

PAWWW: Sprawozdanie projektu II

Serwer reklam

Autor: Pura Dawid

Grupa:PS3

Założenia projektu:

Serwer reklam jest aplikacją webową utworzoną w technologii webowej specyfikacji JEE. Ma on spełniać następujące cechy systemu:

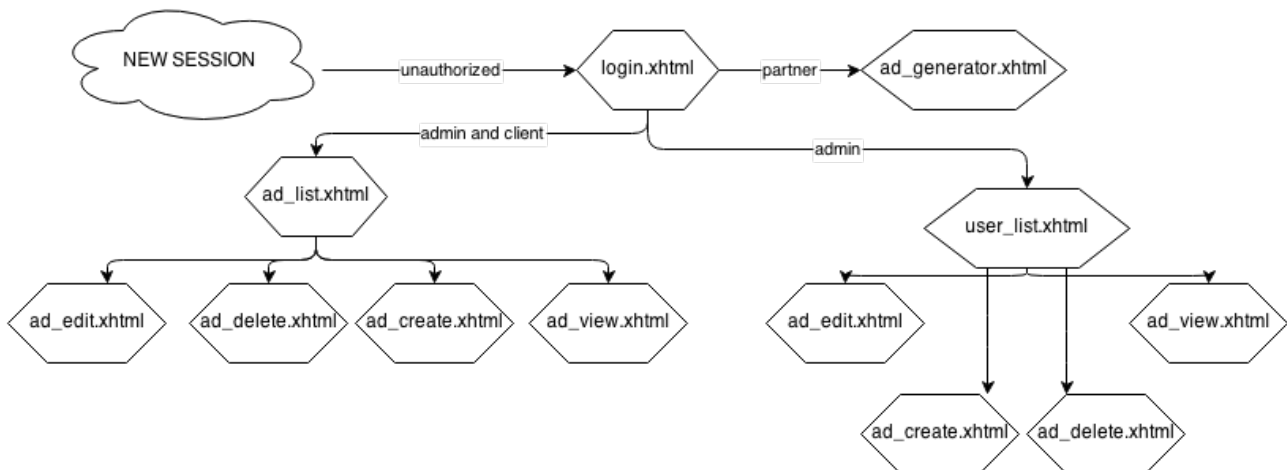
- **Możliwość wgrywania reklam na serwer.** Udostępnianie fragmentu kodu HTML i/lub JavaScript do pobierania i wyświetlania losowych reklam z serwera
- **Baza reklamodawców.** Możliwość rejestracji i logowania się. Reklamodawca może dodawać, usuwać i edytować swoje reklamy
- **Baza partnerów.** Możliwość rejestracji i logowania się. Partner może pobierać fragmenty kodu do umieszczenia na swojej stronie
- **Baza administratorów.** Możliwość logowania się. Administrator może zarządzać wszystkimi zasobami w serwisie
- **Podział reklam według co najmniej dwóch kryteriów**, np. według medium (tekst, obrazek, animacja Flash) i wielkości. Reklamodawca przy wprowadzaniu reklamy podaje jej typ, np. obrazek duży. Partner, podczas uzyskiwania kodu na stronę, ma możliwość wyboru jaki typ reklam ma być wyświetlany
- **Podział reklam względem branż.** Reklamodawca wybiera listę branż, do których kierowana jest dana reklama. Partner podczas definicji strony wybiera dla niej kilka branż. Wyświetlane na stronie reklamy pochodzą tylko z branż podanych przez partnera. Możliwość zarządzania branżami przez administratorów
- **Zhierarchizowany system branż.** System w postaci drzewa, którego korzeniem są "Wszystkie" branże. Uwzględnienie zależności w hierarchii przy wyświetlaniu reklam, np. jeżeli reklamodawca wybrał budownictwo, to reklama powinna się pojawiać m.in. na stronach o budownictwie mieszkaniowym czy wykończeniu wnętrz
- **Licznik odsłon i kliknięć** w daną reklamę
- **Wybór skórek** dla interfejsu użytkownika

Struktura systemu

- **WEB-INF**
 - **HTML** - pliki JSP, które generują strony HTML
 - ad_generator.jsp – widok kreatora kodu reklamy (partner)
 - categories.jsp – widok kategorii (pasteable)
 - index.jsp – strona startowa (login/register) w jednym elemencie
 - login.jsp – widok logowania
 - logout.jsp – widok wylogowania (dropowanie sesji)
 - manage_ads.jsp – widok zarządzania reklamami (admin/client)
 - user_table.jsp – widok zarządzania użytkownikami (administrator)
 - register.jsp – widok rejestracji (pasteable)
 - category_manager.sjp – widok odpowiedzialny za zarządzanie kategoriami
 - **JSF** – pliku XHTML opisujące warstwę prezentacji aplikacji
 - ad_list.xhtml – widok listy dostępnych reklam
 - ad_create.xhtml – widok tworzenia reklamy
 - ad_delete.xhtml – widok usunięcia wybranej reklamy
 - ad_generator.xhtml – widok umożliwiający partnerowi wygenerowanie kodu reklamy
 - ad_view.xhtml – widok przeglądu wybranej reklamy
 - user_list.xhtml – widok listy dostępnych użytkowników
 - user_create.xhtml – widok tworzenia użytkownika
 - user_delete.xhtml – widok usuwania wybranego użytkownika
 - user_view.xhtml – widok przeglądu wybranego użytkownika

- login.xhtml – widok logowania użytkownika
- template.xhtml – szablon widoków
- XML - pliki JSP, które generują dokumenty XML/dane
 - ad_link.jsp – dokument generujący treść reklamy do wklejenia
 - generate_ad.jsp – document reklamy usługi
- jspf - pliki z szablonami HTML
 - footer.jsp – stopka dokumentu HTML
 - header.jsp – nagłówek dokumentu HTML
- <generic package>
 - AdService - servlet serwisu reklam (CRUD, przekierowania)
 - UserServicelet - servlet do zarządzania użytkownikami (CRUD)
 - RESTFilter - filtr do przechwytywania URI i parsowania ich na wywołania servletowe
 - ParseableServlet – servlet bazowy parsujący URI akcji
- pl.pb.edu.adserver.model
 - Ad.java – model JPA reklamy
 - User.java – model JPA użytkownika
 - Category.java – model JPA kategorii
 - UserAction.java – model dostępowy akcji użytkownika
- pl.pb.edu.adserver.model.controller
 - AdJpaController – kontroler JPA reklam
 - UserJpaController – kontroler JPA użytkownika
 - CategoryJpaController – kontroler JPA kategorii
- pl.pb.edu.adserver.model.utils
 - CategoryBuilder – wzorzec builder i strategii do przeglądania/generowania drzew kategorii.

Widoki JSF:



Zasada działania serwera reklam

Akcje CRUD:

- Użytkownik
 - PUT /user/id/{id} – uaktualnia dane użytkownika (admin)
 - POST /user – tworzy użytkownika (admin)
 - DELETE /user/id/{id} – usuwa użytkownika (admin)

- GET /user – tworzy dane i wysyła aplikacje dostępową w HTML5 (admin)
- Reklama
 - PUT /ad/id/\${id} – uaktualnia reklamy (client/admin)
 - POST /ad - tworzy reklamę (client/admin)
 - GET /ad – wyświetla dane i przysyła aplikacje dostępową w HTML5 (admin/partner/client)
 - DELETE /ad/id/\${id} – usuwa reklamę (admin/client)
 - GET /ad?action=getAd – pobiera reklamę z danej kategorii (anonymous)
 - GET /ad?action=redirect – przekierowuje do danej zawartości reklamy (anonymous)

Jak wygląda przetwarzanie reklam?

1. Użytkownik o uprawnieniach klienta tworzy reklamę
2. Użytkownik o uprawnieniach partnera pobiera kod z kreatora i wkleja na swoją stronę
3. Kod wyświetlany wysyła zapytanie do serwisu – GET /ad?action=getAd – system losuje i przysyła reklamę z powrotem do aplikacji która zostaje wyświetlona. Do reklamy zostaje dodane +1 liczba wyświetleń.
4. Po naciśnięciu w reklamie aplikacja odwołuje się do systemu GET /ad?action=redirect który wysyła redirect do strony (poprzez nagłówek HTTP) i dodaje do reklamy liczbę kliknięć.

Jak wygląda dodawanie reklamy?

1. Klient autoryzuje się jako klient :)
2. Na stronie „Manage ads” klient edytuje okno formularza „Add another ads” - wprowadza nazwę, referer link etc i klika dodaj.
3. Od tego momentu reklama jest dostępna w puli reklam z danej kategorii.

Sposób rozwiązywania hierarchii kategorii.

Kategorie jako reprezentacja danych w systemie to oddzielny obiekt trwały zgodny ze specyfikacją JPA. Oprócz nazwy posiada on obiekty dzieci w relacji leniwej do reprezentacji drzewa. Przeszukiwanie (zaznaczanie) drzewa odbywa się przez wskazanie węzła o danej nazwie i przeszukiwanie drzewa w głąb i realizacja funkcjonalności poprzez wdrożone wzorce strategii (budowanie drzewa kategorii i zaznaczanie dzieci).

