

POLITECHNIKA KOSZALIŃSKA

WYDZIAŁ ELEKTRONIKI I INFORMATYKI

STUDIA STACJONARNE, SEMESTR VI, KIERUNEK INFORMATYKA

ROK AKADEMICKI 2015 / 2016

DOKUMENTACJA Z PRZEDMIOTU „PROJEKT ZESPOŁOWY CASE”

TEMAT PROJEKTU

APLIKACJA BAZODANOWA DO OBSŁUGI FIRMY TRANSPORTOWEJ



wykonawcy

Damian Berda

Jakub Wachowiak

Łukasz Stolarczyk

Marcin Chruścicki

Michał Motykowski

spis treści

1. Protokół z zebrania założycielskiego zespołu programistycznego „aplikacja bazodanowa do obsługi firmy transportowej” zwanego dalej zespołem, odbytego w dniu 01.03.2016 r. w Koszalinie.	4
2. Instrumentarium zespołu projektowego.	5
3. Opis wymagań klienta.....	5
3.1. Model klienta.	5
3.2. Wymaganie funkcjonalne.....	5
3.3. Wymagania нефункционалне (ограничения).	5
4. Wstępny harmonogram prac.....	6
5. Specyfikacja wymagań.	6
5.1. Wstęp.....	6
5.1.1. Cele.	6
5.1.2. Zakres.....	6
5.1.3. Definicje, skróty. Opis.	6
5.2. Ogólny opis.....	7
5.2.1. Przydatność projektowanej aplikacji. Walory użytkowe.....	7
5.2.2. Możliwości projektowanego oprogramowania.	7
5.2.3. Ogólne ograniczenia.....	7
5.2.4. Charakterystyka użytkowników.....	7
5.2.5. Środowisko operacyjne.	7
5.3. Specyfika wymagań.	7
5.3.1. Wymagania funkcjonalne.....	7
5.3.2. Wymagania нефункционалне (ограничения).	8
5.4. Diagram przypadków użycia.....	8
6. Raport dla zleceniodawcy.	8
6.1. Cele przedsięwzięcia.	8
6.2. Zakres przedsięwzięcia.....	8
6.3. Systemy zewnętrzne.	9
6.4. Szkicowy opis wymagań.....	9
6.4.1. Wymagania funkcjonalne.....	9
6.4.2. Wymagania нефункционалне (ограничения).	9

6.5. Szkicowy opis rozwiązań.	9
6.6. Szkicowy opis modelu systemu.	9
6.7. Kosztorys projektu.	10
6.8. Wstępny harmonogram prac.....	10
7. Harmonogram prac.....	11
8. Wizytówka.....	12
9. Logo.	13
10. Model konceptualny bazy danych.	13
11. Model fizyczny bazy danych.	13
12. Diagram klas.....	14
13. Diagram komponentów.	14
14. Diagram czynności.	14
15. Diagramy sekwencji.....	14
16. Metodologia. Narzędzia CASE.	14
17. Kosztorys.....	15
18. Narzędzia RAD.	16
19. Testy przeprowadzone w ramach projektu.....	16

1. Protokół z zebrania założycielskiego zespołu programistycznego „aplikacja bazodanowa do obsługi firmy transportowej” zwanego dalej zespołem, odbytego w dniu 01.03.2016 r. w Koszalinie.

W zebraniu wzięły udział następujące osoby:

1. Damian Berda.
2. Jakub Wachowiak.
3. Łukasz Stolarczyk.
4. Marcin Chruścicki.
5. Michał Motykowski.

Uczestnicy zebrania podjęli indywidualną decyzję o przystąpieniu do zespołu programistycznego „aplikacja bazodanowa do obsługi firmy transportowej” zwanego dalej zespołem. Po zapisaniu obecności wszystkich członków oraz wypełnieniu deklaracji członkowskich zespołu przystąpiono do wyboru kierownika zespołu. Podczas głosowania jednogłośnie wybrany został Marcin Chruścicki. Zebrani upoważnili nowo wybranego przedstawiciela do reprezentowania interesów zespołu. Po wybraniu przedstawiciela przystąpiono do ustalenia postanowień ogólnych.

Postanowienia ogólne:

1. Odwołanie przewodniczącego może odbyć się po zgłoszeniu tzw. wotum nieufności względem wyżej wymienionej osoby oraz zatwierdzenie usunięcia poprzez demokratyczne głosowanie.
2. Każdy członek zespołu zobowiązany jest do wykonywania zadań projektowych określonych w harmonogramie zadań.
3. Każdy członek zespołu zobowiązany jest do przestrzegania postanowień ogólnych.
4. Data zakończenia projektu została wstępnie ustalona na dzień 7 VI 2016 r.

Następnie przystąpiono do określenia celów projektu, jak również obowiązków członków zespołu.

Cele projektu:

1. Zaliczenie przedmiotu „Projekt Zespołowy CASE”.
2. Wykonanie projektu oprogramowania, które posłuży zarówno dyżurnemu ruchu PKS, jak i pasażerom.
3. Nauka pracy w zespole projektowym.
4. Rozwiązanie zespołu po pomyślnym zaliczeniu projektu. Przewidywany termin to 7 czerwca 2016 r.

Obowiązki członków zespołu:

1. Damian Berda (grafik, tester).
2. Jakub Wachowiak (analityk, projektant).
3. Łukasz Stolarczyk (dokumentalista).
4. Marcin Chruścicki (kierownik, programista).
5. Michał Motykowski (projektant, programista).

2. Instrumentarium zespołu projektowego.

W ramach pracy zespołowej zostało stworzone repozytorium projektu o nazwie „DamberPol” na serwerach GitHub.com i Trello.com. Do repozytorium dodawane będą kolejne wersje aplikacji oraz dokumentacji, z możliwością przeglądania przez wszystkich członków zespołu projektowego.

Żeby wspomóc i usprawnić wspólną pracę nad projektem, członkowie zespołu będą kontaktować się za pomocą takich aplikacji, jak Skype i Facebook Messenger.

Narzędzia wykorzystywane podczas pracy projektowej:

- Eclipse – producent Eclipse Foundation, licencja Eclipse Public License. (<http://www.eclipse.org>)
- Microsoft Office Excel – producent Microsoft, licencja komercyjna. (<http://www.office.microsoft.com>)
- Microsoft Office Word – producent Microsoft, licencja komercyjna. (<http://www.office.microsoft.com>)
- Microsoft SQL Server – producent Microsoft, licencja Microsoft EULA. (<http://www.microsoft.com/sql>)
- Mozilla Firefox – producent Mozilla Corporation, licencja Mozilla Licensing. (<http://www.mozilla.org>)

3. Opis wymagań klienta.

3.1. Model klienta.

Klientem jest Państwowa Komunikacja Samochodowa w Koszalinie. Działalność tej instytucji polega na gromadzeniu pewnego rodzaju danych i udostępnianiu jej części pasażerom za pośrednictwem Internetu.

3.2. Wymaganie funkcjonalne.

- sprawdzanie połączeń przez pasażerów;
- sporządzanie harmonogramu ruchu po danej trasie (powiązany z wybranym kursem);
- sporządzanie raportów (np. tygodniowych, miesięcznych) na temat ilości pasażerów korzystających z danego kursu;
- zarządzanie (np. dodawanie, usuwanie) listą zatrudnionych kierowców i wykorzystywanych autokarów;
- przygotowywanie chwilowych raportów – jaki kierowca jest w trasie i do jakiego kursu został przypisany;
- administrowanie danymi – wprowadzanie / usuwanie podstawowych obiektów bazy danych;

3.3. Wymagania niefunkcjonalne (ograniczenia).

- dla każdego kierunku (np. Koszalin – Gdańsk) istnieją kursy wyznaczające realizację danego połączenia (np. data, czas przyjazdu / odjazdu);
- na danej trasie (np. Koszalin – Gdańsk) istnieją kursy pośrednie (np. Słupsk – Gdańsk);
- pasażerowie mają możliwość sprawdzenia połączenia poprzez podanie miejscowości początkowej i końcowej oraz dnia, w którym zamierzają podróżować;
- dyżurny powinien mieć możliwość sporządzenia odpowiednich raportów;

- dyżurny ma wgląd do statystyk, w której widoczne są następujące dane: jaka osoba jest aktualnie w trasie, jakim jedzie kursem oraz jaki prowadzi autokar;

4. Wstępny harmonogram prac.

luty:

- założenie grupy projektowej;
- uzgodnienie wstępnych założeń projektu;

marzec:

- określenie wymagań funkcjonalnych projektu;
- tworzenie dokumentacji (diagram klas, przypadków użycia, funkcjonalności);

kwiecień:

- zaprojektowanie graficznego interfejsu użytkownika;
- programowanie kolejnych etapów projektu;
- konsultacje z klientem (prowadzącym zajęcia);

maj:

- dokończenie prac nad projektem;
- testowanie i naprawa ewentualnych błędów w projekcie;
- finalizacja prac związanych z projektem;

czerwiec:

- zaprezentowanie aplikacji;
- zakończenie prac nad projektem;

5. Specyfikacja wymagań.

5.1. Wstęp.

Celem dokumentu jest specyfikacja obsługi bazodanowej aplikacji w firmie transportowej.

5.1.1. Cele.

- zaliczenie przedmiotu „projekt zespołowy CASE”;
- wykonanie projektu aplikacji wspomagającej funkcjonowanie Państwowej Komunikacji Samochodowej;
- nauka pracy w zespole projektowym;
- rozwiązanie zespołu po pomyślnym zakończeniu projektu;
- usprawnienie działania firmy;

5.1.2. Zakres.

5.1.3. Definicje, skróty. Opis.

- dodawanie kierowców / autokarów – dodawanie zatrudnionych kierowców i wykorzystywanych autokarów w firmie;

- dyżurny ruchu – wykwalifikowany pracownik, którego obowiązkiem jest prowadzenie ruchu;
- pasażer – osoba podróżująca pojazdem;
- PKS – Państwowa Komunikacja Samochodowa, firma, dla której opracowywany jest projekt.
- połączenie – łączność między miastami utrzymywana za pomocą środka lokomocji;
- sporządzanie raportów – jaki kierowca jest w trasie, jakim jedzie kursem, jaki autokar prowadzi;
- usuwanie kierowców / autokarów – usuwanie zwolnionych kierowców i niewykorzystywanych autokarów w firmie;

5.2. Ogólny opis.

5.2.1. Przydatność projektowanej aplikacji. Walory użytkowe.

Aplikacja będzie zaprojektowana w taki sposób, aby użytkownik z podstawowymi umiejętnościami obsługi komputera, nie miał jakichkolwiek problemów z obsługą rozkładu jazdy, a dyżurny z nadzorem ruchu.

5.2.2. Możliwości projektowanego oprogramowania.

Aplikacja będzie oferowała:

- dodawanie autokarów;
- dodawanie kierowców;
- generowanie cyklicznych raportów nt. ilości pasażerów korzystających z danego kursu;
- sprawdzanie rozkładu jazdy przez pasażerów;
- usuwanie autokarów;
- usuwanie kierowców;
- wglądu do informacji, w których znajdują się następujące dane: jaki kierowca jest obecnie w trasie i do jakiego kursu został przypisany;

5.2.3. Ogólne ograniczenia.

5.2.4. Charakterystyka użytkowników.

- dyżurny ruchu – osoba zarządzająca aplikacją, posiada największe uprawnienia;
- pasażer – osoba posiadająca ograniczone przywileje, posiada możliwość sprawdzenia rozkładu jazdy autobusów;

5.2.5. Środowisko operacyjne.

5.3. Specyfika wymagań.

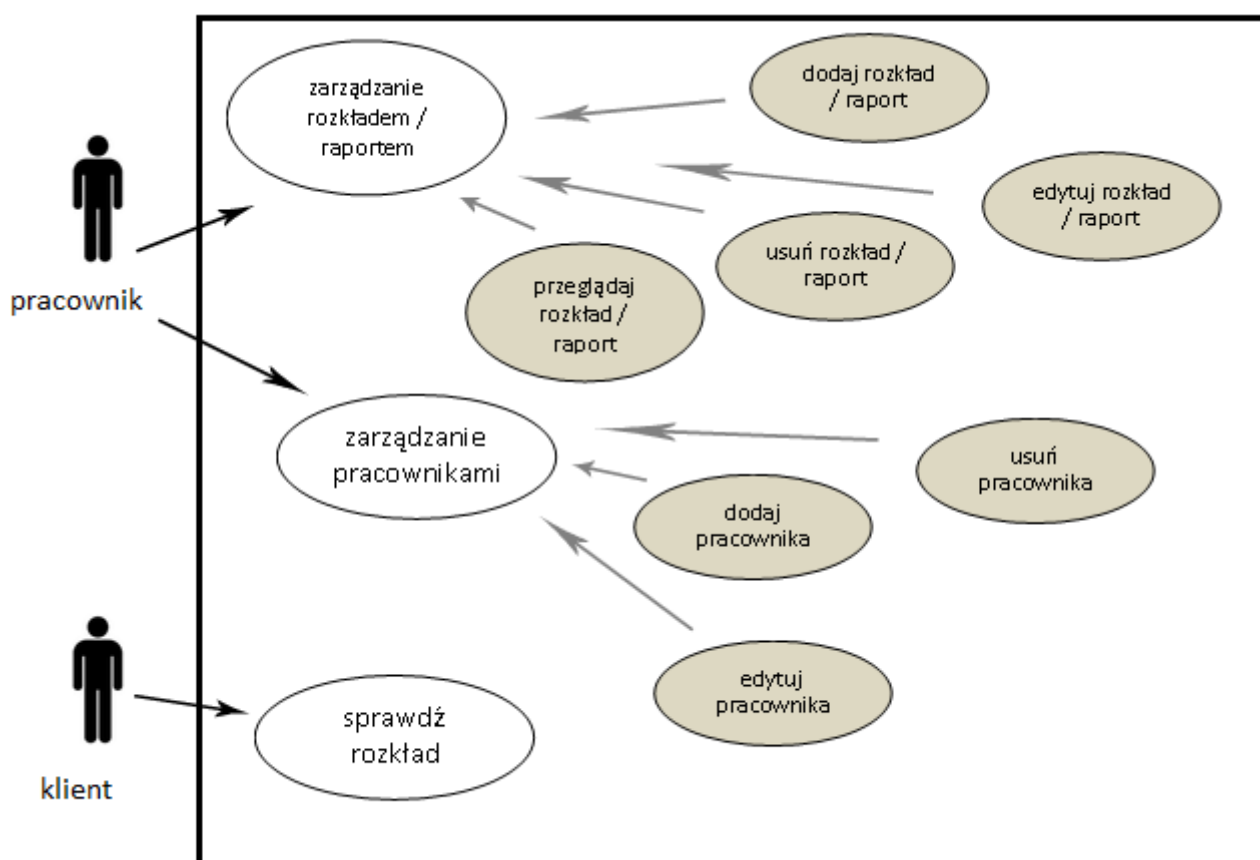
5.3.1. Wymagania funkcjonalne.

- dodawanie i usuwanie autokarów z bazy danych;
- dodawanie i usuwanie kierowców z bazy danych;
- dodawanie i usuwanie z bazy danych informacji o kursie;

- edycja informacji o autokarach;
- edycja informacji o kierowcach;
- edycja wprowadzonych kursów;
- informowanie pasażerów o zmianach w rozkładzie;
- logowanie do bazy;
- wyświetlanie rozkładu jazdy;

5.3.2. Wymagania niefunkcjonalne (ograniczenia).

5.4. Diagram przypadków użycia.



6. Raport dla zleceniodawcy.

6.1. Cele przedsięwzięcia.

Celem projektu jest stworzenie aplikacji wspomagającej działanie firmy transportowej. Program ma na celu ułatwić pracownikom zarządzanie danymi, a czytelnikom zapewnić możliwość przeglądania wybranych zasobów, rozkładu jazdy.

6.2. Zakres przedsięwzięcia.

- określenie wymagań ze strony klienta;
- analiza i projektowanie systemu;

- realizacja projektu;
- tworzenie dokumentacji;
- testowanie;
- wdrożenie aplikacji;

6.3. Systemy zewnętrzne.

System będzie zgodny z systemem operacyjnym Windows (XP, Vista, Windows 7, Windows 8, Windows 10).

System będzie współpracować z systemem bazodanowym MS SQL.

Po stronie klienta system musi współpracować z przeglądarkami internetowymi:

- Internet Explorer wersja 7.0 lub nowsza.
- Mozilla Firefox wersja 3.0 lub nowsza.
- Opera wersja 9.0 lub nowsza.
- Google Chrome wersja 2.0 lub nowsza.

6.4. Szkicowy opis wymagań.

6.4.1. Wymagania funkcjonalne.

- zarządzanie rozkładem jazdy;
- zarządzanie kontami użytkowników;
- zdalne przeglądanie zasobów rozkładu jazdy;
- powiadamianie pasażerów i kierowców firmy o zmianie w rozkładach jazdy;

6.4.2. Wymagania niefunkcjonalne (ograniczenia).

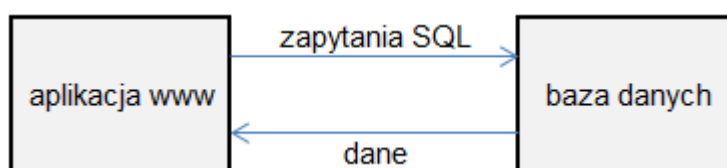
- weryfikacja użytkownika;

6.5. Szkicowy opis rozwiązań.

Realizacja aplikacji WWW (interfejsu) zostanie wykonana w technologii Java.
Do przygotowania bazy danych wykorzystany zostanie Microsoft SQL Server.

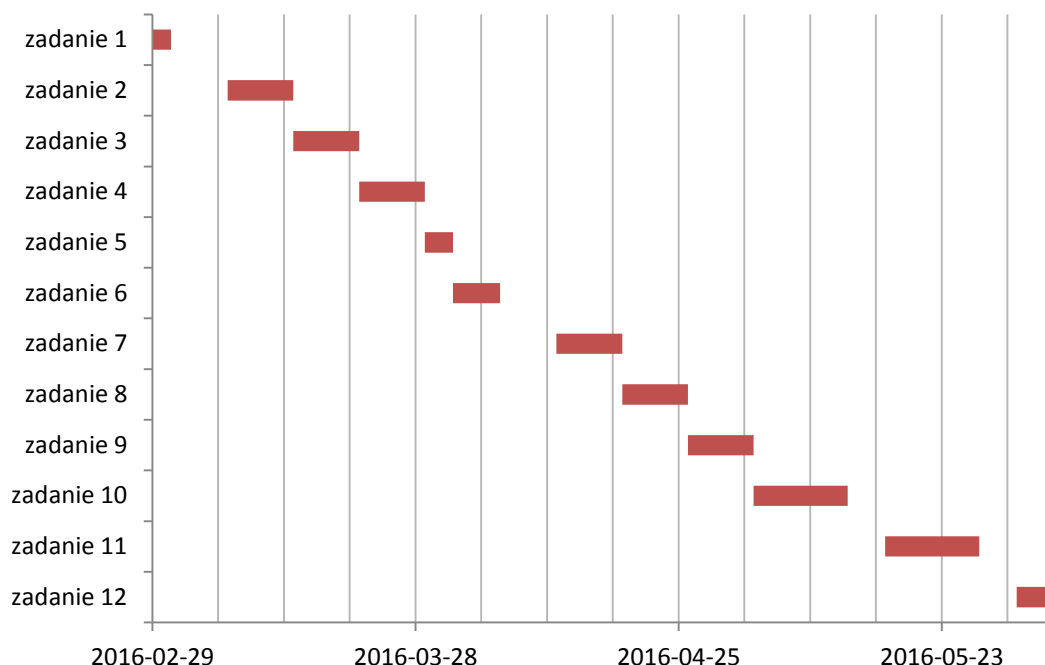
6.6. Szkicowy opis modelu systemu.

System będzie się składał z aplikacji (interfejsu) w postaci strony internetowej dostępnej z poziomu przeglądarki internetowej. Aplikacja będzie się łączyć z bazą danych Microsoft Server.



6.7. Kosztorys projektu.

6.8. Wstępny harmonogram prac.



Zadanie 1 (29.02 – 01.03). Uzgodnienie tematu projektu, wstępna umowa z klientem. Utworzenie grupy projektowej, ustalanie zasad współpracy z członkami grupy. Utworzenie protokołu z zebrania założycielskiego.

Zadanie 2 (08.03 – 15.03). Utworzenie harmonogramu prac. Opracowanie dokumentacji wymagań użytkownika. Raport klienta, opracowanie raportu dla zleceniodawcy (zakres przedsięwzięcia; przygotowanie szkicowego opisu wymagań, modelu systemu, proponowanych rozwiązań, oszacowanie kosztów projektu oraz daty zakończenia prac nad projektem).

Zadanie 3 (15.03 – 22.03). Określenie wymagań funkcjonalnych projektu, opracowanie specyfikacji wymagań (opracowanie słownika pojęć, przygotowanie listy wymagań funkcjonalnych i niefunkcjonalnych, zidentyfikowanie aktorów oraz ich funkcjonalności, przeprowadzenie identyfikacji przypadków użycia, opracowanie diagramu przypadków użycia). Opracowanie stylu firmowego projektu (logo, wizytówka). Zaprojektowanie modelu bazy danych (model konceptualny i model fizyczny), stworzenie bazy danych na potrzeby projektu.

Zadanie 4 (22.03 – 29.03). Wybór odpowiednich narzędzi CASE. Opracowanie zestawu diagramów (przypadków użycia, klas, przepływu danych).

Zadanie 5 (29.03 – 01.04). Wykończenie prac nad stylem firmowym projektu (logo, wizytówka).

Zadanie 6 (01.04 – 06.04). Zaprojektowanie graficznego interfejsu użytkownika. Opracowanie słownika danych zawierający specyfikację modelu.

Zadanie 7 (12.04 – 19.04). Opracowanie diagramu związków encji, diagramu przejść stanów, diagramu aktywności, diagramu sekwencji, diagramu współpracy, diagramu komponentów. Analiza kosztorysu. Konsultacje z klientem.

Zadanie 8 (19.04 – 26.04). Wybór odpowiednich narzędzi do szybkiego rozwijania aplikacji (ang. RAD). Ciąg dalszy prac nad projektem (zaprogramowanie GUI, podstawowych funkcjonalności).

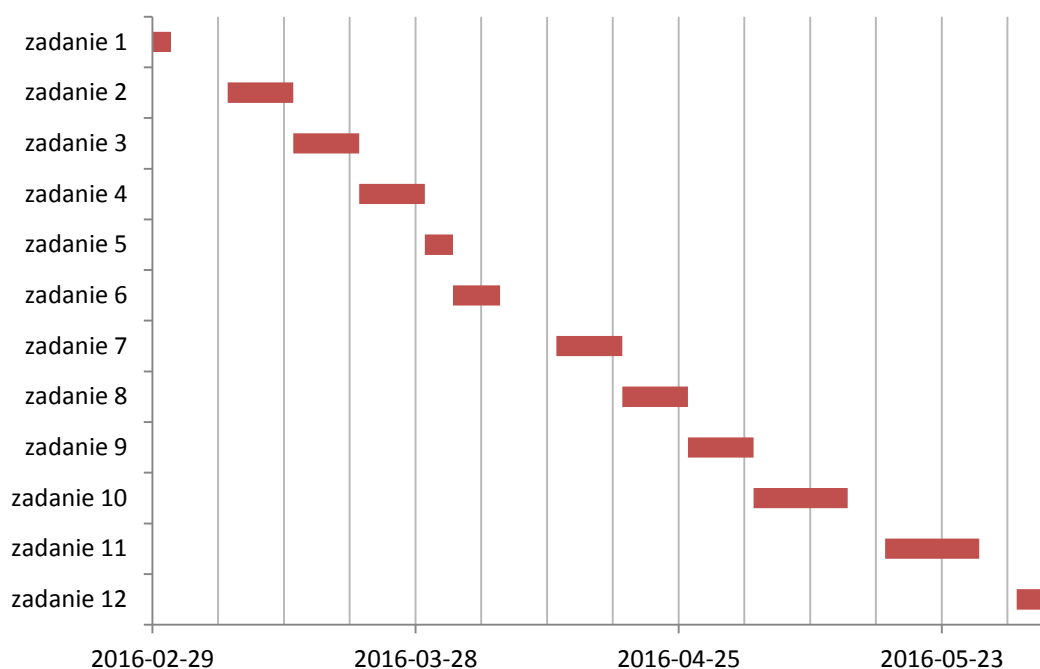
Zadanie 9 (26.04 – 03.05). Prace nad programowaniem dalszych funkcjonalności systemowych.

Zadanie 10 (03.05 – 13.05). Wstępne testowanie aplikacji. Naprawa ewentualnych błędów.

Zadanie 11 (17.05 – 27.05). Zakończenie programowania projektu. Przeprowadzenie szczegółowych testów. Finalizacja projektu. Sprawdzenie i zamknięcie dokumentacji. Przygotowanie projektu do zaprezentowania klientowi.

Zadanie 12 (31.05 – 03.06). Prezentacja projektu klientowi. Zakończenie prac nad projektem.

7. Harmonogram prac.



Zadanie 1 (29.02 – 01.03). Uzgodnienie tematu projektu, wstępna umowa z klientem. Utworzenie grupy projektowej, ustalanie zasad współpracy z członkami grupy. Utworzenie protokołu z zebrania założycielskiego.

Zadanie 2 (08.03 – 15.03). Utworzenie harmonogramu prac. Opracowanie dokumentacji wymagań użytkownika. Raport klienta, opracowanie raportu dla zleceniodawcy (zakres przedsięwzięcia ;

przygotowanie szkicowego opisu wymagań, modelu systemu, proponowanych rozwiązań, oszacowanie kosztów projektu oraz daty zakończenia prac nad projektem).

Zadanie 3 (15.03 – 22.03). Określenie wymagań funkcjonalnych projektu, opracowanie specyfikacji wymagań (opracowanie słownika pojęć, przygotowanie listy wymagań funkcjonalnych i niefunkcjonalnych, zidentyfikowanie aktorów oraz ich funkcjonalności, przeprowadzenie identyfikacji przypadków użycia, opracowanie diagramu przypadków użycia). Opracowanie stylu firmowego projektu (logo, wizytówka). Zaprojektowanie modelu bazy danych (model conceptualny i model fizyczny), stworzenie bazy danych na potrzeby projektu.

Zadanie 4 (22.03 – 29.03). Wybór odpowiednich narzędzi CASE. Opracowanie zestawu diagramów (przypadków użycia, klas, przepływu danych).

Zadanie 5 (29.03 – 01.04). Wykończenie prac nad stylem firmowym projektu (logo, wizytówka).

Zadanie 6 (01.04 – 06.04). Zaprojektowanie graficznego interfejsu użytkownika. Opracowanie słownika danych zawierający specyfikację modelu.

Zadanie 7 (12.04 – 19.04). Opracowanie diagramu związków encji, diagramu przejść stanów, diagramu aktywności, diagramu sekwencji, diagramu współpracy, diagramu komponentów. Analiza kosztorysu. Konsultacje z klientem.

Zadanie 8 (19.04 – 26.04). Wybór odpowiednich narzędzi do szybkiego rozwijania aplikacji (ang. RAD). Ciąg dalszy prac nad projektem (zaprogramowanie GUI, podstawowych funkcjonalności).

Zadanie 9 (26.04 – 03.05). Prace nad programowaniem dalszych funkcjonalności systemowych.

Zadanie 10 (03.05 – 13.05). Wstępne testowanie aplikacji. Naprawa ewentualnych błędów.

Zadanie 11 (17.05 – 27.05). Zakończenie programowania projektu. Przeprowadzenie szczegółowych testów. Finalizacja projektu. Sprawdzenie i zamknięcie dokumentacji. Przygotowanie projektu do zaprezentowania klientowi.

Zadanie 12 (31.05 – 03.06). Prezentacja projektu klientowi. Zakończenie prac nad projektem.

8. Wizytówka.



DamberPol

Jan Kowalski

dział organizacji przewozów

ul. Szeroka 1

75-257 Koszalin

telefon (94) 341 23 45

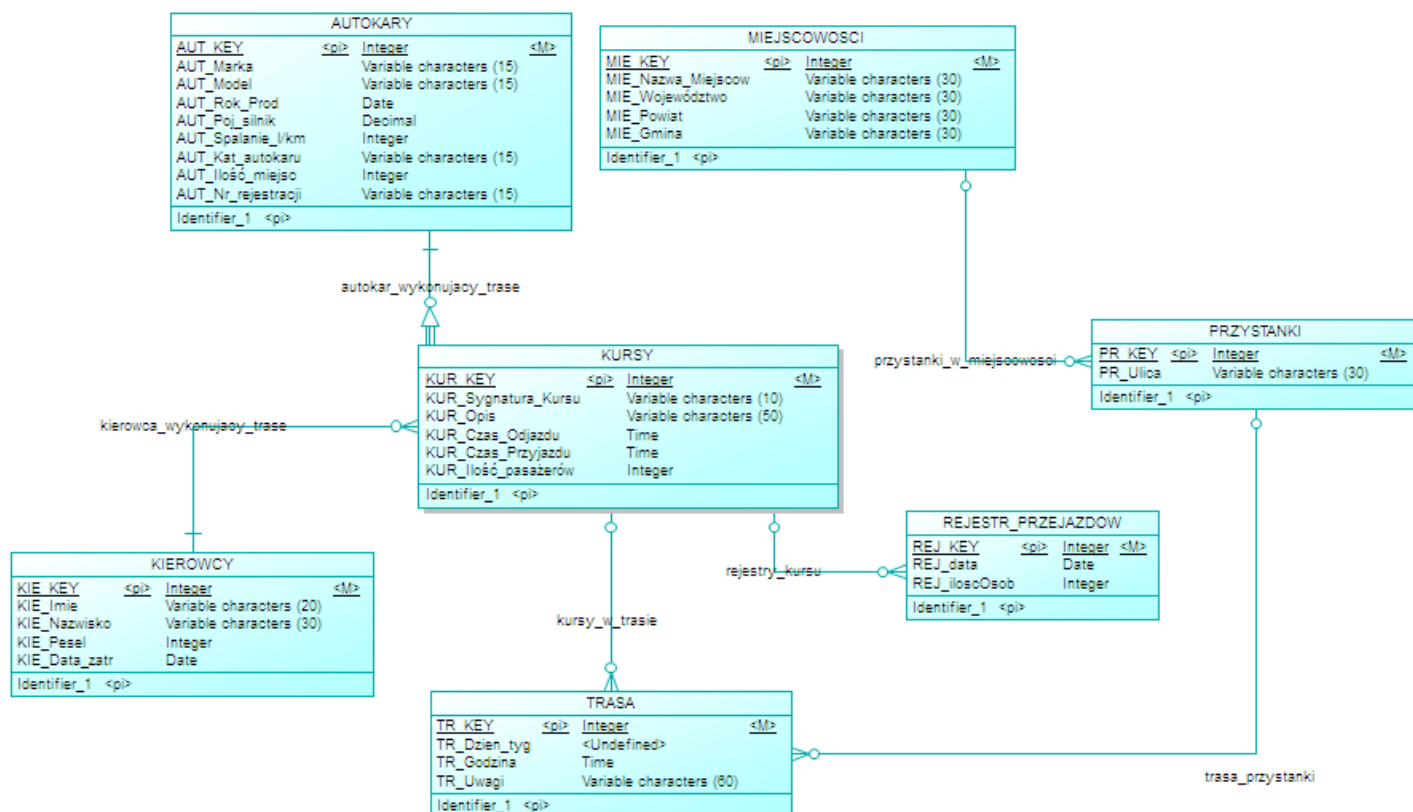
faks (94) 341 23 23

jan.kowalski@firma.com.pl

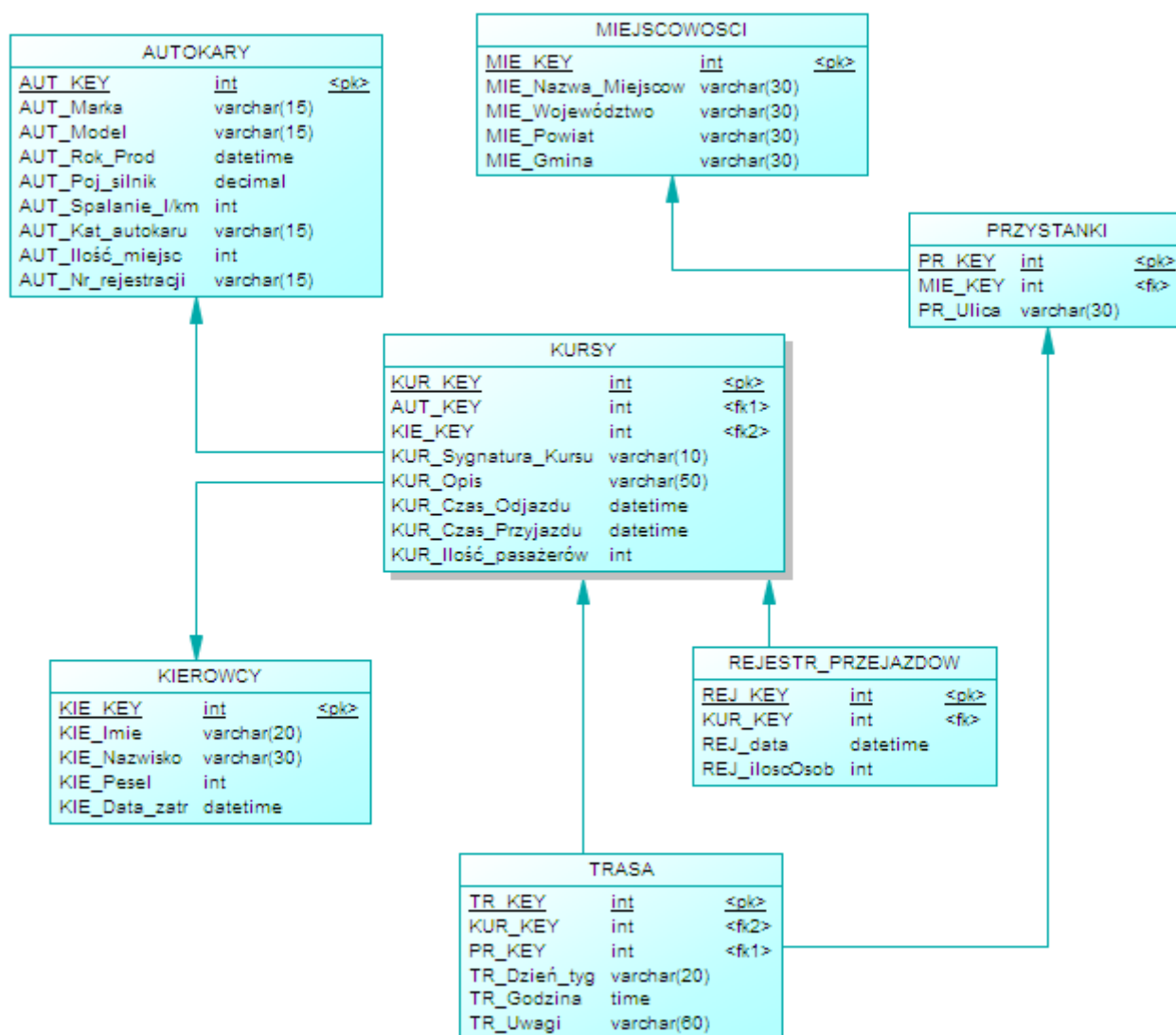
9. Logo.



10. Model konceptualny bazy danych.



11. Model fizyczny bazy danych.



12. Diagram klas.

13. Diagram komponentów.

14. Diagram czynności.

15. Diagramy sekwencji.

16. Metodologia. Narzędzia CASE.

Mianem narzędzi CASE (ang. Computer Aided Software Engineering) określa się systemy komputerowe, przeznaczone do wspomagania rutynowych czynności procesu tworzenia oprogramowania. To właśnie dzięki nim projekty tworzy się precyzyjniej, a praca nad diagramami, weryfikowanie ich poprawności oraz śledzenie wykonanych testów jest jeszcze szybsze i prostsze.

Narzędzia CASE zastosowane w projekcie:

- Eclipse – producent Eclipse Foundation, licencja Eclipse Public License. (<http://www.eclipse.org>)

- Microsoft Office – producent Microsoft, licencja komercyjna. (<http://www.office.microsoft.com>)
- PowerDesigner – producent Sybase, licencja komercyjna. (<http://www.sybaseproducts.com>)
- Microsoft Paint – producent Microsoft, licencja zamknięte oprogramowanie. (<http://www.microsoft.com>)

17. Kosztorys.

Poniżej zamieszczono wydatki uwzględnione w 4-miesięcznym procesie wdrażania projektu.

koszty zakupu oprogramowania							
lp.	nazwa	wersja	licencja	cena	ilość	czas realizacji projektu	suma
1	Microsoft Windows	7	komercyjna	599 zł	5	4 mies.	2 995 zł
2	Microsoft Office	365 Business Premium	komercyjna	45 zł / mies.	5		900 zł
3	Eclipse	IDE for Java EE Dev.	GPL	-	4		-
4	XAMPP	7.0.4	GPL	-	5		-
5	SAP Sybase PowerDesigner	16.5	komercyjna	10 467 zł	1		10 467 zł
6	GitHub	Silver	komercyjna	190 zł / mies.	1		760 zł
7	Git Large File Storage	-	komercyjna	19 zł / mies.	1		76 zł
łączny koszt zakupu oprogramowania							15 198 zł

koszty pracy				
lp.	zadanie	czas realizacji	cena za godz.	suma
1	analiza wymagań	91 godz.	54 zł	4 914 zł
2	projekt systemu	150 godz.	62 zł	9 300 zł
3	realizacja aplikacji	165 godz.	70 zł	11 550 zł
4	testowanie aplikacji	85 godz.	60 zł	5 100 zł
5	tworzenie dokumentacji	90 godz.	45 zł	4 050 zł
łączny koszt pracy				34 914 zł

koszty utrzymania				
lp.	usługa	miesięczny koszt	czas realizacji projektu	suma
1	wynajem lokalu	600 zł	4 mies.	2 400 zł
2	energia elektryczna	400 zł		1 600 zł
3	dostęp do internetu	80 zł		320 zł
4	ogrzewanie	300 zł		1 200 zł
5	dojazdy	600 zł		2 400 zł
6	licencja ZAiKS	81 zł		324 zł
7	komputer (7x)	0 zł (środki własne)		0 zł
8	pracownicy : 2x młodszy programista 2x starszy programista 1x kierownik projektu	12 000 zł / 2x os. 20 000 zł / 2x os. 10 000 zł / os.		168 000 zł
łączny koszt utrzymania				176 244 zł
łączny koszt realizacji projektu				226 356 zł

18. Narzędzia RAD.

Rapid Application Development (RAD) to metodologia polegająca na udostępnieniu programiście dużych możliwości prototypowania oraz dużego zestawu gotowych komponentów (np. zapewniających dostęp do bazy danych). Umożliwia to uzyskanie pewnego efektu już w pierwszych krokach programistycznych, jednocześnie stanowi poważne zagrożenie dla projektów o większych rozmiarach ze względu na łatwość nieprzemyślanego modyfikowania.

Podczas pracy nad projektem wykorzystane zostało narzędzie Oracle Eclipse (framework) do tworzenia aplikacji w języku Java. Narzędzie to zostało wybrane z uwagi na nabyte wcześniej doświadczenie z tym oprogramowaniem.

19. Testy przeprowadzone w ramach projektu.

testowany element	środowisko	rezultat
encje	Sybase PowerDesigner 12.5	pozytywny
połączenia między encjami	Sybase PowerDesigner 12.5	pozytywny
klucze	Sybase PowerDesigner 12.5	pozytywny
relacje pomiędzy encjami	Sybase PowerDesigner 12.5	pozytywny
typy danych atrybutów encji	Sybase PowerDesigner 12.5	pozytywny
dodawanie rekordów do tabel	Microsoft SQL Server 2014	pozytywny
generowanie modelu fizycznego bazy danych na podstawie	Sybase PowerDesigner 12.5	pozytywny

modelu konceptualnego		
-----------------------	--	--

DOKUMENT DETALICZNY PROJEKTU dla firmy transportowej

wersja 1.0

1. Streszczenie.

Niniejszy dokument detaliczny projektu (DDP) prezentuje szczegóły pracy zespołu projektowego, nad stworzeniem aplikacji bazodanowej wspomagającej organizację pracy w firmie transportowej. Pierwsza część dokumentu zawiera opis ogólnych założeń projektowych, druga zaś opisuje wykorzystane w projekcie komponenty.

SPIS TREŚCI

1. Streszczenie
2. Wprowadzenie
 - 2.1. Cel
 - 2.2. Zakres
 - 2.3. Definicje, akronimy, skróty
 - 2.4. Odsyłacze
 - 2.5. Opracowanie
3. Projekt. Standardy, konwencje, procedury
 - 3.1. Standardy projektowe
 - 3.2. Standardy dokumentacyjne
 - 3.3. Konwencje nazewnicze
 - 3.4. Standardy programistyczne
 - 3.5. Narzędzia
4. Specyfikacja komponentów
 - 4.1. Logowanie
 - 4.2. Baza
 - 4.3. Pracownik
 - 4.4. sqljdbc4
5. Załączniki
 - 5.1. Harmonogram prac
 - 5.2. Interfejs aplikacji
 - 5.2.1. Okno powitalne
 - 5.2.2. Logowanie do aplikacji
 - 5.2.3. Aplikacja
 - 5.2.4. Rozkład jazdy

2. Wprowadzenie.

2.1. Cel.

Dokument detaliczny projektu ma za zadanie przedstawić szczegółowo sposób realizowanych prac. Zgromadzone są w nim wszystkie informacje odnośnie budowy i działania oprogramowania. Określa on również założenia projektu, standardy, narzędzia i komponenty wchodzące w skład aplikacji.

2.2. Zakres.

Założeniem projektu jest stworzenie bazodanowej aplikacji *DamberPol*, służącego do wspomagania organizacji pracy w firmie transportowej. System ma służyć do przechowywania różnego rodzaju informacji, m.in. rozkładu jazdy. Aplikacja pozwoli na magazynowanie informacji o kierowcach, pasażerach oraz pracownikach. Umożliwi pasażerom przeglądanie rozkładu jazdy.

2.3. Definicje, akronimy, skróty.

- pasażer – niezarejestrowany osoba firmy transportowej, nieposiadająca własnego loginu i hasła do systemu, posiada możliwość przeglądania zasobów rozkładu jazdy;
- pracownik – pracownik firmy, posiada własny identyfikator i hasło do logowania do systemu, posiada możliwość przeglądania zasobów (raporty, rozkłady jazdy, kierowcy zatrudnieni w firmie itp.), dodawania, edytowania już istniejących pozycji;

2.4. Odsyłacze.

Ustawa z dn. 29 sierpnia 1997 r. o ochronie danych osobowych (Dz. U. 1997 nr 133 poz. 883 z późn. zm.).

2.5. Opracowanie.

Wyżej wymieniony dokument powstał na bazie specyfikacji wymagań systemu określonych podczas pierwszych etapów projektowania systemu. Zawiera on definicje standardów, które będą przestrzegane podczas realizacji projektu. Dalsza część dokumentu zawiera informacje o narzędziach, modułach i komponentach systemu oraz interfejsie graficznym oprogramowania.

3. Projekt. Standardy, konwencje, procedury.

3.1. Standardy projektowe.

Z uwagi na niewielkie doświadczenie zespołu dot. prac projektowych, wykorzystaliśmy tzw. model przyrostowy tworzenia oprogramowania. Model ten gwarantuje częste kontakty z klientem, brak konieczności zdefiniowania z góry całości wymagań, wczesne wykorzystanie przez klienta fragmentów systemu, możliwość elastycznego reagowania na opóźnienia realizacji fragmentu – przyspieszenie prac nad inną / innymi częściami. Wyżej wymienione punkty sprawiają, że ryzyko porażki prac nad całym projektem jest dużo mniejsze.

3.2. Standardy dokumentacyjne.

Wszystkie dokumenty projektu stworzone zostały na podstawie jednego firmowego szablonu (papieru firmowego). W trakcie tworzenia dokumentacji zastosowany został jednolity, przejrzysty język, który jest wspólny dla całego dokumentu. Podczas programowania aplikacji zostały zastosowane komentarze – powodują one, że kod oprogramowania jest bardziej zrozumiały, a także umożliwiają łatwą edycję kodu aplikacji nawet przez osoby niezwiązane z pracą projektową.

3.3. Konwencje nazewnicze.

Zastosowane w projekcie słownictwo jest ukierunkowane na prostotę i jednoznaczność. W implementacji została zachowana konwencja nazewnictwa Java.

3.4. Standardy programistyczne.

System został zaprojektowany i stworzony w oparciu o język programowania Java. W projekcie będzie wykorzystywane podejście obiektowe do programowania oraz wykorzystywany będzie wzorzec projektowy MVC (Model-View-Controller). Aplikacja będzie stworzona zgodnie z wcześniejszymi założeniami i wcześniej ustalonym diagramem klas. Zaletami takiego podejścia jest brak zależności modelu od widoków aplikacji oraz łatwość dodawania i modyfikowania istniejących widoków bez wpływu na kluczową część systemu.

3.5. Narzędzia.

Do zrealizowania oprogramowania wykorzystano język Java. Środowisko RAD w projekcie zastosowano w postaci aplikacji Eclipse. Podczas tworzenia dokumentacji posługiwaliśmy się oprogramowaniem StarUML i oprogramowaniem firmy Microsoft Office 2007. Do stworzenia bazy danych wykorzystaliśmy narzędzie „PowerDesigner” firmy Sybase.

Pełna lista narzędzi wykorzystywanych podczas pracy nad systemem i dokumentacją znajduje się poniżej.

- Eclipse – producent Eclipse Foundation, licencja Eclipse Public License. (<http://www.eclipse.org>)
- Microsoft Office – producent Microsoft, licencja komercyjna. (<http://www.office.microsoft.com>)
- PowerDesigner – producent Sybase, licencja komercyjna. (<http://www.sybaseproducts.com>)
- Microsoft Paint – producent Microsoft, licencja zamknięte oprogramowanie. (<http://www.microsoft.com>)

4. Specyfikacja komponentów.

4.1. Logowanie.

Klasa odpowiedzialna za logowanie do aplikacji. Odpowiada za wygląd okna logowania oraz obsługuje samą operację logowania i sprawdzania poprawności wprowadzonych danych (tj. login i hasło). Klasa ta łączy się z klasą „baza” i poprzez nią odczytuje zawarte w niej loginy i hasła pracowników. Do jej zadań należy również porównanie wpisanych przez użytkownika danych z rekordami bazy danych zawierających loginy i hasła. W przypadku, gdy uwierzytelnienie zostanie zakończone sukcesem, pracownik jest przenoszony do okna aplikacji, która umożliwi zarządzanie bazą danych firmy. Gdy uwierzytelnienie zakończy się niepowodzeniem, użytkownik zostanie poinformowany komunikatem o błędnie wprowadzonych danych.

4.2. Baza.

Klasa odpowiedzialna za połączenie z bazą danych. W niej wykonywane są wszystkie operacje odczytu / zapisu do bazy danych. Klasa ta nie posiada interfejsu graficznego, posiada wyłącznie połączenie z bazą danych.

4.3. Pracownik.

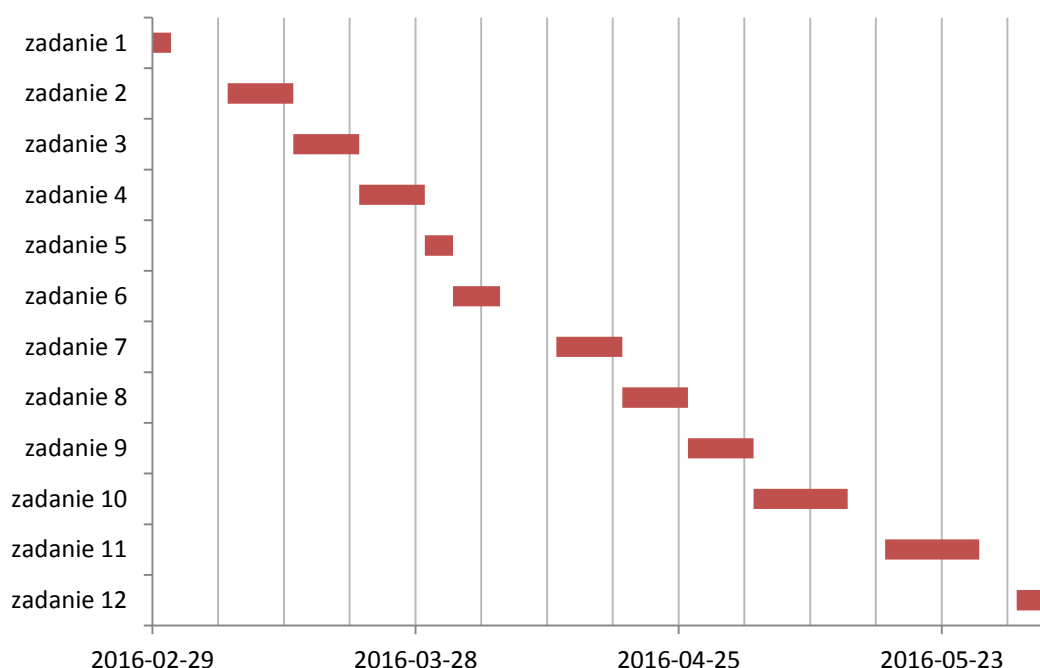
Rozbudowana baza tworząca interfejs użytkownika, przedstawiający wszystkie przyciski i kontrolki potrzebne do zarządzania bazą danych firmy. Klasa ta korzysta z klasy baza podczas łączenia się z bazą danych w celu dodania, usunięcia bądź edytowania rekordów bazy danych.

4.4. sqljdbc4

Klasa odpowiedzialna za połączenie aplikacji z bazą danych.

5. Załączniki.

5.1. Harmonogram prac.



Zadanie 3 (15.03 – 22.03). Zaprojektowanie modelu bazy danych (model konceptualny i model fizyczny), stworzenie bazy danych na potrzeby projektu.

Zadanie 6 (01.04 – 06.04). Zaprojektowanie graficznego interfejsu użytkownika. Opracowanie słownika danych zawierający specyfikację modelu.

Zadanie 8 (19.04 – 26.04). Wybór odpowiednich narzędzi do szybkiego rozwijania aplikacji (ang. RAD). Ciąg dalszy prac nad projektem (zaprogramowanie GUI, podstawowych funkcjonalności).

Zadanie 9 (26.04 – 03.05). Prace nad programowaniem dalszych funkcjonalności systemowych.

Zadanie 10 (03.05 – 13.05). Wstępne testowanie aplikacji. Naprawa ewentualnych błędów.

Zadanie 11 (17.05 – 27.05). Zakończenie programowania projektu. Przeprowadzenie szczegółowych testów. Finalizacja projektu. Sprawdzenie i zamknięcie dokumentacji. Przygotowanie projektu do zaprezentowania klientowi.

Zadanie 12 (31.05 – 03.06). Prezentacja projektu klientowi. Zakończenie prac nad projektem.

5.2. Interfejs aplikacji.

5.2.1. Okno powitalne.



5.2.2. Logowanie do aplikacji.



5.2.3. Aplikacja.

[illegible]

5.2.4. Rozkład jazdy.

Rozkład Jazdy

Przyjazd

Miasto ▾

Odjazd

Miasto ▾

Wyszukaj