

AdvertisementScreen - Dokumentacja

Autorzy

- Andrzej Dybowski
- Stanisław Fiuta
- Damian Wiatrzyk
- Marcin Siwiński

Opis projektu

Projekt powstał w ramach prac na przedmiot Podstawy Teleinformatyki.

Cel projektu

Celem powyższego projektu jest stworzenie systemu do wyświetlania reklam oraz wiadomości na platformie Raspberry PI. Projekt ten został napisany w Javie (część odpowiedzialna za wyświetlanie reklam) oraz w PHP (część odpowiedzialna za web interface). Projekt, jak już wspomniano, dzieli się na 2 części:

- web interface,
- aplikacja wyświetlająca reklamy i wiadomości.

Dlaczego wybrano taki temat

Temat tego projektu pozwala on rozwinąć wiedzę autorów na temat łączenia dwóch odrębnych aplikacji w jeden system. Temat ten również zawiera się w technologiach, w które są już znane członkom zespołu, więc nie istniał problem jej poznawania oraz poświęcania czasu na naukę.

Technologie te są uznawane w firmach jako podstawę, którą musi znać każdy programista. Dodatkowo ugruntował on wiedzę autorów z poniższych technologii oraz rozwinął umiejętność łączenia dwóch odrębnych aplikacji w cały system.

Platforma Raspberry PI umożliwia stworzenie rozwiązań, które nie potrzebują zbyt dużo mocy obliczeniowej, a dzięki temu nie pobierają zbyt dużo prądu.

Wyświetlanie reklam z paskiem wiadomości jest idealnym przykładem takiego systemu. Dzięki temu musimy zapewnić tylko dostęp do sieci internet (np. przez Wi-Fi) oraz ekran, a zarządzanie sprawować zdalnie z przeglądarki internetowej.

Dalszy rozwój projektu

Aby rozwinąć dalej powyższy projekt można wydzielić część administracyjną z bazą danych na osobną maszynę (np. zdalny serwer), i zostawić samą aplikację na Raspberry PI. Daje to możliwość rozwoju platformy z jednego ekranu, na wiele urządzeń sterowanych zdalnie. Sieć takich wyświetlaczy reklam można wykorzystać na uczelniach, sklepach, galeriach handlowych - gdzie konfigurowanie osobno tych urządzeń byłoby bardzo uciążliwe lub niemożliwe albo zbyt kosztowne dla klienta rozwiązania.

Kolejnym rozwinięciem systemu mogłoby być wprowadzenie logowania użytkowników oraz wprowadzenie ACL (Access Control List).

Podział prac

- aplikacja wyświetlająca reklamy i wiadomości:
 - przygotowanie widoków oraz wyświetlania reklam i wiadomości: Stanisław Fiuta
 - przygotowanie obsługi plików oraz komunikacji z webinterface: Damian Wiatrzyk
- web interface:
 - przygotowanie obsług akcji użytkownika: Andrzej Dybowski
 - przygotowanie komunikacji z aplikacją wyświetlającą reklamy: Marcin Siwiński

Funkcjonalności

System musi zapewniać pełną kontrolę nad reklamami i wiadomościami, od strony administratora, oraz prezentację reklam i wiadomości na ekranie.

Wyświetlanie reklam

Reklamy przechowywane w bazie danych wyświetlają się cyklicznie na ekranie bez udziału użytkownika; cykliczność zapewnia konfiguracja, ustawiana przez użytkownika w panelu administracyjnym; każda reklama ma określony czas na wyświetlenie; można wyróżnić kilka typów reklam do wyświetlania: obraz, film, strona WWW, animacja; wyświetlanie reklam ma się odbywać na pełnym ekranie monitora.

Wyświetlanie wiadomości

Wiadomości przechowywane w bazie danych wyświetlają się cyklicznie na pasku, u dołu ekranu; kolejność i czas wyświetlania się wiadomości są zapisane w konfiguracji; konfiguracja ustawiana jest w panelu administracyjnym.

Dodanie reklamy

Funkcjonalność zapewniająca dodanie reklamy każdego typu (film, animacja, obraz, strona WWW) do bazy danych, dodanie źródła samej reklamy (plik wysyłany na serwer albo adres URL do strony/filmu/obrazu umieszczonego w Internecie) oraz ustawienia kolejności jej wyświetlania lub czasu przez jaki będzie widoczna na ekranie.

Dodanie wiadomości

Funkcjonalność zapewniająca dodanie wiadomości do bazy danych oraz jej konfiguracja - czas, przez jaki jest wyświetlana na ekranie, oraz kolejność w której wiadomości do wyświetlenia.

Edycja reklamy

Funkcjonalność zapewniająca edycję istniejącej już reklamy w bazie danych, tj. typ reklamy, źródło, czas wyświetlania oraz kolejność w kolejce.

Edycja wiadomości

Funkcjonalność zapewniająca edycję wiadomości w bazie danych oraz jej parametry, tj. treść i kolejność w wyświetlaniu na pasku wiadomości.

Usunięcie reklamy

Funkcjonalność zapewniająca usunięcie istniejącej reklamy z bazy danych oraz jej źródeł (tylko w przypadku, gdy źródło jest przechowywane lokalnie na serwerze).

Usunięcie wiadomości

Funkcjonalność umożliwiająca usunięcie istniejącej już wiadomości z bazy danych.

Podgląd aktualnego stanu wyświetlanego na ekranie

Funkcjonalność zapewniająca podgląd jakie aktualnie są wyświetlane reklamy oraz wiadomości.

Wybrane technologie

Java: JavaFX, Gson

Wybranie języka Java daje możliwość uruchomienia aplikacji nie tylko na Raspberry PI ale też na innych urządzeniach mających zainstalowaną maszynę wirtualną Javy. Kolejnym czynnikiem, który skłonił do wyboru tego języka jest dostępność bibliotek oraz ich aktualność (a więc też i utrzymanie bezpieczeństwa samego systemu). `JavaFX` daje dostęp do szerokiej gamy klas ułatwiającej wyświetlanie obrazów, wideo czy stron www. Java zapewnia też natywne

wsparcie wątków. W połączeniu z `JavaFX` daje możliwość współbieżnego zarządzania kilkoma elementami, np: czytanie bazy danych, wyświetlanie newsów i reklam.

PHP

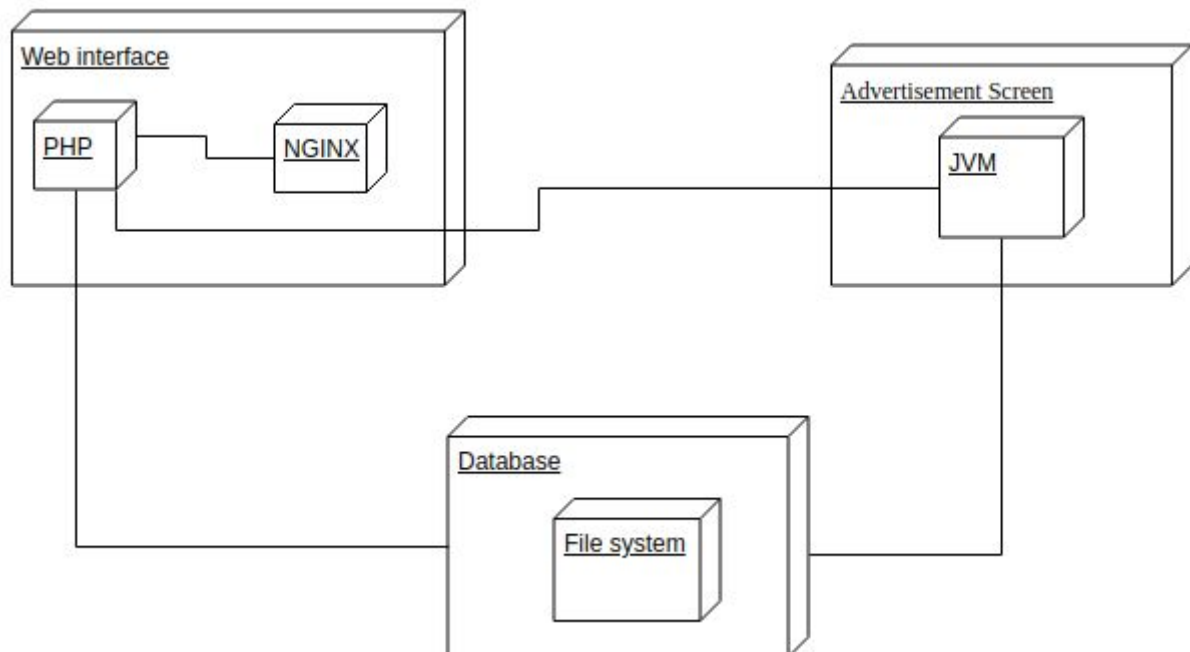
Ten język jest bardzo dobrze opisany oraz posiada dużą bazę frameworków - które są cyklicznie rozwijane i naprawiane z błędów. Interpreter tego języka łatwo instaluje się na systemach wchodzących w skład rodziny Linuxa (w popularnych dystrybucjach jest on dostępny jako paczka w package manager - więc nie należy kompilować go ze źródeł oraz łatwo podnieść do wyższej wersji).

Zastosowanie PHP z NGINX daje możliwość zniwelowania obciążenia do minimum, bez pisania od nowa obsługi protokołów HTTP i wyświetlania widoków w HTML.

Nginx

Jest to jeden z najpopularniejszych serwerów WWW dostępnych na platformę Raspberry PI (Raspbian). Dzięki temu, że do obsługi połączeń przychodzących wykorzystuje wątki zamiast procesów (w porównaniu np z Apache), daje możliwość obsługi większej ilości zapytań HTTP wykorzystując przy tym mniej zasobów serwera. Łatwo też się skaluje. Jest też dostępny jako paczka w Raspbianie - więc nie trzeba kompilować tego programu z kodu źródłowego.

Architektura



System składa się z 2 głównych modułów oraz bazy danych w postaci łącznika. Dane na temat aktualnego stanu przesyłane są za pomocą plików. `Web interface`, czyli panel administracyjny, oraz `Advertisement Screen` są oddzielnymi aplikacjami. Dzięki temu można w późniejszym czasie rozdzielić na osobne maszyny.

Aplikacja wyświetlająca reklamy

Aplikacja wyświetlająca reklamy składa się na 3 główne części, które odpowiadają uruchomionym wątkom w aplikacji:

- `ContentUpdater` - odpowiedzialną za ładowanie konfiguracji z bazy danych oraz wczytywaniem danych,
- `AdvertisementThread` - odpowiedzialną za wyświetlanie reklam,
- `NewsThread` - odpowiedzialną za wyświetlanie wiadomości.

Dzięki takiej architekturze można oddzielić warstwę ładującą dane od samego wyświetlania.

Jest ona uruchamiana wraz ze startem systemu operacyjnego, dzięki czemu restart całego urządzenia nie wymusza na użytkowniku uruchamiania jej. Aplikacja stoi na JVM, dlatego że jest napisana w języku Java.

Komunikacja między panelem administracyjnym, samą aplikacją odbywa się za pomocą plików, gdzie są trzymane dane na temat reklam i wiadomości oraz screenshoty z obecnego stanu aplikacji wyświetlającej reklamy.

Reklamy oraz wiadomości są trzymane w kolejkach, dzięki którym kontroler może iterować po danych bez znajomości ilości przechowywanych danych w danej kolejce. Klasy odpowiedzialne za przechowywanie danych to `Advertisement` - dane o samej reklamie, `AdvertisementType` - klasa enum reprezentująca typ reklamy, `News` - dane samej wiadomości z bazy danych.

ContentUpdater

ContentUpdater
<ul style="list-style-type: none">- lastModifiedNewsFile: Long- lastModifiedAdvertisementFile: Long- root: Parent
<ul style="list-style-type: none">+ run(): void- initLastModified(): void- updateNews(): void- updateAdvertisement(): void- updateScreen(root Parent): void

Posiada niezbędne metody do ładowania danych z bazy danych. Korzysta on z Biblioteki `Gson`, zapewniającą ładowanie danych konfiguracyjnych. Zawiera w sobie również metody ładujące dane samej reklamy (tj. plik obrazu/filmu lub stronę) ze źródeł lokalnych lub zdalnych. Po załadowaniu konfiguracji oraz danych, dane są ładowane w odpowiedniej kolejności do kolejki `AdvertisementThread` i `NewsThread`.

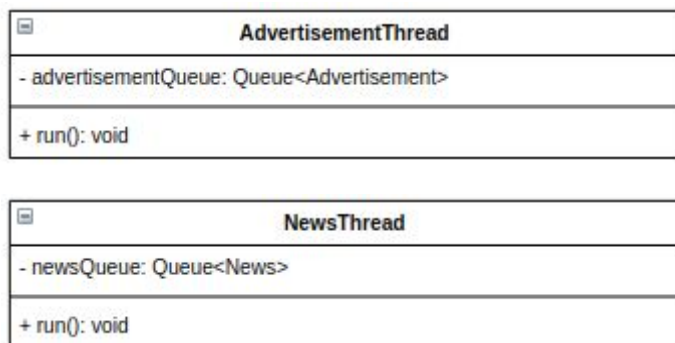
Sczytywanie danych z plików są odpowiedzialne klasy pomocnicze:

`AdvertisementQueueCreator` oraz `NewsQueueCreator`, które wykorzystują odpowiednio `AdvertisementReader` i `NewsReader` - które odpowiadają za implementację czytania plików.

Aby zapewnić aktualność danych, sprawdzany jest czas ostatniej modyfikacji plików przechowujących dane.

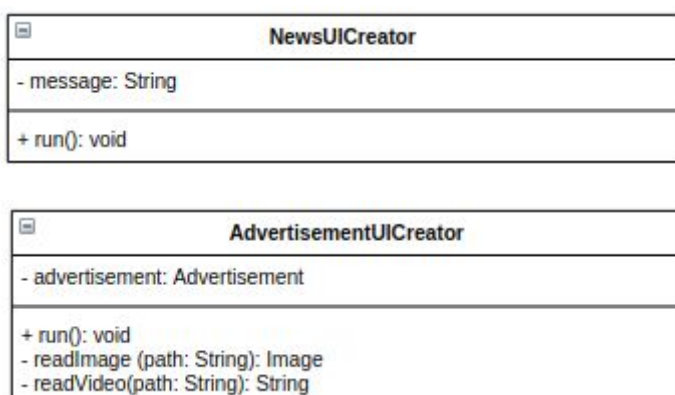
Przy uruchamianiu aplikacji sprawdzany jest ostatni czas modyfikacji, a następnie co 1 sekundę sprawdzane jest czy nie zostały dokonane zmiany i jeśli tak, to przeładowywana jest konfiguracja oraz powiadamiane są wątki odpowiedzialne za odświeżanie widoków.

AdvertisementThread i NewsThread



Są to wątki odpowiedzialne za odświeżanie widoku, odpowiednio, reklam i wiadomości. Przechowują kolejki z wiadomościami i reklamami. Klasy odpowiedzialne za widoki to: `AdvertisementUICreator` i `NewsUICreator`.

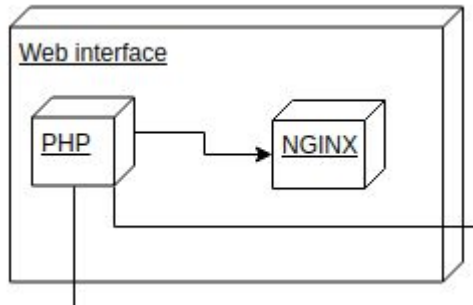
Widoki: AdvertisementUICreator i NewsUICreator



`AdvertisementUICreator` w zależności od typu danej reklamy wybiera odpowiednie klasy z biblioteki JavaFX (obsługujące wideo, strony www oraz obrazy).

NewsUICreator wyświetla wiadomość na pasku.

Panel administracyjny



Jest odpowiedzialny za implementację funkcjonalności dodawania/usuwania oraz edytowania reklam i wiadomości.

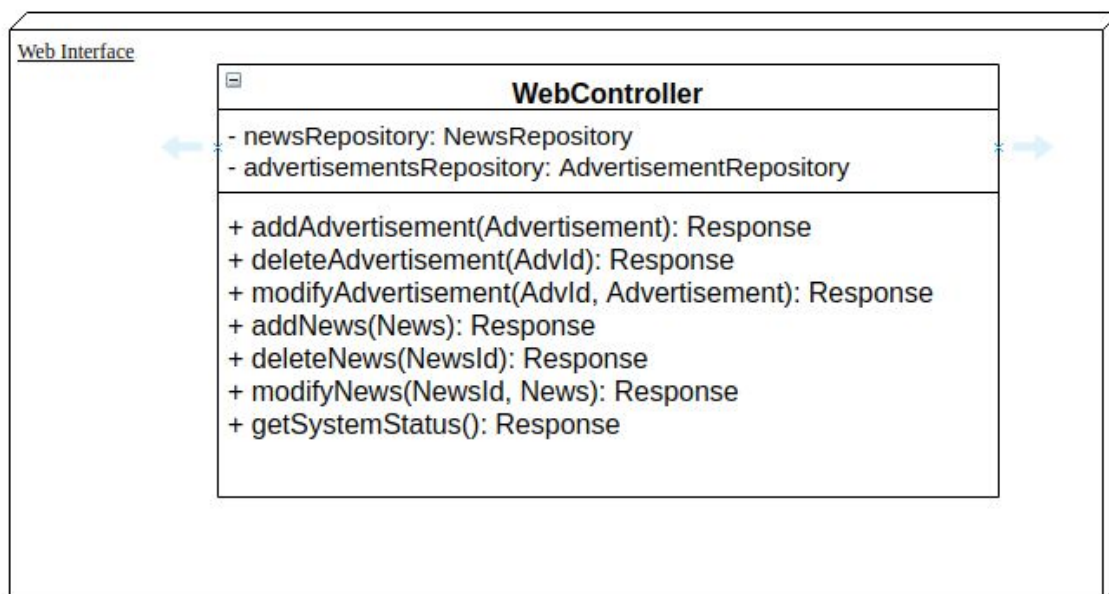
Aplikacja jest zbudowana z dwóch elementów: serwera www - NGINX oraz aplikacji w języku PHP.

Serwer WWW

Jest odpowiedzialny za przyjmowanie ruchu HTTP oraz przekazywanie go do interpretera języka PHP.

Aplikacja jest uruchamiana wraz z uruchomieniem systemu operacyjnego.

Aplikacja w PHP

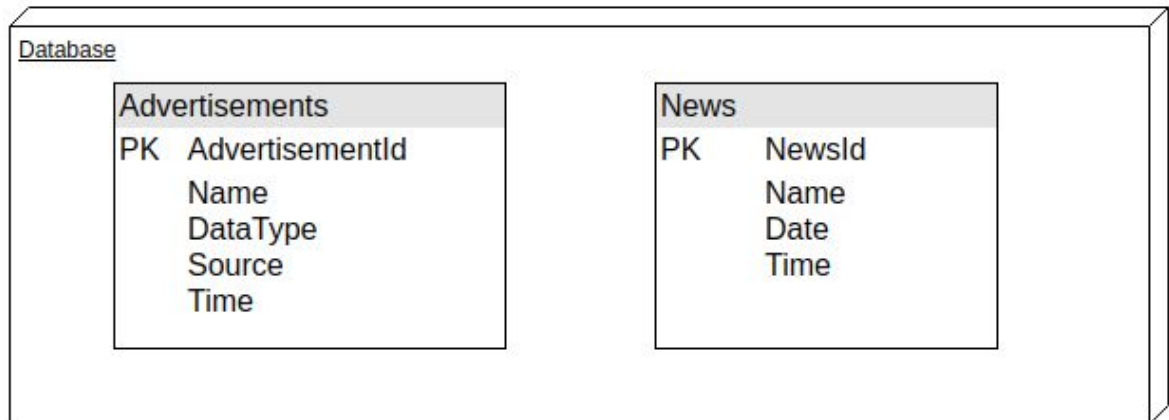


Dane są zapisywane w bazie danych. Panel został zaprojektowany w wzorcu architektonicznym MVC dzięki, któremu komunikacja z bazą danych oraz widoki są oddzielone od siebie. Daje to przejrzystość kodu oraz łatwość w implementacji. W aplikacji występują 2 modele: `Advertisement` i `News`. `Advertisement` jest modelem, który przechowuje dane na temat danej reklamy. `News` przechowuje dane na temat danej wiadomości.

Aplikacja posiada jeden kontroler, który zapewnia akcje: `addAdvertisement` - akcja dodania nowej reklamy do bazy danych, `deleteAdvertisement` - akcja usunięcia reklamy z bazy danych, `modifyAdvertisement` - akcja edycji reklamy w bazie danych, `addNews` - akcja dodania nowej wiadomości do bazy danych, `deleteNews` - akcja usuwania wiadomości z bazy danych, `modifyNews` - akcja edycji wiadomości z bazy danych, `getSystemStatus` - akcja pobrania danych o stanie całego systemu.

Aplikacja jest napisana w języku PHP, przez co skrypty są uruchamiane po wystąpieniu zapytania HTTP.

Baza danych



Ze względu na małą ilość tabel (2 tabele: reklamy oraz wiadomości) zdecydowano, aby bazą danych były pliki JSON. Lista plików:

- `advertisement.json` - plik, w którym przechowywana jest konfiguracja reklam;
- `news.json` - plik, w którym przechowywana jest konfiguracja wiadomości.

Lokalne pliki

Na potrzeby aplikacji pliki obrazów oraz filmów są przechowywane lokalnie odpowiednio w katalogach `video` oraz `img`. Dzięki czemu skrócono czas odczytu i zapisu danych.

Problemy na które się natknęliśmy

Początkowo planowano napisać aplikację w samej Javie z wykorzystaniem MySQL. Jednak ze względów na ograniczenia platformy Raspberry PI porzucono ten pomysł na rzecz bardziej wydajniejszych i szybszych rozwiązań - Panel Administracyjny został napisany w PHP, jako serwer WWW wykorzystano Nginx, bazę danych zastąpiono plikami JSON.

Instrukcja użytkowania aplikacji

Ze względu na to, że użytkownik ma wszystkie akcje w aplikacji są wykonywane tylko w Panelu Administratora, instrukcja będzie dotyczyć tylko interfejsu użytkownika.

Panel Administracyjny

- Dodawanie reklamy:
 - a. Należy kliknąć w przycisk Dodaj reklamę
 - b. Uzupełnić odpowiednio: - typ reklamy - jej źródło - w przypadku: ładowania z Internetu adres URI / pliku: załadować plik na serwer - czas trwania - pozycja w kolejce
 - c. Zatwierdzić operację przyciskiem Dodaj
- Usuwanie reklamy:
 - a. Wybrać z listy interesującą nas reklamę
 - b. Kliknąć w przycisk Usuń
 - c. Potwierdzić operację przyciskiem Ok
- Edycja reklamy
 - a. Wybrać interesującą nas reklamę
 - b. Kliknąć w przycisk Edytuj
 - c. Dokonać zmian pól wypisanych w Dodawanie reklamy ii.
 - d. Zatwierdzić zmiany przyciskiem Zapisz
- Dodawanie wiadomości
 - a. Należy kliknąć w przycisk Dodaj wiadomość
 - b. Uzupełnić odpowiednio:
 - treść wiadomości
 - czas trwania
 - pozycja w kolejce
 - c. Zatwierdzić operację przyciskiem Dodaj
- Usuwanie wiadomości:
 - a. Wybrać z listy interesującą nas wiadomość

- b. Kliknąć w przycisk Usuń
 - c. Potwierdzić operację przyciskiem Ok
- Edycja wiadomości
 - a. Wybrać interesującą nas wiadomość
 - b. Kliknąć w przycisk Edytuj
 - c. Dokonać zmian pól wypisanych w Dodawanie wiadomości ii.
 - d. Zatwierdzić zmiany przyciskiem Zapisz