

Modernizacja aplikacji w Lotus Notes Domino z wykorzystaniem technologii XPages

Lotus Notes Domino jest rozbudowanym środowiskiem bazodanowym zorientowanym na dokumenty. Prócz zadań, z którymi jest najczęściej wiązany, takimi jak obsługa poczty, kalendarz czy wymiana wiadomości, ma on znacznie większą liczbę zastosowań dzięki możliwości szybkiego tworzenia aplikacji na tę platformę (rapid application development). Często jednak szybko zaprojektowane aplikacje, choć spełniały swoje zadanie, wyglądały dość surowo i były mało interaktywne. Poprawa tego stanu wymagała większego nakładu pracy i mieszania warstwy prezentacji z warstwą danych. Od wersji Lotus Notes Domino 8.5 ten problem został rozwiązany przez wprowadzenie technologii XPages.

Aplikacje Notes do tej pory

Podstawowymi elementami aplikacji Notes są formularze i widoki. Pierwsze reprezentują opis pojedynczego dokumentu i pozwalają na jego utworzenie, podgląd zawartości oraz edycję. Drugie natomiast pozwalają na wylistowanie wszystkich dokumentów (lub danych z poszczególnych ich pól) spełniających określone warunki. Dotychczas, to jak zostały zrealizowane te elementy, wpływało zarówno na reprezentację danych przechowywanych w aplikacji, jak i reprezentację tychże w kliencie czy przeglądarce. Ich standardowy wygląd, jak to było już wspomniane, jest dość surowy. Zmiana wyglądu aplikacji bez opisanej w dalszej części artykułu technologii jest pracochłonna, wymaga manewrowania wieloma opcjami i nie wszystkie wprowadzone zmiany są widoczne już na etapie projektowania (niepełny WYSIWYG), niełatwe jest także umieszczanie obiektów dynamicznych, jak na przykład filmów czy interaktywnych wykresów. Dodatkowo zmiany te są inne w przypadku tej samej aplikacji wykorzystywanej w kliencie Lotus Notes oraz przeglądarce. Do tej pory były to niemalże odrębne byty, wymagające od programistów przynajmniej podwójnej pracy, aby w obu środowiskach uzyskać te same lub przynajmniej podobne efekty i sposób działania oraz wygląd.

XPages jest warstwą prezentacji dla aplikacji Notes wykorzystującą najnowsze webowe standardy takie jak: JavaScript, Server-side JavaScript, Ajax, Dojo Toolkit, JavaServer Faces. Dzięki niej wszystkie poprzednio wymienione problemy zostają wyeliminowane.

XPages i jego elementy

Strony XPages projektowane są w podobny sposób jak aplikacje okienkowe .NET czy Java, dlatego każdy kto zetknął się z tego typu programowaniem nie będzie miał problemów z rozpoczęciem pracy w tej technologii. Do dyspozycji programisty oddany jest zestaw podstawowych kontrolki, z których można budować stronę poprzez przeciąganie ich na jej obszar. Są wśród nich między innymi przyciski, pola tekstowe, pola edycji, ramki i wiele innych, widać je na rysunku 1. Jednak głównymi elementami, z których strona jest budowana są kontrolki własne (*Custom Controls*). To obiekty pozwalające grupować pozostałe kontrolki, przez co umożliwiają przygotowanie fragmentów stron spełniających konkretne zadania. Dzięki nim oszczędzany jest nakład pracy przez ponowne wykorzystanie zaprojektowanych fragmentów. Można o nich myśleć jak o analogii klas w programowaniu obiektowym. Każda z kontrolki posiada zestaw właściwości, które można

1981



Czy wiesz, że:

Rok 1981 – W laboratoriach IBM skonstruowano mikroskop tunelowy. Za opracowanie mikroskopu tunelowego dwóch pracowników IBM – Gerd Binnig oraz Heinrich Rohrer – odebrali w roku 1986 Nagrodę Nobla z fizyki.

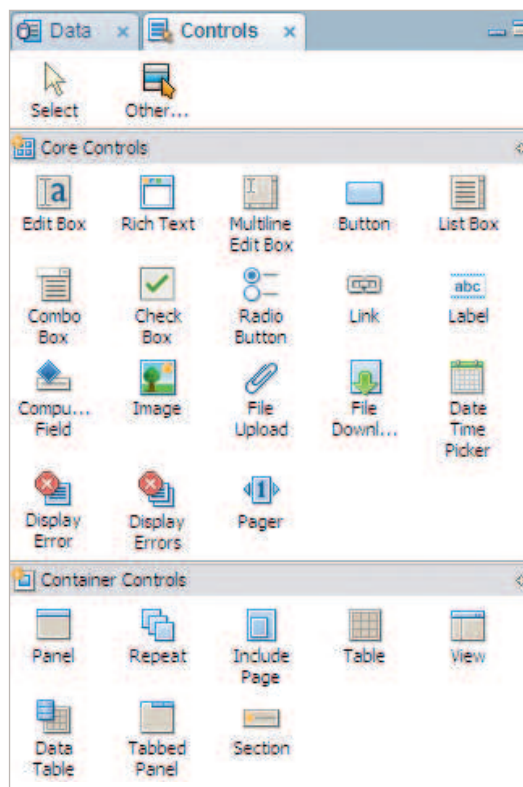
ustawiać, by zmienić ich zachowanie i/lub wygląd. W celu uzyskania pożądanego zachowania wykorzystywany jest system zdarzeń oraz obliczalnych wartości. Ich obsługa pisana jest w języku JavaScript. Natomiast wygląd, zarówno pojedynczych kontrolek jak i całej strony, opisywany jest za pomocą standardowych arkuszy stylów CSS. Językami jakie skrywają się za opisem całej strony są XML i HTML. XML jest głównym językiem opisującym strukturę strony, a jego kod jest automatycznie generowany w trakcie projektowania. HTML natomiast może być wykorzystywany w naturalny sposób do ustalania układu strony czy dodawania innych webowych elementów. Ostatnim zestawem narzędzi, jaki zostaje oddany programiście do wykorzystania we własnych aplikacjach, jest Dojo Toolkit, który jest biblioteką JavaScript zawierającą wiele gotowych elementów dynamicznych takich jak wykresy, formularze, galerie czy obiekty video. Dzięki temu, że XPages jest oparte na standardowych i popularnych technologiach webowych, łatwo jest je poznać i rozpocząć pracę z aplikacją.

Przykład modernizacji aplikacji Notes

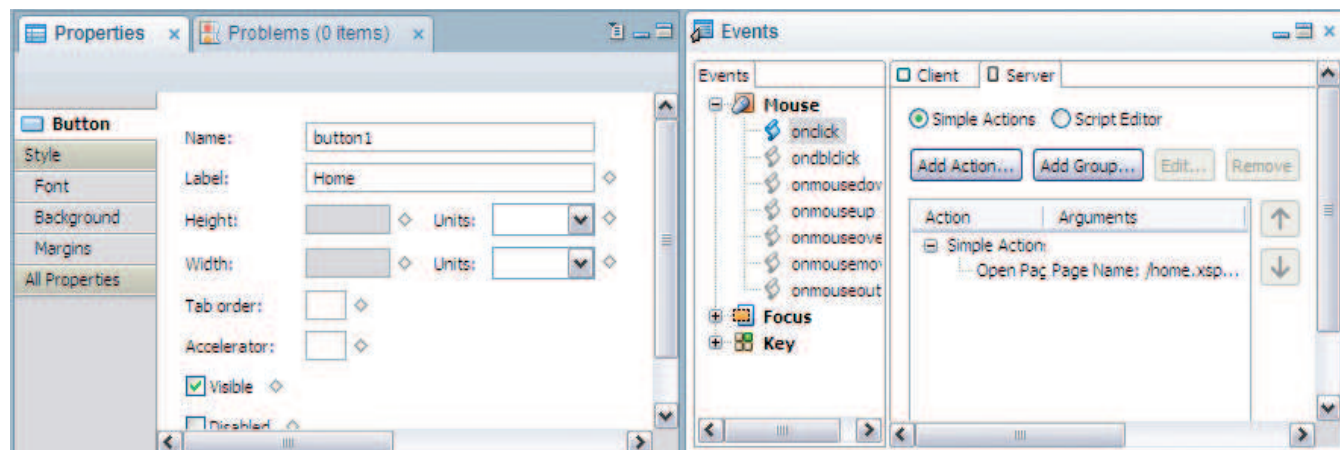
Pracę z XPages najlepiej jest rozpocząć w analogiczny sposób, jak przy projektowaniu zwykłych stron internetowych. Zakładając, że mamy już

działającą aplikację Notes, powinniśmy wyróżnić jej najbardziej ogólne fragmenty. Zazwyczaj będą one reprezentacją formularzy oraz widoków, znajdujących się już w aplikacji, i zostaną one zrealizowane w postaci kontrolek własnych.

Rysunek 1. Paleta kontrolek dla XPages



Rysunek 2. Palety właściwości i zdarzeń dla kontrolek



Znając architekturę aplikacji powinniśmy przygotować projekt układu graficznego strony, ustalając rozmieszczenie menu, list, tabel i innych jej elementów oraz określić z ilu i jakich podstron będzie się ona składała. Dobrym zwyczajem jest przygotowanie najpierw makiety strony przez grafika. Gdy już będziemy wiedzieli, jak ma wyglądać nasza aplikacja następnym krokiem będzie utworzenie wszystkich wspomnianych wcześniej kontrolki własnych, a następnie złożenie z nich stron, opisanie ich zachowań i zastosowanie szablonów stylu.

Za przykład wykorzystania XPages posłuży aplikacja wypożyczalni filmów. Jej pierwotny stan, przed wprowadzeniem jakichkolwiek zmian, widać na rysunku 3. Jest ona prosta, składa się tylko z domyślnej ramki z zestawem widoków, podstawowymi akcjami dodawania, edycji i usuwania elementów. Realizując przedstawione wcześniej kroki wymieniamy elementy składowe aplikacji:

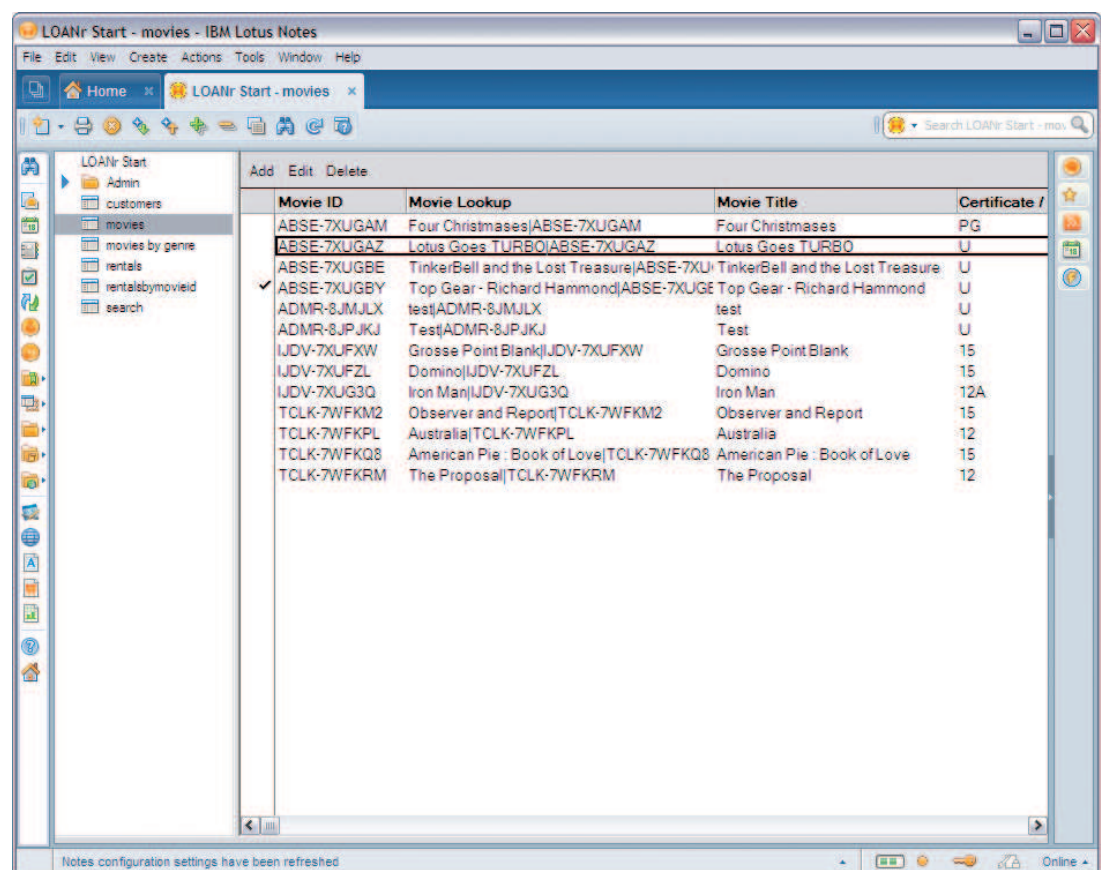
Formularze:

- **customer** – dokument z informacjami o kliencie
- **movie** – dokument z informacjami o filmie
- **RefData** – dokument z informacjami administracyjnymi (gatunki, certyfikaty, kraje)
- **rental** – dokument z informacją o wypożyczeniu

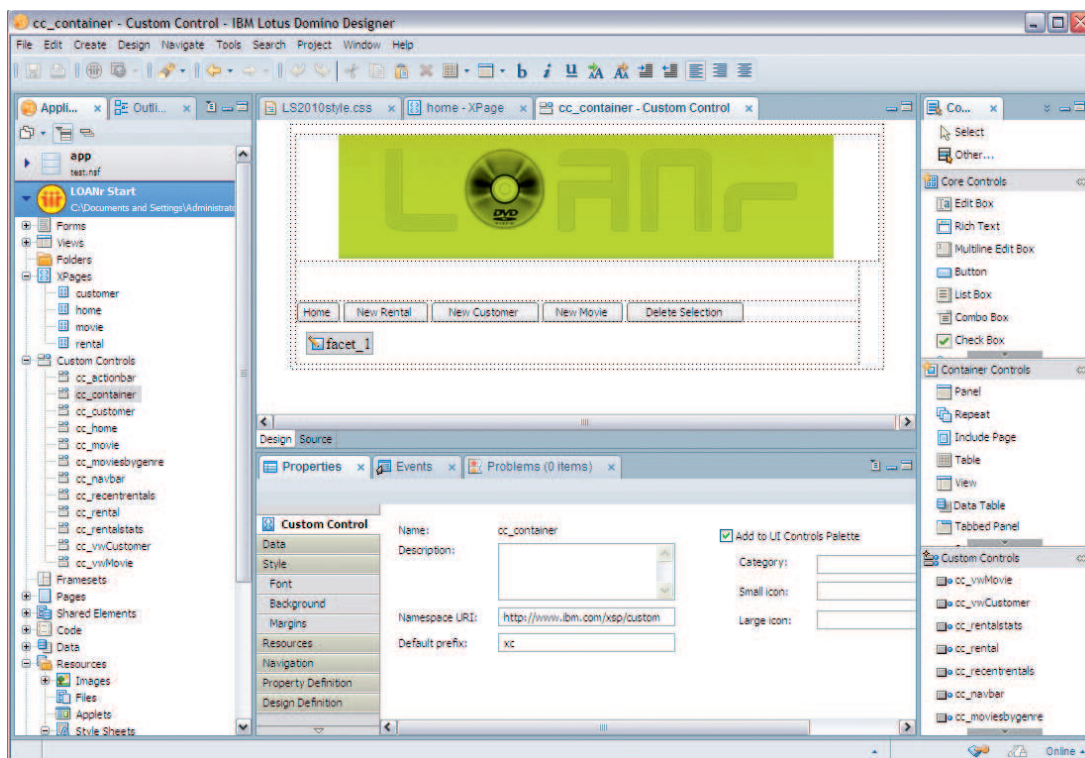
Widoki:

- **Admin\Reference Data** – lista dokumentów administracyjnych
- **customers** – lista klientów
- **movies** – lista filmów
- **movies by genre** – lista filmów grupowana wg gatunków
- **rentals** – lista wypożyczeń
- **rentals by genre** – lista wypożyczeń grupowana wg gatunku filmu
- **search** – przeszukiwanie dokumentów

Jak widać aplikacja jest prosta i łatwo jest sobie wyobrazić, jak powinna wyglądać. W tym przypadku na stronie głównej znajdują się wszystkie listy przełączane zakładkami. Ponadto do każdego dokumentu (formularza) będzie osobna kontrolka własna do jego podglądu i edycji. Potrzebne też będzie menu z opcjami utworzenia i kasowania dokumentów. Pierwszym elementem jaki zostanie utworzony będzie kontrolka *cc_container* (rysunek 4), będąca podstawową formą strony, zawierającą logo, menu główne i obszar edytowalny (*Editable Area*) pozwalający na umieszczenie treści danej strony w ramach tego obiektu. Do odpowiedniego ukształtowania elementów wykorzystane są panele (odpowiednik *div*ów w HTML-u) i style CSS. Plik szablonu stylów można dodać w menu *Styl* panelu właściwości, stamtąd też



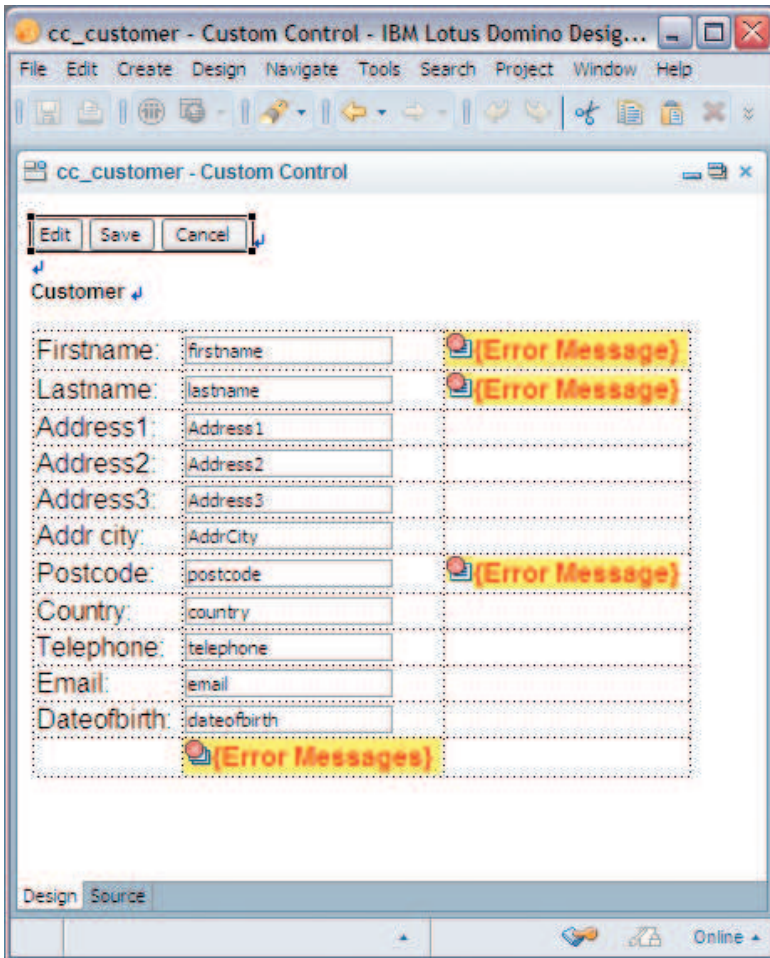
Rysunek 3. Aplikacja LOANr przed zmianami



Rysunek 4.
Kontrolka cc_container
w edytorze Domino

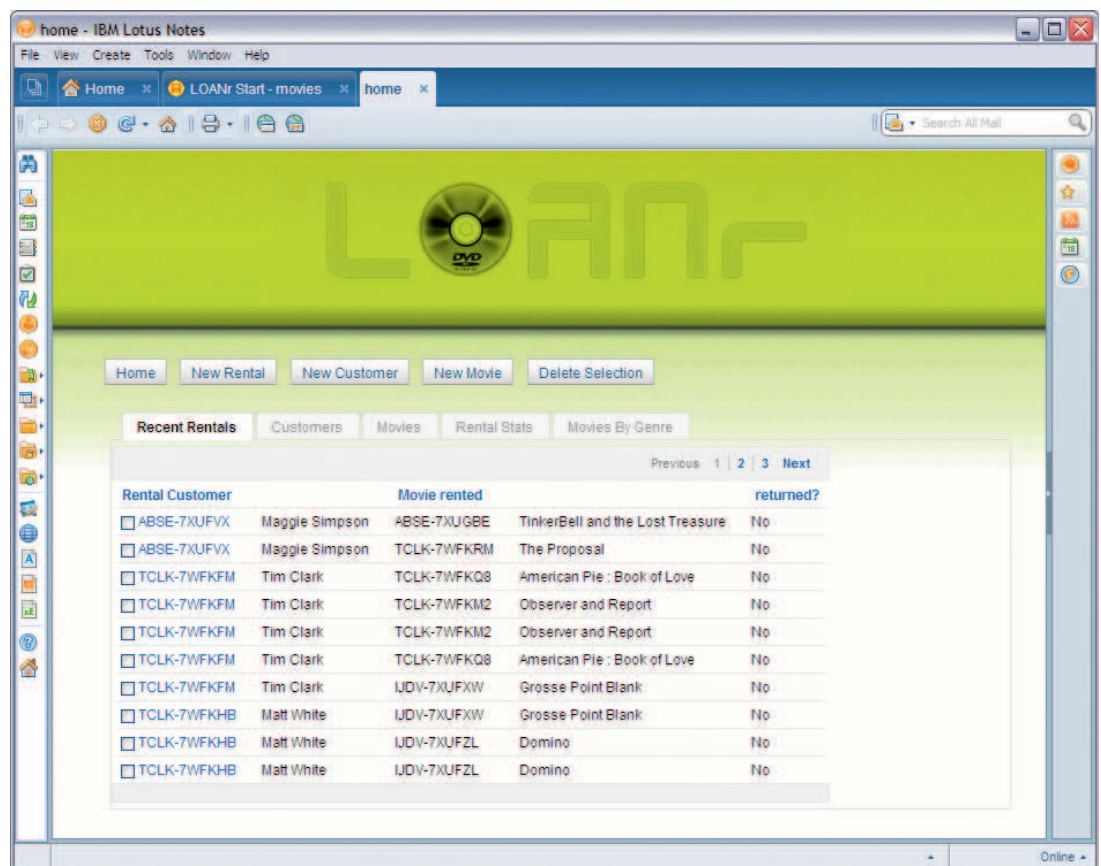
można wybrać klasę, jaka ma być przypisana do zaznaczonego elementu lub też wybrać odpowiednie ustawienia i dodać je jako nową klasę do wybranego arkusza. Menu główne jest kolejną kontrolką własną (*cc_navbar*), składa się z czterech przycisków: *Home*, *New Rental*, *New Customer*, *New Movie* i *Delete Selection*, przy czym ostatni jest wyświetlany warunkowo w zależności od tego czy użytkownik znajduje się na stronie głównej. Efekt ten można uzyskać wyliczając własność „widoczne” (*visible*) przycisku w zależności od zawartości zmiennej określającej aktualną stronę (również należy ją utworzyć). Natomiast do zdarzenia *onclick* przycisków należy dodać polecenie otwarcia odpowiedniej strony (dla pierwszych czterech) i kod usunięcia zaznaczonych na liście elementów (dla ostatniego). Następnie *cc_container* posłuży do utworzenia czterech stron XPages, do których będą prowadziły pierwsze cztery przyciski menu. Będą to strony *home*, *customer*, *movie* i *rental*. *Home* będzie zawierała kontrolkę z widokami list, natomiast pozostałe będą pozwalały na podgląd, edycję i tworzenie odpowiadających sobie dokumentów. Dla przykładu przedstawiona zostanie jedynie budowa *home* i *customer*, ponieważ pozostałe powstają w analogiczny sposób. Dla strony *home* zostanie utworzona kontrolka *cc_home*, jej głównym elementem będzie panel z zakładkami (*Tabbed Panel*), a w każdej z zakładek znajdzie się widok listy dokumentów, również przygotowany w osobnej kontrolce. Przykładowo w pierwszej zakładce,

nazywającej się *Recent Rentals*, umieszczona zostanie kontrolka *cc_recentrentals*. W zakładce *Dane* panelu właściwości każdej kontrolki własnej jest pole *źródła danych*, pozwalające wybrać elementy z których będą czerpane informacje, takich jak widoki czy formularze. Dla *cc_recentrentals* w tym przypadku należy dodać widok *rentals*, a następnie kontrolkę *widok*. Podczas dodawania tej ostatniej pojawi się okno wyboru danych wyświetlanych w tym widoku i lista dostępnych źródeł danych, z których należy wybrać to uprzednio dodane. W ten sposób pierwsza zakładka jest gotowa i analogicznie zostają zrealizowane pozostałe. Na koniec do zdarzenia *onclick* każdej z nich należy dodać kod ustawiający zmienną *TabID*, tak by było wiadomo, z której listy w danej chwili będą kasowane elementy za pomocą przycisku *Delete Selection*. Gdy już strona *home* jest gotowa należy przygotować pozostałe, które będą wyświetlały dokumenty i pozwalały na ich edycję. Dla strony *customer* przygotowana zostanie kontrolka *cc_customer* (rysunek 5). Po jej utworzeniu najpierw należy dodać źródło danych czyli formularz *customer*. Następnie dodany zostanie *cc_actionbar*, jest to kontrolka, którą również należy wcześniej przygotować. Składa się z trzech przycisków *Edit*, *Save* oraz *Cancel* pozwalających odpowiednio przejść do edycji dokumentu, zapisać zmiany oraz anulować je. Po dodaniu przycisków wprowadzony zostanie nagłówek dokumentu *Customer*. Będzie on zwykłym tekstem statycznym, a taki można wpisać bezpośrednio w dokumencie

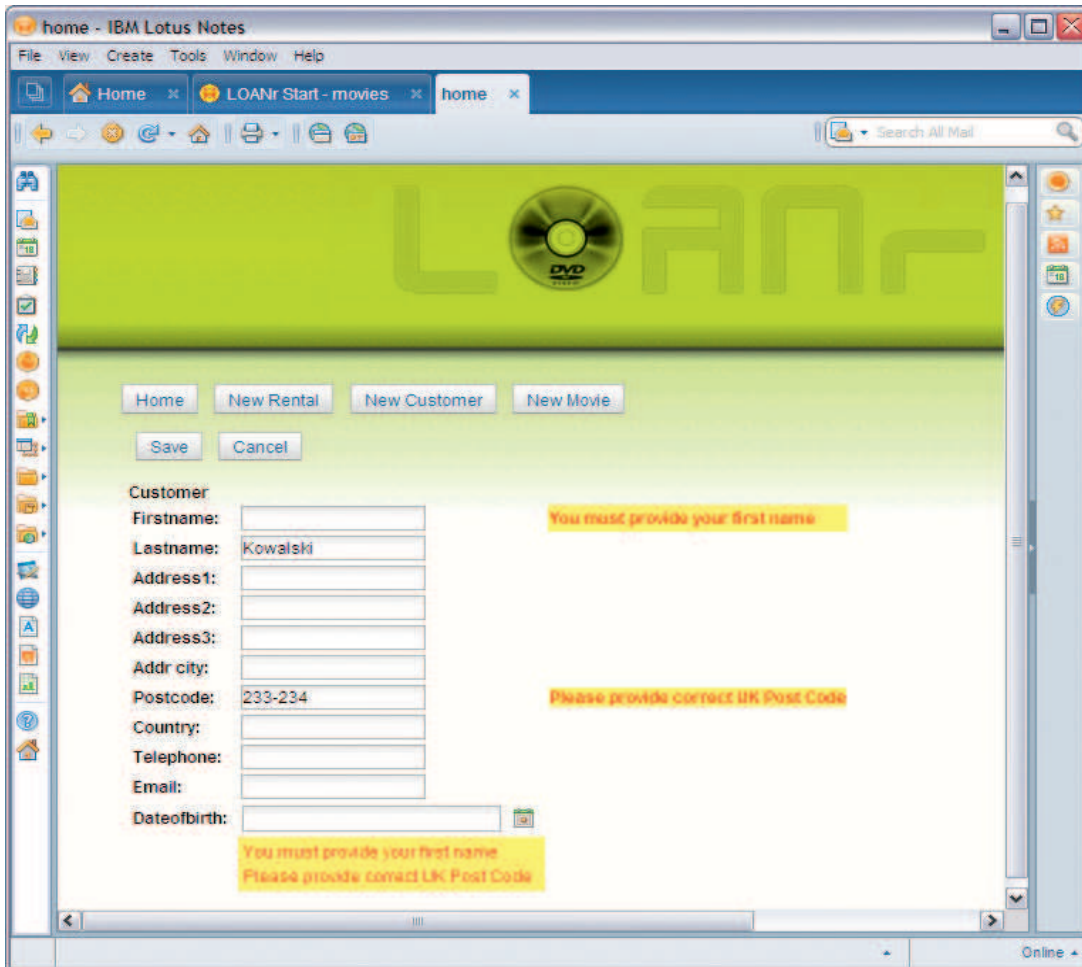


Rysunek 5. Kontrolka
cc_customer

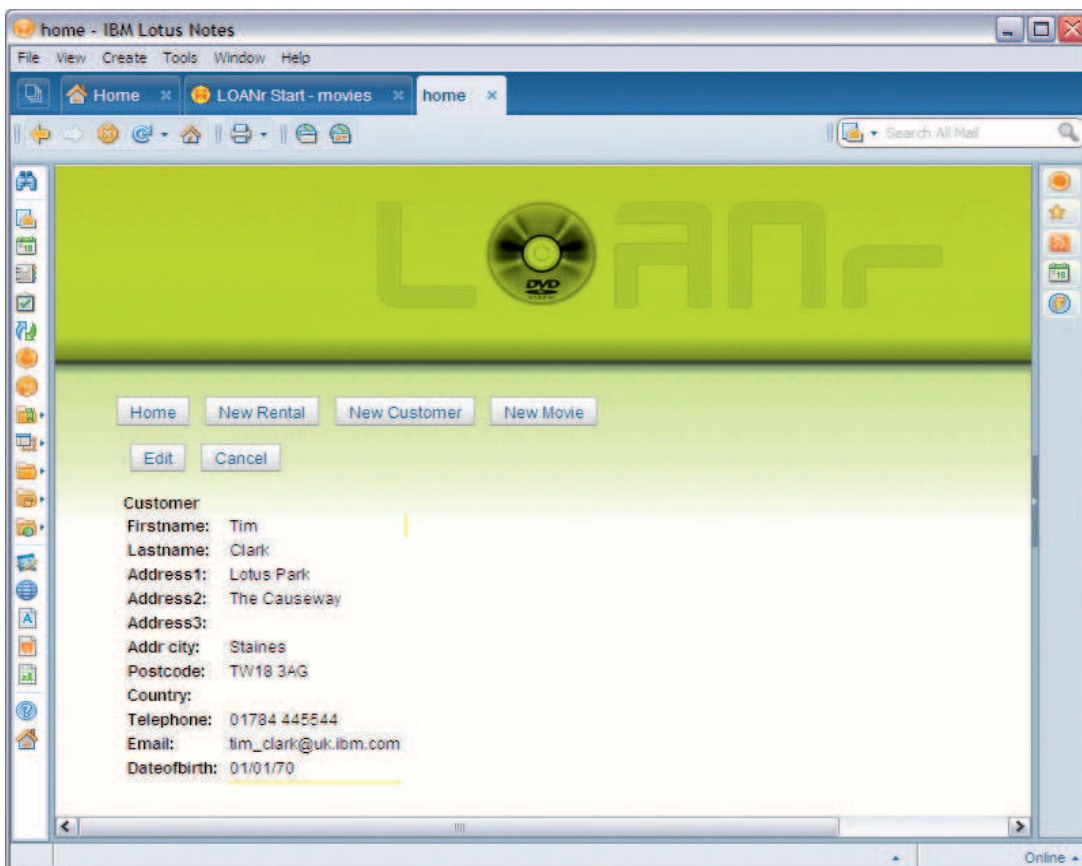
i ustawić jego formatowanie, na przykład na pogrubione lub przejść do kodu XML i wykorzystać znaczniki `<h>`. Kolejnym elementem będzie tabela, w której znajdują się wszystkie pola dokumentu. Będzie ona miała trzy kolumny – w pierwszej znajdzie się nazwa pola wprowadzona jako statyczny tekst, w następnej odpowiednie kontrolki w zależności od danych, jakie będą miały wyświetlać, w ostatniej natomiast komunikaty błędów dla pól wymaganych. Dokumenty mogą przechowywać bardzo szeroką gamę danych, w tym przypadku będą to dane osobowe klienta czyli m.in. imię, nazwisko, data urodzenia itp. Dla danych tekstowych zostanie wykorzystane *pole edycji (Edit Box)*, natomiast dla daty urodzenia *pole wyboru daty i godziny (Date Time Picker)*. W zakładce *Dane* każdej z tych kontrolki należy ustawić do jakiego pola z dokumentu się odwołuje. W zakładce *Sprawdzenie poprawności (Validators)* możemy ustawić to czy pole jest wymagane, ograniczenie znaków i komunikat błędu, bardziej złożone ograniczenia można wprowadzić we właściwości *Validators*, w zakładce *Wszystkie właściwości*. Obok pól wymaganych, w trzeciej kolumnie tabeli, należy dodać kontrolki *Komunikat o błędzie (Display Error)*, a w ich właściwościach wybrać z listy kontrolki, dla których będą one wyświetlały



Rysunek 6. Aplikacja
LOANr po dodaniu
XPages. Widok list



Rysunek 7. Aplikacja LOANr. Dokument klienta w czasie edycji



Rysunek 8. Aplikacja LOANr. Dokument klienta

Adrian Ciż jest studentem IV roku Automatyki i Robotyki na Wydziale Elektroniki Politechniki Wrocławskiej. W ramach praktyk ESI zajmował się oprogramowaniem IBM Lotus Notes Domino.

komunikaty oraz odpowiedni styl, by błąd był widoczny. Po stworzeniu pozostałych stron w analogiczny sposób nasza aplikacja jest gotowa, na koniec można jeszcze dopracować arkusz CSS, by całość odpowiednio się prezentowała i zacząć z niej korzystać. Efekt końcowy widoczny jest na rysunkach 6, 7 i 8.

Informacje z dystrybucji

IBM Storwize V7000 w programie „Wypożycz Demo”



ARROW ENTERPRISE COMPUTING SOLUTIONS

Program „Wypożycz Demo” przeznaczony jest dla Partnerów Arrow ECS. Pozwala na darmowe wypożyczenie sprzętu na potrzeby targów, prezentacji u klienta, szkoleń i testów nowych rozwiązań.

Specyfikacja techniczna IBM Storwize V7000:

W Arrow ECS dostępna jest macierz demo Storwize w następującej konfiguracji:

- IBM Storwize V7000
- 10 dysków 450 GB 2.5 in. 10k HDD
- 2 dyski 300 GB 2.5 in. SSD
- IBM Storwize V7000 External Virtualization Software V6.1
- IBM Storwize V7000 Remote Mirroring Software V6.1
- IBM Storwize V7000 Software V6.1



Kontakt: **Adam Zakrzewski**

IBM System Storage BDM, tel.: +48 22 488 70 11, e-mail: Adam.Zakrzewski@arrowecs.pl

Mariusz Bożyk

FTSS IBM System Storage, tel.: +48 12 616 43 93, e-mail: Mariusz.Bozyk@arrowecs.pl

Szczegóły: www.arrowecs.pl/WWW/News.nsf/id/Wypozycz_DEMO