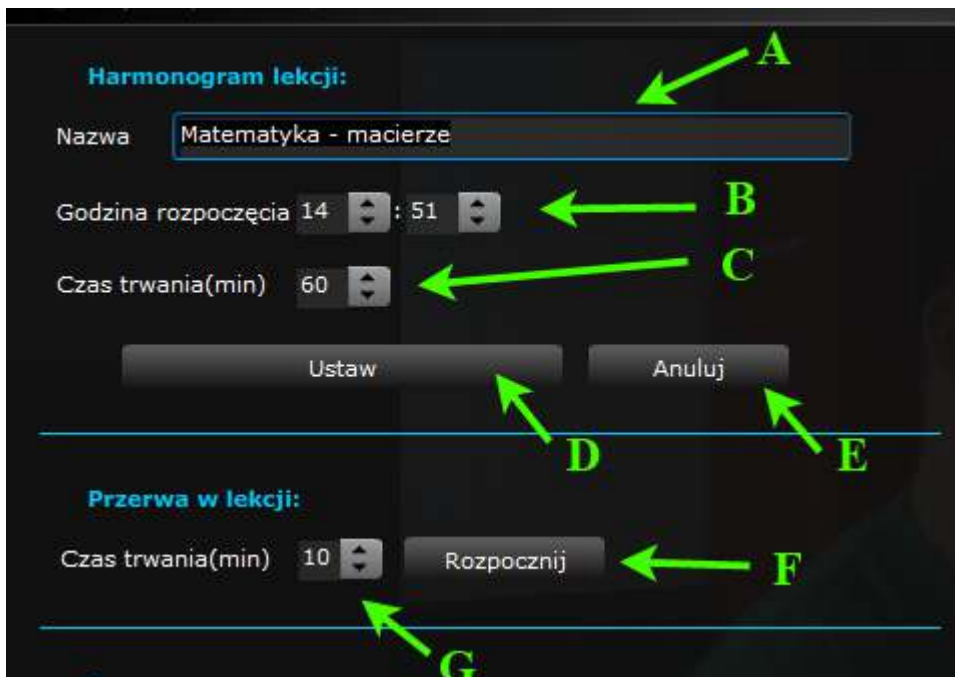
3.4.1 USTAWIENIA LEKCJI

Zarówno przed, jak i w trakcie zajęć nauczyciel ma dostęp do szeregu ustawień lekcji z poziomu *Menu głównego*, *Ustawienia*. Część z tych ustawień ma krytyczne znaczenie ze względu na poprawny przebieg całej lekcji, priorytetowym jest ustawienie harmonogramu lekcji z czasem trwania i tematyką zajęć. Wykorzystanie np. modułu przedstawiania się uczestników lub dokumentów jest opcjonalne, jednak uatrakcyjnia przebieg zajęć oraz kondensuje ich przekaz merytoryczny.



Rys. 3.13 Zakładka Ustawienia.

3.4.1.1 HARMONOGRAM



Rys. 3.14 Harmonogram lekcji.

- A) Pole tekstowe do którego wpisywany jest temat prowadzonych lub mających się rozpocząć zajęć.
- B) Numeryczne Stepper'y w których podajemy godzinę rozpoczęcia zajęć. Po otwarciu zakładki ustawień pobierany jest aktualny czas z serwera. Nie można rozpocząć lekcji w przeszłości, czas rozpoczęcia musi być przynajmniej +1 minutę względem czasu serwera. Jest to zabezpieczenie przed ewentualną pomyłką w ustawieniach oraz nie daje żadnej możliwości manipulacji czasem w przypadku spóźnienia się na zajęcia przez prowadzącego.
- C) Stepper czasu trwania lekcji, domyślna wartość to 60 minut, jednak możemy iterować licznik w zakresie od 15 do 360 minut, ze skokiem co 5 minut.
- D) Po naciśnięciu przycisku Ustaw, lekcja zostaje dodana do harmonogramu, a odpowiednia informacja rozprzestrzenia się po wszystkich uczestnikach. W obszarze zegarów lekcji wyświetla się komunikat o czasie rozpoczęcia i długości trwania zbliżających się zajęć.
- E) Po wywołaniu funkcji zdalnej związanej z przyciskiem Anuluj, następuje całkowite porzucenie ustawionego wcześniej harmonogramu lekcji, a także jeśli lekcja została już rozpoczęta zostanie bezwzględnie zakończona.
- F) W trakcie trwania zajęć może zająć potrzeba przeprowadzenia przerwy, zależy to od woli uczestników, a prowadzący może to formalnie wykonać bez uszczerbku na czasie lekcyjnym. W Liczniku ustawiamy minutową wartość trwania przerwy.
- G) Rozpoczęcie przerwy wiąże się z wstrzymaniem zegara lekcji, natomiast po upływie zadanego czasu, lekcja zostanie automatycznie wznowiona.

3.4.1.2 PRACA GRUPOWA

Praca grupowa jest jednym z ważniejszych aspektów aktywności uczestników i to właśnie tutaj mają najlepszą okazję aby wykazać swoje zaangażowanie i kreatywność. W zakładce *Ustawienia* dostępny jest panel do rozpoczęcia i zarządzania pracą grupową.



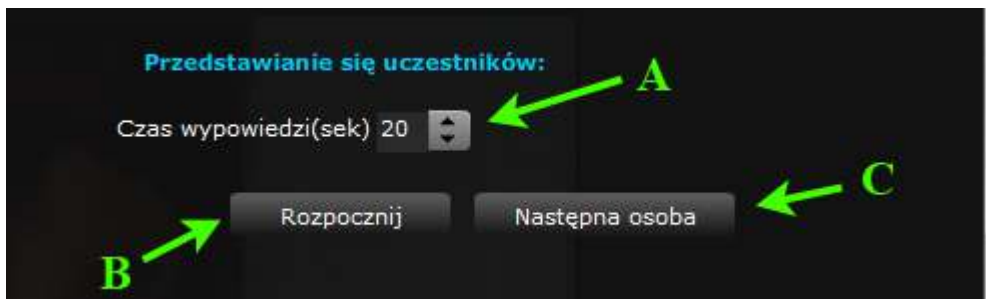
Rys. 3.15 Ustawienia pracy grupowej.

- A) Liczba grup determinuje w jak dużych zespołach będą pracować uczniowie. Dopuszcza się grupy jednoosobowe, jednak praca grupowa nie rozpocznie się jeśli któraś z grup miałaby mieć zero uczestników. W Stepperze podajemy wartość od 1 do 10.
- B) Czas trwania pracy grupowej powinien być odpowiednio dobrany do efektów jakie chcemy uzyskać od grupy. Nie powinno się dawać zbyt wiele czasu aby utworzony dokument był bardziej wynikiem burzy mózgów i pełnej aktywności niż skrupulatnego dopieszczenia utworzonego dokumentu.
- C) Po rozpoczęciu pracy grupowej uaktywnia się dodatkowy zegar, pod zegarem lekcji, który tym razem odlicza do tyłu. Ma to skłaniać do jeszcze większej mobilizacji uczestników grupy.
- D) Przycisk *Pauza/wznów* działa na zasadzie *toggle button*, Pracę grupową można wstrzymać jeśli drużyny mają jakieś pytania lub wątpliwości które chcieliby przedyskutować z prowadzącym zajęcia. Formalnie odbywa się to na zasadach czystej rywalizacji, a więc zarówno pytanie jak i odpowiedź nauczyciela słyszą wszystkie grupy. Przycisk ma też dodatkową funkcję, ponieważ można nim wydłużyć pracę grupową jeśli grupy nalegają na jeszcze kilka chwil do pełnego ukończenia swojego zadania.
- E) Można zakończyć pracę grupową przed czasem przy pomocy przycisku *Zakończ*

3.4.1.3 PRZEDSTAWIENIE UCZESTNIKÓW

Na początku spotkania można przeprowadzić przedstawianie się uczestników dając im czas na powiedzenie kilku słów o sobie. Każdy z uczestników jest pokazywany w głównym kanale wideo i ma możliwość zaprezentowania się innym uczestnikom.

Wystąpienie w publicznym kanale wideo dopinguje też do rzetelnego podejścia i nie lekceważenia zajęć np. odbywając je w szlafroku lub sennej, łózkowej atmosferze. Przedstawienie się uczestników choć nie zajmuje wiele czasu, to ma duży wpływ na lepszą integrację w klasie oraz zwalnia od formalnego przedstawienia się gdy uczestnicy rozpoczynają pracę w grupach.

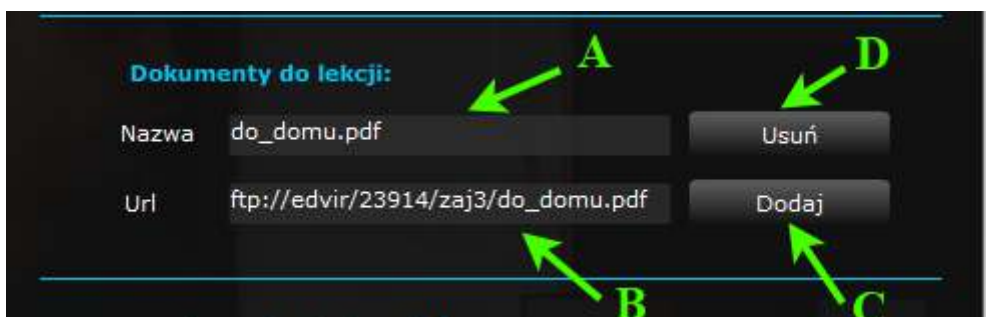


Rys. 3.16 Przedstawianie się uczestników.

- A) Czas wypowiedzi w sekundach ustawiany w Stepperze.
- B) Rozpoczęcie przedstawiania wiąże się z utworzeniem listy aktywnie zalogowanych uczestników, którzy automatycznie z zadany czas wypowiedzi przełączani się na główny kanał wideo.
- C) Jeśli uczestnik sobie tego życzy nauczyciel może wyłączyć jego obraz i przejść do następnej osoby z listy przyciskiem *Następna osoba*.

3.4.1.4 MATERIAŁY DO LEKCJI

Kolejną strategiczną częścią ustawień, ze względu na atrakcyjność zajęć jest dodawanie dodatkowych materiałów. Obok przekazu audio-wideo nauczyciela, użytkownik może przeglądać dokument ze skondensowaną wersją wykładu lub zawierający inną, dowolną treść. Dokumenty precyzuje się poprzez określenie nazwy oraz adresu URL. Nazwa może być dowolna, natomiast adres URL powinien dać się zinterpretować przez popularne przeglądarki internetowe. Dla pewności warto wpisywać pełen adres wraz z nazwą protokołu, np. *https*, *ftp* itp.




Rys. 3.17 Zarządzanie materiałami do lekcji.

- A) Pole tekstowe z nazwą dokumentu. Nazwa będzie wyświetlana jako *label* przycisku w zakładce *Menu główne, Lekcja*.

- B) Adres URL prowadzący bezpośrednio do dokumentu.
- C) Konsekwencją przycisku *Dodaj* jest natychmiastowe wygenerowanie odnośnika do dokumentów w zakładce *Menu główne, Lekcja*. W tym obszarze mieści się do 4 przycisków włącznie, co jest wystarczającą ilością materiałów jak na pojedyncze zajęcia.
- D) Przyciski do dokumentów można usuwać poprzez wypełnienie Pola tekstowego z nazwą i wywołanie przycisku *Usuń*. Wskazany dokument zostanie odnaleziony i usunięty, robiąc tym samym miejsce na dodanie kolejnych dokumentów jeśli zachodzi taka potrzeba.

3.4.1.5 DODAWANIE UŻYTKOWNIKÓW

Dodawanie użytkowników jest samo zapętlającym się mechanizmem ponieważ dla utworzenia ogromnej bazy zarówno nauczycieli jak i uczniów wystarczy jeden użytkownik z prawami nauczyciela. W środowiskach pełnego zaufania wśród nauczycieli i ograniczonego rozprzestrzenienia aplikacji jest to mechanizm z powodzeniem wystarczający.



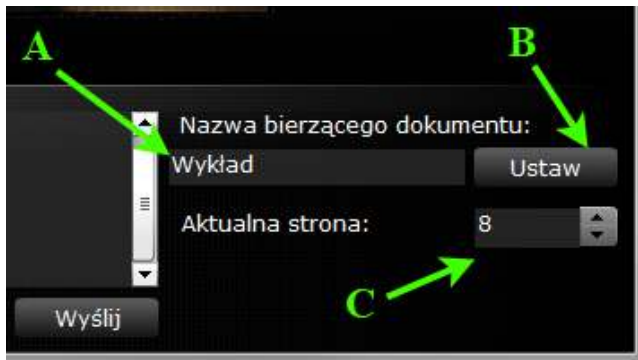
Rys. 3.18 Dodawanie użytkowników

- A) Podajemy imię i nazwisko użytkownika (jest to forma loginu). Dopuszczalne są polskie litery, spacja, a także inne znaki tzw. widzialne.
- B) Miejsce w którym wpisujemy hasło do profilu. Hasła w serwerze są przechowywane w formie tajnej, hashowane jednostronną funkcją skrótu *MD5*.
- C) Flaga determinująca czy utworzony użytkownik będzie miał prawa nauczycielskie.
- D) Przycisk dodania użytkownika do bazy danych.

3.4.2 PRZEBIEG LEKCJI

3.4.2.1 OBSŁUGA MATERIAŁÓW

Moduł obsługi materiałów zewnętrznych znajduje się w obszarze ekranu głównego programu z uprawnieniami nauczyciela. Bez zbędnego zagłębiania się w zakładki Menu nauczyciel może szybko zmienić aktualny dokument oraz inkrementować bieżącą stronę.

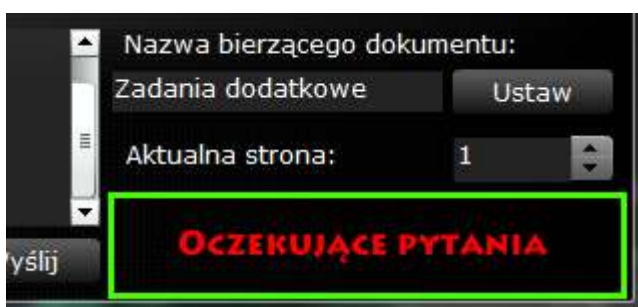


Rys. 3.19 Obsługa materiałów zewnętrznych.

- A) Pole tekstowe dla nazwy dokumentu. Ustawienie pustej zawartości tego pola jest interpretowane jako zakończenie pracy z dokumentem i skutkuje usunięciem informacji o aktualnym dokumencie zewnętrznym z ekranów wszystkich użytkowników.
- B) Przycisk wyzwalający ustawienie dokumentu jako aktywnego.
- C) *NummericStepper* służy do zmiany aktualnie przeglądanej strony, może też wyznaczać początek rozdziału aby nie przewijać go zbyt często.

3.4.2.2 NOTYFIKACJE

W przypadku wysłania pytania do prowadzącego zajęcia przez któregoś z uczestników, na ekranie prowadzącego pojawia się informacja że otrzymał pytanie oczekujące na obsłudze. Jest to jedynie drobny komunikat, nie burzący trwającego wykładu. Nauczyciel w dogodnym dla siebie momencie może przejść do zakładki *Aktywność w Menu głównym* i zająć się otrzymanymi pytaniami.

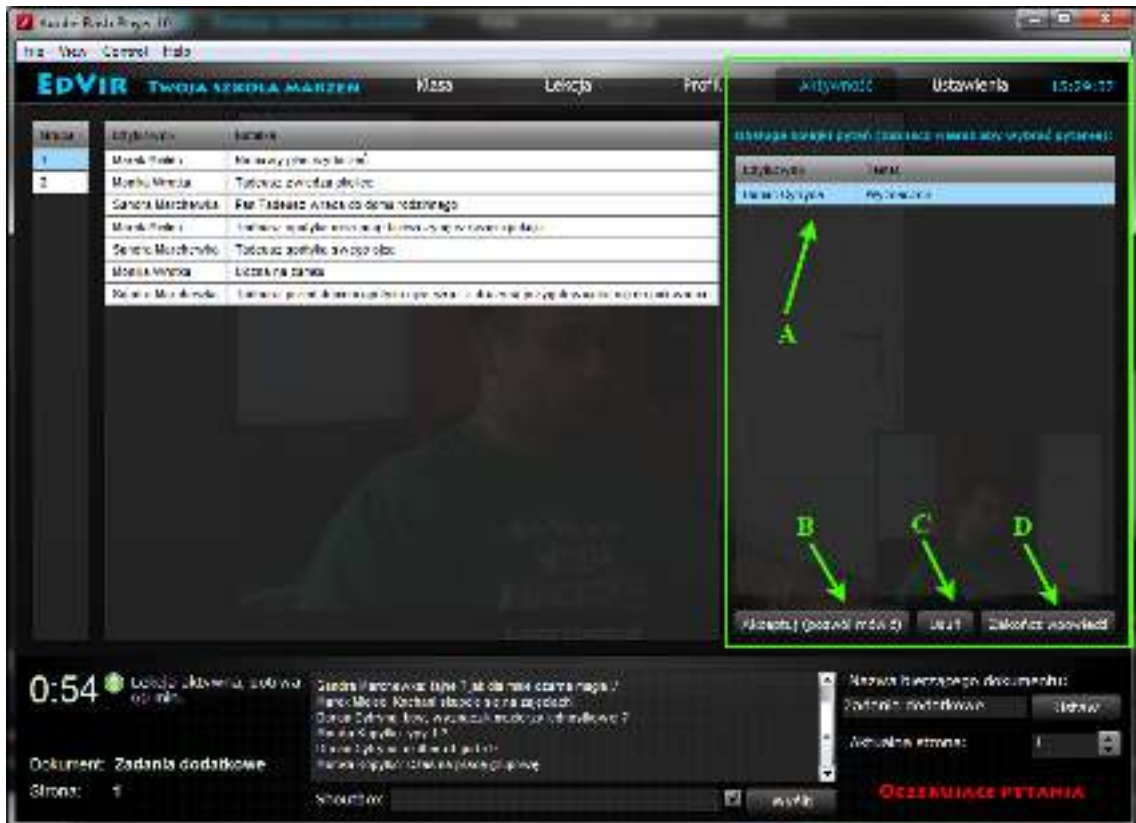


Rys. 3.20 Notyfikacje

3.4.3 AKTYWNOŚĆ UCZNIÓW

3.4.3.1 OBSŁUGA PYTAŃ

Z poziomu *Menu głównego*, zakładka *Aktywność* znajduje się dostęp do modułu obsługi przychodzących pytań od uczniów uczestniczących w zajęciach. W jego wnętrzu wyświetlana jest komponent *DataGrid*, z kolumnami zawierającymi nazwę ubiegającego się o głos użytkownika oraz temat pytania.



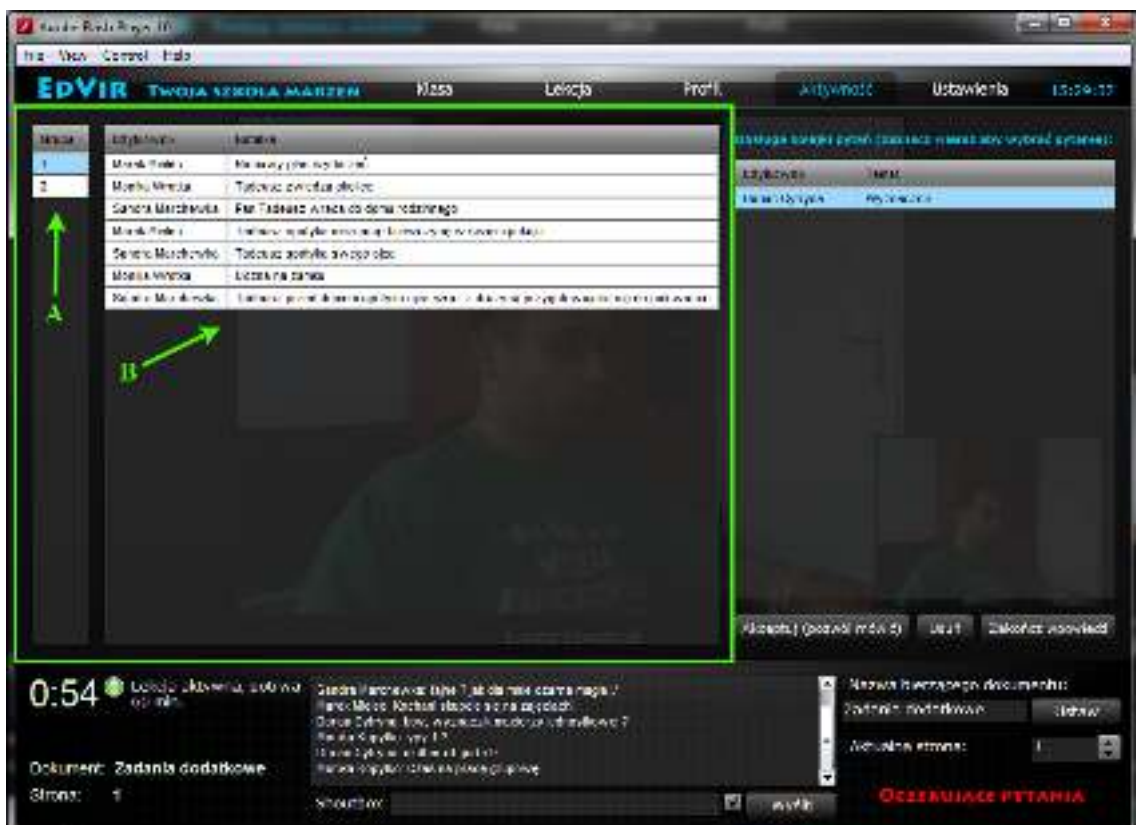
Rys. 3.21 Obsługa pytań.

- A) Naciśnięcie na pytanie skutkuje jego podświetleniem i zamarkowaniem go jako pytanie aktywne, po tej czynności możemy wykonać szereg akcji związanych z obsługą pojedynczego pytania.
- B) Zaakceptowanie pytania automatycznie przenosi obraz z kamery osoby pytającej do drugiego głównego kanału wideo. U wszystkich uczestników wywoływana jest metoda przełączająca tryb wyświetlania ekranu głównego aby obsłużyć dwa publiczne kanały wideo. Obraz z kamery nauczyciela zostaje pomniejszony i przesunięty do prawej krawędzi ekranu, natomiast obraz pytającego ucznia jest większy i wyświetlony przy lewej krawędzi ekranu głównego (Rys. 3.6). Od tej chwili uczeń może zadać pytanie które widzą i słyszą wszyscy inni uczestnicy, a także nauczyciel może na bieżąco odpowiadać, tudzież prowadzić dyskusję z uczniem.

- C) Dowolne pytanie można usunąć bez konieczności rozpatrzenia np. gdy pytanie o podobnym temacie już zostało wysłane.
- D) W momencie wyczerpania tematu przez nauczyciela i potwierdzenia ze strony ucznia że jego wszelkie wątpliwości zostały rozwiane, nauczyciel kończy konwersację przyciskiem *Zakończ wypowiedź*.

3.4.3.2 DOKUMENTY GRUPOWE

Kluczowym wymiarem aktywności uczestników lekcji jest sporządzany przez nich wspólny dokument podczas pracy grupowej. Dokument grupowy działa na zasadzie krótkich notatek dodawanych w dowolnej ilości przez uczestników danej grupy. Notatki można kopiować, usuwać oraz przestawiać ich kolejność względem siebie. Wszystkie dodane notatki powinny stanowić spójną całość, nazywaną dalej dokumentem grupowym. Panel podglądu tworzonych dokumentów grupowych dostępny jest z *Menu głównego*, zakładka *Aktywność* i składa się z 2 obiektów typu *DataGrid*.



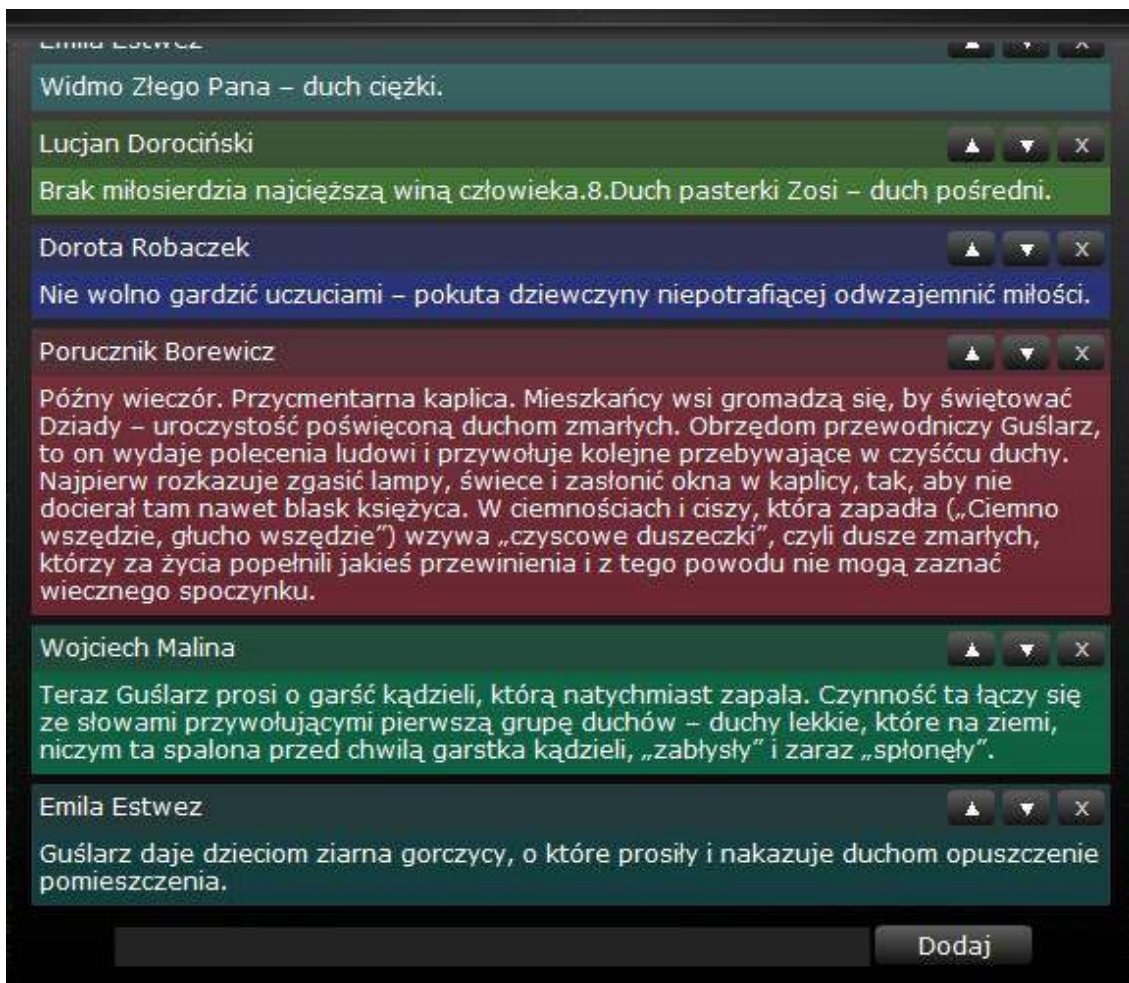
Rys. 3.22 Podgląd dokumentów grupowych.

- A) Pierwszym obiektem jest lista grup które zostały utworzone. Kliknięcie na grupę powoduje wyszczególnienie wyboru niebieskim kolorem oraz wyświetleniem dokumentu grupy w kolejnym *DataGrid* po prawej stronie.
- B) Każda notatka w formie wyświetlanej nauczycielowi jest korelacją nazwy użytkownika oraz treści wysłanej notatki. Kolejność wyświetlanych notatek jest kolejnością jaką życzyli sobie uczestnicy grupy i przyjmuje się ją za poprawną.

- A) Obraz każdego użytkownika jest podpisany imieniem i nazwiskiem.
- B) Obraz z własnej kamerki jest bez podpisu i znajduje się w stałym miejscu – prawym, dolnym rogu ekranu głównego.

3.5.2 DOKUMENT GRUPOWY

Praca grupowa może mieć na celu omówienie problemu i zaproponowanie rozwiązań, wyciągnięcie wniosków z jakiegoś opracowywanego temat lub zbudowaniem np. planu wydarzeń. Notatki wysyłane do tworzonego wspólnie dokumentu grupowego są podpisane przez autora, co pozwala wykazać się każdemu z uczestników z osobna. Dodawanie notatek do dokumentu grupowego odbywa się w sposób przyrostowy, tzn. nowa notatka doklejana jest na sam koniec dokumentu. Użytkownik może zmienić jej położenie, a także np. usunąć poprzedni wpis zastępując go poprawioną wersją. Dla łatwiejszego rozróżniania autorów notatek, na początku zajęć każdy z uczestników otrzymuje wygenerowany losowo kolor w zapisie heksadecymalnym, kolor ten jest używany każdorazowo przy tworzeniu notatki przez użytkownika. Dostęp do modyfikacji notatek ma każdy uczestnik danej grupy, a swoje poczynania wspiera stałą rozmową z innymi uczestnikami.



Rys. 3.24 Przykład dokumentu grupowego.



Rys. 3.25 Tworzenie dokumentu grupowego.

- A) W tytule każdej notatki znajduje się imię i nazwisko jej autora.
- B) Zawartość notatki. Dopuszczalne są długie teksty, wówczas notatka jest odpowiednio skalowana, a zawartość tekstowa zawijana do kolejnych linii.
- C) Naciśnięcie przycisku skutkuje przesunięciem notatki o jedną pozycję wyżej
- D) Przycisk do przesunięcia notatki o jedną pozycję w dół.
- E) Permanentne usunięcie notatki.
- F) Pole tekstowe do wpisywania treści nowej notatki
- G) Przycisk powoduje wysłanie notatki do serwera i podwiązanie jej na końcu dokumentu grupowego.

Rozmiar dokumentu grupowego jest teoretycznie nie ograniczony, jeśli zostanie wypełniony obszar na ekranie głównym programu, dokument grupowy da się przewijać za pomocą kółka myszy, poruszając nim odpowiednio w górę i w dół.

4. REALIZACJA PROJEKTU

4.1 KLIENT PROGRAMU

4.1.1 APLIKACJA SWF

Aplikacja kliencka została w pełni stworzona z wykorzystaniem technologii Flash, oraz obiektowego języka programowania ActionScript 3.0. Skompilowany plik wyjściowy *.swf może być uruchomiony za pomocą dowolnej formy odtwarzacza Adobe Flash Player. Jeśli na urządzeniu użytkownika znajduje się zainstalowana lokalna wersja odtwarzacza to pliki swf są domyślnie z nią powiązane i uruchamiają się w formie indywidualnego programu. Pliki swf można również dobrze uruchamiać z dowolnej przeglądarki internetowej zawierającej Adobe Flash Player w formie dodatku.

Podstawą tworzenia aplikacji w technologii Flash jest wykorzystanie obiektów typu *MovieClip*. Takimi obiektami są np. panele (zakładki): *Klasa*, *Lekcja*, *Profil*, jak również przyciski *Menu głównego*. Obiektami *MovieClip* jest prawie wszystko co wyświetla się na głównej scenie aplikacji Flash. Podstawowy obiekt tego typu jest pusty i niewidoczny, jednak jego zaletą jest możliwość zawierania w sobie innych obiektów i tym samym tworzenie swoistych kontenerów obiektów widzialnych. Takie kontenery mają ujednolicone metody do zarządzania zawartością i w łatwy sposób można je animować, skalować, transformować, a także zmieniać ich zawartość w czasie rzeczywistym.

Dla zwiększenia atrakcyjności aplikacji klienckiej wykorzystano narzędzie *Oś czasu* (ang. *Timeline*). *Oś czasu* działa na zasadzie klatek kluczowych pomiędzy którymi można tworzyć rozmaite animacje. Przykładem może być przesuwanie się obrazów wideo na ekranie głównym po aktywowaniu drugiego publicznego kanału wideo. Zdefiniowanie dwóch, następujących po sobie z określonym interwałem czasowym, klatek kluczowych, poprzez np. zmianę położenia obiektu *MovieClip*, automatycznie generuje pozostałe klatki pomiędzy kluczowymi, tworząc animację ruchu.

Obiekty *MovieClip* są obiektami interaktywnymi, można za ich pomocą reagować na akcje użytkownika, takie jak: przesuwanie, kliknięcie myszą lub klawiszem. Obsługę zdarzeń zapewnia się poprzez dodanie odpowiednich *eventListener*'ów w sekcji kodu ActionScript. W przypadku wystąpienia zdefiniowanego wcześniej zdarzenia (np. *MouseEvent.CLICK*) *eventListener* wywołuje pożądaną funkcję zaimplementowaną w kodzie aplikacji. Z poziomu ActionScript możliwe jest zarządzanie wszystkimi obiektami znajdującymi się na scenie, a także tworzenie i dodawanie nowych obiektów podobnie jak wszystkie elementy dostępne z predefiniowanej biblioteki Flash np. *CheckBox*, *RadioButton*, *ComboBox*, *TextBox*, *ColorPicker*. ActionScript pozwala na pełne kontrolowanie osi czasu: wstrzymywanie, wznowianie przenoszenie się do konkretnych klatek. W aplikacji klienckiej wykorzystano także inne, programowane mechanizmy czasowe dla lepszej i łatwiejszej kontroli zdarzeń.

Wykorzystując wymienione główne zalety budowania aplikacji w technologii Flash, skonstruowano m.in. panel z zalogowanymi uczestnikami, dostępny z poziomu *Menu głównego*, *Klasa*. Na bazie obiektów *MovieClip* stworzono fabrykę obiektów dla pojedynczych pozycji w liście uczestników lekcji. W tym celu skomponowano szablon pojedynczego obiektu zawierającego w sobie miejsce na obraz jpg oraz kilka pól tekstowych odnoszących się do danych uczestnika. Taki szablon został wyeksportowany z biblioteki do klasy, tak aby był dostępny z poziomu kodu ActionScript. W chwili pobierania danych z serwera o aktualnie zalogowanych użytkownikach tworzone są kolejne instancje z bazowego szablonu, a następnie modyfikowane są wartości wewnątrz obiektu *MovieClip* dotyczące Avatara uczestnika, nazwy, statusu oraz grupy. Natomiast dla całego obiektu definiowana jest pozycja *X,Y* względem obiektu będącego jego rodzicem. Gotowe obiekty dodawane są do wspólnego kontenera który również jest typu *MovieClip*. Z wykorzystaniem mechanizmów *Osi czasu* realizowane są animacje pojawienia się gotowej listy na ekranie, elastycznych efektów rozszerzania i zanikania przy chowaniu obiektu.

Obiekty wymagane do połączenia z serwerem multimedialnym, obiekty wideo i obiekty przesyłające strumienie audiowizualne są importowane z pakietów dostarczanych przez firmę Adobe wraz z oprogramowaniem Flash Professional do tworzenia niniejszej aplikacji. Nie było potrzeby posiłkowania się dodatkowymi bibliotekami od zewnętrznych dostawców. Praktyczne zastosowanie tych obiektów jest opisane w rozdziale (4.2.3 Streaming audio-video).

4.1.2 WRAPPER WWW

Wrapper (z angielskiego) jest rodzajem opakowania, obiektu nadrzędnego służącego do przygotowania powłoki pomiędzy wewnętrznym obiektem, a zewnętrznym środowiskiem. Zastosowany w projekcie Wrapper WWW służy do obudowania wyjściowego pliku swf w strukturę kodu HTML. Dzięki temu możliwe jest zbudowanie całego serwisu internetowego wokół aplikacji Flash. W niniejszym projekcie Wrapper daje możliwość użycia Hiperłącza do wtyczki Adobe Flash Player, w przypadku braku obsługi tego standardu w systemie klienta oraz dodanie trywialnych znaczników HTML jak tytuł strony, wyrównanie, czy skalowanie obiektu.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <title>Edvir - Twoja szkoła marzeń</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
    <style type="text/css" media="screen">
      html, body { height:100%; background-color: #000000;}
      body { margin:0; padding:0; overflow:hidden; }
      #flashContent { width:100%; height:100%; }
    </style>
  </head>

```

```
<body>
  <div id="flashContent" align="center">
    <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-
      444553540000" width="1000" height="680"
      id="EdvirClient8" align="middle">
      <param name="movie" value="EdvirClient8.swf" />
      <param name="quality" value="high" />
      <param name="bgcolor" value="#000000" />
      <param name="play" value="true" />
      <param name="loop" value="true" />
      <param name="wmode" value="window" />
      <param name="scale" value="showall" />
      <param name="menu" value="true" />
      <param name="devicefont" value="false" />
      <param name="salign" value="" />
      <param name="allowScriptAccess" value="sameDomain"
    />
    <!--[if !IE]>-->
    <object type="application/x-shockwave-flash"
      data="EdvirClient8.swf" width="1000" height="680">
      <param name="movie" value="EdvirClient8.swf" />
      <param name="quality" value="high" />
      <param name="bgcolor" value="#000000" />
      <param name="play" value="true" />
      <param name="loop" value="true" />
      <param name="wmode" value="window" />
      <param name="scale" value="showall" />
      <param name="menu" value="true" />
      <param name="devicefont" value="false" />
      <param name="salign" value="lt" />
      <param name="allowScriptAccess" />

```

4.2 SERWER MULTIMEDIALNY

Wowza Media Server v2.24 jest uniwersalnym serwerem multimedialnym, zapewniającym funkcjonalność dla większości nowoczesnych systemów e-learning'owych. Możliwe jest udostępnianie materiałów dydaktycznych w formie zapisanych plików filmowych i dźwiękowych w formatach MP4, FLV, MP3. Streaming wideo w czasie rzeczywistym posiada wsparcie dla wielu formatów kompresji oraz protokołów przesyłania danych. Dzięki temu można zoptymalizować ruch sieci a przy okazji zapewnić obsługę zróżnicowanych odbiorców końcowych. Połączenie aplikacji klienckiej z serwerem możliwe jest z urządzeń opartych na platformach: Windows, Linux, Mac OS, iOS, Android. Wowza Media Server w opisywanym projekcie został zainstalowany w systemie Windows 7. Działa na zasadzie uruchomionej usługi, którą można kontrolować za pomocą linii komend lub chociażby z poziomu platformy Eclipse Indigo na której pisana była aplikacja serwerowa.

4.2.1 WYBRANA SPECYFIKACJA

PROTOKOŁY, MULTI-KLIENT STREAMING

Adobe Flash® RTMP (RTMPE, RTMPT, RTMPTE, RTMPS) Adobe Flash HTTP Streaming (HDS) RTSP/RTP	Flash Player Air Quicktime Player VideoLAN VLC player Mobile devices (3GPP) Other compliant players
Apple® HTTP Live Streaming (HLS)	iPhone®, iPod®, iTouch™ (iPhone OS 3.0 or later) iPad® QuickTime® Player (10.0 or later) Safari (4.0 or later on Snow Leopard) Roku® and other HLS compatible players
Microsoft® Smooth Streaming	Silverlight® 3 or later

KODOWANIE W CZASIE RZECZYWISTYM

RTMP	Video: H.264, VP6, Sorenson Spark®, Screen Shared codec Audio: AAC, AAC-LC, HE-AAC, MP3, Speex, NellyMoser ASAO
------	--

OBSŁUGIWANE FORMATY PLIKÓW

Wideo i audio	FLV (Flash Video - .flv) MP4 (QuickTime container - .mp4, .f4v, .mov, .m4v, .mp4a, .3gp, and .3g2) MP3 (.mp3)
---------------	--

WYMAGANIA SYSTEMOWE

Minimalne rekomendowane wymagania sprzętowe	CPU: Single Quad Core, 3.00 GHz or better RAM: 4GB Disk: 2 or more in RAID 0 (striping) Network: 1Gb Ethernet
Obsługiwane systemy operacyjne	Windows® (XP, Vista, 7, Server 2003 & 2008) Linux (all distributions) Solaris Mac® OS Unix

4.2.2 APLIKACJA SERWEROWA

Aplikacja serwerowa to program napisany w języku Java, skompilowany i odpowiednio przygotowany do pracy z serwerem multimedialnym Wowza jako niezbędne środowisko uruchomieniowe. Aplikacja dodawana jest do serwera w formie modułu. Moduły można ze sobą łączyć w bardziej rozbudowane aplikacje, a tym samym współdzielić różne mechanizmy z innymi aplikacjami. Np. do odtwarzania plików, obsługi strumieni audiowizualnych. W projekcie zostały połączone ze sobą trzy moduły: *Base* zapewniający podstawową funkcjonalność (jest konieczny dla każdej aplikacji), *Logging* służący do rejestrowania zdarzeń oraz *EdvirApplication* będący trzonem całej omawianej aplikacji serwerowej. Konfiguracji modułów dokonuje się poprzez utworzenie dokumentu XML z odpowiednimi argumentami i umieszczenie go w głównej lokalizacji serwera Wowza.

Serwer Wowza oferuje obsługę tzw. *Scopes* (często nazywanych też *Rooms*), czyli niezależnych pokoi składających się z głównej klasy serwerowej oraz podległych klas pomocniczych. W prezentowanym systemie takim pojedynczym pokojem jest jedna sesja lekcyjna. Wybór do którego pokoju ma się połączyć klient określa się odpowiednio przygotowanym adresem serwera.

```
nc.connect("rtmp://192.168.1.102/Edvir2/V3",
loginScreen.loginInput.text, loginScreen.passwordInput.text);
```

Tekst umieszczony po „/Edvir2/”, „v3” uwarunkowuje nazwę *Scope’a* z którym połączy się klient. Budowanie adresu może odbywać się z poziomu interfejsu użytkownika. Instancja głównej klasy aplikacji, *EdvirApplication* tworzona jest przy połączeniu pierwszego użytkownika do danego *Scope’a*. Tworzy się niezależne środowisko, od strony uczestnika jest w pełni odizolowane od innych grup. Z technicznego punktu widzenia ogniwnem łączącym jest jedynie baza danych z zapisanymi profilami użytkowników.

Główna klasa aplikacji, *EdvirApplication* dziedziczy po klasie *BaseModule*. Jest to klasa pochodząca z zasobów bibliotek Wowza. Do tej klasy zaimportowane zostały biblioteki dostarczane przez Wowza Media Systems które zapewniają pełną funkcjonalność serwera multimedialnego. Dodawana jest obsługa m.in. obiektów współdzielonych, protokołu *RTMP*, strumieniowania audio-video. W opisywanym systemie klasa pochodna implementuje funkcje *onAppStart()*, gdzie przygotowywane są wszystkie zmienne i mechanizmy do obsługi lekcji. Kolejną implementacją jest metoda *onConnect()*, gdzie odwołując się do bazy danych sprawdzana jest poprawność loginu i hasła, oraz to czy logujący się użytkownik jest nauczycielem. Jeśli nowy uczestnik posiada uprawnienia nauczyciela, to uaktualniane są jego dane w informacjach dotyczących lekcji, a obraz z kamery prowadzącego przekazywany jest do głównego kanału video.

```
public void onConnect(IClient client, RequestFunction function,
AMFDataList params) {
    try{
        EdvirLogger.info("onConnect: " + client.getClientId());
        // sprawdzenie argumentów
        if(params.get(PARAM1) != null &&
params.getString(PARAM2) != null ){
            //weryfikacja loginu i hasła

            if(EdvirDataBase.checkLogin(params.getString(PARAM1),
params.getString(PARAM2))){
                EdvirClient tmpClient = new
EdvirClient(client.getClientId(),
params.getString(PARAM1));
                //identyfikacja czy polaczona osoba jest nauczycielem
                if(EdvirDataBase.checkTeacher(tmpClient.userName)){
                    // zmiana status
                    tmpClient.status =
EdvirVariable.teacherStatusName;
                    // aktualizacja informacji o lekcji
                    publicInfoSharedObj.setProperty(EdvirVariable.le
ssonTeacherNameSOName, tmpClient.userName);

                    publicInfoSharedObj.setProperty(EdvirVariable.no
wSpeaksSOName, tmpClient.userName);

                    publicInfoSharedObj.setProperty(EdvirVariable.pu
blicStreamSOName1, tmpClient.userName + "
Stream");
                    tmpClient.isOnPublic = true;
                }
                clientsList.add(tmpClient);
                client.acceptConnection();
            }else{
                client.rejectConnection("incorrect login or password");
            }

        }else{
            client.rejectConnection("server expects no less than 2
arguments. Add username and password as fallow
parameters");
        }

    }catch (Exception e){
        EdvirLogger.error(e.getMessage());
    }
}
```

Innymi implementacjami metod klasy nadrzędnej są m.in. *onStreamCreate()*, wywoływany podczas uruchamiania transmisji audio-wideo. *onDisconnect()* czyszcząca dane po rozłączonym użytkowniku.

4.2.3 STREAMING AUDIO-VIDEO

Strumienie audio-video w prezentowanym systemie, od przechwycenia obrazu do wyświetlenia w aplikacji klienckiej przechodzi długą drogę i poddawany jest wielu modyfikacjom. Regulowana jest między innymi jakość ze względu na to, jak ważny z merytorycznego punktu widzenia jest dany strumień, a także sposób kodowania i formalne nazewnictwo dla jednoznacznej identyfikacji.

Przebieg strumienia audiowizualnego:

1. Tworzenie strumienia Audio-Video (dalej "AV") rozpoczyna się od ustalenia źródła tj. mikrofonu, kamery lub innego wejścia AV rozpoznanego przez aplikację jako domyślne urządzenie wejściowe obrazu i dźwięku.
2. Na strumień nakłada się parametry dotyczące wysokości i szerokości obrazu, stopnia kompresji, podkładu i ewentualnie innych filtrów obrazu. Dla kamery nauczyciela przyjmuje się 4 razy większą jakość obrazu w porównaniu z transmisją od klienta typu uczeń.
3. Następnie strumień jest lokalnie kodowany przy pomocy On2 VP6 i MP3, możliwe jest kodowanie do formatu H.264/MPEG-4 po poprawnym *shakehands* (jest to rodzaj uzgodnienia czy obie maszyny są w stanie kodować w tym formacie) z serwerem. Rodzaj kodowania dobierany jest przez zamknięte mechanizmy Adobe Flash Player, uwzględniając zadane wcześniej, w pkt.2 parametry.
4. Po zakodowaniu sygnału jest on publikowany za pośrednictwem obiektu *NetConnection* z użyciem niepowtarzalnej nazwy będącej zestawieniem jego loginu i losowego zestawu znaków.
5. Przy otrzymaniu prośby na publikację strumienia Aplikacja serwerowa rozpoznaje dostawcę transmisji i upewnia się m.in. czy dany strumień ma unikatową nazwę oraz czy jest to dodatkowy obraz, a może nadpisanie starego strumienia któregoś z użytkowników. Rozpoznawane jest także to, czy wysyłający klient jest nauczycielem, jeśli tak to podczas rozpoczynającej się lekcji automatycznie ustawiany jest jego strumień w publicznym kanale wideo. Jeśli serwer zgodzi się na publikację, klient rozpoczyna transmisję strumienia na serwer oraz uaktualniane są jego dane w obiektach współdzielonych aby nazwa strumienia była dostępna dla innych użytkowników.
6. Streaming czasu rzeczywistego jest budowany w oparciu o format flv i jest dostępny jedynie w czasie rzeczywistym, żadne informacje nie są archiwizowane a jedynie fragment strumienia jest przechowywany w pamięci tymczasowej. Podczas trwania transmisji. Nie ma też możliwości manualnej ingerencji w jakość sygnału, rozmiar itp. Wszystko to odbywa się dynamicznie, uwzględniając przepustowość łącza oraz możliwości sprzętowe po stronie aplikacji klienckiej.
7. Jeśli któryś z klientów podłączonych do serwera będzie chciał rozpocząć odbieranie strumienia, musi znać jego nazwę, która powinna być unikatowa, wówczas może zainicjować pobieranie strumienia. Nazwy strumieni są przechowywane w tabeli

- Clients* w obiekcie współdzielonym *ClientsSharedObject*, uaktualnianym przy każdorazowych zmianach parametrów zalogowanych użytkowników.
8. Strumień podcina się do obiektu *Video* znajdującego się na scenie. Ten rodzaj obiektu zdolny jest wyświetlić obraz wideo ze strumienia. Może być także wykorzystany do prezentacji utworzonych wcześniej plików wideo w formacie MP4/FLV, np. z instruktazem lub zapisanym wcześniej wykładem do lekcji.
 9. Klient pobiera strumień na swoją lokalną maszynę, tam jest dekodowany za pomocą wbudowanych w wtyczkę Flash, kodeków, a od tego momentu prezentowany na scenie w formie obrazu i dźwięku.

4.2.4 SKALOWALNOŚĆ

Wowza Media System, dostawca serwera multimedialnego Wowza, utrzymuje że pojedynczy komputer serwerowy, spełniający rekomendowane wymagania sprzętowe, zdolny jest do obsługi danych i strumieni audiowizualnych z maksymalną przepustowością 10Gbit/s. Podczas przeprowadzonych testów, program kliencki w trakcie pracy grupowej wykorzystywał około 1-1.2 Mbit/s. Praca grupowa jest najbardziej obciążającym łańcuchem elementem systemu, ponieważ właśnie wtedy na lokalną maszynę jest pobieranych do pięciu obrazów pochodzących od innych użytkowników, a także udostępniany i wysyłany na serwer jest obraz z własnej kamery uczestnika. Przyjmując że wszystkie aktywne lekcje byłyby w trakcie pracy grupowej to maksymalna ilość użytkowników wynosi w przybliżonej liczbie 8 tysięcy jednocześnie. Podczas pobierania pojedynczego strumienia nauczyciela, w trakcie typowej lekcji, generowany ruch wynosi ok 0.7 Mbit/s. Z założeniem że średnio $\frac{1}{4}$ czasu lekcji to praca grupowa, pojedynczy serwer w optymistycznym przypadku jest w stanie obsłużyć do około 12 tysięcy jednoczesnych użytkowników.

Aplikacja serwerowa systemu została przygotowana do pracy na zasadzie obiektów typu *Scope*, gdzie każdy *Scope* jest odrębną lekcją. Ogromną zaletą takiego rozwiązania jest zdolność do ciągłego powielania swojej instancji, a więc bez wiedzy użytkownika można bardzo dokładnie regulować wykorzystywaną przepustowość łącza przez serwer (z krokiem o wartość transferu generowanego przez pełną klasę, około 15 Mbit/s) i uruchamiać aplikację na kolejnej maszynie serwerowej. Możliwe jest zautomatyzowanie procesu balansowania łączy poszczególnych serwerów. W tym celu należałoby określić tzw. serwer *MASTER*, który zawsze przyjmowałby przychodzące połączenia, natomiast w zależności od połączonej liczby użytkowników do poszczególnych serwerów podlegających, typu *SLAVE*, wyznaczałby miejsce utworzenia nowej lekcji.

4.3 KOMUNIKACJA KLIENT-SERWER

4.3.1 METODY ZDALNE

Metody zdalne wywoływane z aplikacji klienckiej uruchamiają na serwerze metodę, która wykonuje jakieś czynności, a następnie może zwrócić rezultat swojego działania w postaci dowolnej, uniwersalnej wartości np. wartość logiczna *Boolean*, *String*, tablica obiektów. W omawianym systemie metody zdalne stanowią podstawę komunikacji klient-serwer ponieważ w aplikacji klienckiej, zwłaszcza z uprawnieniami nauczycielskimi dostępnych jest do wywołania wiele opcji, w jakimś stopniu oddziałujących na innych klientów. Większość wywoływanych metod ściśle powiązana jest z uaktualnieniem danych na serwerze i wykonaniem określonego kodu. Natomiast rzutowanie na innych klientów może odbywać się w inny sposób np. przez obiekty współdzielone lub zmianę strumieni wideo.

Procedura wywołania metody zdalnej rozpoczyna się w aplikacji klienckiej, przykład obrazuje dodanie dodatkowego dokumentu do porządku zajęć. Podczas naciśnięcia przycisku *Dodaj* obsługiwany jest następujący *event*:

```
function onDocAdd(e:Event = null):void{
    if(inputDocName.text != "" && inputDocUrl.text != ""){
        nc.call("insertLessonDocument", new
        Responder(onDefaultResult), inputDocName.text, inputDocUrl.text);
    }
}
```

Metody zdalne wywoływane są za pomocą funkcji *call()* na rzecz obiektu *NetConnection*. Pierwszym argumentem jest nazwa wywoływanej metody w aplikacji serwerowej. Drugi parametr definiuje funkcję odpowiedzialną za odpowiedź ze strony serwera na tę konkretną funkcję, w tym przypadku zwracana będzie wartość *Boolean*, oznaczająca powodzenie lub fiasko wykonania funkcji na serwerze. Kolejnymi parametrami są argumenty do funkcji wywoływanej pierwszym parametrem, mianowicie nazwa dokumentu oraz adres URL doń prowadzący. Na serwerze zostaje wywołana metoda *insertLessonDocument()*.

```
public void insertLessonDocument(IClient client, RequestFunction function,
AMFDataList params){
    try{
        if(params.size() > 1 && lessonDocumentsList.size() <=
        EdvirVariable.maxLessonDocumentsSize){
            EdvirLessonDocument tmpDocument = new
            EdvirLessonDocument(params.getString(PARAM1),
            params.getString(PARAM2));
            lessonDocumentsList.add(tmpDocument);
            rebuildLessonsDocumentsArray();
        }
    }
}
```

```
        sendResult(client, params, true);  
    }else{  
        sendResult(client, params, false);  
    }  
}catch (Exception e){  
    EdvirLogger.error(e.getMessage());  
}  
}
```

Po stronie serwera następuje sprawdzenie czy otrzymano oczekiwaną ilość parametrów oraz czy nie przekroczono już maksymalnej liczby dokumentów możliwych do jednoczesnego udostępnienia. Jeśli warunki zachodzą, jest tworzony nowy obiekt typu *EdvirLessonDocument* oraz uaktualniany jest *SharedObject* z informacjami do lekcji, przy pomocy metody *rebuildLessonsDocumentsArray()*. Rezultat operacji zwracany jest za pośrednictwem *sendResult()*.

Przechwycenie wyniku po stronie klienta realizowane jest funkcją *onDefaultResult* i polega na wyświetleniu zwróconej wartości w oknie konsoli.

```
function onDefaultResult(returnVal:Object):void{  
    trace("onDefaultResult: "+returnVal.toString());  
}
```

4.3.2 FUNKCJE BROADCAST'OWE

Funkcje broadcast'owe są specyficznym rodzajem funkcji, które są wywoływane po stronie klienta za pośrednictwem obiektów *SharedObject*. Funkcja wywołana u jednego z klientów, za pośrednictwem aplikacji serwerowej wywołuje lokalną funkcję u wszystkich innych klientów połączonych z tą samą instancją aplikacji serwerowej. Zaletami funkcji broadcast'owych jest mniej skomplikowana aplikacja serwerowa, a tym samym skoncentrowanie kodu po stronie klienta. Można ją wykorzystać do rozprzestrzeniania informacji, które mają znaczenie tylko dla osób aktywnie zalogowanych. Informacje które szybko się przedawniają i nie są istotne dla osób, które zalogują się do systemu później lub z jakiegoś innego powodu będą miały przerwę w zajęciach. Przykład śledzi realizację funkcji broadcast'owej podczas wysyłania wiadomości do czatu globalnego.

```
so.send("receiveMessage", new Array( loggedUserName + ": " +  
inputMsg.text, textColorPicker.selectedColor.toString(16) ));
```

Pierwszym argumentem jest nazwa funkcji, która ma być wywołana, natomiast drugim argumentem jest obiekt, będący parametrem przekazywanym do wskazanej pierwszym argumentem funkcji. W tym przypadku jest to tablica obiektów, pierwszy typu *String*,

zawierający treść wypowiedzi w czacie. Drugi element to kolor tekstu zapisany w formie heksadecymalnej

U wszystkich połączonych klientów zostanie wywołana funkcja *receiveMessage*, która w przypadku otrzymania dwóch parametrów uwzględni kolor tekstu w przysłanej wiadomości a następnie doda wpis do głównego pola tekstowego czatu *msgArea*.

```
function receiveMessage(params:Array = null):void
{
    if(params[1] != null){
        var tmp:String = "<FONT COLOR=\"#" + params[1].toString() +
            "\">" + params[0].toString() + "</FONT>";
        msgArea.htmlText += tmp;
    }else{
        msgArea.htmlText += params[0].toString();
    }
    msgArea.verticalScrollPosition += 10000;
}
```

4.3.3 OBIEKTY WSPÓLDZIELONE

Obiekty współdzielone są specjalnym typem obiektów, które współdzielą informacje przechowywane na serwerze pomiędzy wszystkimi połączonymi programami klienckimi. Obiekty współdzielone nazywane *SharedObject's* zapewniają aplikacji klienckiej funkcjonalność do tworzenia swoich instancji, modyfikowania zawartości, a także informowania innych uczestników o zmianach w obiekcie. *SharedObjects* mogą być naturalnie tworzone za pośrednictwem aplikacji serwerowej. Obsługa zmian odbywa się poprzez odpowiedni *EventListener*.

Wszystkie obiekty współdzielone tworzone są w aplikacji serwerowej, dla przykładu utworzenie obiektu z podstawowymi danymi publicznymi o przebiegu lekcji wykonuje się następującą sekwencją kodu:

```
ISharedObjects sharedObjects = appInstance.getSharedObjects(false);
publicInfoSharedObj =
sharedObjects.getOrCreate(EdvirVariable.publicInfoSharedObjName);
publicInfoSharedObj.acquire();
initialPublicData();
```

Początkowo pobierany jest zbiór obiektów współdzielonych z instancji aplikacji za pomocą metody *appInstance.getSharedObjects()*, następnie na rzecz tego obiektu wywołuje się metodę *getOrCreate()* która jako parametr przyjmuje wartość *String*, odpowiadającą za nazwę nowego obiektu współdzielonego. Funkcja *initialPublicData()* inicjalizuje zmienne publiczne, aby aplikacje klienckie nie odwoływały się do obiektów typu *null*.

```
publicInfoSharedObj.setProperty(  
    EdvirVariable.groupWorkInProgressSOName, false);
```

Inicjalizacja oraz modyfikacja parametrów następuje za pomocą funkcji *setProperty()*, która za pierwszy argument przyjmuje String nadając nazwę zmiennej, natomiast drugi jest dowolną wartością lub tablicą obiektów.

Aby zautomatyzować procedurę uaktualniania programu klienckiego nowymi danymi z obiektu współdzielonego, należy taki obiekt zainicjalizować na podstawie aktywnego połączenia.

```
so3 = SharedObject.getRemote("publicInfoSharedObj", nc.uri, false);  
so3.client = this;  
so3.connect(nc);  
so3.addListener( SyncEvent.SYNC, alertSO3 );
```

Pobranie *SharedObject* z instancji aktualnie połączonego *NetConnection* jest wykonywane funkcją *getRemote(.)*. Dla bezpośredniej obsługi jakichkolwiek zmian na obiekcie współdzielonym dodaje się do niego *EventListener*, który w chwili synchronizacji obiektu współdzielonego wywoła funkcję *alertSO3()*.

```
if (so3.data.groupWorkInProgress == true){  
    if(so3.data.groupWorkTime.length > 4){  
        groupWorkTimeL.htmlText = "<FONT COLOR=#E2FFE0\"  
FACE=\"Arial\" SIZE=\"25\"> " + so3.data.groupWorkTime + "</FONT>";  
    }else{  
        groupWorkTimeL.htmlText = "<FONT COLOR=#E2FFE0\"  
FACE=\"Arial\" SIZE=\"25\"> " + so3.data.groupWorkTime + "</FONT>";  
    }  
    if(so3.data.groupWorkPause == true){  
        showGroupLight("red");  
        groupWorkInfoL.text = "Praca grupowa wstrzymana.";  
    }else{  
        showGroupLight("green");  
        groupWorkInfoL.text = "Trwa praca grupowa.";  
    }  
}
```

Bezpośrednio w kodzie mamy dostęp do danych przechowywanych w obiekcie współdzielonym za pomocą notacji kropkowej *so3.data*. W przykładzie interpretowane są informacje dotyczące pracy grupowej. Sprawdzane są m.in. flagi czy praca grupowa trwa, czy jest wstrzymana a także obsługiwane jest pole wyświetlające zegar pracy grupowej i kolorowa dioda statusu.

4.4 BAZA DANYCH

4.4.1 STRUKTURA XML

Informacje konieczne do stałego przetrzymywania na serwerze, niezależnie od jego pracy, zostają zapisane do odpowiednio sformatowanego dokumentu XML. Format XML pozwala na przechowywanie danych w jasny i ustrukturyzowany sposób. Dzięki swej popularności i zuniwersalizowaniu silników XML możliwe jest przenoszenie danych pomiędzy różnymi platformami i wykorzystanie z różnymi językami programowania. Dokumenty w tym formacie są łatwe do przeglądania a także bardzo podatne na edycję i możliwość rozbudowy.

4.4.2 PRZECHOWYWANIE DANYCH

Dane użytkowników jakie są przechowywane w omawianym projekcie zawierają informację o nazwie użytkownika, zapisanym hasle oraz jego statusie rozpoznawanym w systemie i determinującym jego prawa w programie klienckim. Podczas logowania do systemu wyszukiwany jest użytkownik wg. *username*, następnie podane hasło jest kodowane funkcją skrótu i porównywane z hasłem zapisanym w dokumencie. Po poprawnym dopasowaniu haseł zwracana jest informacja o poprawnym użytkowniku, w każdym innym przypadku tj. nie odnalezienie użytkownika lub nie dopasowania haseł funkcja odpowiedzialna za sprawdzenie użytkownika zwraca wartość negatywną. Sprawdzenie statusu odbywa się za pośrednictwem innej funkcji i wykonywana jest w następnej kolejności po logowaniu. Rezultat sprawdzenia powraca do użytkownika wraz z jego tymczasowym (sesyjnym) numerem *clientId*.

Format dokumentu na przykładzie jednego użytkownika:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Edvir>
  <user>
    <username>
      Łukasz Sawaściuk
    </username>
    <password>
      0000000000000000ad1a0c016daa3cadb5d04bcc28a96bcc
    </password>
    <status>
      Teacher
    </status>
  </user>
</Edvir>
```

4.4.3 BEZPIECZEŃSTWO

Bezpieczne przechowywanie haseł jest zapewnione poprzez wykorzystanie algorytmu MD5. Jest to kryptograficzna, jednokierunkowa funkcja skrótu, która z dowolnego ciągu znaków generuje unikatowy skrót. W aplikacji serwerowej zaimplementowano generator funkcji *MD5* tworzący skrót o długości 48 bitów. Metoda tworzenia skrótu jest wywoływana przy tworzeniu nowego użytkownika, a następnie każdorazowo podczas sprawdzania czy podane hasło jest zgodne z tym zdefiniowanym w dokumencie XML.

5. PODSUMOWANIE

5.1 REZULTAT KOŃCOWY

Edvir to uniwersalna platforma, wspomagająca przeprowadzanie zdalnych lekcji, z wykorzystaniem nowoczesnych technologii do komunikacji. Najnowsze narzędzia do tworzenia oprogramowania klienckiego w technologii Flash pozwoliły na stworzenie przejrzystego, ergonomicznego i łatwego w obsłudze interfejsu użytkownika. Zarówno uczniowie, jak i nauczyciele będący użytkownikami systemu niemal natychmiast są w stanie zorientować się w jego obsłudze. Jedną z zalet wyróżniających prezentowany system, na tle innych rozwiązań do e-learningu jest stała transmisja audiowizualna pomiędzy nauczycielem i uczniami. Stały kontakt wzrokowy, komunikacja werbalna i niewerbalna odwołują się do tradycyjnych form nauczania i przekazywania wiedzy, wzmacniają poczucie jedności grupy oraz budują silną więź pomiędzy wszystkimi uczestnikami zajęć. W trakcie sesji lekcyjnej również uczniowie często występują publicznie więc jest to komunikacja w pełni dwukierunkowa.

Aby uatrakcyjnić typową lekcję, gdzie osoba prowadząca referuje przygotowany materiał, uczniowie mają możliwość pracy grupowej i własnego opracowywania dowolnych zagadnień. Podczas takiego trybu, wszyscy członkowie grupy wzajemnie się widzą i słyszą. Dostają zadanie do wykonania, a wyniki ich pracy sporządzane są we wspólnym dokumencie grupowym. Każdy z uczestników ma możliwość dodawania i modyfikowania treści dokumentu. O tym co się w nim znajdzie, uczniowie na bieżąco dyskutują i wprowadzają zmiany do dokumentu, na podstawie którego później będzie zweryfikowana ich aktywność.

W trakcie trwania zajęć uczniowie mają stały dostęp do zewnętrznych materiałów, którymi mogą być dowolne pliki, np. wykłady lub ćwiczenia zapisane w formacie PDF. W ramach materiałów zewnętrznych nauczyciele mogą udostępniać dowolne odnośniki, np. do filmów instruktorzowych. Uczniowie automatycznie są informowani z którego materiału powinni aktualnie korzystać i którą stronę dokumentu omawia prowadzący zajęcia. W dowolnym momencie lekcji uczniowie mogą też zadać pytanie

prowadzącemu. Dyskusja z nauczycielem jest prowadzona na dwóch publicznych kanałach wideo, dzięki czemu wszyscy przyglądają się rozmowie, w której omawiane zagadnienia mogą być przydatne również dla nich. Podczas pracy grupowej upublicznienie rozmów z nauczycielem sprzyja rywalizacji na zasadach fair play.

W aplikacji końcowej nacisk położono na jakość interfejsu, odpowiednie zorganizowanie i zachowanie się obiektów na ekranie głównym, w tym animacje ruchu, efekty przenikania i transformacji. Funkcje które nie są w danym momencie istotne, automatycznie są ukrywane, dzięki temu użytkownicy mają dostęp tylko do aktualnie przydatnych czynności i mogą skupić się na treści merytorycznej kursu.

Oprogramowanie serwerowe Wowza Media Server oraz zintegrowane środowisko programistyczne Eclipse Indigo IDE pozwoliło na szybkie i efektywne tworzenie kodu aplikacji serwerowej. Jest ona spoiwem łączącym rozproszonych klientów i to tutaj wykonywana jest największa praca związana z transmisją audiowizualną oraz logiką i funkcjonalnością całego systemu. Do należytej koordynacji rozproszonych klientów końcowych, wykorzystywany jest szeroki wachlarz technik komunikacji, zaliczają się do nich metody uruchamiane zdalnie na serwerze, funkcje broadcast'owe, a także obiekty współdzielone o rozmaitych przeznaczeniach. Odpowiednie zastosowanie tych metod umożliwia optymalne zorganizowanie ruchu sieciowego i zapewnienie wymaganej niezawodności przekazu newralgicznych informacji. Samo środowisko uruchomieniowe aplikacji serwerowej, Wowza Media Server, umożliwia transmisję materiałów w popularnych formatach audio-video, co w połączeniu z równie rozpowszechnionym standardem obsługi aplikacji Flash umożliwia uruchomienie klienta systemu na zdecydowanej większości współczesnych komputerów, tabletów i urządzeń mobilnych.

5.2 ZASTOSOWANIE

Rezultat który udało się osiągnąć, z powodzeniem podołałby prowadzeniu zdalnych warsztatów, szkoleń firmowych lub innych, dowolnych kursów internetowych. System mógłby też częściowo uniezależnić nauczanie ustawowe od tradycyjnego uczestnictwa w zajęciach realizowanych w budynkach szkolnych. Przyczyniłoby się to do zwiększenia dostępności edukacji, poprawienia jakości nauczania oraz wyrównania szans zdobycia dobrego wykształcenia na terenach słabo zaludnionych. W połączeniu z mechanizmami przeprowadzania testów oraz dostępem do rozbudowanej sieci intranet, stworzony system mógłby być solidnym fundamentem wirtualnej szkoły z prawdziwego zdarzenia. Multimedialną platformą, która pod względem przekazywania i weryfikacji zdobytej wiedzy, sprostałaby wymaganiom tradycyjnej szkoły.

BIBLIOGRAFIA

- [1] *Adobe Flash CS4/CS4 PL Professional. Biblia*, Helion, 2010
- [2] *Wowza Media Server 2: User's Guide*, Wowza Media Systems, Inc, 2010
- [3] *Wowza Media Server 2: Server Side API*, Wowza Media Systems, Inc, 2010
- [4] *Wowza IDE: User's Guide*, Wowza Media Systems, Inc, 2007
- [5] ActionScript 3.0 Language and Components Reference
http://help.adobe.com/pl_PL/AS3LCR/Flash_10.0/
- [6] Publish and play a live stream (RTSP/RTP based encoder)
<http://www.wowza.com/forums/content.php?354-How-to-publish-and-play-a-live-stream-%28RTSP-RTP-based-encoder%29>
- [7] On2's Flix Live encoder with Wowza Server (RTMP)
<http://www.wowza.com/forums/content.php?106-Using-On2-s-Flix-Live-encoder-with-Wowza-Server-%28RTMP%29>
- [8] Modifying XML file in Java – (JDOM Parser)
<http://www.mkyong.com/java/how-to-modify-xml-file-in-java-jdom>
- [9] Install Wowza Media Server (video tutorial)
<http://www.wowza.com/forums/content.php?288-Getting-Started-%28Video-Tutorial%29>
- [10] Red5 – Beginning Rooms and Shared Objects
<http://www.stumbleupon.com/url/lelandcope.com/red5-beginning-rooms-and-shared-objects/>
- [11] Dynamic streaming in Flash Media Server 3.5 - ActionScript 3.0 dynamic stream API
http://www.adobe.com/devnet/flashmediaserver/articles/dynstream_advanced_pt2.html
- [12] flashZONE forum <http://www.flashzone.pl/forum.php>

Poprawność wszystkich odnośników została sprawdzona w dniu 12.01.2012.

OPIS ILUSTRACJI

- Rys. 2.1 Adobe Flash Professional CS5.5 edycja Klienta systemu. 6
- Rys. 2.2 Eclipse Indigo IDE 3.7.1 – otwarty projekt aplikacji, uruchomiony serwer Wowza. 7

Rys. 3.1 Klient aplikacji w trakcie testów w środowisku sieciowym.....	8
Rys. 3.2 Ekran logowania.	9
Rys. 3.3 Ekran główny.....	10
Rys. 3.4 Komponenty wideo.....	11
Rys. 3.5 Czat globalny.	12
Rys. 3.6 Uczeń podczas rozmowy z nauczycielem.	13
Rys. 3.7 Panel zadawania pytań.....	13
Rys. 3.8 Lekcja i klasa.	14
Rys. 3.9 Przebieg lekcji.	15
Rys. 3.10 Materiały dodatkowe.	16
Rys. 3.11 Zakładka Profil	17
Rys. 3.12 Ekran główny nauczyciela.....	17
Rys. 3.13 Zakładka Ustawienia.	18
Rys. 3.14 Harmonogram lekcji.....	19
Rys. 3.15 Ustawienia pracy grupowej.	20
Rys. 3.16 Przedstawianie się uczestników.....	21
Rys. 3.17 Zarządzanie materiałami do lekcji.....	21
Rys. 3.18 Dodawanie użytkowników	22
Rys. 3.19 Obsługa materiałów zewnętrznych.....	23
Rys. 3.20 Notyfikacje	23
Rys. 3.21 Obsługa pytań.....	24
Rys. 3.22 Podgląd dokumentów grupowych.	25
Rys. 3.23 Komunikacja wewnątrz grupy.....	26
Rys. 3.24 Przykład dokumentu grupowego.....	27
Rys. 3.25 Tworzenie dokumentu grupowego.	28

Wszystkie ilustracje pochodzą z własnego opracowania.