

## Układanie fundamentów

Teraz, gdy przekonałeś klienta, że możesz stworzyć fajną stronę internetową, która uzupełni jego działalność, czas przestać świętować i zacząć myśleć o tym, jak wprowadzić w życie wszystkie złożone obietnice. Jak zwykle, kiedy przełożysz na papier wymagania techniczne, które musisz spełnić, wszystko zaczyna wydawać się nieco bardziej skomplikowane, niż początkowo zakładano. Aby zapewnić sukces tego projektu, musisz wymyślić sprytny sposób na wdrożenie tego, na co zgodziłeś się przy podpisywaniu umowy. Chcesz sprawnie i szybko rozwijać projekt, ale ostatecznym celem jest upewnienie się, że klient jest zadowolony z Twojej pracy. W związku z tym powinieneś dążyć do zapewnienia rosnącej liczbie odwiedzających witrynę pozytywnych doświadczeń internetowych, tworząc przyjemną, funkcjonalną i responsywną witrynę internetową. Wymagania są wysokie, ale w dzisiejszych czasach jest to normalne w przypadku witryn e-commerce. Aby zmaksymalizować szanse na sukces, przeanalizujemy i przewidzimy jak najwięcej wymagań technicznych i wdrożymy rozwiązania w sposób, który wspiera zmiany i uzupełnienia przy minimalnym wysiłku. Twoje cele w tej części to:

- Analizuj projekt z technicznego punktu widzenia
- Przeanalizuj i wybierz architekturę dla swojej aplikacji
- Zdecyduj, jakich technologii, języków programowania i narzędzi użyć
- Rozważ konwencje nazewnictwa i kodowania

## Projektowanie dla wzrostu

Słowo „projekt” w kontekście aplikacji internetowej może oznaczać wiele rzeczy. Jego najpopularniejsze zastosowanie prawdopodobnie odnosi się do projektowania wizualnego i interfejsu użytkownika witryny internetowej. Ten aspekt jest kluczowy, ponieważ, spójrzmy prawdzie w oczy, odwiedzający jest często bardziej pod wrażeniem wyglądu witryny i łatwości jej użytkowania niż tego, jakie technologie i techniki są używane za kulisami lub jaki system operacyjny działa na serwerze sieciowym. Jeśli witryna jest powolna, trudna w użyciu lub łatwa do zapomnienia, po prostu nie ma znaczenia, jaką technologię raketową wykorzystano do jej stworzenia. Niestety ta prawda sprawia, że wielu niedoświadczonych programistów nie docenia znaczenia sposobu implementacji niewidocznej części strony - kodu, bazy danych i tak dalej. Wizualna część witryny na początku wzbudza zainteresowanie odwiedzających, ale jej funkcjonalność sprawia, że wracają. Witryna internetowa może czasami zostać zaimplementowana bardzo szybko w oparciu o pewne początkowe wymagania, ale jeśli nie jest odpowiednio zaprojektowana, jej zmiana może być trudna, jeśli nie niemożliwa. W przypadku każdego projektu o dowolnej wielkości, przed rozpoczęciem kodowania należy wykonać pewne przygotowania. Jednak bez względu na to, jak wiele prac przygotowawczych i projektowych zostanie wykonanych, zdarzają się nieoczekiwane sytuacje, a ukryte pułapki, nowe wymagania i zmieniające się zasady zawsze wydają się działać wbrew terminom. Nawet bez tych nieoczekiwanych czynników projektanci witryn są często proszeni o zmianę lub dodanie funkcji wiele razy po zakończeniu i wdrożeniu projektu. Tak będzie również w przypadku TShirtShop, który zostanie wdrożony w trzech oddzielnych etapach. Dowiesz się, jak stworzyć witrynę internetową, aby witryna (lub Ty) nie rozpadła się, gdy funkcjonalność zostanie rozszerzona lub dokonywane są aktualizacje. Ponieważ mówmy o programowaniu, zamiast koncentrować się na projektowaniu interfejsu użytkownika lub technikach marketingowych, zwrócimy szczególną uwagę na zaprojektowanie kodu, który sprawi, że będą działać. Wyrażenie „projektowanie kodu” może mieć różne znaczenia; na przykład będziemy potrzebować krótkiej rozmowy o konwencjach nazewnictwa. Jednak najważniejszym aspektem, któremu musimy się przyjrzeć, jest architektura aplikacji. Architektura odnosi się do sposobu, w jaki dzielisz kod na mniejsze komponenty (na przykład funkcja wyszukiwania

produktów) w celu uzyskania prostej funkcjonalności. Chociaż może być łatwiej wdrożyć tę funkcjonalność tak szybko i tak prosto, jak to możliwe w jednym komponencie, zyskujesz duże długoterminowe korzyści, tworząc mniejsze, prostsze komponenty, które współpracują ze sobą, aby osiągnąć pożądany rezultat. Zanim zaczniesz mówić o samej architekturze, musisz określić, czego chcesz od tej architektury.

### **Spełnianie długoterminowych wymagań przy minimalnym wysiłku**

Oprócz tego, że chcesz mieć szybką witrynę internetową, każda z faz rozwoju, o której mówiliśmy w Części 1, niesie ze sobą nowe wymagania, które należy spełnić. Za każdym razem, gdy przechodzisz do nowego etapu, chcesz mieć możliwość ponownego wykorzystania większości już istniejącego rozwiązania. Przeprojektowanie całej witryny (nie tylko części wizualnej, ale także kodu!) byłoby bardzo nieefektywne tylko dlatego, że musisz dodać nową funkcję. Możesz ułatwić ponowne wykorzystanie rozwiązania, planując z wyprzedzeniem, dzięki czemu każda nowa funkcja, która musi zostać dodana, może być łatwo podłączona, a nie każda zmiana powodująca nowy ból głowy. Podczas tworzenia strony internetowej wdrożenie elastycznej architektury złożonej z wtykowych komponentów umożliwia dodawanie nowych funkcji – takich jak koszyk, lista działań lub funkcja wyszukiwania produktów – poprzez zakodowanie ich jako oddzielnych komponentów i podłączenie ich do istniejącej aplikacji. Osiągnięcie dobrego poziomu elastyczności jest jednym z głównych celów związanych z architekturą aplikacji, a ta część pokazuje, jak można to zastosować w praktyce. Zobaczysz, że poziom elastyczności jest proporcjonalny do ilości czasu potrzebnego do zaprojektowania i wdrożenia, więc postaramy się znaleźć kompromis, który zapewni najlepsze korzyści bez zbytniego komplikowania kodu. Innym ważnym wymaganiem wspólnym dla wszystkich aplikacji internetowych jest skalowalna architektura. Skalowalność jest definiowana jako zdolność do zwiększania zasobów w celu uzyskania liniowego wzrostu wydajności usług. Innymi słowy, w idealnym przypadku w skalowalnym systemie stosunek (proporcja) między liczbą żądań klientów a zasobami sprzętowymi wymaganymi do obsługi tych żądań jest stały, nawet gdy liczba klientów wzrasta. System nieskalowalny nie poradzi sobie z rosnącą liczbą klientów, bez względu na ilość dostępnych zasobów sprzętowych. Ponieważ jesteśmy optymistycznie nastawieni do liczby klientów, musimy mieć pewność, że strona będzie w stanie dostarczyć swoją funkcjonalność dużej liczbie klientów bez wyrzucania błędów lub powolnego działania. Niezawodność jest również krytycznym aspektem aplikacji e-commerce. Za pomocą spójnej strategii obsługi błędów i potężnej relacyjnej bazy danych możesz zapewnić integralność danych i zapewnić właściwą obsługę niekrytycznych błędów bez rzucania witryny na kolana.

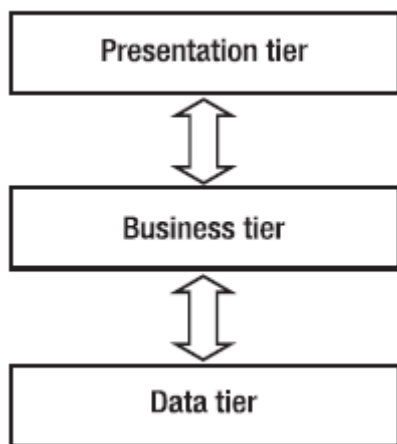
### **Magia trójpoziomowej architektury**

Ogólnie rzecz biorąc, architektura odnosi się do sposobu, w jaki dzielimy kod, który implementuje funkcję aplikacji na oddzielne komponenty w oparciu o to, co robią i grupujemy każdy rodzaj komponentu w pojedynczą warstwę logiczną. W szczególności architektura trójwarstwowa odnosi się do architektury opartej na tych warstwach:

- Poziom prezentacji
- Poziom biznesowy
- Warstwa danych

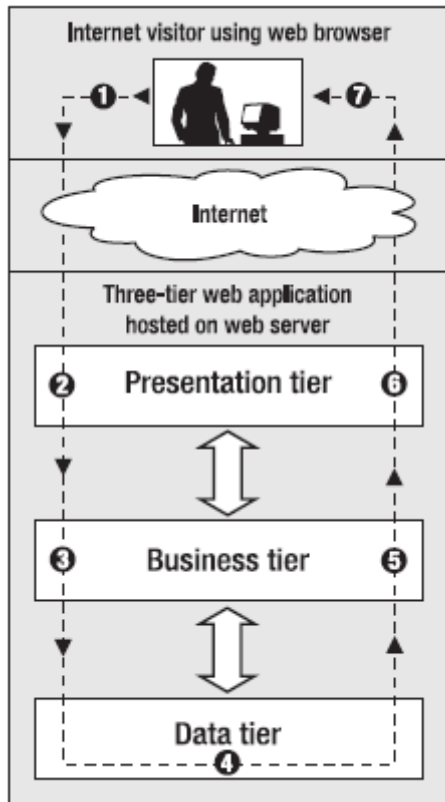
Warstwa prezentacji zawiera elementy interfejsu użytkownika witryny i zawiera całą logikę, która zarządza interakcją między odwiedzającym a firmą klienta. Ta warstwa sprawia, że cała witryna wydaje się żywa, a sposób jej projektowania ma kluczowe znaczenie dla sukcesu witryny. Ponieważ aplikacja

jest witryną sieci Web, jej warstwa prezentacji składa się z dynamicznych stron sieci Web. Warstwa biznesowa (nazywana również warstwą środkową) odbiera żądania z warstwy prezentacji i zwraca wynik do warstwy prezentacji w zależności od zawartej logiki biznesowej. Prawie każde zdarzenie, które ma miejsce w warstwie prezentacji, zwykle powoduje wywołanie warstwy biznesowej (wykorzystanie), z wyjątkiem zdarzeń, które mogą być obsługiwane lokalnie przez warstwę prezentacji, takich jak prosta weryfikacja danych wejściowych i tak dalej. Na przykład, jeśli odwiedzający przeprowadza wyszukiwanie produktów, warstwa prezentacji wywołuje warstwę biznesową i mówi: „Proszę odeślij mi produkty spełniające to kryterium wyszukiwania”. W większości przypadków warstwa biznesowa musi wywołać warstwę danych w celu uzyskania informacji, aby móc odpowiedzieć na żądanie warstwy prezentacji. Warstwa danych (czasami nazywana warstwą bazy danych) jest odpowiedzialna za zarządzanie danymi aplikacji i wysyłanie ich do warstwy biznesowej na żądanie. W przypadku witryny e-commerce TShirtShop musisz przechowywać dane o produktach (w tym ich kategoriach i działach), użytkownikach, koszykach i tak dalej. Prawie każde żądanie klienta powoduje w końcu odpytanie warstwy danych o informacje (z wyjątkiem sytuacji, gdy wcześniej pobrane dane zostały zbuforowane na poziomie warstwy biznesowej lub warstwy prezentacji), dlatego ważne jest, aby mieć szybki system bazy danych. Później dowiesz się, jak zaprojektować bazę danych pod kątem optymalnej wydajności. Te warstwy są czysto logiczne - nie ma ograniczeń co do fizycznej lokalizacji każdej warstwy. Teoretycznie możesz umieścić całą aplikację i domyślnie wszystkie jej warstwy na pojedynczym serwerze lub możesz umieścić każdą warstwę na osobnym komputerze, jeśli aplikacja na to pozwala. Część 22 wyjaśnia, jak zintegrować funkcje z innych witryn internetowych za pomocą usług XML Web Services. Usługi XML sieci Web umożliwiają łatwą integrację funkcji na wielu serwerach bez konieczności dostosowywania kodu. Ważnym ograniczeniem w modelu architektury trójwarstwowej jest to, że informacje muszą przepływać w kolejności między warstwami. Warstwa prezentacji ma dostęp tylko do warstwy biznesowej i nigdy nie może uzyskać bezpośredniego dostępu do warstwy danych. Warstwa biznesowa to mózg pośrodku, który komunikuje się z innymi warstwami i procesami oraz koordynuje cały przepływ informacji. Gdyby warstwa prezentacji miała bezpośredni dostęp do warstwy danych, zasady programowania architektury trójwarstwowej zostałyby złamane. Te zasady mogą początkowo wyglądać na ograniczenia, ale korzystając z architektury, musisz być konsekwentny i przestrzegać jej zasad, aby czerpać korzyści. Trzymanie się architektury trójwarstwowej zapewnia łatwą aktualizację lub zmianę witryny oraz zapewnia większą kontrolę nad tym, kto lub co może uzyskać dostęp do Twoich danych. W tej chwili może się to wydawać niepotrzebnym obciążeniem; Jednak przestrzeganie tego systemu może w przyszłości przynieść znaczne korzyści, gdy będziesz musiał zmienić działanie lub logikę witryny. Rysunek przedstawia prostą reprezentację sposobu przekazywania danych w aplikacji, która implementuje architekturę trójwarstwową.



**Prosty przykład wykorzystujący architekturę trójwarstwową**

Łatwiej jest zrozumieć, w jaki sposób dane są przekazywane i przekształcane między warstwami, jeśli przyjrzy się bliżej prostemu przykładowi. Aby przykład był jeszcze bardziej adekwatny do naszego projektu, przeanalizujemy sytuację, która faktycznie wydarzy się w TShirtShop. Ten scenariusz jest typowy dla aplikacji trójwarstwowych. Podobnie jak większość witryn e-commerce, TShirtShop będzie miał koszyk na zakupy, który omówimy w dalszej części. Na razie wystarczy wiedzieć, że odwiedzający doda produkty do koszyka, klikając przycisk Dodaj do koszyka. Rysunek pokazuje, w jaki sposób informacje przepływają przez aplikację po kliknięciu tego przycisku.

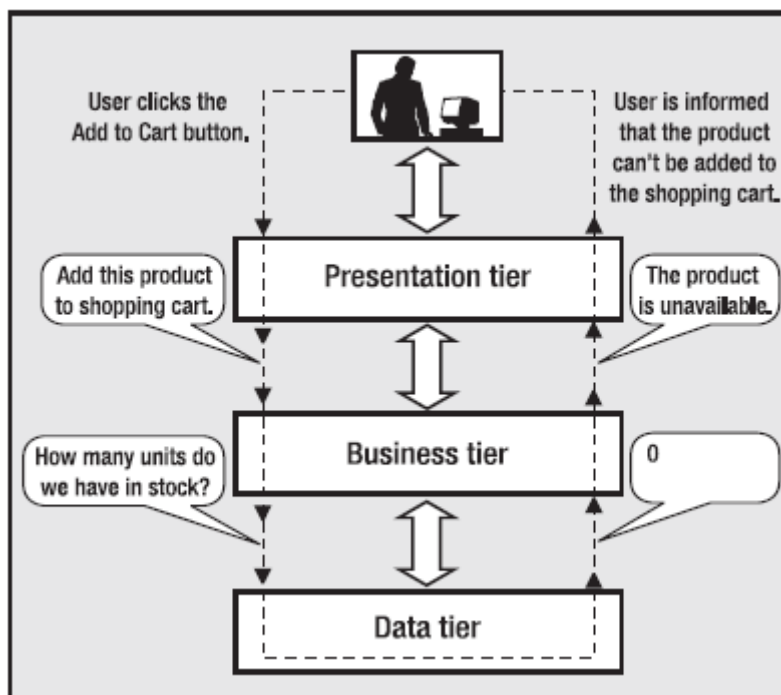


W kroku 1 użytkownik klika przycisk Dodaj do koszyka dla określonego produktu. W kroku 2 warstwa prezentacji (zawierająca przycisk) przekazuje żądanie do warstwy biznesowej „Hej, chcę dodać ten produkt do mojego koszyka!” W kroku 3 warstwa biznesowa otrzymuje żądanie, rozumie, że użytkownik chce dodać określony produkt do koszyka i obsługuje żądanie, nakazując warstwie danych zaktualizowanie koszyka na zakupy odwiedzającego poprzez dodanie wybranego produktu. Warstwa danych musi zostać wywołana, ponieważ przechowuje i zarządza danymi całej witryny internetowej, w tym informacjami o koszyku zakupów użytkowników. W kroku 4 warstwa danych aktualizuje bazę danych i ostatecznie zwraca kod powodzenia do warstwy biznesowej. W kroku 5 warstwa biznesowa obsługuje kod powrotu i wszelkie błędy, które mogły wystąpić w warstwie danych podczas aktualizowania bazy danych, a następnie zwraca dane wyjściowe do warstwy prezentacji. W kroku 6 warstwa prezentacji generuje zaktualizowany widok koszyka. W kroku 7 wyniki wykonania są pakowane poprzez generowanie strony internetowej w języku Hypertext Markup Language (HTML), która jest zwracana odwiedzającemu, gdzie zaktualizowany koszyk może być widoczny w przeglądarce odwiedzającego. Zwróć uwagę, że w tym prostym przykładzie warstwa biznesowa nie wykonuje dużo przetwarzania, a jej logika biznesowa nie jest bardzo złożona. Jeśli jednak pojawią się nowe reguły biznesowe dla Twojej aplikacji, zmienisz warstwę biznesową. Jeśli na przykład logika biznesowa określa, że produkt może zostać dodany do koszyka tylko wtedy, gdy jego ilość w magazynie jest większa niż zero, zostanie wykonane dodatkowe wywołanie warstwy danych w celu określenia ilości.

Warstwa danych zostanie poproszona o zaktualizowanie koszyka zakupów tylko wtedy, gdy produkty są w magazynie. W każdym przypadku warstwa prezentacji jest informowana o stanie i zapewnia odwiedzającemu informację zwrotną czytelną dla człowieka.

### Co jest w liczbie?

Warto zauważyć, że każdy poziom różnie interpretuje te same informacje. W warstwie danych przechowywane w niej liczby i informacje nie mają znaczenia, ponieważ ta warstwa jest aparatem, który zapisuje, zarządza i pobiera liczby, ciągi znaków lub inne typy danych oraz je pobiera - dla warstwy danych te dane są tylko arbitralną informacją, a nie ilościami produktów lub nazwy produktów. W kontekście poprzedniego przykładu ilość produktu równa zero reprezentuje prostą, zwykłą liczbę bez znaczenia dla warstwy danych (jest to po prostu zero, 32-bitowa liczba całkowita). Dane nabierają znaczenia dopiero wtedy, gdy warstwa biznesowa je odczytuje. Gdy warstwa biznesowa pyta warstwę danych o ilość produktu i otrzymuje wynik „0”, jest to interpretowane przez warstwę biznesową jako „Hej, brak produktów w magazynie!” Te dane są ostatecznie opakowane w ładną, wizualną formę przez warstwę prezentacji, taką jak etykieta z napisem „Przepraszamy, w tej chwili nie można zamówić tego produktu”. Nawet jeśli jest mało prawdopodobne, że chcesz zabronić klientowi dodawania produktu do koszyka, jeśli nie ma go w magazynie, przykład (opisany na rysunku) jest wystarczająco dobry, aby w jeszcze inny sposób przedstawić, jak każdy z trzech poziomów mają inny cel.



### Właściwa logika dla właściwego poziomu

Ponieważ każda warstwa zawiera własną logikę, czasami trudno jest zdecydować, gdzie dokładnie narysować linie między warstwami. W poprzednim scenariuszu zamiast odczytywać ilość produktu w warstwie biznesowej i decydować, czy produkt jest dostępny na podstawie tej liczby (co ostatecznie skutkuje dwoma wywołaniami bazy danych), możesz mieć jedną procedurę składowaną o nazwie `add_product_if_available`, która dodaje produkt do koszyka tylko wtedy, gdy jest dostępny w magazynie. W tym scenariuszu część logiki jest przenoszona z warstwy biznesowej do warstwy danych. W wielu innych okolicznościach możesz mieć możliwość umieszczenia logiki w jednej lub drugiej warstwie, a może w obu. W większości przypadków nie ma jednego najlepszego sposobu na wdrożenie architektury trójwarstwowej i musisz dokonać kompromisu lub wyboru opartego na osobistych

preferencjach lub ograniczeniach zewnętrznych. Co więcej, zdarzają się sytuacje, w których nawet jeśli znasz właściwy sposób (w odniesieniu do architektury), aby coś zaimplementować, możesz zdecydować się na złamanie zasad, aby uzyskać wzrost wydajności. Zgodnie z ogólną zasadą, jeśli wydajność można poprawić w ten sposób, można nieco złamać ściśle limity między warstwami (na przykład dodać niektóre reguły biznesowe do warstwy danych lub odwrotnie), jeśli te reguły są niezmienne w czasie. W przeciwnym razie preferowane jest zachowanie wszystkich reguł biznesowych w warstwie środkowej, ponieważ generuje czystsza aplikację, która jest łatwiejsza w utrzymaniu. Na koniec nie ulegaj pokusie, aby uzyskać dostęp do warstwy danych bezpośrednio z warstwy prezentacji. To powszechny błąd, który jest najkrótszą drogą do skomplikowanego, trudnego w utrzymaniu i nieelastycznego systemu. W wielu samouczkach dostępu do danych lub materiałów wprowadzających zobaczysz, jak wykonywać podstawowe operacje na bazie danych za pomocą prostej aplikacji interfejsu użytkownika. W tego typu programach cała logika jest prawdopodobnie napisana w krótkim, pojedynczym pliku, a nie w oddzielnych warstwach. Chociaż materiały mogą być bardzo dobre, pamiętaj, że większość z tych tekstów ma na celu nauczenie Cię, jak wykonywać różne poszczególne zadania (na przykład dostęp do bazy danych), a nie jak poprawnie tworzyć elastyczną i skalowalną aplikację.

### **Trójpoziomowa architektura dla TShirtShop**

Wdrożenie trójwarstwowej architektury witryny internetowej TShirtShop pomoże osiągnąć cele wymienione na początku rozdziału. Dyscyplina kodowania, narzucona przez system, który na pierwszy rzut oka może wydawać się sztywny, zapewnia doskonały poziom elastyczności i rozszerzalności na dłuższą metę. Podział głównych części aplikacji na oddzielne mniejsze elementy zachęca do ponownego użycia. Niejednokrotnie dodając nowe funkcje do witryny, zobaczysz, że możesz ponownie wykorzystać niektóre z już istniejących bitów. Dodanie nowej funkcji bez konieczności zmiany wielu elementów już istniejących jest samo w sobie dobrym przykładem ponownego wykorzystania. Kolejną zaletą architektury trójwarstwowej jest to, że przy prawidłowym wdrożeniu cały system jest odporny na zmiany. Gdy bity w jednym z poziomów zmieniają się, pozostałe poziomy zwykle pozostają nienaruszone, czasami nawet w skrajnych przypadkach. Na przykład, jeśli z jakiegoś powodu system baz danych zaleca zostanie zmieniony (powiedzmy, że kierownik zdecyduje się użyć PostgreSQL zamiast MySQL), wystarczy zaktualizować warstwę danych, a może tylko trochę warstwy biznesowej.

### **Dlaczego nie użyć większej liczby poziomów?**

Architektura trójwarstwowa, o której mówiliśmy do tej pory, jest szczególną (i najpopularniejszą) wersją architektury n-warstwowej. Architektura n-warstwowa odnosi się do podziału rozwiązania na liczbę (n) warstw logicznych. W złożonych projektach czasami sensowne jest podzielenie warstwy biznesowej na więcej niż jedną warstwę, co skutkuje architekturą składającą się z więcej niż trzech warstw. Jednak w przypadku naszej witryny internetowej najbardziej sensowne jest trzymanie się trójwarstwowego projektu, który oferuje większość korzyści, a jednocześnie nie wymaga zbyt wielu godzin projektowania ani złożonej hierarchii kodu frameworka do obsługi architektury. Być może dzięki bardziej zaangażowanej i złożonej architekturze można osiągnąć jeszcze wyższy poziom elastyczności i skalowalności aplikacji, ale przed rozpoczęciem implementacji potrzeba znacznie więcej czasu na projektowanie. Podobnie jak w przypadku każdego projektu programistycznego, musisz znaleźć odpowiednią równowagę między czasem wymaganym do zaprojektowania architektury a czasem poświęconym na jej wdrożenie. Architektura trójwarstwowa najlepiej nadaje się do projektów o średniej złożoności, takich jak witryna internetowa TShirtShop. Możesz również zadać przeciwne pytanie: „Dlaczego nie użyć mniejszej liczby poziomów?” Dwupoziomowa Architektura, zwana również architekturą klient-serwer, może być odpowiednia dla mniej złożonych projektów. Krótko mówiąc, architektura dwupoziomowa wymaga mniej czasu na planowanie i pozwala na szybszy rozwój na

początku; jednak generuje aplikację, która na dłuższą metę jest trudniejsza w utrzymaniu i rozbudowie. Ponieważ spodziewamy się, że będziemy musieli rozszerzać aplikację w przyszłości, architektura client-server nie jest odpowiednia dla naszej aplikacji, więc nie będzie ona dalej omawiana. Teraz, gdy ogólna architektura jest znana, zobaczymy, jakie technologie i narzędzia będziesz musiał użyć, aby go zaimplementować. Pokróćce omówimy te technologie, a w Części 3 stworzysz podstawę prezentacji i warstw danych, tworząc pierwszą stronę witryny i bazę danych zaplecza. Realną funkcjonalność na każdym z trzech poziomów opisanych w Części 4 rozpoczniesz, gdy zaczniesz tworzyć katalog produktów w witrynie sieci Web.

### **Wybór technologii i narzędzi**

Bez względu na wybraną architekturę, głównym pytaniem, które pojawia się w każdym projekcie deweloperskim, jest to, jakie technologie, języki programowania i narzędzia zostaną użyte, pamiętając, że wymagania zewnętrzne mogą poważnie ograniczyć Twoje możliwości. Tu tworzymy witrynę internetową przy użyciu PHP, MySQL i powiązanych technologii. Naprawdę lubimy te technologie, ale niekoniecznie oznacza to, że są najlepszym wyborem dla każdego rodzaju projektu, w każdych okolicznościach. Dodatkowo istnieje wiele sytuacji, w których ze względu na wymagania klienta konieczne jest użycie określonych technologii. Etap analizy wymagań, który jest obecny w większości procesów tworzenia oprogramowania, określi, jakich technologii należy użyć do stworzenia aplikacji. Chociaż zakładamy pewne wcześniejsze doświadczenia z PHP i MySQL, przyjrzymy się im szybko i zobaczymy, jak pasują do naszego projektu i architektury trójwarstwowej.

### **Używanie PHP do generowania dynamicznych treści internetowych**

PHP to technologia open source do tworzenia dynamicznych, interaktywnych treści internetowych. Jego krótki opis (na oficjalnej stronie PHP, <http://www.php.net>) brzmi: „PHP to powszechnie używany język skryptowy ogólnego przeznaczenia, który jest szczególnie odpowiedni do tworzenia stron internetowych i może być osadzony w HTML”. PHP oznacza PHP: Hypertext Preprocessor (tak, jest to akronim rekurencyjny) i jest dostępny do bezpłatnego pobrania na oficjalnej stronie internetowej. Historia PHP, którego korzenie sięgają 1994 roku, jest udana. Wśród czynników, które doprowadziły do jego sukcesu są:

- PHP jest bezpłatne; szczególnie w połączeniu z oprogramowaniem serwerowym Linux, PHP może okazać się bardzo opłacalną technologią do tworzenia dynamicznych treści internetowych.
- PHP ma krótszą krzywą uczenia się niż inne języki skryptowe.
- Społeczność PHP jest zwinna. Wiele przydatnych bibliotek pomocniczych lub nowe wersje istniejących bibliotek są opracowywane, a nowe funkcje są często dodawane.
- PHP działa bardzo dobrze na różnych serwerach internetowych i systemach operacyjnych (platformy typu Unix, Windows i Mac OS).

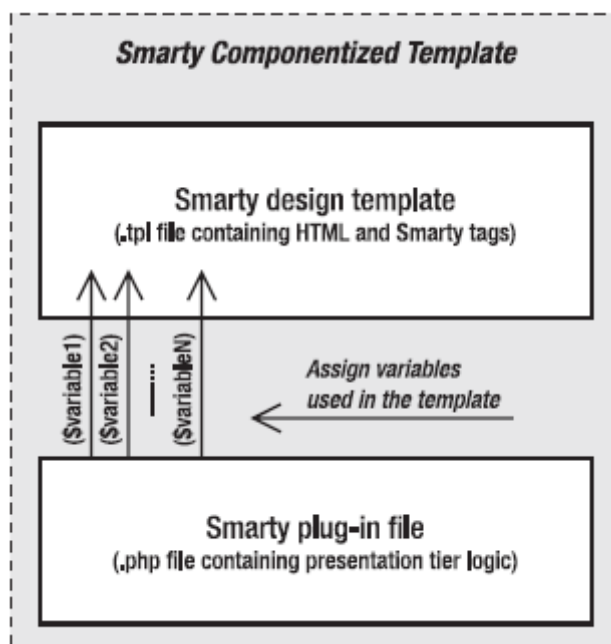
Jednak PHP nie jest jedynym językiem skryptowym po stronie serwera do tworzenia dynamicznych stron internetowych. Do najpopularniejszych konkurentów należą JavaServer Pages (JSP), Perl, ColdFusion i ASP.NET. Wśród tych technologii jest wiele różnic, ale także pewne zasadnicze podobieństwa. Na przykład strony napisane za pomocą dowolnej z tych technologii składają się z podstawowego kodu HTML, który rysuje statyczną część strony (szablon) oraz kodu generującego część dynamiczną.

### **Używanie Smarty do oddzielania układu od kodu**

Ponieważ PHP jest proste i łatwe do rozpoczęcia, zawsze kuszące było rozpoczęcie kodowania bez odpowiedniego zaprojektowania architektury i frameworka, które byłyby korzystne na dłuższą metę. Co gorsza, prostą metodą budowania stron PHP jest łączenie instrukcji PHP z HTML, ponieważ PHP nie ma domyślnie oczywistej techniki oddzielania kodu PHP od informacji o układzie HTML. Mieszanie logiki PHP z HTML ma dwie ważne wady:

- Ta technika często prowadzi do długiego, skomplikowanego i trudnego w zarządzaniu kodu. Być może widziałeś te kilometrowe pliki źródłowe z nieprzyjemną mieszanką PHP i HTML, które są trudne do odczytania i niemożliwe do zrozumienia po tygodniu.
- Te mieszane pliki są przedmiotem pracy zarówno projektantów, jak i programistów, co komplikuje współpracę bardziej niż to konieczne. Zwiększa to również szanse projektanta na tworzenie błędów w logice kodu podczas pracy nad kosmetycznymi zmianami.

Tego rodzaju problemy doprowadziły do opracowania silników szablonów, które oferują frameworki oddzielające logikę prezentacji od statycznego układu HTML. Smarty to najpopularniejszy i najpotężniejszy silnik szablonów dla PHP. Jego głównym celem jest zaoferowanie prostego sposobu oddzielenia logiki aplikacji (kod PHP) od kodu prezentacji (HTML). Ta separacja pozwala programiście i projektantowi szablonów pracować niezależnie na tej samej aplikacji. Programista może zmieniać logikę PHP bez konieczności zmiany plików szablonów, a projektant może zmieniać szablony bez dbania o to, jak działa kod, który je ożywia. Rysunek przedstawia relację między plikiem szablonu projektu Smarty a jego plikiem wtyczki Smarty.



Szablon projektu Smarty (plik .tpl zawierający układ HTML oraz tagi i kod specyficzne dla Smarty) oraz jego plik wtyczki Smarty (plik .php zawierający powiązany kod szablonu) tworzą szablon skomponowany na komponenty Smarty. W praktyce nie utworzymy pliku wtyczki Smarty dla każdego szablonu, jak pokazano na rysunku. Zamiast tego stworzymy ogólną wtyczkę Smarty, która integruje się ze wszystkimi szablonami Smarty, ładując niezbędne obiekty prezentacji. Obiekty prezentacji to klasy, które dostarczają plikom szablonów potrzebnych danych. Dowiesz się więcej o tym, jak działa Smarty, budując witrynę e-commerce.



**Uwaga:** dodanie Smarty'ego lub innego silnika szablonów do architektury aplikacji internetowej zwiększa początkowy wysiłek związany z kodowaniem, a także wiąże się z krzywą uczenia się. Jednak i tak powinieneś spróbować, ponieważ zalety korzystania z tak nowoczesnej techniki programowania okażą się znaczące w dalszej części procesu.

### **A co z alternatywami?**

Smarty nie jest jedynym silnikiem szablonów dostępnym dla PHP. Możesz znaleźć wiele innych, wyszukując w Google „silniki szablonów PHP”. Najpopularniejsze z nich można znaleźć ładnie wymienione przez Justina Silvertona w jego artykule „Top 25 PHP Template Engines” (Twoja ulubiona wyszukiwarka pomoże ci ponownie znaleźć artykuł). Chociaż wszystkie silniki szablonów działają zgodnie z tymi samymi podstawowymi zasadami, zdecydowaliśmy się użyć Smarty w projekcie PHP e-commerce dla tej książki ze względu na jego bardzo dobre wyniki wydajności, potężne funkcje (takie jak kompilacja szablonów i buforowanie) oraz szeroką akceptację w branży .

### **Używanie MySQL do przechowywania danych witryn internetowych**

Większość danych, które odwiedzający zobaczą podczas przeglądania witryny internetowej, zostanie pobrana z relacyjnej bazy danych. System zarządzania relacyjnymi bazami danych (RDBMS) to złożony program, którego celem jest przechowywanie, zarządzanie i pobieranie danych tak szybko i niezawodnie, jak to tylko możliwe. W przypadku witryny internetowej TShirtShop będzie przechowywać wszystkie dane dotyczące produktów, działów, użytkowników, koszyków i tak dalej. Wiele systemów RDBMS jest dostępnych do użycia z PHP, w tym MySQL, PostgreSQL, Oracle i tak dalej. Jednak zarówno zwykłe ankiety, jak i praktyka w świecie rzeczywistym pokazują, że MySQL jest naprawdę wiodącym wyborem bazy danych dla projektów opartych na PHP. MySQL to najpopularniejsza na świecie baza danych typu open source, bezpłatna (do użytku niekomercyjnego), szybka i niezawodna baza danych. Kolejną ważną zaletą jest to, że wielu dostawców usług hostingowych oferuje dostęp do bazy danych aMySQL, co ułatwia życie z nowo utworzoną witryną e-commerce. Użyjemy MySQL jako wewnętrznej bazy danych podczas tworzenia witryny e-commerce TShirtShop. Językiem używanym do komunikacji z relacyjną bazą danych jest SQL (SQL Query Language lub, zgodnie ze starszymi specyfikacjami, Structured Query Language). Jednak każdy silnik bazy danych rozpoznaje określony dialekt tego języka. Jeśli zdecydujesz się użyć innego RDBMS niż MySQL, prawdopodobnie będziesz musiał zaktualizować niektóre zapytania SQL.

### **Kontakt z MySQL**

Rozmawiasz z serwerem bazy danych, formułując zapytanie SQL, wysyłając je do silnika bazy danych i pobierając wyniki. Zapytanie SQL może powiedzieć wszystko, co ma związek z danymi witryny internetowej lub jej strukturami danych, na przykład „pokaż mi listę działów”, „usuń numer produktu 223”, „utwórz tabelę danych” lub „przeszukaj katalog w poszukiwaniu żółtego koszulki.” Bez względu na to, co mówi zapytanie SQL, potrzebujemy sposobu na przesłanie go do MySQL. MySQL jest dostarczany z prostym interfejsem tekstowym (o nazwie mysql), który umożliwia wykonywanie zapytań SQL i zwraca wyniki. Jeśli uważasz, że jest to trudne w użyciu, nie martw się; istnieją alternatywy dla interfejsu wiersza poleceń. Kilka bezpłatnych narzędzi do administrowania bazami danych innych firm umożliwia manipulowanie strukturami danych i wykonywanie zapytań SQL za pomocą łatwego w użyciu interfejsu graficznego. Wiele firm hostingowych oferuje dostęp do bazy danych za pośrednictwem phpMyAdmin (który jest najczęściej używanym interfejsem klienta internetowego MySQL), co jest kolejnym dobrym powodem, aby zapoznać się z tym narzędziem. Możesz jednak skorzystać z wybranego klienta wizualnego. Popularnym narzędziem komputerowym do interakcji z bazami danych MySQL jest Toad for MySQL . Oprócz konieczności interakcji z MySQL za pośrednictwem bezpośredniego interfejsu do jego silnika, musisz także nauczyć się programowo

uzyskiwać dostęp do MySQL z kodu PHP. To wymaganie jest oczywiste, ponieważ witryna e-commerce będzie musiała przeszukiwać bazę danych, aby pobrać informacje z katalogu (działy, kategorie, produkty itd.) podczas tworzenia stron dla odwiedzających. Jeśli chodzi o zapytania do baz danych MySQL za pomocą kodu PHP, narzędziem, na którym możesz polegać, jest rozszerzenie PHP Data Objects (PDO).

### **Wdrażanie integracji bazy danych za pomocą PDO**

PDO (PHP Data Objects) to natywna biblioteka abstrakcji dostępu do danych, która jest dostarczana z PHP od wersji 5.1 i jest oferowana jako rozszerzenie PECL dla PHP 5.0 (PECL to repozytorium rozszerzeń PHP). PDO oferuje jednolity sposób dostępu do różnych źródeł danych. Korzystanie z PDO zwiększa przenośność i elastyczność aplikacji, ponieważ w przypadku zmiany wewnętrznej bazy danych wpływ na kod dostępu do danych jest minimalny (w wielu przypadkach wystarczy zmienić parametry połączenia dla nowej bazy danych). Po zapoznaniu się z warstwą abstrakcji dostępu do danych PDO możesz użyć tych samych technik programowania w innych projektach, które mogą wymagać innego rozwiązania bazodanowego. Aby zademonstrować różnicę między dostępem do bazy danych za pomocą starych funkcji PHP a PDO, przyjrzyjmy się dwóm krótkim fragmentom kodu PHP. Poniżej przedstawiono dostęp do bazy danych przy użyciu natywnych funkcji PHP (specyficznych dla MySQL):

```
// Connecting to MySQL
$link = mysql_connect('localhost',&nbsp;$username, $password);
if (!$link)
{
die ('Could not connect: ' . mysql_error());
}
$db_selected = mysql_select_db('tshirtshop', $link);
if (!$db_selected)
{
die ('Could not select database : ' . mysql_error());
}
// Execute SQL query
$queryString = 'SELECT * FROM product';
$result = mysql_query($queryString);
if (!$result)
{
die ('Query failed : ' . mysql_error());
}
// Close connection
mysql_close($link);
```

**Uwaga** : Jeśli nadal planujesz używać rozszerzenia PHP MySQL zamiast PDO w swoich projektach, powinieneś rozważyć użycie ulepszonych rozszerzenia PHP MySQL (mysqli).

Następnie pokazana jest ta sama akcja, tym razem przy użyciu PDO:

```
try
{
// Create a new PDO instance
$database_handler =
new PDO('mysql:host=localhost;dbname=tshirtshop',
$username, $password);
// Build the SQL query
$sqlQuery = 'SELECT * FROM product';
// Execute SQL query
$stmt_handler = $database_handler->query($sqlQuery);
// Fetch data
$result = $stmt_handler->fetchAll(PDO::FETCH_ASSOC);
// Clear the PDO object instance
$database_handler = null;
}
catch (PDOException $e)
{
/* If something goes wrong we catch the exception thrown by
the object, print the message, and stop the execution of
script */
print 'Error! <br />' . $e->getMessage() . '<br />';
exit;
}
```

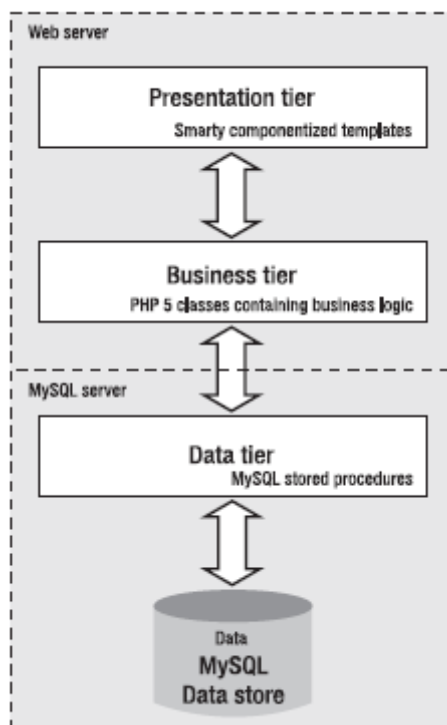
Wersja kodu używająca PDO jest dłuższa, ale zawiera potężny mechanizm obsługi błędów - bardzo pomocne narzędzie podczas debugowania aplikacji. Jeśli te koncepcje brzmią obco, jeszcze raz poczekaj do dalszych rozdziałów, w których umieścimy PDO w pracy, a dowiesz się tam więcej na ten temat. Ponadto, korzystając z PDO, nie będziesz musiał zmieniać kodu dostępu do danych, jeśli na przykład zdecydujesz się na użycie PostgreSQL zamiast MySQL. Z drugiej strony, pierwszy fragment kodu, który używa funkcji specyficznych dla MySQL, musiałby zostać całkowicie zmieniony (użyj `pg_connect` i `pg_query` zamiast `mysql_connect` i `mysql_query` itd.). Ponadto niektóre funkcje specyficzne dla PostgreSQL mają inne parametry niż podobne funkcje MySQL. Korzystając z warstwy abstrakcji bazy

danych (takiej jak PDO), prawdopodobnie będziesz musiał zmienić ciąg połączenia tylko podczas zmiany zaplecza bazy danych. Zauważ, że tutaj mówimy tylko o kodzie PHP, który współdziała z bazą danych. W praktyce może zajść potrzeba zaktualizowania niektórych zapytań SQL, jeśli silniki bazy danych obsługują różne dialekty SQL.

**Uwaga:** Aby zapytania SQL były jak najbardziej przenośne, staraj się, aby ich składnia była jak najbardziej zbliżona do standardu SQL-92

### MySQL i architektura trójwarstwowa

Jest już jasne, że MySQL jest w jakiś sposób powiązany z warstwą danych. Jeśli jednak do tej pory nie pracowałeś z bazami danych, może być mniej niż oczywiste, że MySQL jest czymś więcej niż zwykłym magazynem danych. Oprócz rzeczywistych danych przechowywanych w środku, MySQL może również przechowywać logikę w postaci procedur składowanych, utrzymywać relacje między tabelami, zapewniać przestrzeganie różnych zasad integralności danych i tak dalej. Możesz komunikować się z MySQL poprzez SQL, który jest językiem używanym do interakcji z bazą danych. SQL służy do przesyłania do bazy danych instrukcji, takich jak „wyślij mi 10 ostatnich zamówień” lub „usuń numer produktu 123”. Chociaż możliwe jest tworzenie instrukcji SQL w kodzie PHP, a następnie przesyłanie ich do wykonania, jest to generalnie zła praktyka, ponieważ wiąże się z karami w zakresie bezpieczeństwa, spójności i wydajności. W naszym rozwiązaniu będziemy przechowywać całą logikę warstwy danych za pomocą funkcji bazy danych. Rolę serwera MySQL w architekturze trójwarstwowej opisano na rysunku.



### Wybór standardów nazewnictwa i kodowania

Chociaż standardy kodowania i nazewnictwa mogą z początku wydawać się nieistotne, zdecydowanie nie należy ich pomijać. Nieprzestrzeganie zestawu reguł dotyczących kodu prawie zawsze spowoduje, że kod będzie trudny do odczytania, zrozumienia i utrzymania. Z drugiej strony, gdy podążasz spójnym sposobem kodowania, możesz prawie powiedzieć, że Twój kod jest już w połowie udokumentowany,

co jest ważnym wkładem w łatwość utrzymania projektu, zwłaszcza gdy wiele osób pracuje jednocześnie nad tym samym projektem.

**Wskazówka** : Niektóre firmy mają własne zasady dotyczące standardów kodowania i nazewnictwa, podczas gdy w innych możesz swobodnie korzystać z własnych preferencji. W obu przypadkach złota zasada, której należy przestrzegać, to spójność w sposobie kodowania. Komentowanie kodu to kolejna dobra praktyka, która poprawia długoterminową konserwację kodu.

Konwencje nazewnictwa odnoszą się do wielu elementów w projekcie, ponieważ prawie wszystkie elementy projektu mają nazwy: sam projekt, pliki, klasy, zmienne, metody, parametry metod, tabele bazy danych, kolumny bazy danych i tak dalej. Bez pewnej dyscypliny przy nazywaniu wszystkich tych elementów, po tygodniu kodowania nie zrozumiesz ani jednej linijki tego, co napisałeś. Podczas tworzenia TShirtShop kierowaliśmy się zestawem konwencji nazewnictwa, które są popularne wśród programistów PHP. Niektóre z najważniejszych zasad zostały podsumowane tutaj oraz w poniższym fragmencie kodu:

- Nazwy klas i nazwy metod należy pisać z użyciem wielkości liter Pascal (wielkie litery dla pierwszej litery w każdym słowie), np. WarZone.
- Nazwy atrybutów klas publicznych podlegają tym samym zasadom, co nazwy klas, ale powinny być poprzedzone znakiem „m”. Tak więc prawidłowe nazwy atrybutów publicznych wyglądają tak: \$mSomeSoldier.
- Nazwy atrybutów klas prywatnych podlegają tym samym regułom, co nazwy atrybutów klas publicznych, z wyjątkiem tego, że są również poprzedzone podkreśleniem, na przykład \$\_mSomeOtherSoldier.
- W nazwach argumentów metod należy używać wielkich liter (wielkie litery dla pierwszej litery w każdym słowie z wyjątkiem pierwszego), np. \$someEnemy, \$someOtherEnemy.
- Nazwy zmiennych należy pisać małymi literami, z podkreśleniem jako separatorem słów, np. \$master\_of\_war.
- Obiekty bazy danych używają tych samych konwencji, co nazwy zmiennych (kolumna id\_wydziału).
- Spróbuj wciąć swój kod, używając stałej liczby spacji (powiedzmy cztery) dla każdego poziomu. (Kod w tej książce używa dwóch spacji ze względu na fizyczne ograniczenia przestrzeni).

Oto przykładowy fragment kodu:

```
class WarZone
{
    public $mSomeSoldier;
    private $_mSomeOtherSoldier;
    function SearchAndDestroy($someEnemy, $someOtherEnemy)
    {
        $master_of_war = 'Soldier';
        $this->mSomeSoldier = $someEnemy;
```

```
$this->_mSomeOtherSoldier = $someOtherEnemy;  
  
}  
  
}
```

Wśród decyzji, które należy podjąć, jest to, czy używać cudzysłowów dla ciągów. JavaScript, HTML i PHP pozwalają na używanie zarówno pojedynczych cudzysłowów, jak i podwójnych cudzysłowów. W kodzie w tej książce użyjemy podwójnych cudzysłowów w kodzie HTML i JavaScript oraz pojedynczych cudzysłowów w PHP. Chociaż w przypadku JavaScript to kwestia gustu (możesz używać pojedynczych cudzysłowów, o ile używasz ich konsekwentnie), w PHP pojedyncze cudzysłowy są przetwarzane szybciej, są bezpieczniejsze i rzadziej powodują błędy programistyczne.

### **Podsumowanie**

Hej, omówiliśmy wiele rzeczy, prawda? Rozmawialiśmy o architekturze trójwarstwowej i o tym, jak pomaga ona tworzyć świetne, elastyczne i skalowalne aplikacje. Zobaczyliśmy również, jak każda z technologii wykorzystanych przez nas pasuje do architektury trójwarstwowej. Jeśli czujesz się przytłoczony, nie martw się. W następnej części zaczniemy tworzyć pierwszą część naszej strony. Wyjaśnimy każdy krok na bieżąco, dzięki czemu będziesz miał jasne zrozumienie każdego elementu aplikacji.