

Koncepcja wykonania systemu

Serwis internetowy Koła Studentów Informatyki Uniwersytetu
Wrocławskiego

Autorzy:

Kamil Markiewicz

Anna Szymanek

1. Scenariusze przypadków użycia

Scenariusz #1

Dodawanie zawartości na stronie głównej odbywa się z poziomu panelu administracyjnego.

Funkcjonalność: dodawanie oraz edycja treści na stronie głównej.

Dany jest zalogowany użytkownik z uprawnieniami do edycji strony głównej.

Kiedy doda zawartość na stronę główną z poziomu panelu administracyjnego,
wtedy zostanie ona wyświetlona na stronie głównej.

Scenariusz #2

Wyświetlanie katalogu galerii składającej się z galerii ze zdjęciami dostępne jest dla każdej osoby odwiedzającej stronę.

Funkcjonalność: publiczny katalog galerii

Dany jest niezalogowany użytkownik na stronie głównej.

Kiedy wybierze podstronę "Galeria" i naciśnie na nią,
wtedy zostanie wyświetlony widok katalogu galerii.

Scenariusz #3

Tworzenie wpisu w aktualnościach.

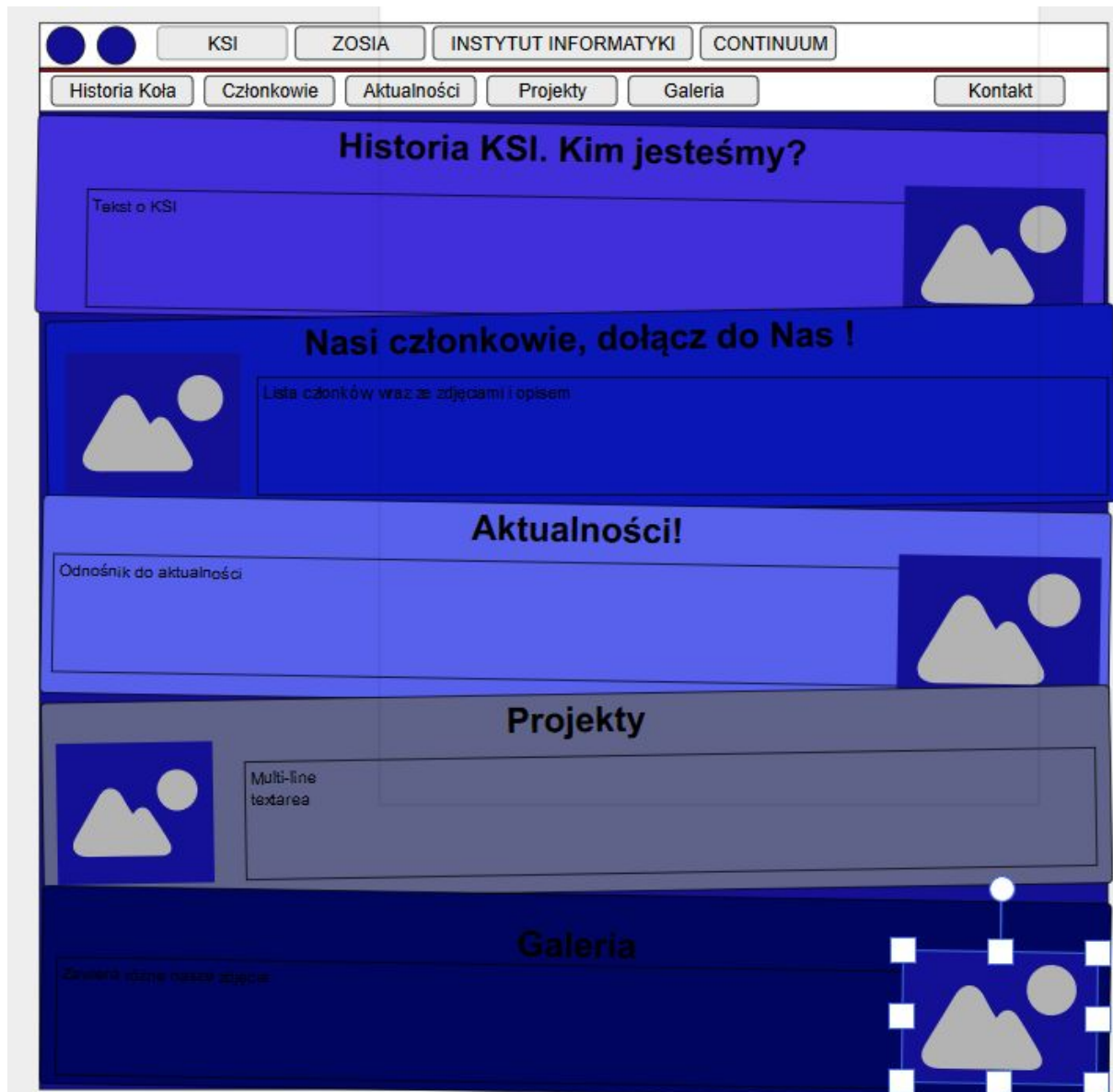
Funkcjonalność: aktualności

Dany jest zalogowany użytkownik z uprawnieniami do dodawania aktualności w panelu administracyjnym.

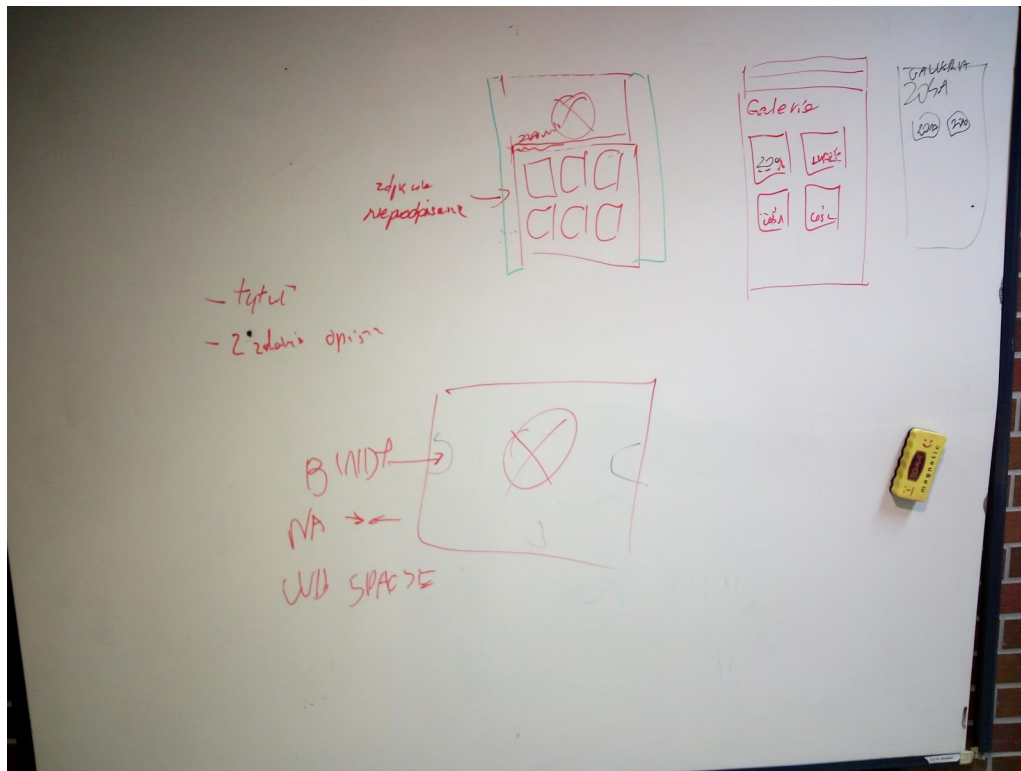
Kiedy doda treść aktualności z poziomu panelu administracyjnego,
wtedy zostanie ona wyświetlona na podstronie dostępnej w zakładce "Aktualności".

2. Projekt ekranów

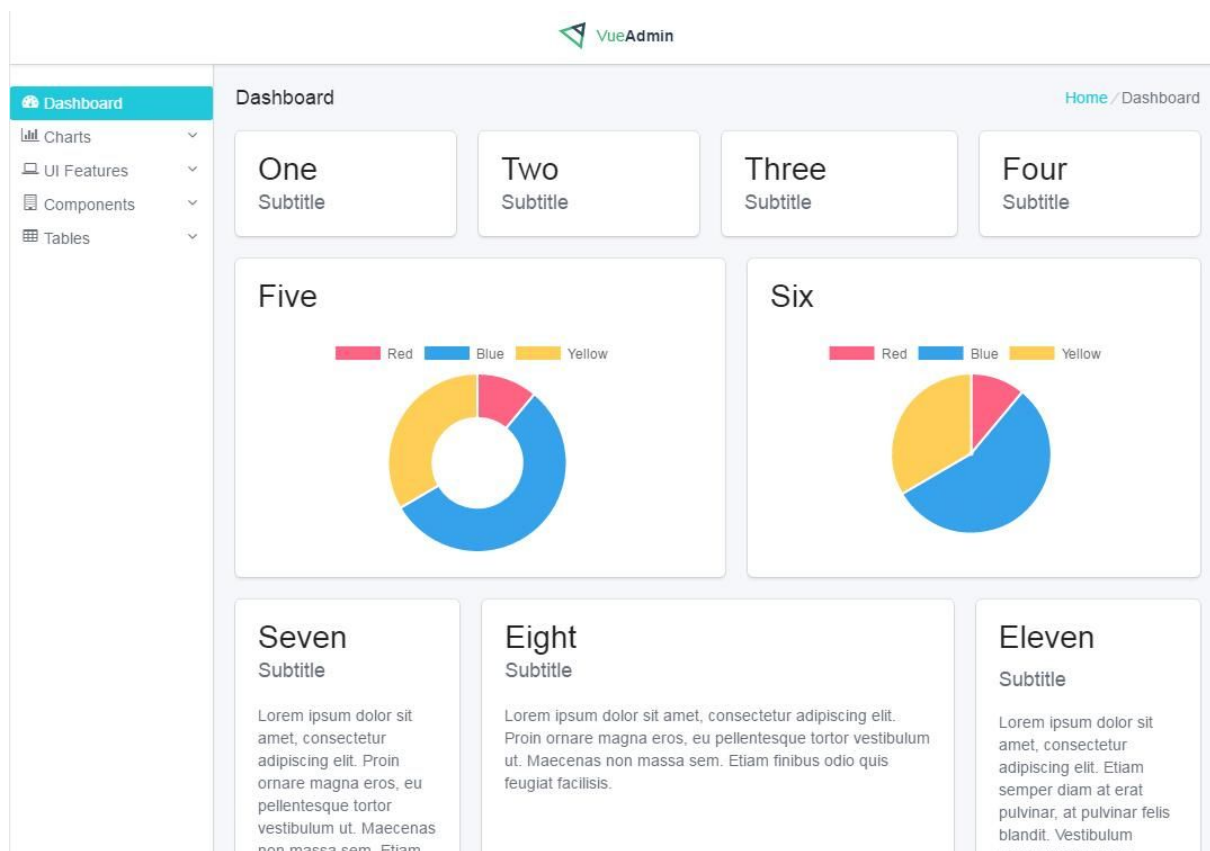
Ekran strony głównej



Ekran katalogu galerii



Ekran panelu administracyjnego



3. Projekt architektury







Serwis ma wykorzystywać serwer udostępniony przez KSI, na którym jest zainstalowany system operacyjny z jądrem Linux wspierany przez oprogramowanie Docker.

W pracy nad serwisem głównymi narzędziami programistycznymi są IDE IntelliJ IDEA firmy JetBrains w wersji Community, Visual Studio Code z wtyczkami ułatwiającymi pracę nad częścią kliencką, Yarn jako menadżer pakietów oraz przeglądarki Google Chrome i Mozilla Firefox.

System dzieli się na 3 główne części.

1. Część kliencka:

Ta część bazuje na możliwościach frameworka Vue.js. Językiem dynamicznym po stronie przeglądarki jest ECMAScript 6 lub wyższej. Nasz serwis docelowo będzie wspierał poniższe przeglądarki:

 Android	 Firefox	 Chrome	 IE	 iPhone	 Edge	 Safari
6.0  * ✓	56  7 ✓	62  7 ✓	9  7 ✓	10  10.12 ✓	15  10 ✓	8  10.10 ✓
			10  8 ✓			
			11  8.1 ✓			

Wykorzystujemy pakiet narzędziowy Bootstrap 4 z pakietem integracyjnym bootstrap-vue w celu implementacji technologii Responsive Design, pozwalającej osiągnąć satysfakcjonujący wygląd strony w różnych rozdzielczościach. Skrypt konfiguracyjny narzędzia Yarn polega na innym narzędziu, jakim jest Webpack - służący do łączenia modułów projektu. Zbudowane pliki stają się statyczne w sensie nie wymagają dalszej dynamicznej obróbki, dzięki czemu mogą być udostępniane na prostym serwerze treści statycznych.

2. Część serwerowa:

Serwer jest napisany w języku Kotlin pracującym na platformie wirtualnej JVM. Wystawia CRUD REST API w celu pośredniczenia między bazą danych, uwierzytelniania i zarządzania sesjami użytkowników poprzez mechanizm ciasteczek. Część ta działa niezależnie od plików części klienckiej, co pozwala na pełną izolację obu części.

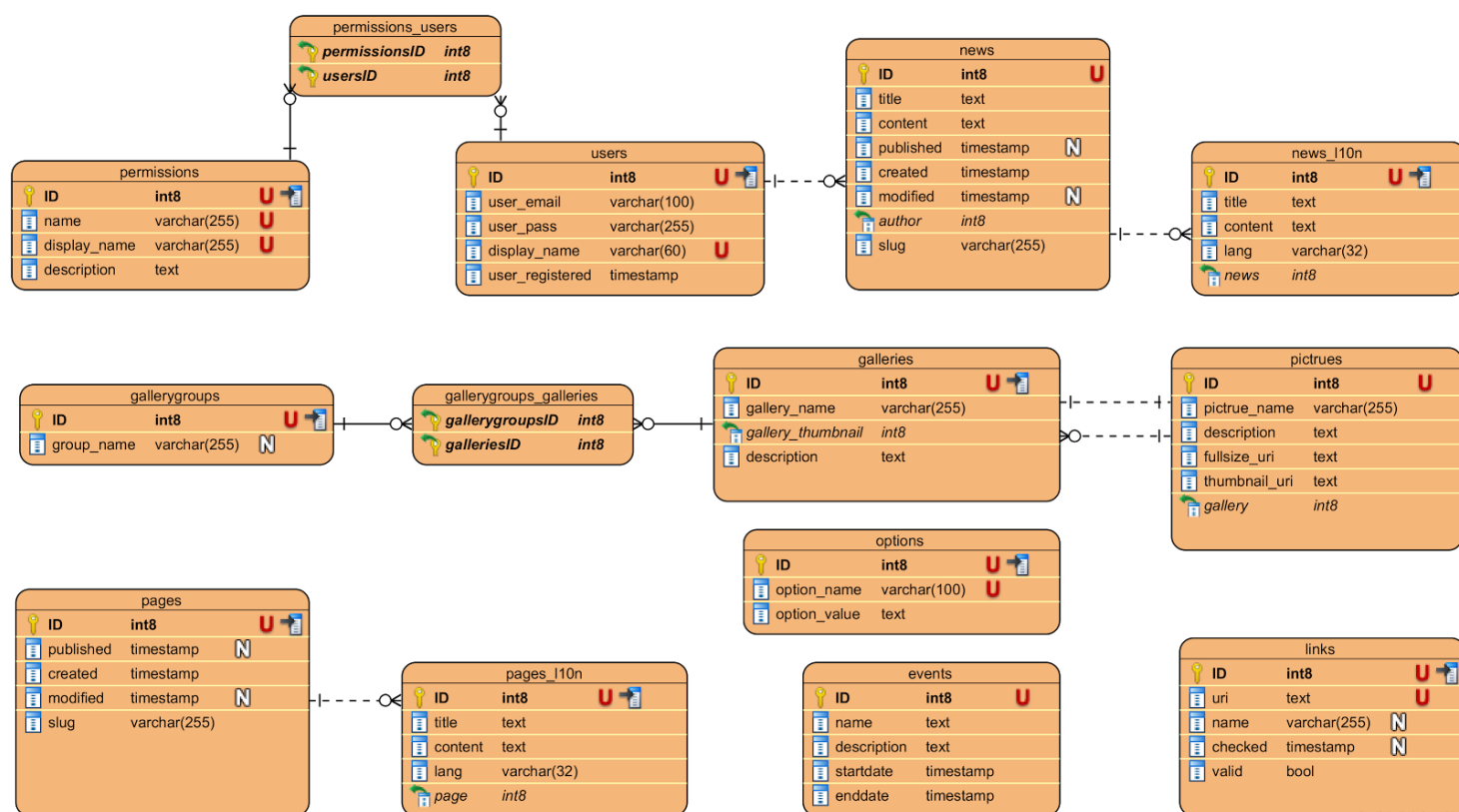
3. Baza danych:

Dedykowana zdokeryzowana instancja systemu bazodanowego PostgreSQL w wersji 10.

Część serwerowa wymaga na niej jednej schemy z użytkownikiem, z pełnym uprzywilejowaniem dla operacji SELECT, INSERT oraz UPDATE na danej schemie.

Dostępny jest drugi użytkownik z uprawnieniami pozwalającymi na sporządzanie kopii zapasowej. Na serwerze jest jeszcze trzeci użytkownik, z pełnym pakietem uprawnień, przeznaczony do celów administracyjnych. Jego dane dostępowe posiada tylko administrator i nie są używane do żadnego innego celu.

4. Diagram E-R bazy danych



5. Główne zasady kodowania

Część serwerowa podlega zasadom Kotlin Coding Conventions¹. Rozwój części klienckiej wymaga przestrzegania zasad dostępnych w Vue.js Style Guide².

Projekt jest tworzony w publicznym repozytorium Git hostowanym na GitHubie, wykorzystuje funkcjonalności platformy w zakresie zgłaszania i śledzenia zagadnień o nazwie “Issues” oraz mechanizm “Pull requests”. Każda zmiana wymaga utworzenia nowej gałęzi i stworzenia prośby o połączenie. Przed scaleniem z główną gałęzią zmiana musi poprawnie zakończyć proces budowania i testowania całości projektu przez zewnętrzne narzędzie ciągłej integracji Drone CI. Prace przebiegają w tygodniowych sprintach, gdzie po każdym następuje zebranie podsumowujące dany cykl pracy, prowadzące do ustalenia przyszłej pracy na bazie backlogu jak i określenie nowych zadań lub podział aktualnych na podzadania.

6. Identyfikacja i zasady zarządzania ryzykiem

Ustalona architektura wymusza podział projektu na logiczne części, jednak wiąże się z pewnymi kompromisami. Jeśli ostateczna postać serwisu, która w dużej mierze przerzuca narzut związany z obsługą żądań do serwera na przeglądarkę użytkownika okaże się

¹ <https://kotlinlang.org/docs/reference/coding-conventions.html>

² <https://vuejs.org/v2/style-guide/>

miejscami za mało wydajna to jesteśmy gotowi przenieść część pracy na serwer wykorzystując technologię server-side rendering. Szacujemy prawdopodobieństwo zajścia takiego zdarzenia na niskim poziomie ze względu na zakres projektu. W celu uniknięcia tego ryzyka możemy podjąć środki prewencyjne, uważać na obciążenie przeglądarek podczas pracy nad funkcjonalnościami oraz przygotować testy sprawnościowe.

Jeśli część serwerowa okaże się powodować zatory, jesteśmy gotowi położyć większy nacisk na optymalizację mechanizmu keshowania danych. Wiąże się to z usprawnieniem pracy bazy danych. Zmusi nas to do odroczenia w czasie prac nad funkcjonalnościami. Można temu zapobiec poprzez testy sprawnościowe. Prawdopodobieństwo jest umiarkowane, ponieważ pisząc funkcjonalności w dużej mierze będziemy polegać na domyślnych możliwościach keshowania włączonych w framework.

Ryzykujemy, że nie zdążymy dostarczyć wszystkich funkcjonalności w ustalonym terminie. Będziemy zmuszeni wtedy ograniczyć zakres projektu, przez co część funkcjonalności trafi do naszego klienta dopiero w kolejnej wersji serwisu.

7. Ocena zgodności z tablicą koncepcyjną i specyfikacją wymagań

Zostaliśmy nawiedzeni przez demona, który przedstawił się jako CZAS. Powiedział, że przekroczyliśmy czas na domknięcie części projektowej. Ekipa projektowa nie zraziła się jednak i systematycznie podejmuje pracę przybliżającą nas do wizji.

Wszyscy członkowie KSI korzystając z testowej wersji będą dostarczali nam informacji zwrotnej oraz sprawdzą zgodność.

Osiągnięte rezultaty zgadzają się z wizją projektu przedstawioną na tablicy koncepcyjnej. Udało zmieścić się w budżecie oraz zachować jakość. Reszta parametrów również jest zachowana.