

Inżynieria oprogramowania 2017-18
ćwiczenia/pracowania

II FAZA WYTWARZANIA
OPROGRAMOWANIA- FAZA
OPRACOWANIA

Walentyn Baliuk
Anton Lunyov
Igor Tryhub

1. Rozwiązanie znajduje się w dodatkowo dołączonym pliku .xlsx.

2. Przypadek #1:

Dany przypadek demonstruje działanie funkcji WYSIWYG redaktora do postów. Po zalogowaniu się użytkownik dostaje dostęp do przycisku "Create post". Po naciśnięciu na ten przycisk otwiera się okienko w którym znajduje się redaktor działający na zasadzie WYSIWYG, tzn to co jest widoczne w oknie redaktora będzie pokazywane jako wynik. Redaktor składa się z paneli instrumentów (np dodaj link, dodaj zdjęcie, etc) oraz pola, zawierającego zawartość postu. Użytkownik oprócz wpisywania tekstu ma możliwość dodawania elementów do postu metodą drag-and-drop (po dropie wyskakuje okienko, w którym można wpisać parametry elementu, np, link do zdjęcia). Użytkownik wpisuje jakiś tekst, ewentualnie dodaje parę elementów media i klika na przycisk "Submit". Po naciśnięciu post okazuje się w kolejce do moderacji.

Przypadek #2:

Dany przypadek demonstruje działanie funkcji nawigacji. Załóżmy, że użytkownik jest na stronie głównej. Na wierzchu strony znajduje się menu nawigacji, w którym są różne elementy menu, wśród nich element "Categories". Po najechaniu kursorem na ten element, w podmenu wyjeżdżają główne kategorie sklepu. Niektóre z nich również mogą się rozwijać w głąb drzewa kategorii. Użytkownik wybiera potrzebną kategorię klikając w nią.

Przypadek #3

Dany przypadek demonstruje działanie funkcji responsywnego dizajnu strony. Użytkownik otwiera dowolną stronę portalu w dowolnej przeglądarce. Początkowo przeglądarka jest rozwinięta na pełny ekran i strona ma ustawienie elementów zwykle dla dużego ekranu. Użytkownik zaczyna postępowo zmniejszać rozmiar okna przeglądarki. Podczas tego można postrzegać zmianę ustawienia elementów strony odpowiednio do rozmiaru okna w sposób optymalny.

3. Sprzęt - serwery w chmurze Google Cloud.

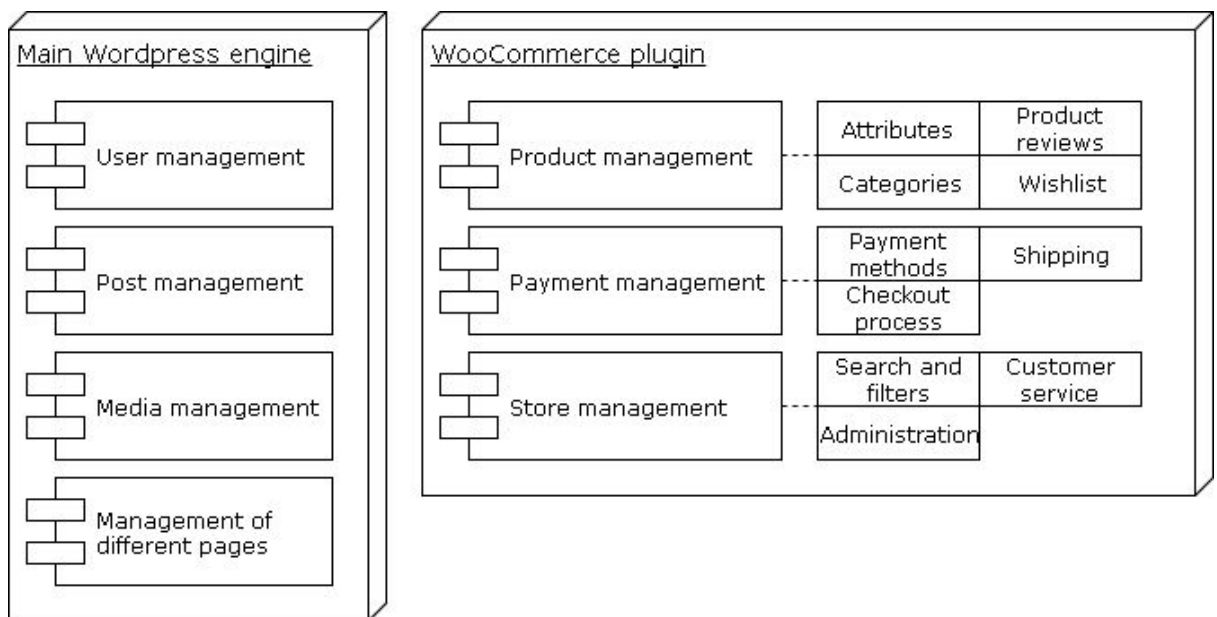
Oprogramowanie systemowe - system operacyjny Linux, dystrybucja Ubuntu 16.04.

Baza danych - MySQL (narzucona przez Wordpress).

Narzędzia programistyczne - Wordpress oraz plugin WooCommerce do tworzenia sklepów.

Oprogramowanie do automatycznego testowania - Cucumber, Selenium.

Struktura oprogramowania - bardzo trywialna przez to, że narzędzia załatwiają implementację wszystkich funkcji.



4. Nie musimy się martwić o schemacie bazy danych, ponieważ on jest wygenerowany automatycznie przez Wordpress.

5. Pliki tekstowe / źródłowe muszą przestrzegać kilku konwencji:

- Wszystkie pliki są kodowane w UTF-8.
- Należy zadbać o odgadywanie ustawień wcięcia i długości linii (80 lub 120 linii dla współczesnych wyświetlaczy).
- Ostatni wiersz musi zawierać koniec linii.
- W nazwach plików nie powinny występować znaki spoza zestawu ASCII, ponieważ prowadzą one do problemów wynikających z niespójności w systemach operacyjnych.

Inżynieria oprogramowania uczy nas, że poszczególne problemy należy dzielić na mniejsze zadania. Pozwala to zaoszczędzić kod i przewidzieć wszystkie możliwe zakończenia. Dzięki temu już na etapie planowania możemy zauważyć możliwe problemy i je rozwiązać. Wskutek takiego podejścia, pozornie trudne i wielkie zadania stają się proste i przyjemne w wykonaniu.

Pisanie kodu etapami i sprawdzanie na każdym czy działa jest dobrym nawykiem. W każdym momencie pisania kodu jest wiadomo, która część kodu sprawia pojawianie się błędów. Oszczędza się dzięki temu czas i nerwy.

Bardzo przydatne w celu czytelności i skalowalności aplikacji webowych należy oddzielać osobne języki webowe. Czyli PHP jest w plikach .php, funkcje JS jest załączone na osobnych plikach .js a jakiegokolwiek stylowania niech będą w plikach .CSS. Plik HTMLa bez CSSa powinien być całkowicie rozsypany, ogołocony, żadnych stylów. Takie oddzielanie plików na pewno zaowocuje w przyszłości. Projekt stanie się czytelniejszy, łatwiejsze w modyfikacji czy rozbudowie.

Generalnie, należy pragnąć programować w taki sposób aby kod sam w sobie był bardzo czytelny i nie wymagał komentarzy. Tylko zwykle nikt nie ma czasu przeanalizować całego pliku, żeby domyśleć się jak działa program. Dlatego należy stosować komentarze, możliwie krótkie i dokładne jednocześnie. Będzie to pomocne zarówno w przypadkach, gdy projekt pójdzie do szuflady i autor wróci do niego za rok, zarówno jak w sytuacji, gdy ktoś inny musi przejąć opiekę nad kodem.

Reguły nazywania zmiennych też są dość proste:

- Stałe - wszystkie stałe (czyli treść, która nigdy nie powinna się zmieniać, nawet jeśli stałe nie istnieją w JS) powinny być nazwane wielkimi literami, a słowa muszą być oddzielone podkreślnikami, np. `THIS_IS_A_CONSTANT`.
- Obiekty - każdy konstruktor (czyli coś, co musisz wywołać za pomocą nowego) musi być nazwany w przypadku wielbłąda (camelCase) i zaczyna się od dużej litery, np. `MyObjectName`.
- Zmienne i cała reszta - wszystko inne ma postać wielbłąda (camelCase) i nie może zaczynać się od wielkiej litery, np. `myVariableName`.

Zaleca się napisanie nazw zmiennych, funkcji i komentarzy w języku angielskim. Jest on językiem programistycznym. Nie warto zaniechiwać prawdopodobieństwem, że w przyszłości nad naszym projektem będą pracować cudzoziemcy.

W kodzie źródłowym znajdują się dwa rodzaje białych znaków:

- Wcięcie: Używamy po to aby od razu było widać, która funkcja jest zagnieżdżona w której. Niezależnie od wybranego wcięcia, zachowaj spójność w całym projekcie.
- Ciągłe spacje: Często niektóre spacje zostaną znalezione po samej linii. To jest nie zalecane, ponieważ może prowadzić do trudnych do wykrycia błędów.

6. Ryzyko jest najistotniejszym (bo najmniej przewidywalnym) wrogiem sukcesu projektu. Zarządzanie ryzykiem ma podstawowe znaczenie dla osiągnięcia celów przedsięwzięcia, a zastosowanie objętych nim technik pozwala minimalizować negatywny wpływ ryzyka, przy jednoczesnej maksymalizacji efektów projektu. Należy zwrócić uwagę iż zarządzanie ryzykiem jest procesem cyklicznym i tylko regularne realizowanie zadań z nim związanych gwarantuje odpowiednią skuteczność i efekty w całym cyklu życia przedsięwzięcia.

Zidentyfikowaliśmy następujące grupy ryzyk i zagrożeń dla naszego portalu kawowego wraz z prawdopodobieństwem ich występowania i potencjalnymi konsekwencjami ich zajścia:

Ryzyko	Zdarzenie	Ppb (0-10)	Konsekwencje
Trudność realizacji logiki biznesowej	Nasze wymagania funkcjonalne są dość wygórowane i rozbudowane. Może się zdażyć tak, że narzędzia, które zamierzamy użyć nie będą wspierały wszystkich funkcjonalności, a żeby je zrealizować trzeba będzie zainwestować nieodpowiednio duży wysiłek lub zmienić wcale stosowane narzędzia.	8	Krytyczne
Projektowanie interfejsu strony internetowej i jej implementacja	Jak wyżej, może się okazać, że Wordpress nie pozwala na realizację pewnych pomysłów design'owych dotyczących wyglądu stron portalu.	7	Marginalne
Błędy harmonogramowania	Projekt może zostać opóźniony w pewnym momencie, przez co trzeba będzie zatrudniać nowych pracowników, płacić stawki ponadnormowe, lub liczyć się z konsekwencjami prawnymi niedostarczenia aplikacji na czas.	4	Krytyczne
„Pełzający” zakres	Typowym niekorzystnym działaniem jest systematyczne rozszerzanie zakresu o dodatkowe wymagania lub swobodne modyfikowanie istotnych założeń projektowych, bez zwracania uwagi na konsekwencje. W takich działaniach celują często nieodpowiedzialni klienci, którzy „usztyniając” cenę i czas realizacji, oczekują spełnienia wszystkich, nawet najbardziej absurdalnych wymagań.	4	Katastroficzne
Czynnik ludzki	Pracownicy mogą opuszczać zespół przed zakończeniem projektu, wymagając nagle szukać zastępstwa na ich miejsce.	6	Pomijalne

Wydajność	Różnice pomiędzy zakładaną, a rzeczywistą wydajnością pracy mogą być duże, np. z powodu niedoszacowania na etapie planowania.	5	Marginalne
Nastawienie do pracy	Angażowanie do projektu pracowników o niewłaściwych lub niewystarczających kompetencjach do realizacji prac.	8	Krytyczne
Współpraca z klientem	Brak lub niewystarczające zaangażowanie klienta w prace projektowe, zagrażające pozytywnej akceptacji projektu przy końcowym odbiorze.	4	Krytyczne

7. Analizując wszystkie dotychczas wykonane prace, można zauważyć, że wizja naszego projektu portalu kawowego bardzo dynamicznie się rozwija. Potrafiliśmy nie tylko zachować początkową idee przedstawioną w tablicy koncepcyjnej, ale również rozszerzyć i pogłębić poziom specyfikacji wymagań funkcjonalnych oraz нефункциональных, wyglądu stron portalu. Udoskonalliliśmy koncepcję interaktywności portalu. Przedstawiliśmy proste, ale bardzo sformalizowane wytyczne, których należy przestrzegać wytwarzając oprogramowanie w naszym projekcie aby ułatwić pracę w zespole. Także została przeprowadzona bardziej realistyczna analiza ryzyka, co umożliwiło wyodrębnienia najkosztowniejszych zagrożeń i ułatwiło wyprodukowanie efektywnych scenariuszy zarządzania ryzykiem.