

Projektowanie oprogramowania

grupa poniedziałek 17:05

Maja Kędras

Mateusz Wesołowski

29 stycznia 2017

Spis treści

1 Wstęp	3
2 Uproszczony opis wycinka rzeczywistości	3
3 Wizja	3
4 Słownik	8
5 Model domenowy	8
6 Reguły biznesowe	9
7 Specyfikacja wymagań	10
8 Diagram przypadków użycia	12
9 Specyfikacja przypadków użycia	13
10 Model informacyjny	15
11 Prototyp interfejsu	15
12 Architektura systemu	16
12.1 Architektura logiczna	16
12.2 Architektura fizyczna	16
13 Model logiczny danych	17
14 Realizacja przypadku użycia	17
15 Implementacja interfejsu	19
16 Dokumentacja kodu	25
17 Testy jednostkowe	26
18 Przypadki testowe dla przypadków użycia	27
19 Automatyzacja testów funkcjonalnych.	28
20 Badanie jakości projektu	30
21 Wnioski	32

1 Wstęp

Celem kursu projektowanie oprogramowania było zapoznanie się z zadaniami realizowanymi w ramach podstawowych procesów cyklu życia oprogramowania, poznanie dobrych praktyk projektowych oraz wyrobienie umiejętności opracowania i przedstawiania w języku technicznym dokumentacji projektowej. W ramach ich realizacji zostało wybrane zagadnienie tworzenia internetowego biura podróży.

2 Uproszczony opis wycinka rzeczywistości

Pierwszym krokiem w stronie zaprojektowania systemu było utworzenie uproszczonego opisu wycinka rzeczywistości. Wyglądał on następująco:

Właściciele hoteli oraz przewoźnicy (Wsp), z którymi biuro podróży (BPod) nawiązało współpracę otrzymując unikalny identyfikator oraz hasło, dzięki którym będą mogli zalogować się do systemu poprzez aplikację webową. Zaraz po zalogowaniu się wymagana jest zmiana dostarczonego hasła. Kierownik biura podróży (Kbp) po zalogowaniu do systemu ma możliwość wygenerowania zapytania o dostępność usług współpracowników w podanym przez niego terminie. W zależności od typu usługi ma możliwość doprecyzowania warunków współpracy, przykładowo jakość hotelu, dostępność wyżywienia czy rodzaj transportu. Proces wygląda następująco:

- Kbp wysyła zapytanie o usługi z sprecyzowanymi szczegółami
- Zapytanie o usługę trafia do odpowiednich Wsp jeżeli spełniają warunki wyspecyfikowane przez Kbp
- Wsp odrzuca lub akceptuje zapytanie
- Kbp wybiera usługę spośród zaakceptowanych usług przez Wsp
- Kbp tworzy ofertę wycieczki na podstawie wybranych usług
- Wsp zostają poinformowani czy ich oferta została zaakceptowana
- Kbp uzupełnia szczegóły wycieczki

Pod koniec każdego miesiąca system podlicza kwotę jaką BPod jest winne Wsp, a księgowy jest zobowiązany uiścić opłatę do połowy następnego miesiąca. Oferty Last Minute są generowane automatycznie na podstawie zdefiniowanej przez Kbp liczby dni do wyjazdu oraz zadanej zniżki. Istnieje też możliwość ręcznego zmieniania typu oferty na Last Minute z wartościami innymi niż wpisane w systemie. Wszystkie dostępne oferty, są możliwe do przejrzenia zarówno przez system jak i w placówkach BPod. System oferuje klientowi filtrowanie i sortowanie ofert, aby wyszukiwanie stało się łatwiejsze. Kiedy klient jest zainteresowany wycieczką może ją zarezerwować od razu na stronie internetowej, po czym jest zobowiązany do uiszczenia zaliczki w wysokości 20% kwoty całkowitej, a następnie do 30 dni przed wyjazdem pozostałą część. W przypadku gdy oferta jest ofertą Last Minute, klient musi zapłacić całkowitą kwotę w ciągu 24h.

3 Wizja

Następnie przeszliśmy do wypełnienia dokumentu wizji.

Biuro podróży	
Wizja	Data: 16/Oct/16

Biuro podróży Wizja

1. Wprowadzenie

Produktem, który chcemy zaprojektować, a następnie zaimplementować będzie aplikacja internetowa wspierająca działalność biura podróży. Głównym celem jest zwiększenie liczby potencjalnych klientów, których zachęcić ma możliwość wyszukania i zakupu najlepszych dostępnych ofert bez wychodzenia z domu poprzez wyspecyfikowanie indywidualnych potrzeb. Aplikacja ta dodatkowo będzie wspomagać biuro dzięki zautomatyzowanej funkcji generowania ofert Last Minute oraz planowania porozumień z przewoźnikami i hotelami.

2. Pozycjonowanie

2.1 Sformułowanie problemu

Problem	Nie każdy klient ma czas żeby przyjechać do biura i osobiście ustalić szczegóły wyjazdu.
Dotyczy	Klient
Wpływ problemu	Mniej ludzi korzysta z usług biura
Pomyślne rozwiązanie	Umożliwienie przeglądania oferty oraz zakupu wyjazdu online

Problem	Placówki biura znajdują się tylko w większych miastach.
Dotyczy	Klient, właściciel biura
Wpływ problemu	Potencjalni klienci wybierają lokalne, często droższe oferty konkurencji.
Pomyślne rozwiązanie	Umożliwienie przeglądania oferty oraz zakupu wyjazdu online

Problem	Ręczne zmienianie ofert, które wkrótce przestaną być aktualne, na Last Minute zajmuje wiele czasu
Dotyczy	Właściciel biura
Wpływ problemu	Marnowanie czasu
Pomyślne rozwiązanie	Zaimplementowanie automatycznego zmieniania ofert na Last Minute przy zadanym czasie wygaśnięcia oferty

Problem	Ustalanie grafiku z hotelami i przewoźnikami zajmuje wiele czasu i często generuje błędy
Dotyczy	Właściciel biura, współpracownicy biura
Wpływ problemu	Marnowanie czasu, problemy organizacyjne pojawiające się przy niepoprawnym przepisaniu danych bądź problemach organizacyjnych, któregoś z współpracowników
Pomyślne rozwiązanie	Zaimplementowanie formularzy ułatwiających ustalenie

Biuro podróży	
Wizja	Data: 16/Oct/16

	wspólnego grafiku
--	-------------------

2.2 Opis pozycji produktu

Dla	Właściciela biura podróży
Który	Chce zautomatyzować część procesu sprzedaży wyjazdów
GoodTrip	Jest aplikacją webową do sprzedaży internetowej
Który	Poza samą sprzedażą obsługuje kontakt ze współpracownikami i generowanie ofert Last Minute
Inaczej niż	Aplikacje do sprzedaży internetowej ogólnego zastosowania
Nasz produkt	Obsługuje kontakt ze współpracownikami i generowanie ofert Last Minute oraz nie zajmuje pamięci na obsługę magazynu, która w tym przypadku nie jest potrzebna

3. Opis udziałowców i użytkowników

3.1 Podsumowanie udziałowców

Nazwa	Opis	Odpowiedzialności
Biuro podróży Trampik	Biuro podróży, które jest zainteresowane zakupem aplikacji wspomagającej sprzedaż i organizację	Decyduje o rozwoju funkcjonalności Zapewnia, że napisane oprogramowanie będzie użyteczne dla innych biur Finansuje projekt

3.2 Podsumowanie użytkowników

Nazwa	Opis	Odpowiedzialności
Kierownik biura	Osoba zajmująca się tworzeniem ofert oraz porozumieniami z przewoźnikami i hotelami	Dostarcza informacje o procesach zachodzących podczas wykonywania jego obowiązków
Współpracownik biura	Osoba dostarczająca usługi dla biura np. przeloty samolotem czy noclegi w hotelu	Dostarcza informacje o procesach zachodzących podczas wykonywania jego obowiązków
Właściciel biura	Osoba posiadająca biuro podróży	Akceptuje plan finansowania projektu

Biuro podróży	
Wizja	Data: 16/Oct/16

4. Opis produktu

4.1 Potrzeby i cechy

Potrzeba	Priorytet	Cechy	Planowane wydanie
Publikowanie oferty wyjazdów online	M	Szybkie, dokładne, szczegółowe, możliwość dodawania ofert do bazy danych	1.0
Przeglądanie oferty wyjazdów online	M	Czytelne, intuicyjne, możliwość odczytywania ofert z bazy danych	1.0
Dokonywanie zakupu wyjazdu	S	Intuicyjne, szybkie, łatwe, bezpieczne, możliwość płatności online, automatycznie zaksięgowanie wpłaty	1.0
Zapytanie o ofertę współpracowników	M	Możliwość wysłania zapytania o ofertę usług w danym terminie w postaci formularza Filtrowanie odbiorców oferty	1.0
Odrzucenie lub akceptacja oferty biura przez współpracowników	M	Intuicyjne, szybkie, łatwe, generowanie formularza odpowiedzi	1.0
Tworzenie ofert wycieczki	S	Klarowne, intuicyjne, wybór usługi współpracownika spośród zaakceptowanych ofert, możliwość dodania szczegółów na temat oferty wycieczki	1.0
Generowanie ofert Last Minute	S	Łatwe, dokładne, niezawodne, szybkie, automatyczne zmniejszanie cen ofert, które kończą się za zadany okres czasu	1.0
Rejestracja kont klientów	C	Szybkie, łatwe, możliwość założenia konta i zalogowania się do systemu	2.0
Program lojalnościowy dla klientów	C	Automatycznie przydzielanie punktów lojalnościowych, generowanie informacji o przydzielonych zniżkach	2.0

4.2 Inne wymagania produktowe

Biuro podróży	
Wizja	Data: 16/Oct/16

Wymaganie	Priorytet	Planowane wydanie
Serwer	M	1.0
Komputery z systemem operacyjnym Windows 7 lub nowszym	M	1.0
Dostęp do internetu 10Mbit/s lub wyższa	M	1.0
Przeglądarka internetowa Google Chrome 53.0 lub nowsza	S	1.0
Urządzenia peryferyjne komputera	S	1.0
Szkolenie	C	1.0
Instrukcja użytkownika	C	1.0

W

Największym błędem jaki został popełniony w tym etapie było dopisanie planów na drugą wersję oprogramowania. Naszą motywacją było pokazanie, że myślimy przyszłościowo na temat projektu jednak podczas dalszych czynności projektowych niepotrzebne cechy produktu zaczęły być problematyczne, gdyż musiały być wszędzie uwzględnianie, a ich implementacja czy bardziej szczegółowe zaprojektowanie nigdy nie było naszym celem.

4 Słownik

Kolejnym krokiem w uporządkowaniu naszych pomysłów na temat planowanego systemu było stworzenie słownika, który miał zapobiec wszelkim nieporozumieniom związanym z różnym rozumieniem danych pojęć.

Najwięcej problemów sprawił fakt, że w kontekście biura podróży słowa usługa, oferta i wyjazd są często używane zamiennie, a w kontekście naszego systemu oznaczają różne pojęcia.

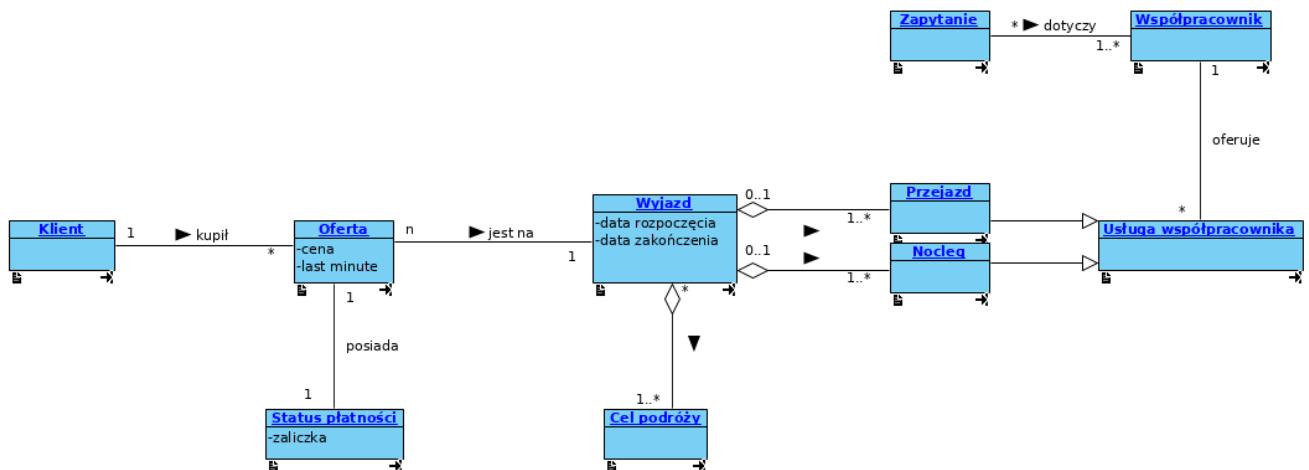
Podczas dalszych etapów projektowania okazał się on również zbyt lakoniczny i wybrakowany o niektóre pojęcia.

Name	...	Description
Kierownik biura	...	Osoba zajmująca się kontaktem ze współpracownikami oraz zarządzaniem całym biurem
Klient	...	Osoba nabywająca wyjazd w biurze
Wyjazd		Usługa oferowana przez biuro zawierająca w sobie dojazd do celu podróży, noclegi oraz opcjonalnie wyżywienie i/lub dodatkowe atrakcje
Przejazd		Usługa oferowana przez współpracowników polegająca na dostarczeniu klientów z punktu zbiórki do celu podróży
Współpracownik		Osoba lub instytucja współpracująca z biurem, przykładowo hotel, przewoźnik czy linia samolotowa
Oferta		Usługa zakupu wyjazdu, która może zmieniać swoją cenę w przypadku stania się ofertą Last Minute
Nocleg		Usługa oferowana przez współpracownika polegająca na udostępnieniu miejsc noclegowych
Cel podróży		Miejsce, do którego chcą pojechać klienci
Harmonogram biura		Harmonogram zawierający wszystkie zaplanowane wyjazdy
Oferta Last Minute		Specjalna oferta, o obniżonej cenie, stworzona przez generator LM na podstawie zmiennych podanych przez kierownika biura
Generator LM		Skrypt odpowiedzialny za tworzenie ofert Last Minute
Zapytanie		Formularz z propozycją współpracy zawierający szczegóły potrzebnej usługi współpracownika
Usługa współpracownika		Usługa oferowana przez zarejestrowanego współpracownika w postaci noclegu lub przejazdu
Katalog		Zbiór dostępnych ofert
Status płatności		Informacja o tym kiedy, czy i kiedy klient zapłacił

5 Model domenowy

W końcu przystąpiliśmy do zaprojektowania naszego pierwszego diagramu jakim był model domenowy.

Nie sprawił on większych problemów.



6 Reguły biznesowe

W ramach projektu zostały utworzone następujące reguły biznesowe:

1. Ograniczenia struktury

- Każdy wyjazd musi mieć przypisany przynajmniej jeden przejazd
- Każdy wyjazd musi mieć przypisany przynajmniej jeden nocleg
- Każdy wyjazd musi mieć przypisany przynajmniej jeden cel podróży
- Każdy wyjazd musi mieć ustaloną datę rozpoczęcia oraz zakończenia
- Z jednym wyjazdem może być powiązane tylko tyle ofert ile jest miejsc na wyjeździe
- Każda oferta musi być powiązana z jednym wyjazdem
- Jedna oferta może być zakupiona przez co najwyżej jedną osobą
- Jedna osoba może zakupić wiele ofert

2. Ograniczenia operacji

- Aby dokonać rezerwacji zwykłej oferty należy opłacić zaliczkę
- Wyjazd musi być w pełni opłacony przez klienta nie później niż 30 dni przed datą wyjazdu jeżeli oferta nie była typu Last Minute
- W przypadku ofert Last Minute należy przelać całą kwotę od razu
- Zaliczka nie podlega zwrotowi w momencie rezygnacji z wyjazdu
- Współpracownik musi aktualizować swój harmonogram przy każdej jego zmianie
- Koszt wyjazdu nie jest zwracany w przypadku rezygnacji 3 dni przed terminem wyjazdu lub później

3. Wnioski

- Osoba fikcyjna musi być uznana za klienta w momencie dokonania rezerwacji
- Wyjazd musi być wyświetlany w katalogu jeżeli są jeszcze wolne oferty
- Oferta nie może być wyświetlany w katalogu jeżeli w jej statusie płatności jest opłacona zaliczka
- Jeżeli kwota całkowita nie zostanie opłacona w odpowiednim terminie oferta ponownie trafia do katalogu

4. Obliczenia

- Zaliczka musi być obliczona jako 20% kwoty całkowitej
- Cena oferty Last Minute musi być obliczona jako procent zadany przez kierownika kwoty całkowitej

5. Wyzwalacze

- Kierownik biura może zdefiniować w jakim momencie oferty stają się ofertami typu Last Minute i jaką zniżkę to gwarantuje
- Oferty z zapłaconą zaliczką nie podlegają zmianie na Last Minute
- Jeżeli do ostatecznego terminu zapłaty zostały dwa tygodnie, tydzień lub jeden dzień system musi wysłać maila z przypomnieniem

Dużym i z perspektywy całego projektu mocno rzucającym się w oczy błędem jest brak jakichkolwiek reguł biznesowych związanych z zapytaniami wymienianymi pomiędzy kierownikiem, a współpracownikami. Okazało się to bardzo problematyczne, gdyż oba przypadki użycia, którymi się zajmowaliśmy były związane właśnie z zapytaniami.

7 Specyfikacja wymagań

Specyfikację wymagań zapisaliśmy w postaci historyjek użytkownika, które wyglądają następująco:

- Właściciel biura

1. Chcę mieć możliwość zalogowania się do systemu aby mieć dostęp do wszystkich funkcji systemu.
2. Chcę mieć możliwość dodawania nowych ofert do bazy danych aby klienci mogli je przeglądać.
3. Chcę mieć możliwość wysyłania zapytań do współpracowników aby dowiedzieć się kto może zaofrować mi daną usługę w podanym terminie..
4. Chcę mieć możliwość odbierania odpowiedzi na zapytania od współpracowników aby wiedzieć czy mogę wpisać ich usługi do wyjazdu.
5. Chcę mieć możliwość ręcznego generowania ofert Last Minute aby móc wybierać oferty, które nie spełniają globalnych warunków bycia ofertami LM.
6. Chcę mieć możliwość automatycznego generowania ofert Last Minute aby nie musieć robić tego ręcznie.

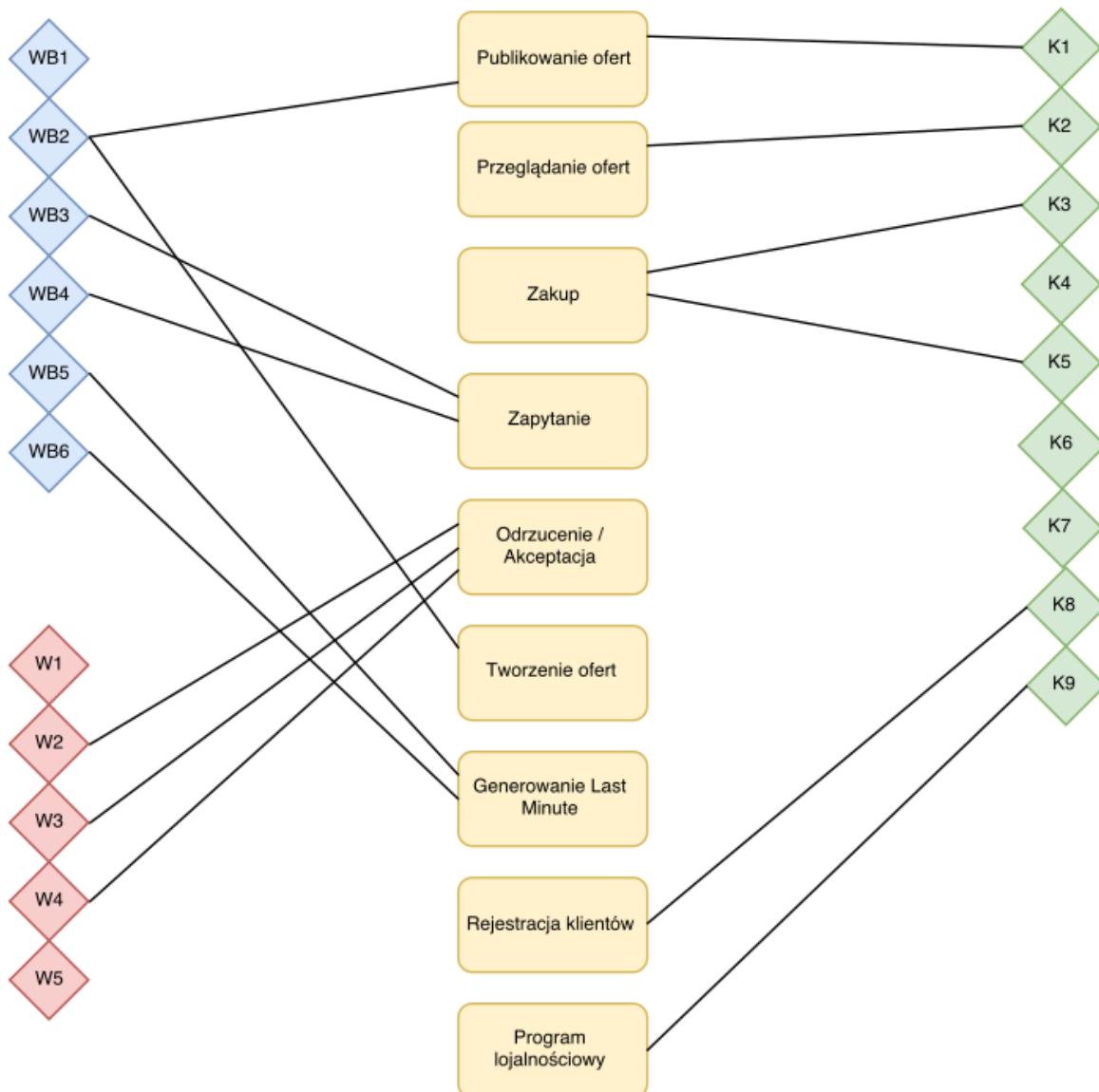
- Współpracownik

1. Chcę mieć możliwość zalogowania się do systemu aby mieć dostęp do przypisanych mi przez właściciela biura funkcji systemu.
2. Chcę mieć możliwość odbierania zapytań od właściciela biura aby móc zaproponować mu swoje usługi.
3. Chcę mieć możliwość akceptacji zapytania właściciela biura aby poinformować go o mojej dyspozycyjności.
4. Chcę mieć możliwość odrzucenia zapytania właściciela biura aby poinformować go o mojej dyspozycyjności.
5. Chcę mieć możliwość powiadomienia właściciela biura o anulowaniu usługi w przypadku wydarzeń losowych aby nie musieć do niego dzwonić.

- Klient

1. Chcę mieć możliwość przeglądania oferty wyjazdów online aby nie musieć jechać do biura podróży.
2. Chcę mieć możliwość filtrowania interesujących mnie ofert aby nie musieć przeglądać tych, które mnie nie interesują.
3. Chcę mieć możliwość rezerwacji wyjazdów online aby mieć pewność, że nikt nie wykupi oferty, którą jestem zainteresowany.
4. Chcę mieć możliwość rezygnacji z wyjazdu online żeby móc odzyskać zaliczkę.
5. Chcę mieć możliwość płatności online za wyjazd żeby nie musieć jechać do biura żeby zapłacić.
6. Chcę otrzymywać email z potwierdzeniem rezerwacji i szczegółami wyjazdu aby mieć wszystkie informacje w jednym miejscu.
7. Chcę otrzymywać emaile z przypomnieniem o upływającym terminie zapłaty za wyjazd aby nie zapomnieć.
8. Chcę mieć możliwość rejestracji do systemu aby móc korzystać z programów lojalnościowych.
9. Chcę mieć możliwość korzystania z programów lojalnościowych aby mieć lepsze oferty od zwykłych klientów.

W ramach sprawdzenia śladowalności do wizji utworzyliśmy następujący diagram:

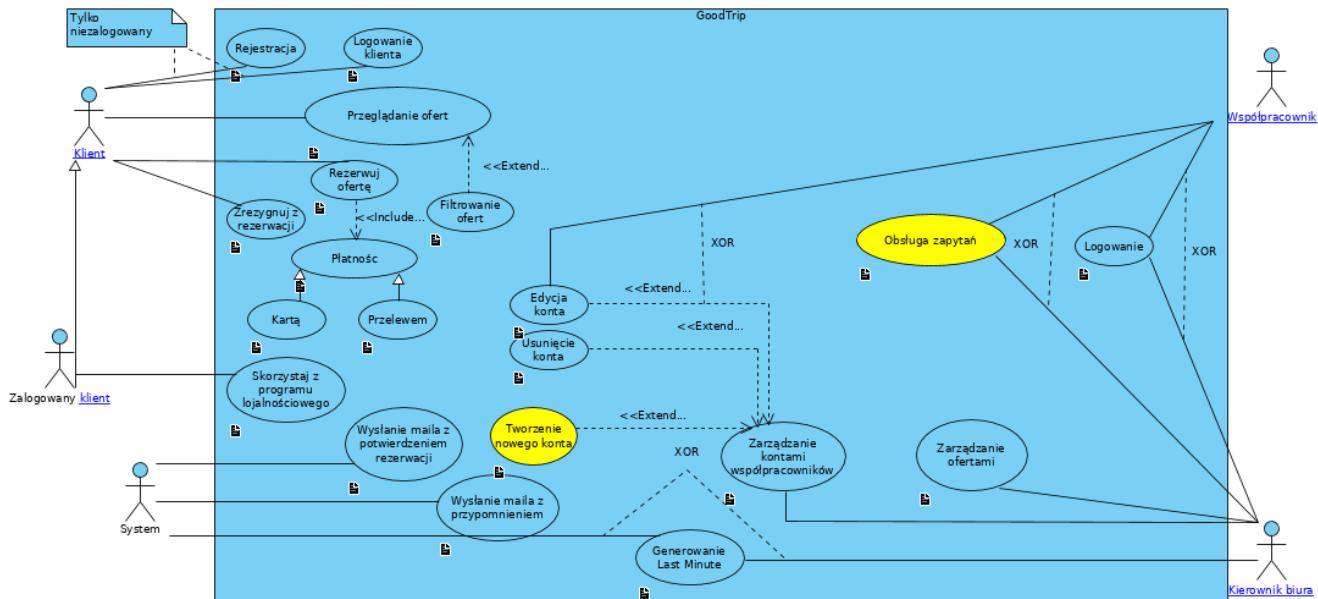


Pośrodku wypisane są potrzeby, a po bokach historyjki użytkownika (odpowiednio WB - właściciel biura, W - współpracownik i K - klient)

Jak widać pojawiło się więcej historyjek użytkownika niż było zdefiniowanych potrzeb. Wynika to z tego, że im dłużej pracowaliśmy nad tym projektem tym więcej dostrzegaliśmy potencjalnych zastosowań. Dalej też ciągną się tu za nami niepotrzebnie zdefiniowane potrzeby powiązane z kolejną wersją systemu.

8 Diagram przypadków użycia

Po ustaleniu czego potencjalni użytkownicy będą wymagali od systemu nadeszła pora na utworzenie diagramu przypadków użycia. Wygląda on następująco:



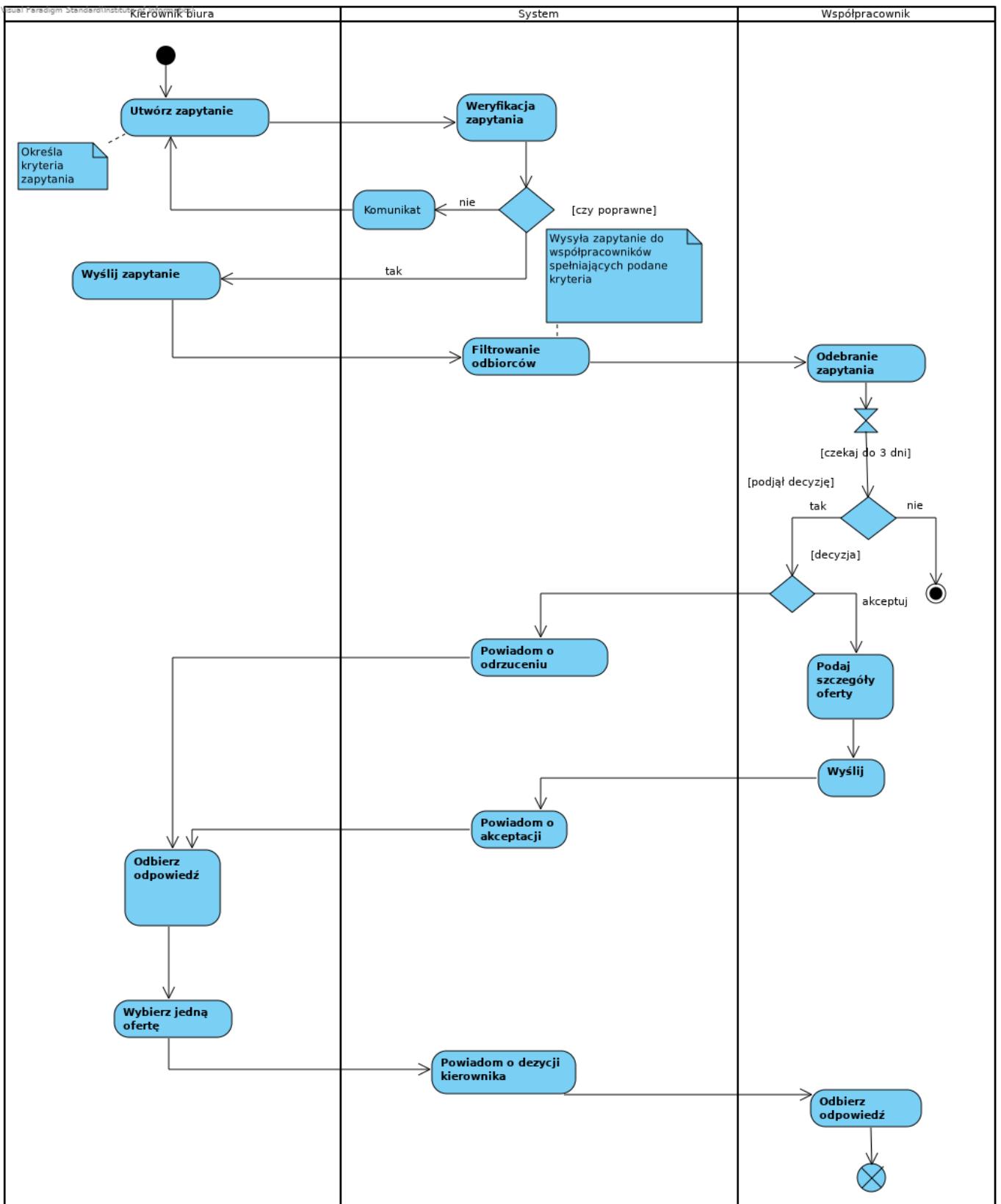
Z punktu widzenia osób, które zaimplementowały dwa z przedstawionych przypadków użycia jesteśmy w stanie dostrzec następujące błędy i niedociągnięcia:

- Współpracownik i kierownik biura dzielą przypadek użycia nazwany obsługą zapytań. O ile faktycznie obydwie mają dostęp do takowej funkcjonalności to wygląda ona dla nich zupełnie inaczej, dlatego powinien być on rozdzielony na dwa osobne przypadki użycia.
- Dodatkowo już sam przypadek użycia o nazwie obsługa zapytań jest zbyt ogólny, nawet gdyby dotyczyły jednego typu użytkownika. Należało go rozbić, podobnie jak obsługę kont współpracowników, na tworzenie, edycje i usuwanie, a do tego wszystkie przypadki związane z ich akceptacją i odrzucaniem.
- Nasze przypadki użycia nie uwzględniają możliwości edycji i usuwania ofert. Obie te funkcjonalności zostały podpięte pod lakoniczne edycja konta, co nie było dobrym rozwiązaniem.
- Ponownie pojawiają się czynności związane z potrzebami zdefiniowanymi dla drugiego wydania systemu, które nic nie dają, a tylko zaczerniają diagram.

9 Specyfikacja przypadków użycia

Po wybraniu dwóch przypadków użycia, na których chcieliśmy się dalej skupić stworzyliśmy dla nich specyfikację. Została ona utworzona na dwa sposoby, jedna w wersji tekstopowej, a druga w postaci diagramu aktywności.

Diagram aktywności wyglądał następująco:



Wersja tekstowa przedstawia się za to następująco:

Aktorzy: Kierownik biura

Warunki wstępne:

- Kierownik musi być zalogowany

Warunki wykonania:

- Nie istnieje w bazie pracownik o podanym telefonie i adresie e-mail
- Spełnienie wszystkich ograniczeń

Warunek końcowy:

- Utworzenie nowego konta

Nr	Kierownik biura	System
1	Wybiera opcję utworzenia nowego konta dla współpracownika	
2		Wyświetla formularz do utworzenia nowego konta
3	Wprowadza dane na temat współpracownika	
4		Weryfikuje dane
5	Zatwierdza dane	
6		Tworzy nowe konto
7		Wyświetla stronę dodawania usług do właśnie utworzonego konta
8	Wybiera typ usługi	
9		Wyświetla formularz dla usługi o wybranym typie
10	Wprowadza dane na temat usługi	
11		Weryfikuje dane
12	Zatwierdza dane	
13		Tworzy nową usługę przypisaną do konta
14	Kończy proces tworzenia konta klienta	

3a Dane niepoprawne →3

3b Dane poprawne

3b1 Rezygnuje →14

3b2 Zapisuje formularz

3b2a Telefon i/lub e-mail istnieją już w bazie danych →3

3b2b Poprawne dane →3

10. Wprowadza dane na temat usługi współpracownika.

10a Dane niepoprawne →8

10b Dane poprawne

10b1 Rezygnuje →14

10b2 Zapisuje formularz

10b2a Chce dodać nową usługę →8

10b2b Kończy proces dodawania →14

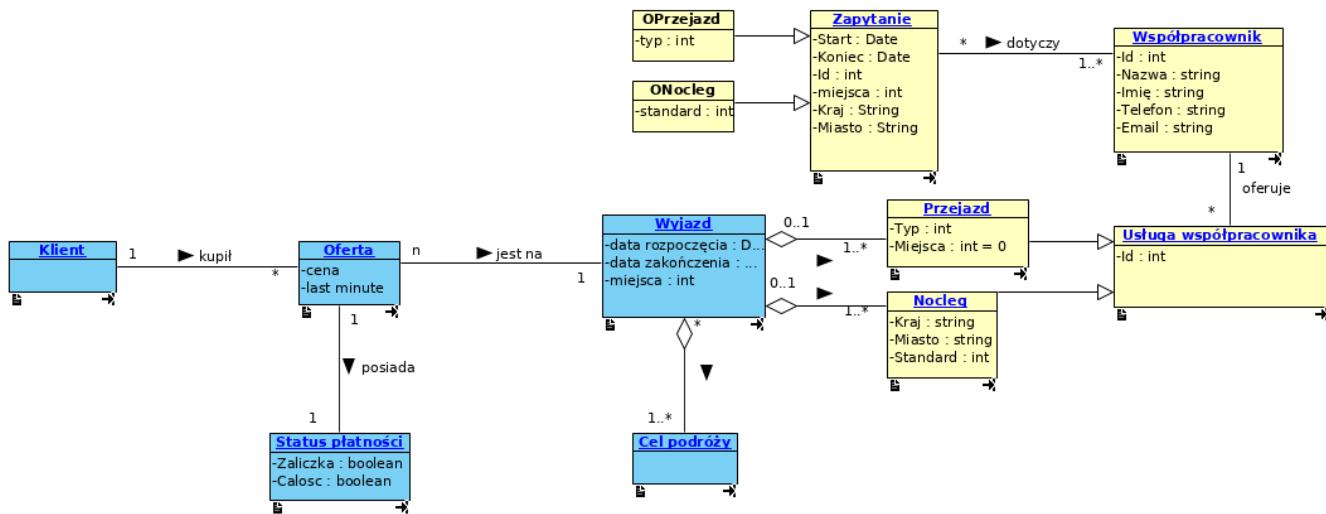
6. Koniec

Oczywistym błędem był tu oczywiście brak opisania jakichkolwiek warunków w przypadku diagramu aktywności.

Porównując oba rodzaje zapisu stwierdziliśmy, że diagram aktywności jest dużo czytelniejszy jeżeli chodzi o ścieżki alternatywne, a w wersji tekstopowej łatwiej dojrzeć ścieżkę główną.

10 Model informacyjny

W kolejnym kroku uszczegółowiliśmy model domenowy tworząc model informacyjny. Wyglądał on następująco:



W ramach uszczegóławiania modelu domenowego dodaliśmy dwa rodzaje zapytań, dziedziczące po ogólnym zapytaniu, co bardzo ułatwiło późniejszą implementację.

11 Prototyp interfejsu

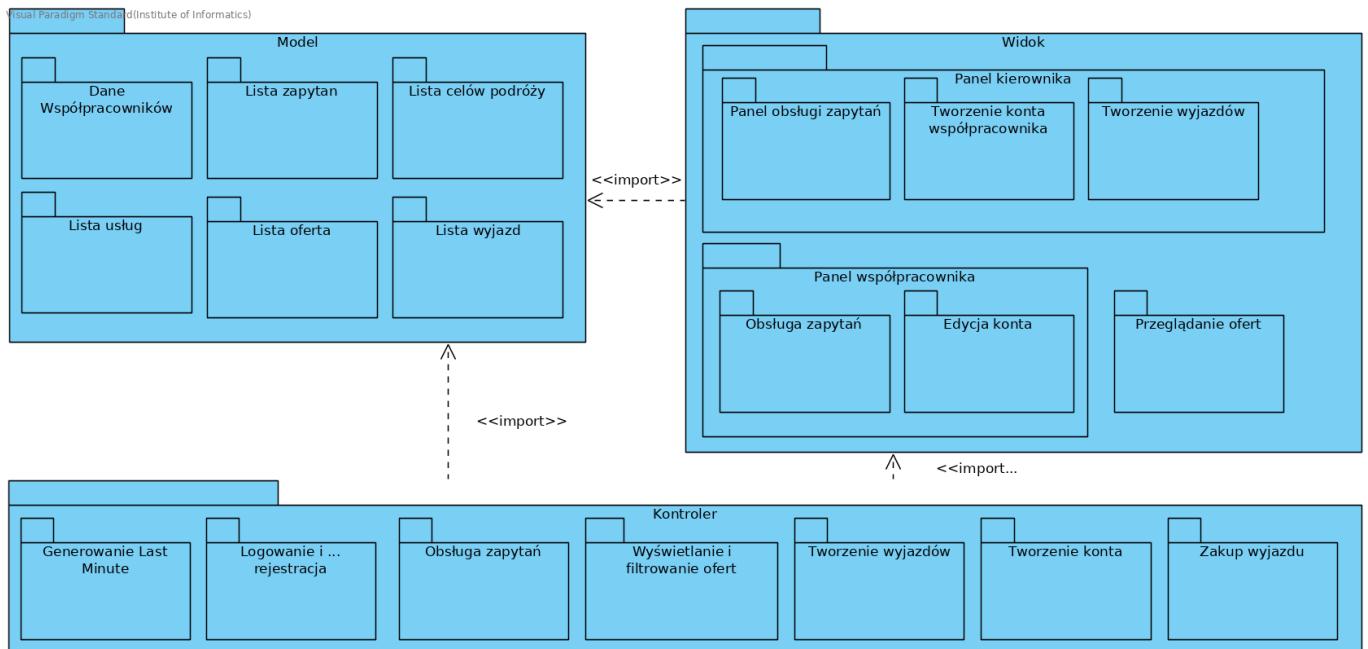
Następnie zaprojektowaliśmy interfejs aplikacji. Jego graficzna reprezentacja zostanie przedstawiona przy porównaniu go z końcową implementacją.

Prototyp interfejsu był pierwszym z kroków, który mocno zaczął weryfikować nasze poprzednie założenia. Jest to drugi, po etapie implementacji, etap podczas którego wykryliśmy najwięcej błędów i nieścisłości w naszym projekcie. Wynika to prawdopodobnie z tego, że nie jesteśmy przyzwyczajeni do stawiania się w pozycji analityka, więc nie byliśmy w stanie dostrzec pewnych rzeczy z jego perspektywy. Za to jesteśmy przyzwyczajeni do patrzenia na projekt z punktu widzenia programisty, a przy projektowaniu interfejsu można było się do tego punktu widzenia zbliżyć.

12 Architektura systemu

12.1 Architektura logiczna

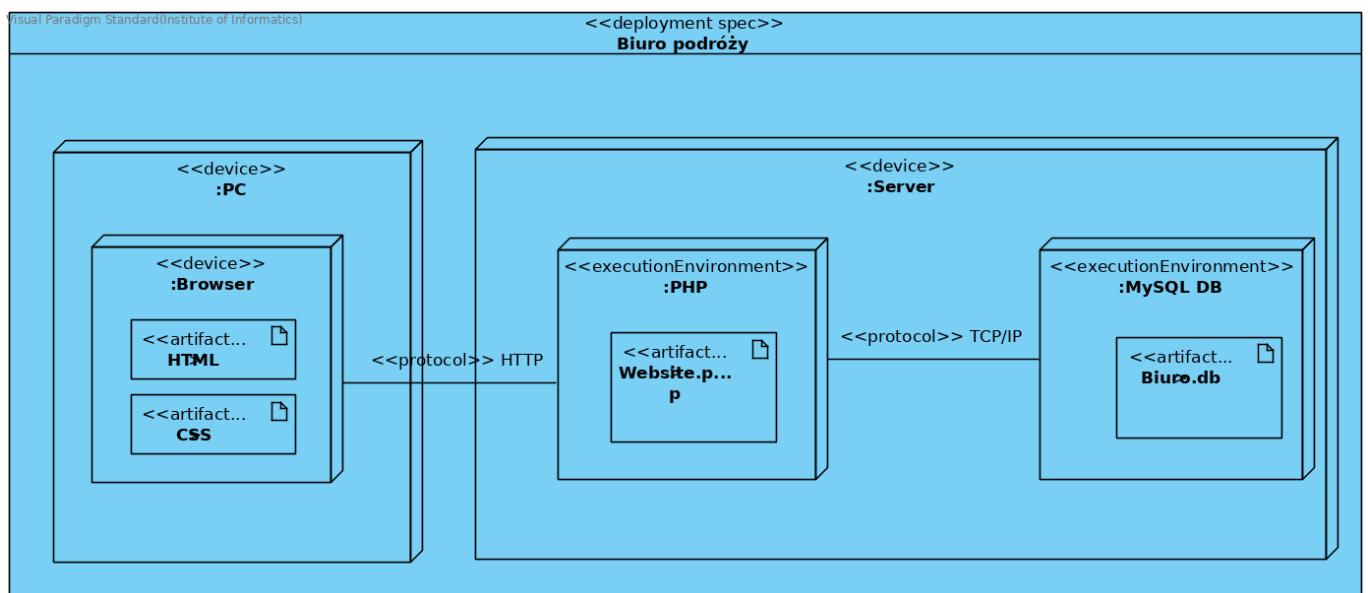
Jako wzorzec architektury logicznej wybraliśmy wzorzec MVC (Model, View, Controller). Podział naszego systemu prezentuje poniższy diagram pakietów.



Największym problemem z tym diagramem jest fakt, że nie pokrywa wszystkich przypadków użycia. Przykładowo brakuje tutaj widoku związanego z logowaniem.

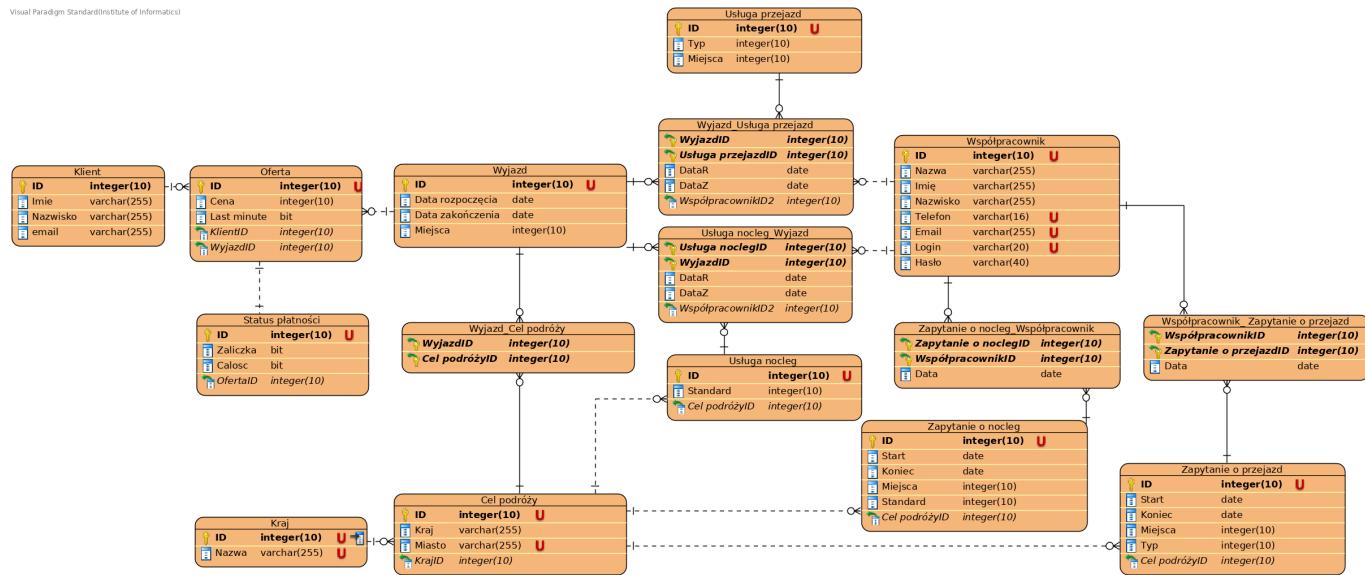
12.2 Architektura fizyczna

W ramach architektury fizycznej zdecydowaliśmy się na architekturę Klient-Serwer. Podział naszego systemu prezentuje poniższy diagram rozmieszczenia.



13 Model logiczny danych

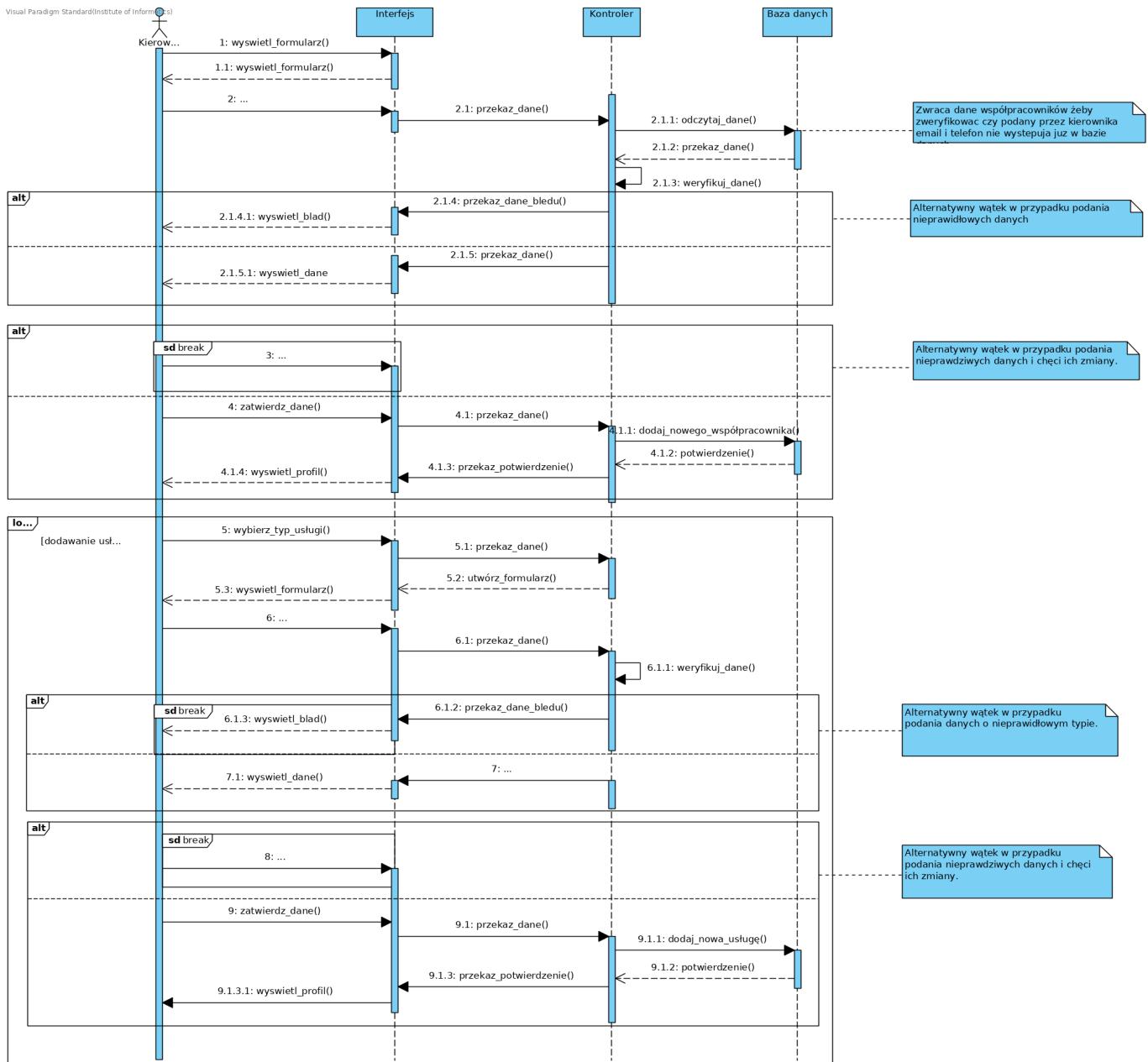
Następnym krokiem było utworzenie logicznego modelu danych. Nasz przedstawia się następująco:

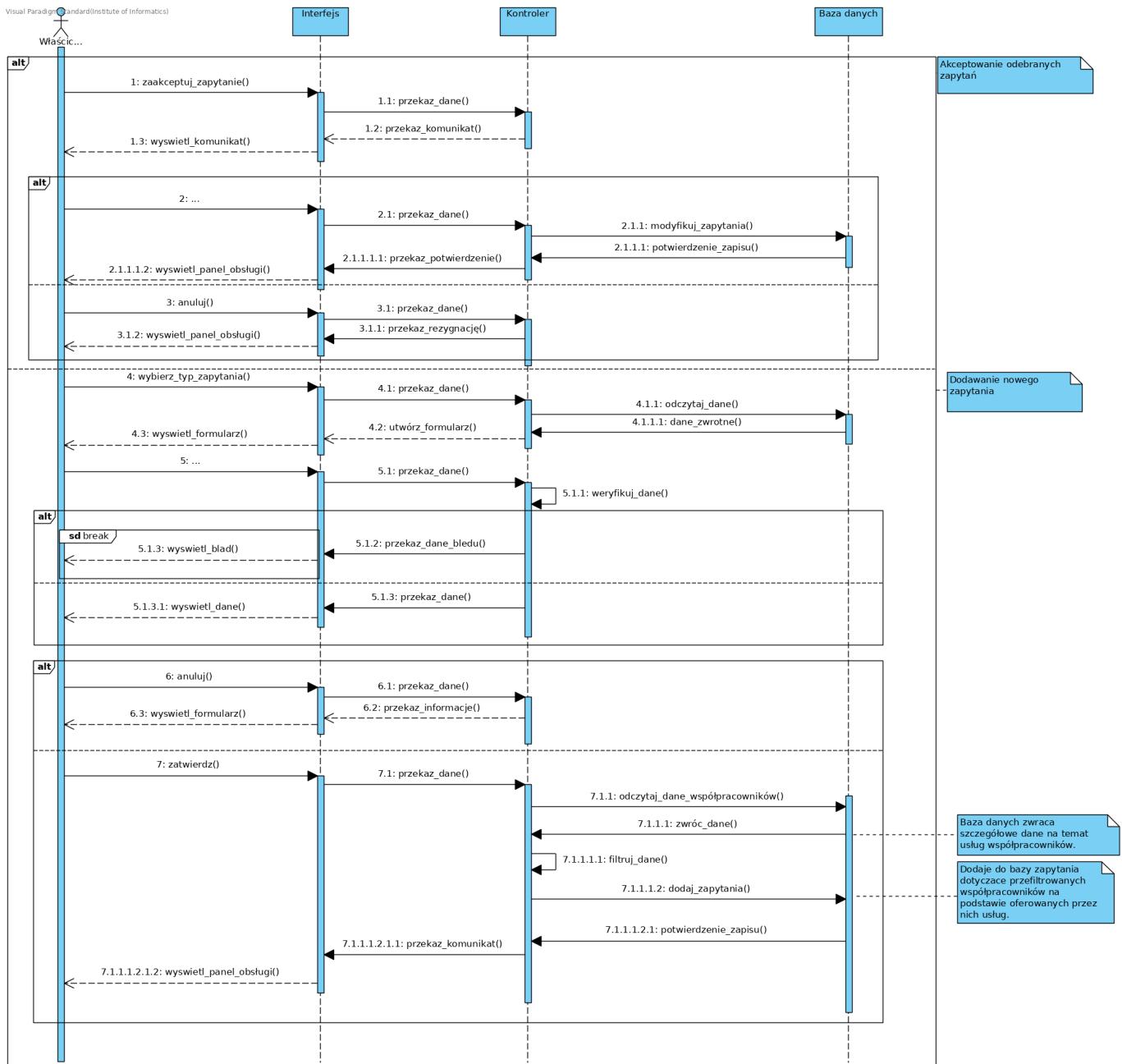


Został on wygenerowany automatycznie na podstawie modelu informacyjnego, a następnie nanieśliśmy na niego poprawki. Możliwość ta zdawała się z początku praktyczna i wygodna jednak poskutkowało to mnóstwem błędów, więc gdybyśmy mieli robić to ponownie zrobilibyśmy to ręcznie. Największym błędem, który sprawił nam dość duże problemy podczas procesu implementacji, jest brak bezpośredniego połączenia tablicy współpracownik z tablicami usług.

14 Realizacja przypadku użycia

Ponownie skupiliśmy się na dwóch, wybranych wcześniej przypadkach użycia, tym razem tworząc dla nich diagramy sekwencji.





Po raz kolejny naszym błędem była zbytnia ogólność. Przykładowo każda z funkcji przekaz_dane() jest odpowiedzialna za inny rodzaj danych, więc powinny mieć co najmniej różne nazwy. Ponownie pojawia się też problem zbyt ogólnych przypadków użycia, co powoduje zaciemnianie kolejnych diagramów.

15 Implementacja interfejsu

Etapem rozpoczynającym proces implementacji była implementacja interfejsu. Poniżej przedstawiamy porównanie prototypu interfejsu oraz jego końcowej implementacji

Przypadek użycia: Dodawanie Współpracownika

Lista współpracowników

Nazwa firmy
Hotel Jana Pawła II
Polonia Wrocław
Qubus Hotel Wrocław
Radisson SAS Wrocław
Sofitel Wrocław

Dodawanie nowego współpracownika

Nazwa firmy	<input type="text" value="Hotel Marriott"/>
Imię	<input type="text" value="Barbara"/>
Nazwisko	<input type="text" value="Grajewska"/>
Telefon	<input type="text" value="665432678"/>
Email	<input type="text" value="b.grajewska@gmail.com"/>

Dodawanie nowego współpracownika

Nazwa firmy	<input type="text" value="Hotel Marriott"/>
Imię	<input type="text" value="Barbara"/>
Nazwisko	<input type="text" value="Grajewska"/>
Telefon	<input type="text" value="665432678"/>
Email	<input type="text" value="b.grajewska@gmail.com"/>

Podano błędne dane
 Numer telefonu jest niepoprawny.

Dane nowego współpracownika

Nazwa firmy	<input type="text" value="██████████"/>
Imię	<input type="text" value="██████████"/>
Nazwisko	<input type="text" value="██████████"/>
Telefon	<input type="text" value="██████████"/>
Email	<input type="text" value="██████████"/>

Zatwierdź poprawność danych

Nazwa Współpracownika

Nazwa	<input type="text" value="██████████"/>
Imię	<input type="text" value="██████████"/>
Nazwisko	<input type="text" value="██████████"/>
Telefon	<input type="text" value="██████████"/>
Email	<input type="text" value="██████████"/>

Uslugi współpracownika:

Nazwa usługi	Typ usługi
██████████	██████████
██████████	██████████

Lista współpracowników

Nazwa firmy
LOT
RYANAIR
POLSKI BUS
Marriott
Tipton
Ibs
Polonia

Dodawanie nowego współpracownika

Nazwa firmy	<input type="text" value="Hotel Wrocław"/>
Imię	<input type="text" value="Mariusz"/>
Nazwisko	<input type="text" value="Hotelski"/>
Telefon	<input type="text" value="666666666"/>
Email	<input type="text" value="m.hotelski@op.pl"/>

Dodawanie nowego współpracownika

Nazwa firmy	<input type="text" value="Hotel Wrocław"/>
Imię	<input type="text" value="Mariusz"/>
Nazwisko	<input type="text" value="Hotelski"/>
Telefon	<input type="text" value="666666666"/>
Email	<input type="text" value="m.hotelski@op.pl"/>

Podany numer telefonu jest już w bazie danych, proszę podać inną.

Dane nowego współpracownika

Nazwa firmy	<input type="text" value="Hotel Wrocław"/>
Imię	<input type="text" value="Mariusz"/>
Nazwisko	<input type="text" value="Hotelski"/>
Telefon	<input type="text" value="666666665"/>
Email	<input type="text" value="m.hotelski@op.pl"/>

Zatwierdź poprawność danych

Hotel Wrocław

Dane osoby kontaktowej

Imię	<input type="text" value="Mariusz"/>
Nazwisko	<input type="text" value="Hotelski"/>
Telefon	<input type="text" value="666666665"/>
Email	<input type="text" value="m.hotelski@op.pl"/>

Uslugi współpracownika

Nazwa usługi	Typ usługi
██████████	██████████
██████████	██████████

Nazwa Współpracownika

Nazwa	[REDACTED]
Imię	[REDACTED]
Nazwisko	[REDACTED]
Telefon	[REDACTED]
email	[REDACTED]

Uslugi współpracownika:

Nazwa usługi	Typ usługi
[REDACTED]	[REDACTED]

Nazwa Współpracownika

Nazwa	[REDACTED]
Imię	[REDACTED]
Nazwisko	[REDACTED]
Telefon	[REDACTED]
email	[REDACTED]

Uslugi

Nozwo

Dodawanie usługi transportowej

Typ środka lokomocji	Samolot
Dostępna ilość miejsc	70

Nazwa Współpracownika

Nazwa	[REDACTED]
Imię	[REDACTED]
Nazwisko	[REDACTED]
Telefon	[REDACTED]
email	[REDACTED]

Uslugi

Nozwo

Dodawanie usługi transportowej

Błd!: [REDACTED]

Typ środka lokomocji	Samolot
Dostępna ilość miejsc	70

Nazwa Współpracownika

Nazwa	[REDACTED]
Imię	[REDACTED]
Nazwisko	[REDACTED]
Telefon	[REDACTED]
email	[REDACTED]

Uslugi

Nozwo

Dodawanie usługi noclegowej

Kraj	Francja
Miasto	Marsylia
Standard	★★★☆☆

Hotel Wrocław

Dane osoby kontaktowej

Imię Mariusz
Nazwisko Hotelski
Telefon 666666665
Email m.hotelksi@op.pl

Uslugi współpracownika

Nazwa usługi	Typ usługi
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]

Hotel Wrocław

Dodawanie przejazdu

Dane osoby kontaktowej

Imię Mariusz
Nazwisko Hotelski
Telefon 666666665
Email m.hotelksi@op.pl

Uslugi współpracownika

Nazwa usługi	Typ usługi
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]

Hotel Wrocław

Dodawanie przejazdu

Dane osoby kontaktowej

Imię Mariusz
Nazwisko Hotelski
Telefon 666666665
Email m.hotelksi@op.pl

Uslugi współpracownika

Nazwa usługi	Typ usługi
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]

Hotel Wrocław

Dodawanie usługi noclegowej

Dane osoby kontaktowej

Imię Mariusz
Nazwisko Hotelski
Telefon 666666665
Email m.hotelksi@op.pl

Uslugi współpracownika

Nazwa usługi	Typ usługi
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]

Przypadek użycia: Obsługa zapytań

Panel obsługi zapytań o usługi

Odpowiedzi na zapytania

Zapytanie o nocleg w Francja - Paryż, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

Tipton	0 złotych	Oczekiwane na odpowiedź
Ibis	14500 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 5

Mariot	5000 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

[Stwórz nowe zapytanie](#)

Powrót

Panel obsługi zapytań

Odpowiedzi na zapytania

Zapytanie o nocleg w Francja - Paryż, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

Tipton	0 złotych	Oczekiwane na odpowiedź
Ibis	14500 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 5

Mariot	5000 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

[Stwórz nowe zapytanie](#)

Panel obsługi zapytań o usługi

Odpowiedzi na zapytania

Zapytanie o nocleg w Francja - Paryż, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

Tipton	0 złotych	Oczekiwane na odpowiedź
Ibis	14500 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 5

Mariot	5000 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

[Stwórz nowe zapytanie](#)

Powrót

Panel obsługi zapytań

Odpowiedzi na zapytania

Zapytanie o nocleg w Francja - Paryż, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

Tipton	0 złotych	Oczekiwane na odpowiedź
Ibis	14500 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 5

Mariot	5000 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

[Stwórz nowe zapytanie](#)

Panel obsługi zapytań o usługi

Odpowiedzi na zapytania

Zapytanie o nocleg w Francja - Paryż, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

Tipton	0 złotych	Oczekiwane na odpowiedź
Ibis	14500 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 5

Mariot	5000 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

[Stwórz nowe zapytanie](#)

Powrót

Panel obsługi zapytań

Odpowiedzi na zapytania

Zapytanie o nocleg w Francja - Paryż, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

Tipton	0 złotych	Oczekiwane na odpowiedź
Ibis	14500 złotych	Oferta wybrana
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 5

Mariot	5000 złotych	Zakceptuj
Ibis	0 złotych	Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

[Stwórz nowe zapytanie](#)

Panel obsługi zapytań o usługę		Odpowiedzi na zapytania	
Zapytanie o		<p>Zaakceptuj odrzucona Zaakceptuj</p>	
Zapytanie o		<p>oczekiwanie na odpowiedz</p>	
<p>oczekiwanie na odp</p>		<p>o usłudze transportowej o usłudze kolejowej</p>	
<p>Stwórz nowe zapytanie</p>		<p>odrzucona</p>	

Tworzenie nowego zapytania o ofertę noclegową	
Kraj	<input type="text" value="Francja"/> ▼
Miasto	<input type="text" value="Paryż"/> ▼
Standard	
Liczba miejsc	<input type="text" value="30"/>
Data początkowa	
Data końcowa	
<input type="button" value="Anuluj"/>	
<input type="button" value="Wyślij"/> Alt	

Tworzenie nowego zapytania o ofertę noclegową

Kraj	Francja
Wprowadzone dane:	
Kraj	Francja
Miasto	Paryż
Standard	2-gwiazdkowy
Liczba miejsc	30
Data początkowa	20-04-2017
Data końcowa	25-04-2017
Anuluj	Wyślij zapytanie

Tworzenie nowego zapytania o ofertę noclegową

Kraj	Francja						
Miasto	Paryż						
Standard							
Lokalizacja	Komunikat błędu Wystąpił następujący błąd						
Daty	<table border="1"> <tr> <td>←</td> <td>→</td> </tr> <tr> <td>S 6 13 20 27</td> <td>← 21 22 23 24 25 26 28 29 → 30</td> </tr> <tr> <td>20 21 22 23 24 25 26 27 28 29 → 30</td> <td>← 20 21 22 23 24 25 26 27 28 29 → 30</td> </tr> </table>	←	→	S 6 13 20 27	← 21 22 23 24 25 26 28 29 → 30	20 21 22 23 24 25 26 27 28 29 → 30	← 20 21 22 23 24 25 26 27 28 29 → 30
←	→						
S 6 13 20 27	← 21 22 23 24 25 26 28 29 → 30						
20 21 22 23 24 25 26 27 28 29 → 30	← 20 21 22 23 24 25 26 27 28 29 → 30						
	Powrót						
	<input type="button" value="Anuluj"/> <input type="button" value="Wyślij"/> Alt						

Tworzenie nowego zapytania o ofertę transportową

Kraj	Francja
Miasto	Paryż
Środek komunikacji	samolot
Liczba miejsc	120
Data wyjazdu	
Data powrotu	
<input type="button" value="Anuluj"/> <input type="button" value="Wyślij"/> <input type="button" value="Alt"/>	

Powrót

Panel obsługi

Odpowiedzi na zapytania

Zapytanie o nocleg w **Francja - Paryż**, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 4, wymagany standard: 4

Transportowa Noclegowa

Ibis 14500 złotych Oferta wybrana

Ibis 0 złotych Propozycja odrzucona

Mariot 5000 złotych Zapisz zmiany

Ibis 0 złotych Propozycja odrzucona

Zapytanie o nocleg w **Niemcy - Berlin**, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 5

Stwórz nowe zapytanie o usługę:

Transportowa Noclegowa

Stwórz nowe zapytanie

Dodawanie nowego zapytania o usługę noclegową

Kraj	Francja
Miasto	Parzy
Standard	★★★★★
Liczba miejsc	30
Data wyjazdu	01/24/2017
Data powrotu	01/23/2017

Anuluj
Dodać

◀ January 2017 ▶

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Wprowadzone dane:

Wybrany kraj	<i>Francja</i>
Miasto	<i>Paryż</i>
Standard	<i>2-gwiazdkowy</i>
Liczba miejsc	<i>30</i>
Data wyjazdu	<i>01/24/2017</i>
Data powrotu	<i>01/30/2017</i>

Zatwierdź poprawność danych

[Wróć i popraw](#) [Zatwierdź](#)

Dodawanie nowego zapytania o usługę przejazdową

Kraj	Francja
Miasto	Paryż
Środek komunikacji	Samolot
Liczba miejsc	30

« January 2017 »

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Data wyjazdu Data powrotu

Anuluj Dodaj

Tworzenie nowego zapytania o ofertę transportową

Kraj	Francja
Wprowadzone dane:	
Kraj	Francja
Miasto	Paryż
Środek komunikacji	Samolot
Liczba miejsc	120
Data wyjazdu	20-04-2017
Data powrotu	25-04-2017
Anuluj	
Wyślij zapytanie	
Anuluj	Wyślij
Alt.	

Wprowadzone dane:

Wybrany kraj	Francja
Miasto	Paryż
Standard	Samolot
Liczba miejsc	30
Data wyjazdu	01/24/2017
Data powrotu	01/30/2017

Zatwierdź poprawność danych

Wróć i popraw

Zatwierdź

Tworzenie nowego zapytania o ofertę transportową

Kraj	Francja
Miasto	Paryż
Komunikat błędu	
Wystąpił następujący błąd	
	
Anuluj	
Wyślij	
Alt.	

Dodawanie nowego zapytania o usługę przejazdową

Kraj	Rosja
Środek komunikacji	Samolot
Liczba miejsc	30
Data wyjazdu	01/24/2017
Data powrotu	01/17/2017
Anuluj	
Dodać	

Panel obsługi zapytań o usługi

Odpowiedzi na zapytania

Zapytanie o	Francja - Paryż
oczekiwanie na odpowiedz	oczekiwanie na odpowiedz
oczekiwanie na odpowiedz	oczekiwanie na odpowiedz
oczekiwanie na odpowiedz	oczekiwanie na odpowiedz
Zapytanie o	Francja - Paryż
Stwórz nowe zapytanie	

Panel obsługi zapytań

Odpowiedzi na zapytania

Zapytanie o nocleg w Francja - Paryż, w okresie: 2017-01-24 - 2017-03-31, potrzebnych miejsc: 25, wymagany standard: 4

Tipton
0 złotych
Oczekiwanie na odpowiedz

Ibis
0 złotych
Oczekiwanie na odpowiedz

Ibis
0 złotych
Oczekiwanie na odpowiedz

Zapytanie o nocleg w Francja - Paryż, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 4

Tipton
0 złotych
Oczekiwanie na odpowiedz

Ibis
14500 złotych
Oferta wybrana

Ibis
0 złotych
Propozycja odrzucona

Zapytanie o nocleg w Niemcy - Berlin, w okresie: 2016-04-10 - 2016-04-20, potrzebnych miejsc: 30, wymagany standard: 5

Stwórz nowe zapytanie

Jak widać końcowa implementacja jest na tyle zgodna z prototypem na ile pozwala na to wybrane narzędzie. Jedyną znaczącą różnicą jest nie zaimplementowanie ekranu alternatywnego dla procesu dodawania usługi noclegu, gdyż podawane dane są w postaci dwóch list rozwijanych i liczby gwiazdek, co sprawia, że podanie nieprawidłowych danych z poziomu interfejsu jest niemożliwe.

16 Dokumentacja kodu

Na potrzeby zadania została wygenerowana dokumentacja za pomocą narzędzia phpDocumentor. Następnie została ona uszczegółowiona ręcznie.

Przykładowe fragmenty dokumentacji wyglądają następująco:

`generujStringa($length)`

Zwraca wartość string o podanej długości

Parameters

int	\$length	długość wygenerowanego stringa
-----	----------	--------------------------------

 **pobierzZapytania()**

`pobierzZapytania()`

Zwraca zapytania wraz z odpowiedziami

 **pobierzOdpowiedzi()**

`pobierzOdpowiedzi($id, $typ)`

Zwraca odpowiedzi o podanym typie usługi dla zapytania o podanym indeksie.

Parameters

int	\$id	indeks zapytania w bazie
int	\$typ	typ usługi

 **rozeslijZapytanieNocleg()**

`rozeslijZapytanieNocleg($id)`

Wysyła zapytanie o nocleg o podanym indeksie do wybranych współpracowników

Parameters

int	\$id	indeks zapytania o nocleg w bazie
-----	------	-----------------------------------

 **rozeslijZapytaniePrzejazd()**

`rozeslijZapytaniePrzejazd($id)`

Wysyła zapytanie o przyjazd o podanym indeksie do wybranych współpracowników

Funkcje

Summary

Methods	Properties	Constants
 zaladujBaze() pobierzWspolpracownikow() pobierzUslugi() pobierzCelPodrozy() pobierzKraj() pobierzMiasto() pobierzKraje() pobierzDane() sprawdzEmailCzyWBazie() sprawdzTelefonCzyWBazie() sprawdzCzyWBazie() czytajWartosc() generujStringa() pobierzZapytania() pobierzOdpowiedzi() rozeslijZapytanieNocleg() rozeslijZapytaniePrzejazd() generujListeKrajow() pobierzZapytaniaWspolpracownikow() obslozZapytanie()	<i>No public properties found</i>	<i>No constants found</i>
 <i>No protected methods found</i>	<i>No protected properties found</i>	N/A
 <i>No private methods found</i>	<i>No private properties found</i>	N/A

17 Testy jednostkowe

Do sprawdzenia poprawności działania systemu zostało zdefiniowanych osiem testów jednostkowych.

Przykładowy test wygląda następująco:

```
public function testPobierzCelPodorozy ()  
{  
    $funkcje = new Funkcje();  
    $id = 3;  
    $oczekiwanie = "<b>Niemcy</b> - Essen";  
    $this->assertEquals($oczekiwanie, $funkcje->pobierzCelPodrozy($id));  
}
```

Oczywiście nasz kod przechodzi wszystkie zaimplementowane testy.

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\mateu>cd C:\xampp\htdocs\biuro-po

C:\xampp\htdocs\biuro-po>phpunit funkcjeTest.php

PHPUnit 3.7.21 by Sebastian Bergmann.

.....
Time: 0 seconds, Memory: 1.75Mb
OK (8 tests, 8 assertions)

C:\xampp\htdocs\biuro-po>
```

18 Przypadki testowe dla przypadków użycia

Kolejnym sposobem na sprawdzenie czy nasza implementacja jest poprawna są przypadki testowe. Nasze wyglądają następująco:

1. Dodawanie współpracownika

Aby dodać współpracownika należy wejść na podstronę z współpracownikami. Zostanie wyświetlona lista dodanych współpracowników.

Aby zobaczyć profil współpracownika lub dodać jemu nowe usługi należy kliknąć w jego nazwę. Natomiast by dodać nowego współpracownika należy nacisnąć przycisk umiejscowiony w prawym dolnym rogu. Użytkownik zostanie przekierowany do formularza, w którym musi uzupełnić wszystkie pięć pól. Na nazwę nie są narzucone żadne wymagania. Imię i nazwisko nie może zawierać cyfr i znaków specjalnych oprócz ‘-’. Telefon i email mają narzucone specjalne dla nich formaty.

W razie nie uzupełnienia jakiegoś pola lub wpisania danych w nieodpowiednim formacie użytkownik jest o tym informowany.

W przypadku gdy wszystkie dane zostaną wprowadzone pomyślnie, użytkownik jest przekierowany do podstrony, w której ma możliwość zatwierdzenia wpisanych danych lub powrotu do poprzedniej strony. Zatwierdzenie wiąże się z przejęciem do profilu nowo utworzonego współpracownika, gdzie można dodać nowe usługi.

2. Obsługa zapytań

Użytkownik udaje się do panelu obsługi zapytań, gdzie może przeglądać wysłane zapytania. Zapytanie danego współpracownika może być w trzech stanach:

- Oczekивание на odpowiedź współpracownika
- Propozycja odrzucona przez współpracownika
- Propozycja zaakceptowana przez współpracownika z ceną usługi

Użytkownik ma możliwość wyboru zaakceptowanych usług poprzez naciśnięcie przycisku „Zaakceptuj”, a następnie potwierdzenia i rezygnacji z wyboru.

Aby dodać nowe zapytanie użytkownik naciska przycisk „Stwórz nowe zapytanie” i wybiera rodzaj zapytania: o usługę transportową i noclegową.

Następnie wyświetla się podstrona z formularzem, gdzie należy wybrać cel podróży, w przypadku hotelu standard, a w przypadku usługi transportowej środek komunikacji. Dodatkowo należy określić maksymalną liczbę potrzebnych miejsc oraz daty wyjazdu i powrotu.

Jeżeli użytkownik wprowadzi datę wyjazdu późniejszą od daty powrotu zostanie o tym poinformowany przy próbie dodania nowego zapytania.

Jeżeli wszystkie dane zostaną wprowadzone poprawnie, użytkownik zostanie przekierowany do strony, gdzie będzie miał możliwość potwierdzenia wpisanych danych lub powroto do edycji.

Po dodaniu nowego zapytania jest przekierowany do panelu obsługi zapytań.

Podobnie jak w przypadku testów jednostkowych, nasz system zachowuje się zgodnie z przypadkiem testowym. Jest on również zgodny z interfejsem.

19 Automatyzacja testów funkcjonalnych.

Ostatnim rodzajem testów, którymi sprawdzaliśmy poprawność naszej implementacji są automatyczne testy funkcjonalne zaimplementowane w narzędziu Selenium.

Testy razem z ich wynikami wyglądają następująco:

The screenshot shows the Selenium IDE interface with a test case titled "Dodawanie...". The test steps listed in the table are:

Command	Target	Value
open	/biuro-po/index.html	
clickAndWait	//button[@onclick="location.href = 'wspolpracownicy.php']	
clickAndWait	link=LOT	
click	id=noclegBtn	
select	id=ctrSelect	label=Niemcy
click	css:option[value="2"]	
click	id=star4	
select	id=ctySelect	label=Hanburg
clickAndWait	id=btn	
assertText	css=td	Niemcy - Hanburg - 4 gwiazdki

The log window shows the execution details:

```
[info] Playing test case Dodawanie_uslugi_hotelowej
[info] Executing: |open| /biuro-po/index.html |
[info] Executing: |clickAndWait| //button[@onclick="location.href = 'wspolpracownicy.php'] |
[info] Executing: |clickAndWait| link=LOT |
[info] Executing: |click| id=noclegBtn |
[info] Executing: |select| id=ctrSelect |label=Niemcy|
[info] Executing: |click| css:option[value="2"] |
[info] Executing: |click| id=star4 |
[info] Executing: |select| id=ctySelect |label=Hanburg|
[info] Executing: |clickAndWait| id=btn |
[info] Executing: |assertText| css=td |Niemcy - Hanburg - 4 gwiazdki|
[info] Test case passed
```

The main browser window displays the LOT application's contact addition form and the service selection page.

Nowa nazwa

Dane osoby kontaktowej

Imię	Marcin
Nazwisko	Kowalski
Telefon	123432109
Email	marcin@gmail.com

Usługi współpracownika

Nazwa usługi

Powrót do listy współpracowników

aj usługę

```

[info] Playing test case Dodawanie_wspolpracownika
[info] Executing: [open | /biuro-po/index.html |]
[info] Executing: [clickAndWait | //button[@onclick="location.href = 'wspolpracownicy.php'] |]
[info] Executing: [type | id=firma | Nowa nazwa |]
[info] Executing: [type | id=imie | Marcin |]
[info] Executing: [type | id=nazw | Kowalski |]
[info] Executing: [type | id=tel | 123432109 |]
[info] Executing: [type | id=email | marcin@gmail.com |]
[info] Executing: [clickAndWait | id=submit |]
[info] Executing: [clickAndWait | xpath=//button[@type='button'][2] |]
[info] Test case passed
    
```

Powrót

Lista

Nazwa firmy

- LOT
- RYANAIR
- POLSKI BUS
- Marriot
- Tipton
- Ibis
- Polonia

Search

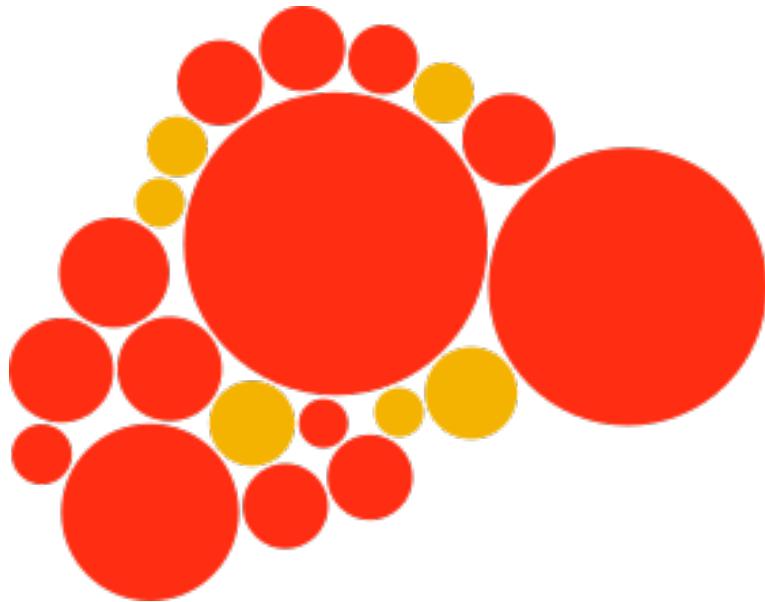
aj usługi

```

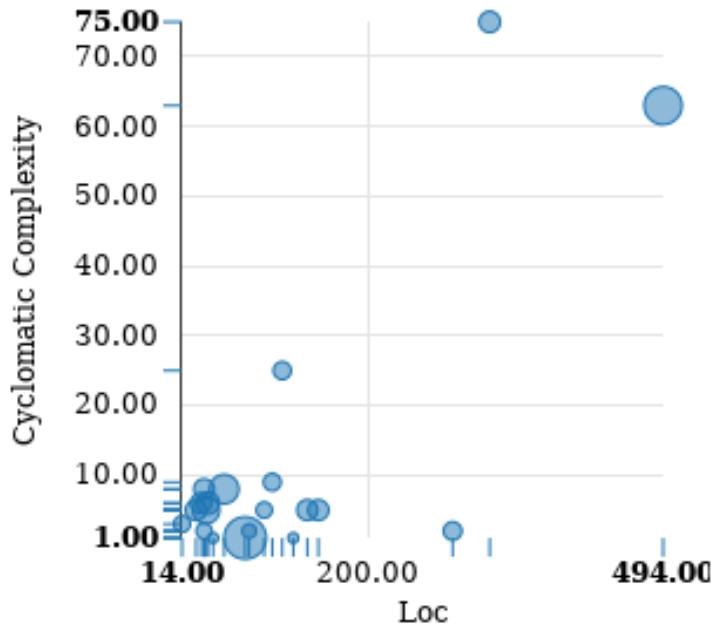
[info] Playing test case Stawianie_bazy
[info] Executing: [open | http://localhost:1234/biuro... |]
[info] Executing: [clickAndWait | //button[@onclick="location.href = 'wspolpracownicy.php'] |]
[info] Executing: [assertText | link=LOT | LOT |]
[info] Test case passed
    
```

20 Badanie jakości projektu

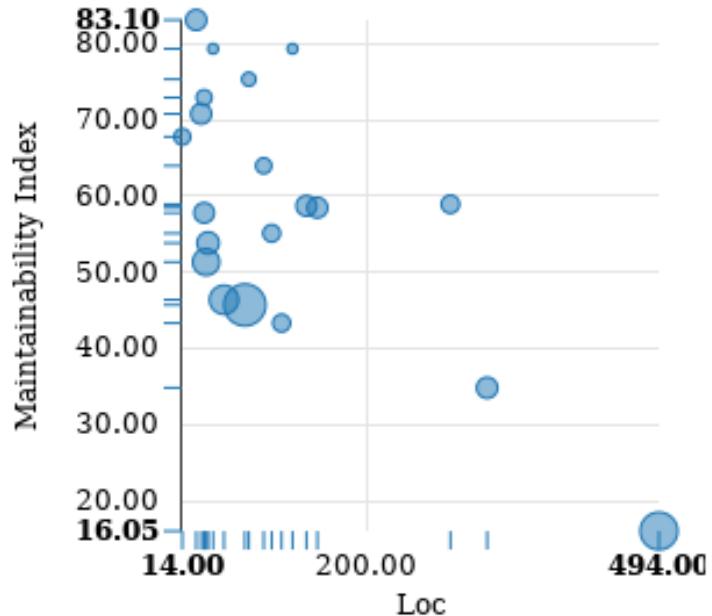
Do zbadania jakości projektu zostały wykorzystane metryki wyliczone przez narzędzie PhpMetrics.



Pierwszy diagram pokazuje reprezentacje pliki jako koła. Według opisu PhpMetrics duże, czerwone koła mogą sprawiać trudności z utrzymaniem. Jak widać w naszym projekcie znajdują się trzy potencjalnie problematyczne pliki.

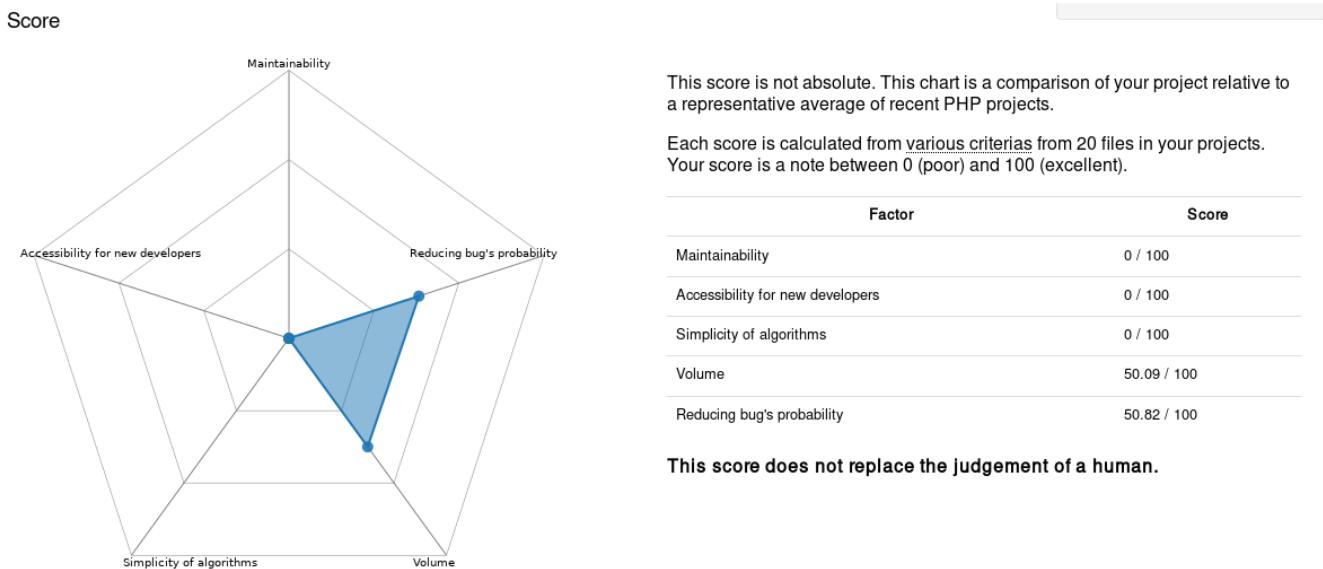


Powyższy diagram pokazuje zależności pomiędzy liczbą linii kodu a złożoność cyklomatyczną. Najgorzej wypada tutaj plik db_insert.php, ponieważ każde zapytanie otoczone jest ifem, który sprawdza czy połączenie do bazy danych przebiegło poprawnie. Sprawia to, że mimo iż został przekroczony najbardziej ryzykowny poziom (50 punktów) kod ten nie powinien nikomu sprawiać problemów. Większy sens ma ona w przypadku drugiego najgorszego pliku zgodnie z tą metryką - funkcje.php, gdyż faktycznie zawiera on wszystkie funkcje, z których korzystamy w naszym systemie i podzielenie go na kilka osobnych plików mogłoby podnieść poziom stabilności systemu.



Podobnie, problemy z plikiem funkcje.php, pokazuje wskaźnik utrzymywalności kodu.

Warto w tym momencie zauważać, że z powodu, iż praktycznie wszystkie funkcje są zdefiniowane, w pliku funkcje.php, wszelkiego rodzaju metryki udostępniane przez PhpMetric dają właśnie temu plikowi najwięcej punktów. Poza wymienionymi już wcześniej metrykami, najwięcej punktów otrzymuje on również w metrykach Halsteada, które są wyliczane na podstawie długości pliku i ilości zdefiniowanych zmiennych, co jest jak najbardziej uzasadnione. Sprawdzają się też tutaj metryki Carda i Agrestia, które poza ilością przetwarzanych danych biorą pod uwagę liczbę wywoływanych metod.



PhpMetric ocenia też cały system w pięciu kategoriach przedstawionych na powyższym zrzucie ekranu. Wyniki nie są szczególnie zadowalające, ale nie są też szczególnym zaskoczeniem biorąc pod uwagę, że nie zdecydowaliśmy się na skorzystanie z jakiegokolwiek frameworka oraz bardzo krótki czas przeznaczony na implementację.

PhpMetrics report

Overview Evaluation Relations map **Repartition** Explore Help

Repartition

Files	20	Relative system complexity	0.11	Lack of cohesion of methods	0.55
Lines of code	2311 (1156 by class, 83 by method)	Relative data complexity	0.06	Efferent Coupling	0.1
Logical lines of code	550 (275 by class, 20 by method)	Relative structure complexity	0.05	Afferent Coupling	0.05
Classes	2 0 interfaces (0 %) 2 concrete classes (100 %) 0 abstract classes (0 %) 0 anonymous classes (0 %)			Abstractness	0
Methods	28 (14 by class)				

Z powyższej tabeli można odczytać, że nasz system nie jest szczególnie skomplikowany, co nie dziwi biorąc pod uwagę, że zaimplementowaliśmy tylko dwa przypadki użycia.

Można też z niej odczytać, że wskaźnik braku kohezji metod. Najlepsza wartość jaką można osiągnąć to 1, więc nasza jest dosłownie średnia.

Dodatkowo mamy tutaj też wskaźniki Efferent coupling, czyli liczbę typów o której wie jedna klasa oraz Afferent coupling czyli ile innych klas średnio zależy od konkretnej klasy. Obie wartości są bardzo niskie, co nie dziwi biorąc pod uwagę, że wszystkie funkcje są zdefiniowane w jednej klasie, a reszta plików odwołuje się do tej klasy. Metryki te są najprawdopodobniej i tak zawyżone gdyż podczas ich wyliczania zostały wzięte pod uwagę testy.

Ostatnią metryką przedstawioną w powyższej tabelce jest abstrakcyjność. Brak jakiejkolwiek klasy abstrakcyjnej czy też interfejsów sprawił, że wynosi ona 0. Ponownie, nie jest to w żaden sposób zaskakujące.

Name	Lines of code	Logical lines of code	Comment weight	Maintainability index	Bugs	Lack of cohesion of methods
<input type="checkbox"/> . (1)	2311	550	3.7	58.65	0.39	0.55

Z ostatniego zestawienia możemy jeszcze odczytać dokładną wartość wskaźnika utrzymywalności kodu, który wynosi 58.65. Biorąc pod uwagę, że wartości niższe niż 64 uznaje się za złe, nasz kod nie będzie dobry w utrzymaniu, ale jesteśmy bliżej granicy średniej jakości kodu niż zupełnie nie nadającego się do utrzymywania.

21 Wnioski

- Każda część systemu, którą dodaje się do projektu powinna być przemyślana, a jej dodanie powinno być uzasadnione. Czasami lepiej zrobić mniej funkcjonalności, ale mających sens w kontekście projektu.
- Dobrą praktyką, której zabrakło w naszym projekcie, jest wracanie do poprzednich etapów po ukończeniu każdego z nich, celem sprawdzenia czy na pewno wszystkie omówione wcześniej elementy znalazły się w następnej części projektu. Szczególnie pomocne byłoby wprowadzenie większej ilości diagramów sprawdzających pokrycie poprzednich etapów, tak jak w przypadku historyjek użytkownika i potrzeb użytkownika zdefiniowanych w wizji.
- Diagramy ze zdefiniowaną semantyką i syntaktyką przewyższającą zapis tekstowy, gdyż mamy pewność, że i projektant, i osoba czytająca projekt będą rozumieć zapis dokładnie w ten sam sposób.
- Reguły biznesowe okazują się zaskakująco ważne, w dalszych etapach, gdyż przynajmniej z założenia, powinny być one przewidywać wszelkiego rodzaju ograniczenia związane z dziedziną, których nie jesteśmy w stanie sami wymyślić. Należało na ten etap poświęcić zdecydowanie więcej czasu.
- Znalezienie złotego środka pomiędzy zbyt szczegółowymi opisami, a zbyt dokładnymi jest bardzo trudne i zdecydowanie wymaga wyrobienia sobie pewnych intuicji. Niestety odnosimy wrażenie, że jeden projekt nie wystarczy żeby to osiągnąć.
- Wszelkiego rodzaju automatyczne generowanie części projektu, a nawet części samej implementacji zdaje się bardzo praktyczną funkcjonalnością, jednak w przypadku nawet drobnych błędów, które zdarzyły się nam w większości przypadków automatycznej generacji, ich wykrycie i naprawienie zajęło dużo więcej czasu niż zajęłoby napisanie danej części od początku. Dlatego też należy bardzo uważać przy korzystaniu z funkcjonalności tego typu i w razie potrzeby od razu po wykryciu nietrywialnych błędów rezygnować z wygenerowanej części.
- Lepiej dopasować narzędzie do architektury niż naciągać architekturę do narzędzia, które nie jest do niej dopasowane. Boleśnie przekonaliśmy się o tym pisząc w czystym php.
- Na wszelkiego rodzaju błędy i niedociągnięcia we wcześniejszych etapach należy reagować jak najszybciej, gdyż będą nam one sprawiać problemy do samego końca projektu. W szczególności może się okazać, że coś co było drobnym niedociągnięciem w początkowej fazie projektu może w ciągu całego procesu urosnąć do skomplikowanego problemu, z którym będzie bardzo ciężko sobie poradzić.
- Dwa przypadki użycia to zdecydowanie za mało żeby ocenianie ich metrykami miało sens, gdyż przy tak małej ilości kodu nie występuje potrzeba podzielenia go na kilka lub nawet kilkanaście plików, a właśnie do systemów napisanych w ten sposób owe metryki powstały.
- Nie jesteśmy przyzwyczajeni do myślenia o projekcie w abstrakcyjny sposób charakterystyczny dla analityków. Nawet w przypadku diagramów nie mających bezpośredniego związku z końcową implementacją najłatwiej było nam je tworzyć odwołując się do hipotetycznej implementacji. Spowodowało to również, że bardzo dużo błędów wykryliśmy podczas prototypowania interfejsu, co skłoniło nas do wniosku, że dobrym rozwiązaniem mogłoby być zrobienie prototypu dużo wcześniej, gdyż pozwolił on nam skorzystać z naszych programistycznych intuicji.
- Praca w grupie nad tego rodzaju projektem pomaga znajdować więcej błędów i skłania do dyskusji, co zdecydowanie pozytywnie wpływa na jego jakość.