

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

ZASTOSOWANIA INFORMATYKI W GOSPODARCE - PROJEKT

PRZEPŁYW INFORMACJI FIRMY SERWISOWEJ

Autorzy:

SAMIR SENHADRI 123456

ADAM SZADY 200890

MATEUSZ CHUDZIK 200755

DAWID OLEJNIK 200275

MACIEJ BOŻEMÓJ 200641

MACIEJ MOŚCIŃSKI 200893

Praca wykonana pod przewodnictwem:

DR INŻ. MAREK WODA

Wrocław, 7 czerwca 2016

Spis treści

1	Wstęp	2
1.1	Geneza projektu	2
1.2	Analiza stanu rynku	2
2	Zakres projektu	3
2.1	Cel projektu	3
2.2	Funkcjonalności podstawowe	3
2.3	Funkcjonalności rozszerzone	4
2.4	Ryzyka projektowe	4
3	Plan projektu	7
3.1	Kamienie milowe	7
3.2	Wykres Gantta	8
3.3	Rzeczywisty nakład pracy i koszty	8
4	Implementacja i wdrożenie projektu.	9
4.1	Diagram technologii wykorzystanych w projekcie	10
4.2	Instalacja projektu i wymagania sprzętowe	11
4.3	Instrukcja obsługi systemu	12
5	Podsumowanie.	12
	Spis rysunków	13
	Spis tablic.	14
	Literatura	15

1 Wstęp

Niniejszy dokument stanowi całościowe sprawozdanie z prac projektowych wykonanych w ciągu ostatniego semestru z zamiarem utworzenia wszechstronnego i otwartego systemu wspomagającego zarządzanie zleceniami w firmie serwisowej.

1.1 Geneza projektu

Idea projektu narodziła się w dość naturalny i przypadkowy sposób krótko po zawiązaniu grupy projektowej. Jeden z członków grupy usiłował od dwóch miesięcy dokonać naprawy gwarancyjnej swojego telefonu komórkowego w autoryzowanym serwisie producenta - abstrahując nawet od faktu, iż za pierwszym razem telefon nie został naprawiony poprawnie i musiał zostać ponownie oddany do naprawy, to czas reparacji urządzenia wydaje się niewspółmiernie duży w stosunku do uszkodzenia telefonu (zepsute gniazdo ładowania). Co więcej serwis nie potrafił właściwie określić, na jakim etapie naprawy znajdują się obecnie, przez co klientowi nie pozostało nic innego jak zaprzestać telefonów i cierpliwie czekać. Przypuszczalnie sytuacja taka wynikała po części z faktu, że serwis nie stosował żadnego usystematyzowanego procesu dotyczącego zarówno samych napraw jak i komunikacji z klientem. Na kanwie złych doświadczeń z serwisem powstał pomysł realizacji opensourcowego systemu upraszczającego i ujednolicającego przepływ informacji pomiędzy serwisem a jego klientem z wykorzystaniem nowoczesnych technologii webowych i mobilnych.

1.2 Analiza stanu rynku

Rynek aplikacji i stron internetowych powiązanych z tematyką serwisowania i naprawy sprzętu jest dość rozległy, jednakże większość z istniejących systemów wypada niekorzystnie w różnych aspektach w porównaniu z założonym przez nas planem działającego systemu. Najczęstszym problemem jest brak otwartości stosowanego oprogramowania, stosowanie archaicznych technologii oraz pomijanie aplikacji dedykowanej klientom serwisu. Tabela 1.1 zawiera zestawienie przykładowych systemów dotyczących tematyki serwisowania oraz porównanie z projektowanym przez nas systemem.

Tablica 1.1: Zestawienie wybranych systemów dostępnych na rynku

Nazwa systemu	Wady w porównaniu z projektowanym systemem
eSerwisowanie.pl	limit pojemności bazy danych, brak aplikacji mobilnej
MyIT CRM	brak aplikacji dla klienta, stosowane oprogramowanie adware
RepairShopr	zamknięty i płatny system
Repair TRAQ	tylko aplikacja desktopowa, naprawa wyłącznie PC, zegarków i biżuterii
ServiceMax	brak aplikacji dla klienta
RepairsLab	wyłącznie aplikacja desktopowa niedostępna dla klienta
SERWISANT	zamknięty system brak aplikacji dla klienta

2 Zakres projektu

Podstawową fazą projektu było obmyślenie wymaganych funkcjonalności systemu, z podziałem na podstawowe (muszą koniecznie zostać zrealizowane aby system działał poprawnie) i rozszerzone (powiększą możliwości lub komfort użytkowania systemu, ale nie są niezbędne do jego funkcjonowania). Następnym krokiem było stworzenie schematu bazy danych SQL, a następnie równoległe utworzenie komponentu serwerowego komunikującego się z bazą danych i aplikacji webowej i mobilnej wykorzystującej API wystawione przez część serwerową. Aplikacja mobilna w zamierzeniu ma być narzędziem wykorzystywanym wyłącznie przez klientów serwisu, przez co jej funkcjonalność jest odpowiednio ograniczona.

2.1 Cel projektu

Celem projektu było wytworzenie intuicyjnego systemu komunikacji pomiędzy firmą serwisującą urządzenia elektroniczne a jej klientem. System miał być dostępny zarówno z poziomu przeglądarki internetowej jak i aplikacji zainstalowanej na telefonie komórkowym. Docelowy produkt powinien być na tyle uniwersalny aby uniknąć konieczności dopasowania do wymagań konkretnego serwisu urządzeń elektronicznych chcącego skorzystać z usług projektowanego produktu, aby usystematyzować proces napraw i poprawić wizerunek u klienta.

2.2 Funkcjonalności podstawowe

W projekcie wyróżnione zostały następujące funkcjonalności podstawowe:

- rejestracja użytkowników w systemie,
- dodanie zgłoszenia naprawy,

- aktualizacja stanu naprawy,
- finalizowanie zgłoszenia,
- generowanie rachunku,
- przeglądanie aktualnych i archiwalnych napraw,
- odzyskiwanie hasła.

Z poziomu aplikacji webowej mamy dostęp do wszystkich funkcjonalności systemu, z poziomu aplikacji mobilnej możemy wyłącznie przeglądać naprawy wraz z ich statusami.

2.3 Funkcjonalności rozszerzone

W projekcie wyróżnione zostały następujące funkcjonalności rozszerzone:

- powiadomienie o zmianie statusu przez pocztę elektroniczną,
- wybór elementów zamiennych przez klienta,
- powiadomienie o zmianie statusu przez SMS.

Zarówno aplikacja webowa jak i mobilna umożliwiają klientowi wybór elementów zamiennych.

2.4 Ryzyka projektowe

Tablice 2.1, 2.2 i 2.3 przedstawiają przewidywane ryzyka projektowe uszeregowane zgodnie z potencjalnym zagrożeniem.

Tablica 2.1: Przewidywane ryzyka projektowe cz. 1 - wysoki priorytet

Ryzyko i konsekwencje	Prawdopodobieństwo wystąpienia [0-100]%	Wpływ na projekt [1-5]	Priorytet [wpływ * prawdopodobieństwo]	Rozwiązanie
Zmiana wizji projektu	60%	4	2.4	Krótsze sprinty
Niedotrzymanie terminu	40%	5	2	Odpowiednia komunikacja i zarządzanie zespołem. Reagowanie na czas
Niedostateczne zabezpieczenie danych personalnych klientów i członków zespołu	50%	4	2	Zakup certyfikatu SSL, przeprowadzenie testów penetracyjnych

Tablica 2.2: Przewidywane ryzyka projektowe cz. 2 - średni priorytet

Ryzyko i konsekwencje	Prawdopodobieństwo wystąpienia [0-100]%	Wpływ na projekt [1-5]	Priorytet [wpływ * prawdopodobieństwo]	Rozwiązanie
Niedoszacowanie godzin wymaganych do ukończenia projektu	75%	2	1.5	Estymacja na krótszych odcinkach czasu, przeszacowanie, szkolenie członków zespołu
Błędy programistów	30%	5	1.5	Przeprowadzanie regularnych testów
Brak motywacji	50%	3	1.5	Imprezy integracyjne
Brak innowacji w projekcie	70%	2	1.4	Dogłębny research, zmiana koncepcji
Niekompetentne zarządzania	40%	3	1.2	Zmiana lidera, dodatkowe szkolenia dla lidera
Brak synchronizacji między etapami	40%	3	1.2	Odpowiednie zaplanowanie pracy, komunikacja
Brak kompetencji członków zespołu	20%	5	1	Dodatkowe szkolenia, zmiana pracownika
Uszkodzenie sprzętu członków zespołu	20%	5	1	Regularna konserwacja, komunikacja

Tablica 2.3: Przewidywane ryzyka projektowe cz. 3 - niski priorytet

Ryzyko i konsekwencje	Prawdopodobieństwo wystąpienia [0-100]%	Wpływ na projekt [1-5]	Priorytet [wpływ * prawdopodobieństwo]	Rozwiązanie
Nieoptymalne rozwiązania techniczne	30%	3	0.9	Konsultacje ze specjalistami
Utrata członka zespołu	15%	5	0.75	Odpowiednia komunikacja i zarządzanie zespołem. Reagowanie na czas
Utrata kodu źródłowego, serwera lub bazy danych	15%	5	0.75	Kopie zapasowe, serwery bliźniacze
Dublowanie pracy	20%	2	0.4	Odpowiednia komunikacja i zarządzanie zespołem. Reagowanie na czas
Zmiany prawne powodujące konieczność implementacji bardziej skomplikowanych zabezpieczeń	10%	3	0.3	Odpowiedni research, konsultacja prawnicza
Brak dokumentacji technicznej	30%	1	0.3	Audyt jakości w trakcie pracy
Projekt jest niewykonalny technicznie	1%	0.5	0.05	Zmiana architektury, tworzenie 'proof of conceptów'

3 Plan projektu

Harmonogram projektu powstawał równolegle z procesem wyboru funkcjonalności podstawowych i rozszerzonych projektowanego systemu. Zostały wydzielone kamienie milowe projektu, oszacowano czas potrzebny na realizację poszczególnych zadań oraz stworzono wykres Gantta jako przejrzysty i prosty sposób na zarządzanie terminowością całego projektu.

Planowany czas realizacji poszczególnych funkcjonalności:

- stworzenie schematu bazy danych - 4h,
- rejestrowanie użytkowników w systemie - 10h,
- wprowadzenie zgłoszenia naprawy - 11h,
- aktualizowanie stanu naprawy - 8h,
- finalizowanie zgłoszenia wraz z generowaniem rachunku - 9h,
- przeglądanie napraw - 9h,
- odzyskiwanie hasła - 5h.

Sumaryczny przewidywany czas realizacji funkcjonalności wyniósł 56h.

3.1 Kamienie milowe

Wydzielone zostały trzy kamienie milowe projektu:

- project kickoff (01.03.2016) - oficjalny start projektu, moment wieńczący etap działań koncepcyjnych, na które składały się m.in wybór tematu, określenie funkcjonalności systemu i podział ról w zespole,
- prototyp (26.04.2016) - zakończenie prac nad prototypem aplikacji webowej i mobilnej, działająca baza danych i serwer aplikacji, zrealizowane wszystkie funkcjonalności podstawowe,
- etap finalny (07.06.2016) - zakończenie implementacji projektu, system w pełni sprawny, zrealizowane wszystkie funkcjonalności podstawowe i jak największy procent funkcjonalności rozszerzonych, ukończona dokumentacja projektowa.

3.2 Wykres Gantta

Wykres Gantta wykorzystany w projekcie pozwolił kierownikowi projektu na sprawne szacowanie postępów prowadzonych prac oraz określaniu problemów osiągnięciem wcześniej wyznaczonych terminów. Pierwsza wersja diagramu Gantta powstała przed pierwszym krokiem milowym projektu i została przedstawiona na rysunku 3.1. Zielone linie na diagramie symbolizują drugi i trzeci kamień milowy.

Druga wersja diagramu Gantta powstała krótko przed osiągnięciem drugiego kamienia milowego. Został uzupełniony postęp poszczególnych funkcjonalności aplikacji, a do listy zadań dodane zostały wybrane funkcjonalności rozszerzone. Poprawiony diagram znajduje się na rysunku 3.2.

Trzecia wersja diagramu Gantta powstała na sam koniec prac projektowych. Ponownie zaktualizowany został postęp poszczególnych funkcjonalności aplikacji, większość których została już ukończona w 100%. Finalną wersję diagramu przedstawia rysunek 3.3.

3.3 Rzeczywisty nakład pracy i koszty

Ze względu na otwarty charakter dostarczanego oprogramowania i brak planowanych bezpośrednich zysków z przedsięwzięcia zdecydowano o nieokreśleniu średniej stawki godzinowej pracy członka zespołu w projekcie. Konsekwentnie nie zostały obliczone sumaryczne koszty wytworzenia kompletnego produktu, jedynie nakład pracy wyrażony w godzinach. Rzeczywisty sumaryczny nakład pracy w projekcie różni się od antycypowanego w znaczącym stopniu ze względu na to, że podczas oryginalnego szacunku pominięto implementację funkcjonalności rozszerzonych aplikacji oraz stworzenie aplikacji mobilnej na urządzenia z systemem Android. Pierwotny plan czasu pracy okazał się jednak być bardzo celny w zakresie funkcjonalności podstawowych systemu. Rzeczywisty czas realizacji poszczególnych funkcjonalności był następujący (w nawiasach wartości przewidywane):

- stworzenie schematu bazy danych - 4h (4h),
- rejestrowanie użytkowników w systemie - 8h (10h),
- wprowadzenie zgłoszenia naprawy - 9h (11h),
- aktualizowanie stanu naprawy - 11h (8h),
- finalizowanie zgłoszenia wraz z generowaniem rachunku - 8h (9h),
- przeglądanie napraw - 8h (9h),
- odzyskiwanie hasła - 5h (5h).
- powiadomienie o statusie przy użyciu e-mail - 3h (brak predykcji)

- powiadomienie o statusie przy użyciu SMS - 12h (brak predykcji)
- implementacja aplikacji mobilnej - 35h (brak predykcji)

Sumaryczny przewidywany czas realizacji funkcjonalności wyniósł 56h, a rzeczywisty czas realizacji funkcjonalności które zostały uwzględnione w predykcjach był o 3 godziny krótszy (spadek o 5.4 %). Sumaryczny czas wszystkich prac implementacyjnych wynosi 103h, co daje wzrost względem wartości planowanej o 83.9 %, jednak należy powtórnie podkreślić że tak duża różnica wynika z nieuwzględnienia prac nad aplikacją mobilną w trakcie pierwotnych predykcji.

4 Implementacja i wdrożenie projektu

W ramach realizacji projektu zaimplementowana została wielowarstwowa aplikacja internetowa z rozszerzeniem o aplikację mobilną. Model bazy danych stworzono zgodnie z podejściem „Database First” z wykorzystaniem technologii ADO.NET. Jako narzędzie do mapowania obiektowo relacyjnego wykorzystano platformę Entity Framework, natomiast zapytania do bazy danych zostały napisane w języku LINQ. Warstwa dostępu do danych została odseparowana od logiki aplikacji z wykorzystaniem warstwy logiki biznesowej. Utworzone zostały modele transferu danych odzwierciedlające modele warstwy dostępu do danych oraz modele widoku je zawierające, po to aby uniknąć korzystania z modeli warstwy dostępu do danych w warstwie aplikacji.

Warstwa logiki aplikacji oraz warstwa prezentacji zostały zrealizowane w oparciu o platformę ASP.NET MVC 5. Autentykację użytkowników uzyskano z wykorzystaniem ASP.NET Identity. Do tworzenia widoków aplikacji wykorzystano składnię Razor, a sterowanie aplikacją zrealizowano za pomocą składnika platformy MVC o nazwie HTML Helper; wykorzystano również składniki biblioteki jQuery UI. Od strony wizualnej widoki zostały zaprojektowane z wykorzystaniem biblioteki Bootstrap oraz autorskich stylów CSS.

Dzięki ogromnym możliwościom dostarczonym przez Entity Framework pomimo relatywnie skomplikowanej struktury bazy danych operacje typu CRUD nie przysparzają żadnego problemu. Wielką zaletą stosowanego narzędzia jest fakt, że obiekty bazodanowe zawierają referencje do powiązanych ze sobą tabel, dzięki czemu w bardzo prosty sposób można pobrać z bazy dane z powiązanych ze sobą tabel. Do mapowania obiektów warstwy dostępu do danych na obiekty transferowe wykorzystano Automapper. Podejście takie daje gwarancję że w razie rozbudowy bazy danych wystarczy uzupełnić obiekty warstwy transferu danych o nowe pola, nie trzeba natomiast przejmować się mapowaniem.

Aplikacja mobilna została napisana w języku Android i została zaprojektowana na urządzenia z minimalną wersją systemu operacyjnego 4.0 (Ice Cream Sandwich). Do komunikacji z serwerem wykorzystywane jest własnoręcznie stworzone API bazujące na metodach HTTP POST i GET, a dane są przesyłane w formacie JSON. Aplikacja dzieli się na graficzny interfejs użytkownika (zawarty w plikach

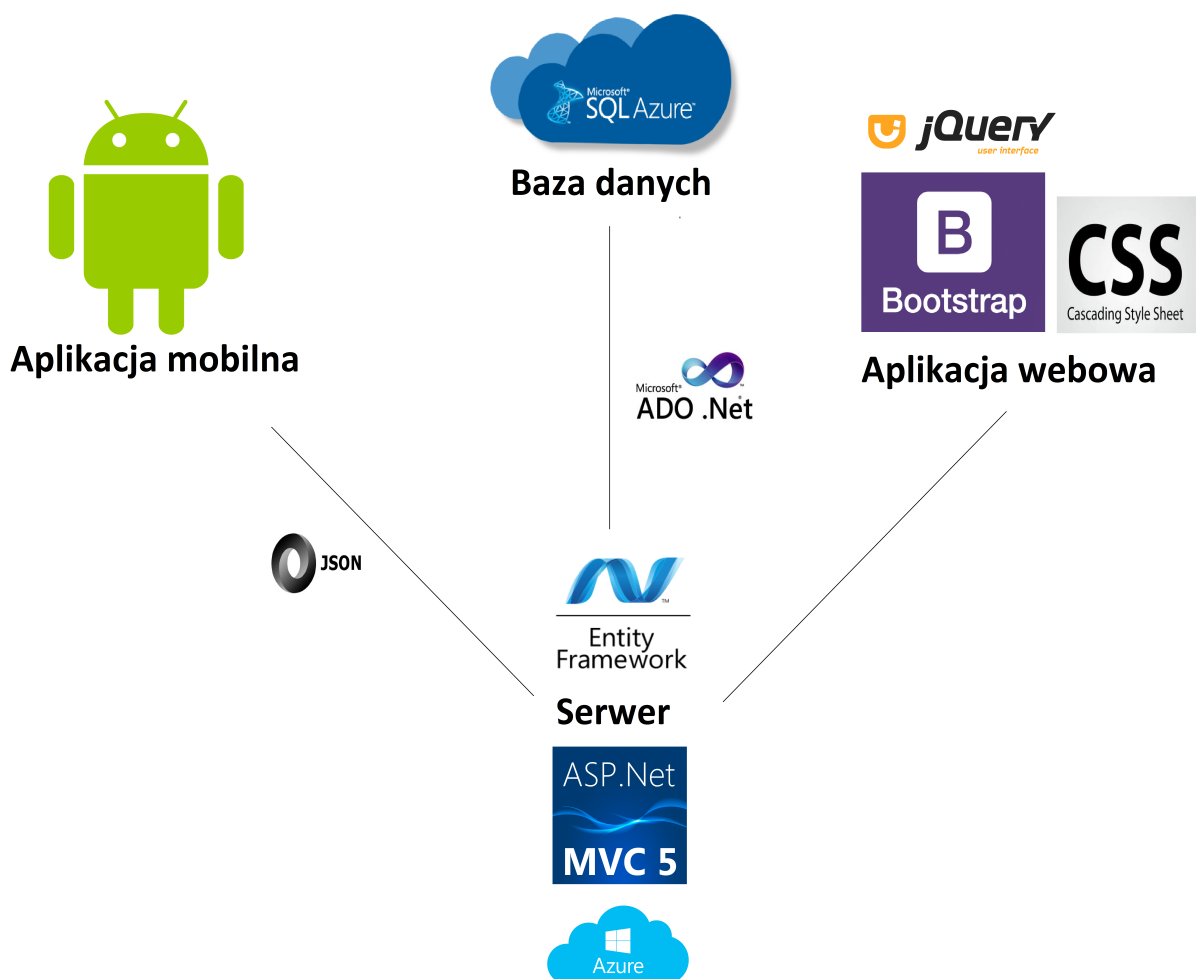
xml) oraz część backendową (pliki java). Pierwszym widokiem otwieranym bezpośrednio po uruchomieniu aplikacji jest `activity_login.xml`, a klasy i metody tej aktywności zostały zaimplementowane w pliku `LoginActivity.java`. Wszystkie kolejne aktywności zostały zaimplementowane analogicznie jak w powyższym przykładzie.

Wśród najważniejszych aktywności aplikacji mobilnej należy wyróżnić:

- `activity_login.xml` - ekran logowania aplikacji wyświetlający logo aplikacji i pozwalający na zmianę języka,
- `activity_panel_components.xml` - wyświetlenie dostępnej listy części zamiennych,
- `activity_panel_services.xml` - wyświetlenie listy zgłoszeń użytkownika pokolorowanych w zależności od ich statusu.

4.1 Diagram technologii wykorzystanych w projekcie

Rysunek 4.1 przedstawia schemat połączeń pomiędzy poszczególnymi komponentami projektu i wykorzystane technologie.



Rysunek 4.1: Diagram wykorzystanych technologii

4.2 Instalacja projektu i wymagania sprzętowe

Ze względu na charakter projektu (open source) fizyczny proces wdrożenia aplikacji we własnym serwisie napraw leży w gestii właściciela serwisu. Jeśli posiada on własny serwer z bazą danych MS SQL i działającą usługą internetową IIS wystarczy wykonać następujące kroki:

- folder zawierający stronę internetową umieścić w folderze `C:\inetpub\wwwroot`,
- przejść do `start -> uruchom -> inetmgr` aby otworzyć okno zarządzania aplikacjami internetowymi i uruchomić serwer (patrz rysunek 4.2).

Jeśli jednak użytkownik nie posiada takiej możliwości zachodzi konieczność wykupienia hostingu. Przykładowym portalem oferującym zarówno serwer usług internetowych IIS jak i serwer bazodanowy MS SQL jest witryna somee.com. Po ukończeniu procesu rejestracji i wykupieniużądanego przez nas

planu usługowego możemy utworzyć swoją stronę internetową, co przedstawiają rysunki 4.3 i 4.4. Następnym krokiem jest instalacja naszej aplikacji na zdalnym serwerze. Proces ten znacząco może uprościć IDE Visual Studio, wymagając od użytkownika jedynie kliknięcia opcji **publish** i wybrania profilu publikacji. Przedstawiają to rysunki 4.5 i 4.6. Ostatnim brakującym elementem układanki jest połączenie serwera z bazą danych. Analogicznie jak w przypadku samej aplikacji, jeśli nie mamy możliwości umieszczenia jej na własnym serwerze jesteśmy zmuszeni skorzystać z hostingu wykonując proces zbliżony do publikowania samej aplikacji. Dodatkowo aby zapewnić komunikację pomiędzy aplikacją a bazą danych należy zdefiniować odpowiedni connection string w pliku konfiguracyjnym `Web.config`. Czynności te zostały zaprezentowane na rysunkach od 4.7 do 4.9.

Wdrożenie aplikacji mobilnej jest relatywnie prostym procesem. Po umieszczeniu aplikacji webowej na wybranym serwerze należy zmienić wykorzystywany URL w czterech plikach źródłowych aplikacji (`services_details.java`, `panel_services.java`, `panel_components.java` i `LoginActivity.java`) na adres działającego własnego serwera, a następnie zbudować całą aplikację od początku. Otrzymany plik `*.apk` możemy już udostępnić do pobrania klientom serwisu np. poprzez istniejącą stronę internetową.

4.3 Instrukcja obsługi systemu

5 Podsumowanie

Spis rysunków

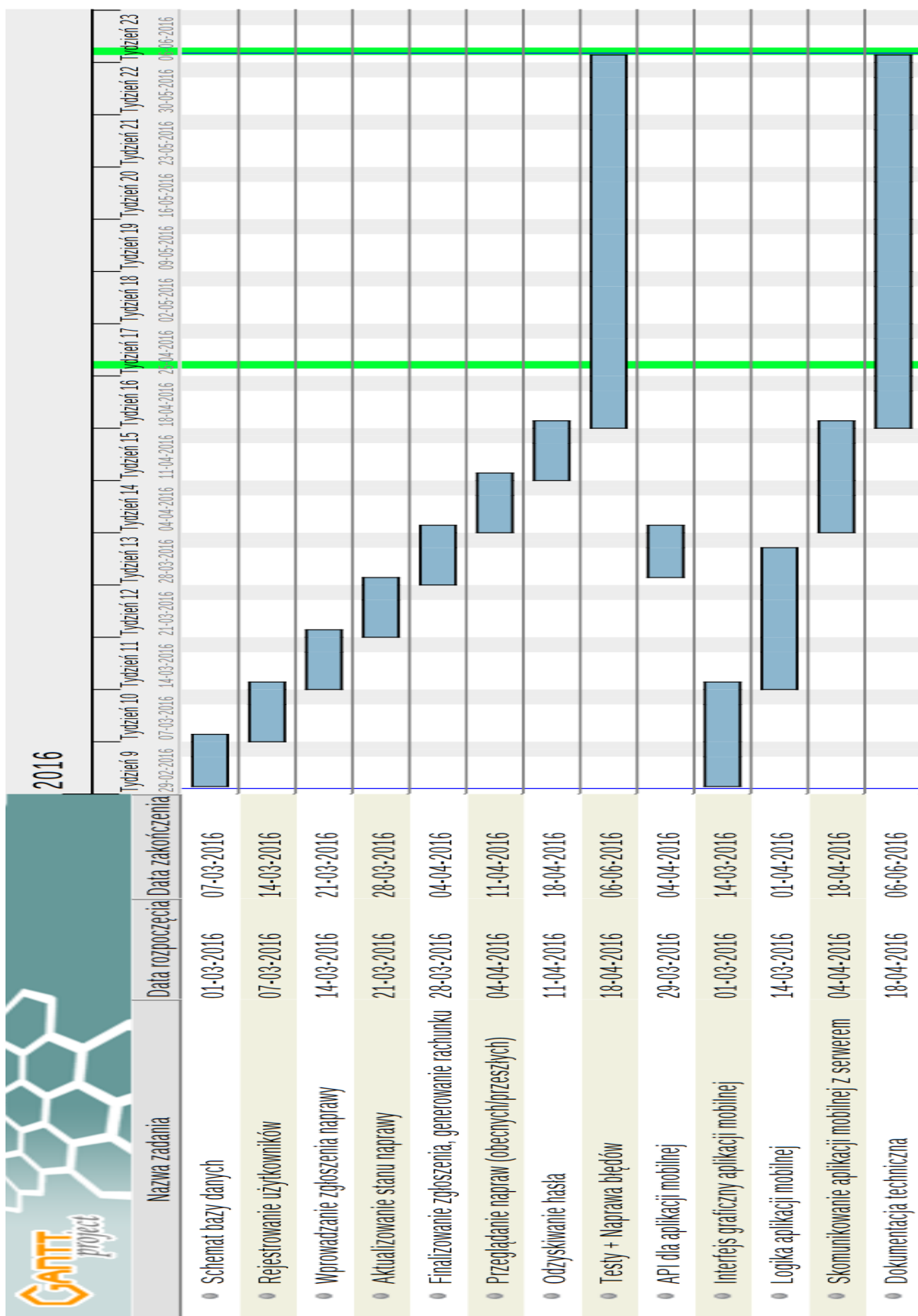
4.1	Diagram wykorzystanych technologii	11
3.1	Wykres Gantt'a - pierwszy kamień milowy	16
3.2	Wykres Gantt'a - drugi kamień milowy	17
3.3	Wykres Gantt'a - trzeci kamień milowy	18
4.2	Ręczne uruchomienie serwera na lokalnej maszynie	19
4.3	Zakładanie strony internetowej cz.1	20
4.4	Zakładanie strony internetowej cz.2	21
4.5	Tworzenie profilu	22
4.6	Wybór adresu serwera i publikacja projektu	23
4.7	Hosting bazy danych	24
4.8	Ustawienia połączenia	25
4.9	Publikacja bazy danych	26

Spis tablic

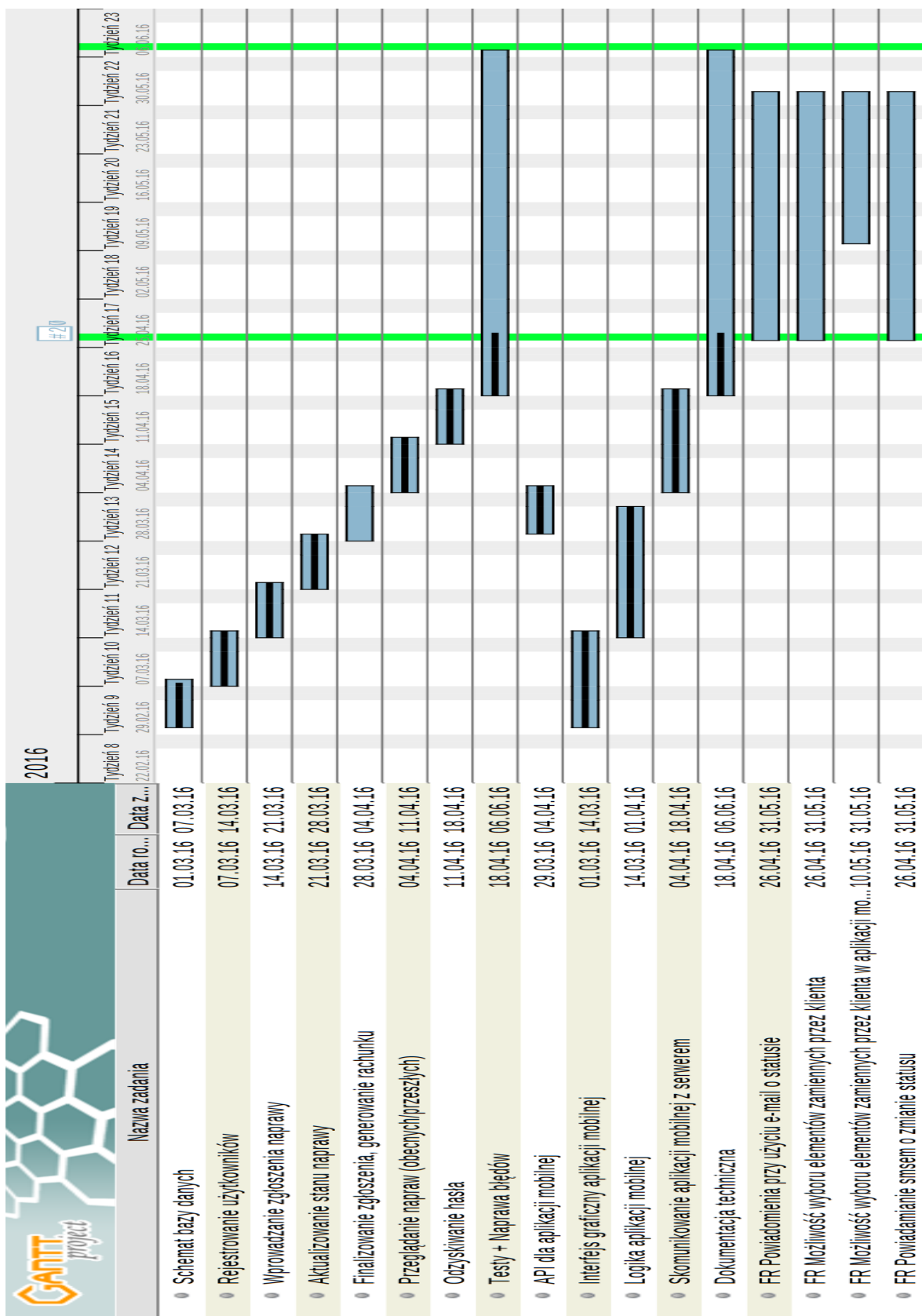
1.1	Zestawienie wybranych systemów dostępnych na rynku	3
2.1	Przewidywane ryzyka projektowe cz. 1 - wysoki priorytet	5
2.2	Przewidywane ryzyka projektowe cz. 2 - średni priorytet	5
2.3	Przewidywane ryzyka projektowe cz. 3 - niski priorytet	6

Literatura

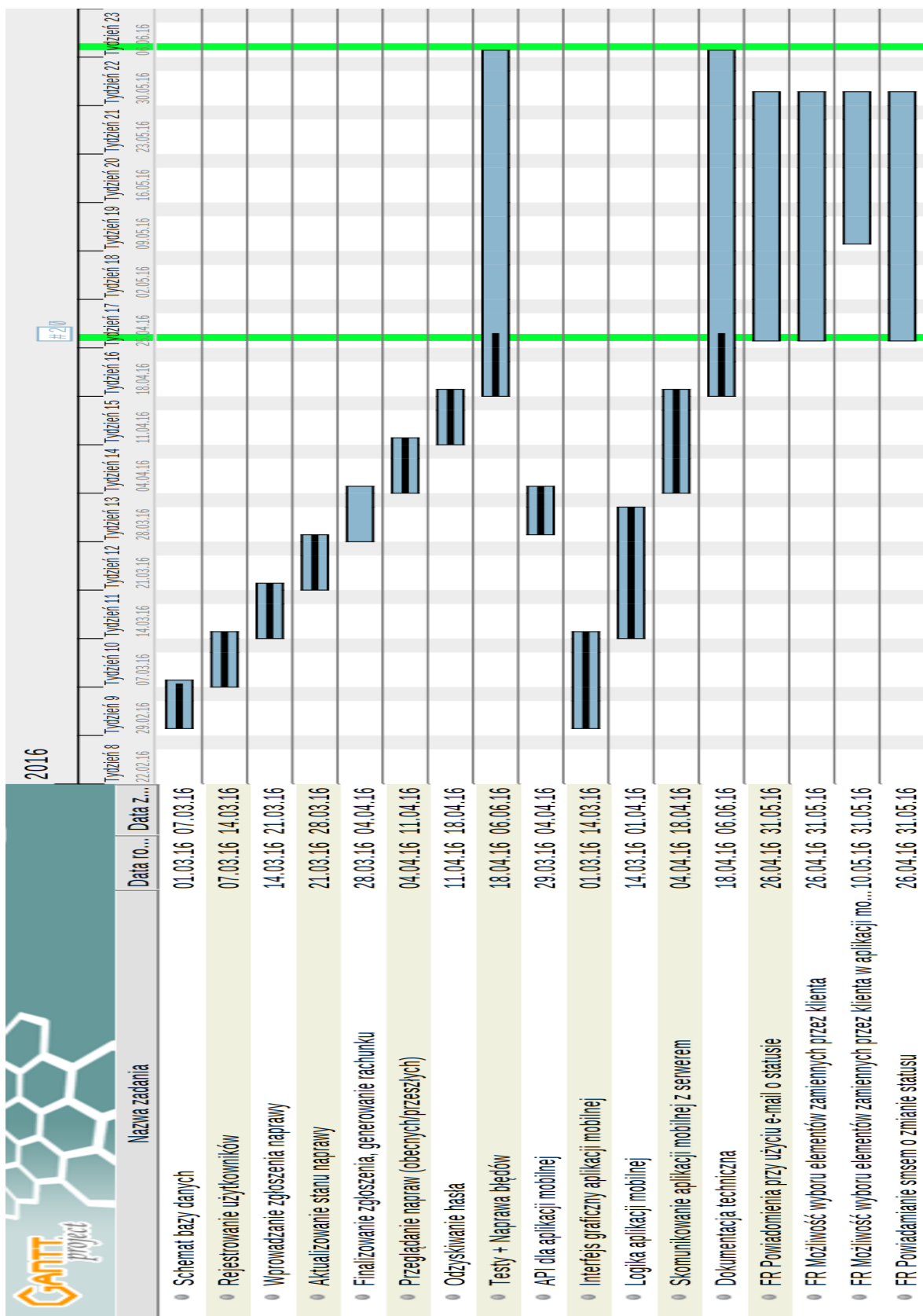
- [1] Cormen T., Leiserson C., Rivest R., Wprowadzenie do algorytmów, WNT, 2001.
- [2] Błażewicz J., Problemy optymalizacji kombinatorycznej, PWN, Warszawa, 1996.
- [3] Strona internetowa: <http://www.cs.put.poznan.pl/wkotlowski/teaching/8-ts.pdf>, 25.01.2013r.



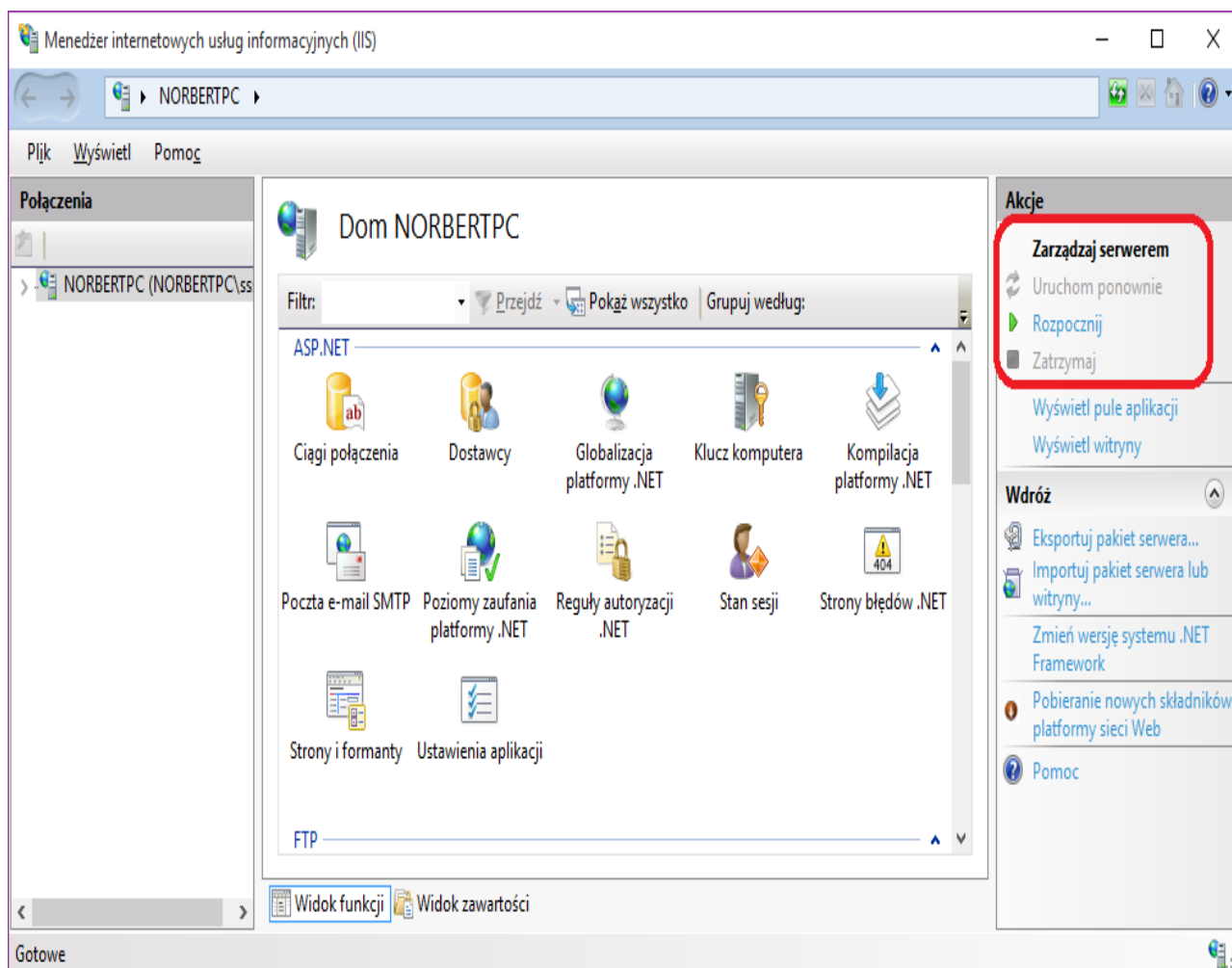
Rysunek 3.1: Wykres Gantt'a - pierwszy kamień milowy



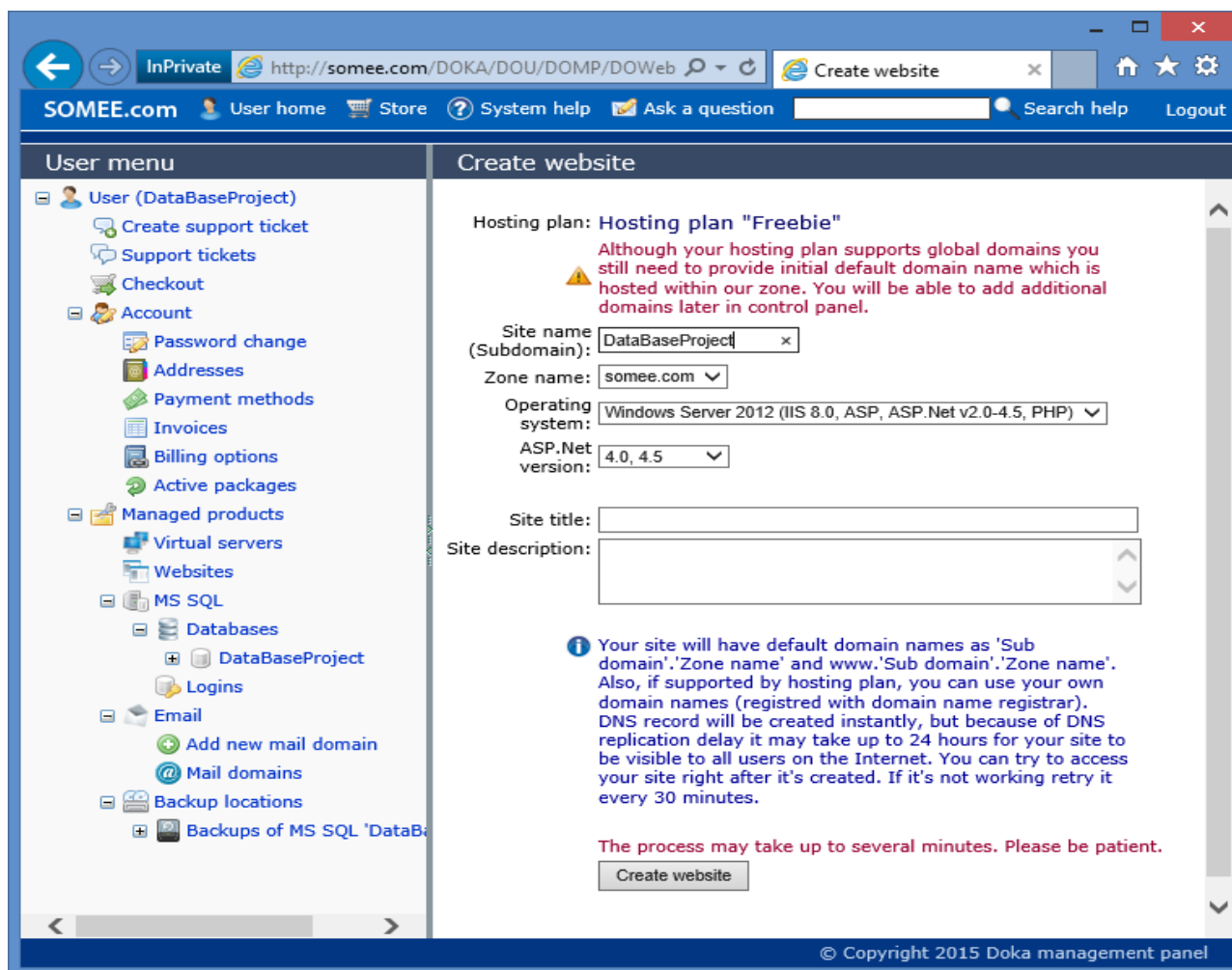
Rysunek 3-2: Wykres Gantt'a - drugi kamień milowy



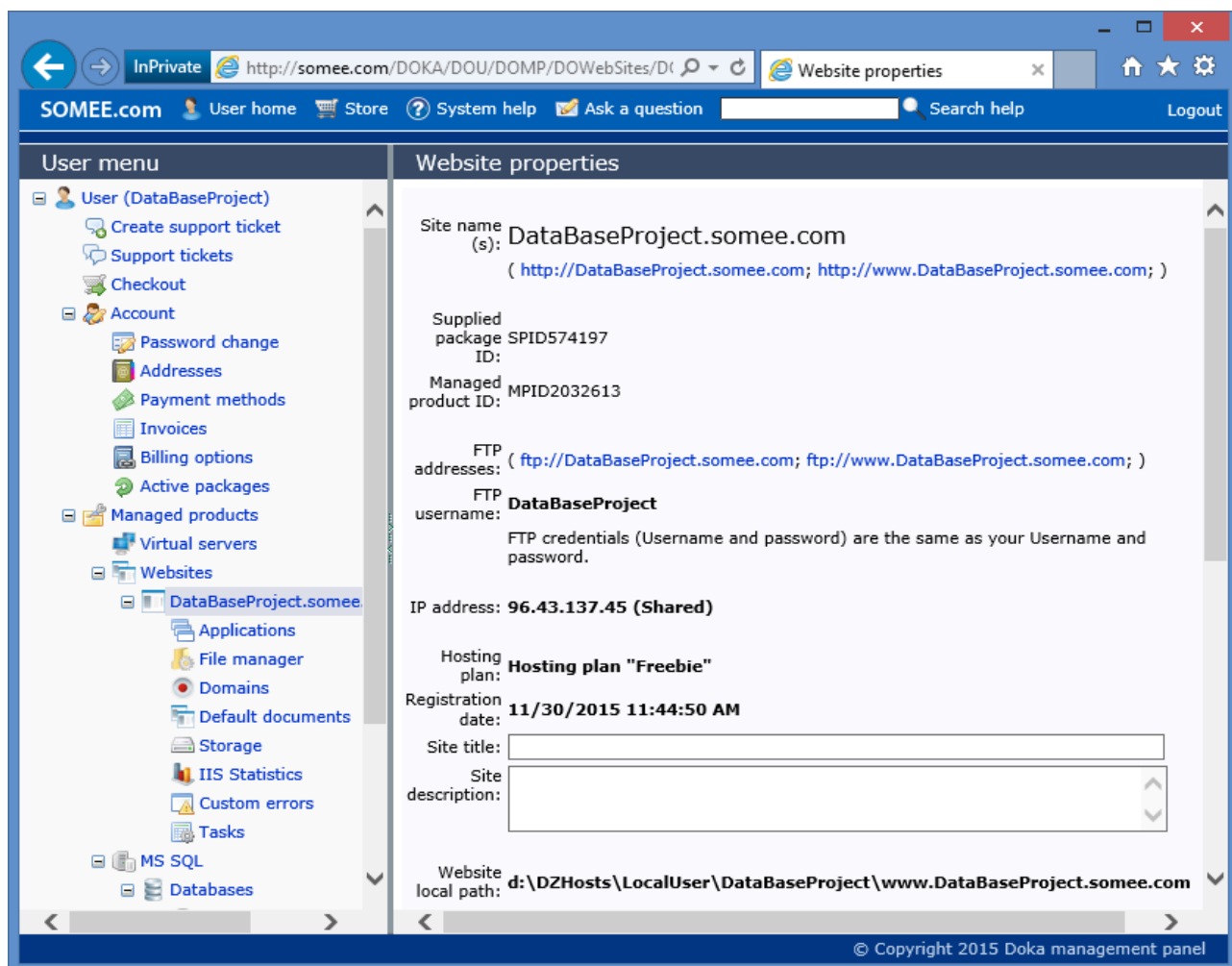
Rysunek 3-3: Wykres Gantt'a - trzeci kamień milowy



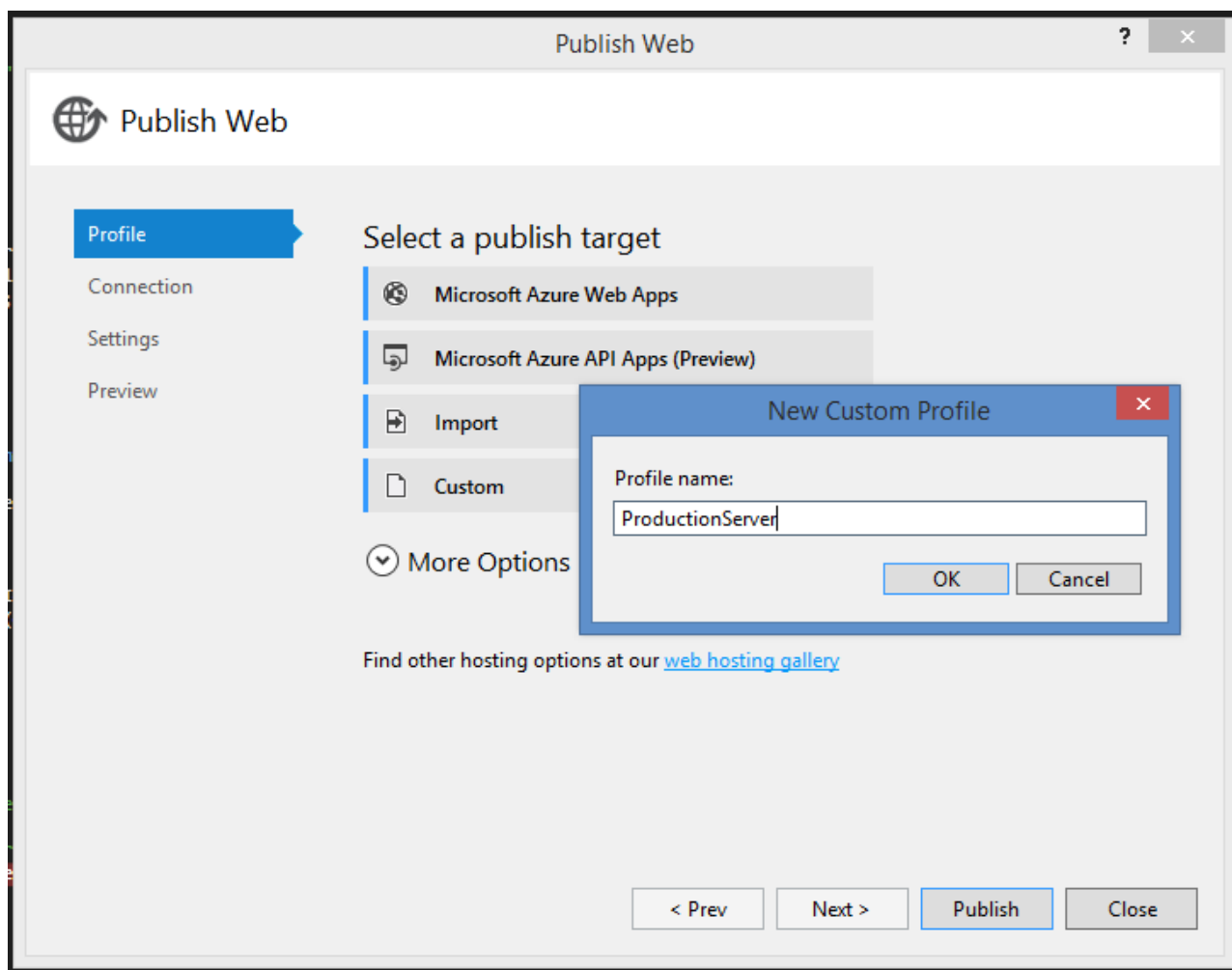
Rysunek 4.2: Ręczne uruchomienie serwera na lokalnej maszynie



Rysunek 4.3: Zakładanie strony internetowej cz.1




Rysunek 4.4: Zakładanie strony internetowej cz.2



Rysunek 4.5: Tworzenie profilu

Publish Web

 Publish Web

Profile

Connection

Settings

Preview

ProductionServer

Publish method: FTP

Server: ftp://DataBaseProject@www.databaseproject.somee.com/

Site path: www.DataBaseProject.somee.com/dbProject


☒ Passive mode

User name: DataBaseProject

Password: ●●●●●●

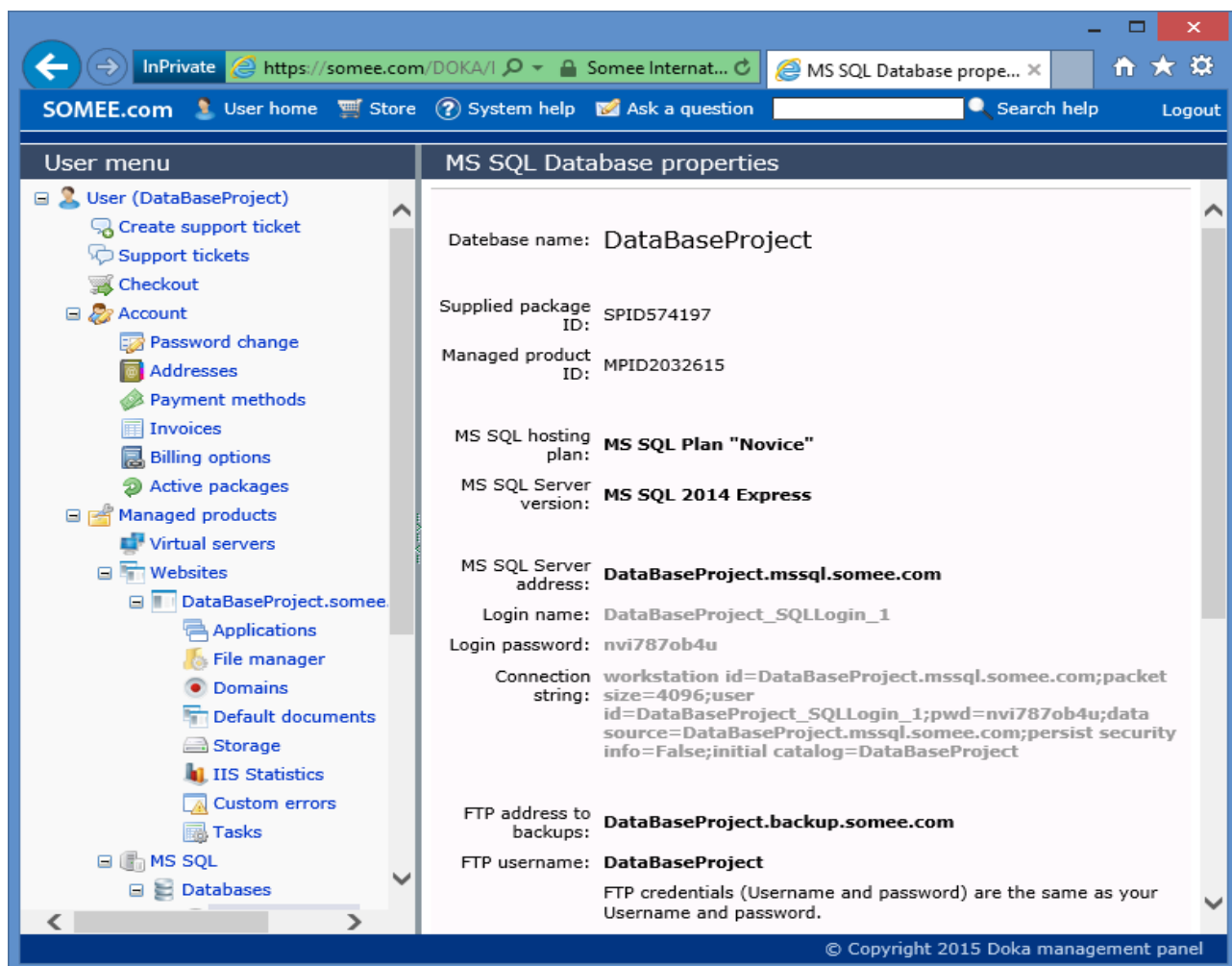
☐ Save password

Destination URL: e.g. http://www.contoso.com

Validate Connection 

< Prev Next > Publish Close

Rysunek 4.6: Wybór adresu serwera i publikacja projektu



Rysunek 4.7: Hosting bazy danych

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
DataBaseProject.mssql.somee.com Refresh

Log on to the server

☐ Use Windows Authentication

☒ Use SQL Server Authentication

User name: DataBaseProject_SQLLogin_1

Password: ••••••••

☐ Save my password

Connect to a database

☒ Select or enter a database name:
DataBaseProject

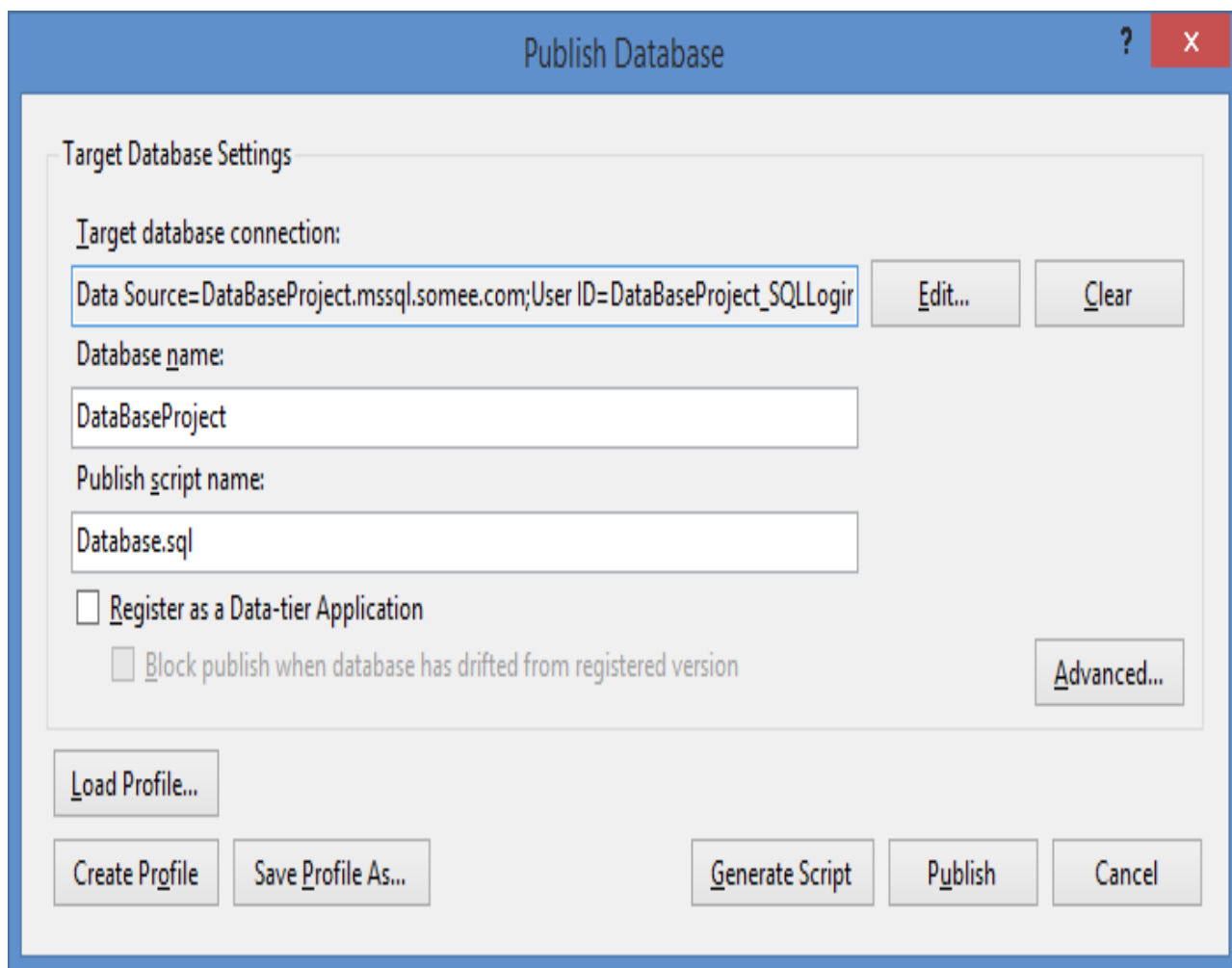
☐ Attach a database file:
Browse...

Logical name:

Advanced...

Test Connection OK Cancel

Rysunek 4.8: Ustawienia połączenia



Rysunek 4.9: Publikacja bazy danych