Alternatywny system synchronizacji.8

Projekt PPZ

Autorzy:
Jakub Ciałkowski
Łukasz Pijarczyk
Przemysław Sendek
Tomasz Wójcik
Paweł Zembrzuski

Warszawa, dn. 08.05.09

Spis treści:

1. Informacje o systemie.

Program ASS.8 powstał jako projekt na zajęcia Pracownia Projektowania Zespołowego w Wyższej Szkole Informatyki Stosowanej i Zarządzania w Warszawie. Głównym zadaniem programu jest umożliwienie współdzielenia plików między użytkownikami oraz synchronizacja danych za pośrednictwem serwera. Użytkownik ma możliwość współdzielenia danych dzięki klientowi usługi natomiast portal WWW daje mu możliwość logowania się i aktywnego uczestnictwa w społeczności tworzonej przez użytkowników.

1.1. Specyfikacja systemu.

System składa się z czterech części:

• klienta – programu instalowanego na komputerze użytkownika. Program ten ma za zadanie ułatwiać organizację plików, określanie ich praw dostępu dla innych użytkowników systemu oraz wyszukiwanie innych użytkowników systemu.

• serwer – który łączy się z bazą danych oraz jest pośrednikiem między klientem a bazą danych.

• bazy danych – gdzie przechowywane informacje o użytkownikach i ich plikach.

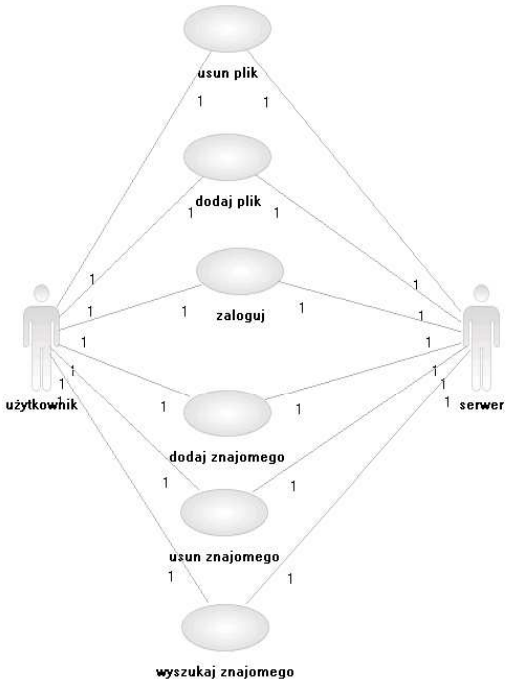
• serwisu WWW – strony na której dany użytkownik systemu może się zalogować, dzięki czemu uzyskuje dostęp do swojego profilu widocznego dla innych użytkowników. Osoba niezalogowana może przeglądać profile zarejestrowanych użytkowników i ma dostęp do plików oznaczonych jako publiczne.

Pliki mają trzy poziomy dostępności dla użytkowników:

• publiczny – każdy użytkownik systemu oraz osoba niezarejestrowana ma dostęp do pliku.
• prywatny – dostęp do pliku mają tylko przyjaciele danego użytkownika.
• osobisty – dostęp do pliku ma tylko jego właściciel.

Zgodnie z przedstawionym poniżej diagramem system daje użytkownikowi następujące możliwości:

1. Logowanie się do systemu.
2. Wyszukiwanie znajomych.
3. Dodawanie znajomych.
4. Usuwanie znajomych.
5. Dodawanie plików.
6. Usuwanie plików.



1.2. Przegląd zastosowanych narzędzi.

W celu osiągnięcia jak najlepszych efektów podczas tworzenia systemu ASS.8, do jego wykonania zostały wykorzystane następujące narzędzia:

- Języki programowania C++ i C# przy tworzeniu serwera oraz klienta usługi.
- MySQL przy tworzeniu bazy danych.
- Framework Django przy tworzeniu serwisu WWW.

1.2.1. Język programowania C++.

C++ to język programowania ogólnego przeznaczenia.
Umożliwia abstrakcję danych oraz stosowanie kilku paradygmatów programowania: proceduralnego, obiektowego i generycznego. Charakteryzuje się wysoką wydajnością kodu wynikowego, bezpośrednim dostępem do zasobów sprzętowych i funkcji systemowych, łatwością tworzenia i korzystania z bibliotek (napisanych w C++, C lub innych językach), niezależnością od konkretnej platformy sprzętowej lub systemowej (co gwarantuje wysoką przenośność kodów źródłowych) oraz niewielkim środowiskiem uruchomieniowym. Podstawowym obszarem jego zastosowań są aplikacje i systemy operacyjne.

C++ został zaprojektowany przez Bjarne'a Stroustrupa jako rozszerzenie języka C o obiektowe mechanizmy abstrakcji danych i silną statyczną kontrolę typów. Zachowanie zgodności z językiem C na poziomie kodu źródłowego pozostaje jednym z podstawowych celów projektowych kolejnych standardów języka.

Źródło: http://pl.wikipedia.org/wiki/C%2B%2B

1.2.2. Język programowania C#.

C# - obiektowy język programowania zaprojektowany przez zespół pod kierunkiem Andersa Hejlsberga dla firmy Microsoft.
Program napisany w tym języku kompilowany jest do języka Common Intermediate Language (CIL), specjalnego kodu pośredniego wykonywanego w środowisku uruchomieniowym takim jak .NET Framework, Mono lub DotGNU. Wykonanie skompilowanego programu przez system operacyjny bez takiego środowiska nie jest możliwe.

Cechy języka:
Język C# ma wiele cech wspólnych z językami programowania Object Pascal, Delphi, C++ i Java.

- osadzane języki proceduralne wykonywane przez bazę danych (plperl, pl/perlu, plpgsql, plpython, pltcl, pl/tcl)
- obiektowość z hierarchią o jednym elemencie nadrzędnym: podobnie jak w Javie, kod programu jest zbiorem klas. W C# podobnie tak jak w Javie/Object Pascalu hierarchia dziedziczenia opiera się na istnieniu jednej klasy object (System.Object), która stanowi element nadrzędny tej hierarchii. W szczególności oznacza to, że również typy proste (int, double, itd.) są obiektami z właściwymi sobie metodami, np. int i=1; string s = i.ToString();
- odśmiecanie pamięci: zarządzaniem pamięcią zajmuje się środowisko uruchomieniowe. Oznacza to, że nie ma konieczności ani potrzeby samodzielnego zajmowania się zwalnianiem pamięci po obiektach, które przestają być używan
- refleksje i atrybuty klas: w czasie pracy programu istnieje możliwość analizy struktury kodu z poziomu tego kodu. Umożliwia to tworzenie wysoce uniwersalnych mechanizmów operujących na strukturze kodu nie znanej w czasie kompilacji. Mechanizm ten wykorzystywany jest m.in. w bibliotekach ORM, narzędziach do analizy i weryfikacji kodu czy rozszerzeniach AOP. Mechanizm atrybutów został z C# zapożyczony do języka Java w wersji 1.5 (adnotacje), jakkolwiek samo Reflection API istniejące od pierwszego wydania języka stanowiło inspirację dla twórców C#
- typy ogólne (generics - dostępne od wersji .NET 2.0): mechanizm zbliżony swoją ogólnością do szablonów w C++, jednak tu typ ogólny jest przenoszony do modułu binarnego i możliwy jest do wykorzystania bez konieczności posiadania kodu źródłowego (szablony w C++ to w uproszczeniu rozbudowane makrodefinicje)
- dynamiczne tworzenie kodu: biblioteki .NET umożliwiają dynamiczne tworzenie kodu w czasie działania programu i włączanie go do kodu aktualnie wykonywanego. Możliwe jest zarówno dynamiczne tworzenie kodu wykonywalnego ze źródeł C# jak i tworzenie dynamicznych modułów w języku pośrednim (MSIL).
- bogata biblioteka klas BCL, umożliwiająca rozwijanie aplikacji konsolowych, okienkowych (System.Windows.Forms), bazodanowych (ADO.NET), sieciowych (System.Net), w architekturze rozproszonej (WebServices) czy dynamicznych aplikacji internetowych (ASP.NET)

Źródło: http://pl.wikipedia.org/wiki/C_Sharp

1.2.3. Język programowania MySQL.

MySQL był pisany raczej z myślą o szybkości niż kompatybilności ze standardem SQL – przez dłuższy czas MySQL nie obsługiwał nawet transakcji, co było zresztą głównym argumentem przeciwników tego silnika bazodanowego. MySQL obsługuje większą część obecnego standardu ANSI/ISO SQL (tj. SQL:2003). Wprowadza również swoje rozszerzenia i nowe elementy języka

Serwer MySQL dostępny jest dla wszystkich popularnych platform systemowych i różnorakich architektur procesorów. Jest dostępny także w wersji źródłowej, co umożliwia skompilowanie go dla dowolnej innej platformy.

Źródło: http://pl.wikipedia.org/wiki/MySQL

1.2.4. Język programowania Django.

Django - wysokopoziomowy, opensource'owy framework służący do tworzenia aplikacji internetowych, napisany w Pythonie. Powstał w środowisku dziennikarskim więc maksymalnie skupia się na szybkości tworzenia takich aplikacji oraz automatyzacji powtarzających się czynności tego procesu. Jego nazwa pochodzi od imienia gitarzysty Django Reinhardta.

- automatycznie generowany i kompletny panel administracyjny, z możliwością dalszego dostosowywania.
- przyjazne adresy dokumentów z możliwością dowolnego ich kształtowania.
- prosty lecz funkcjonalny system szablonów czytelny zarówno dla grafików jak i dla programistów.
- oddzielenie logiki aplikacji (kontroler) logiki biznesowej (model) wyglądu (szablony) oraz baz danych.
- wsparcie dla wielojęzycznych aplikacji.
- bardzo duża skalowalność i wydajność pod obciążeniem.
- wydajne systemy keszowania, obsługa memcache.
- własny, prosty serwer do testowania aplikacji.
- współpracuje z Apache poprzez mod_python oraz z innymi serwerami poprzez protokoły FastCGI i SCGI.
- DRY czyli zasada „nie powtarzaj się” w odniesieniu do tworzenia aplikacji, (np. strukturę baz mysqla django generuje ze zwykłych klas pythona).
- posiada ORM wysokiego poziomu pozwalający na łatwe i bezpieczne operowania na bazach danych bez użycia SQL.
- obsługuje PostgreSQL, MySQL, SQLite oraz Oracle.
- rozpowszechniany jest na liberalnej licencji BSD.

Źródło: http://pl.wikipedia.org/wiki/Django

2. Struktura systemu.

Poniższy diagram klas przedstawia uproszczoną strukturę systemu ASS.8. Klasy typu baza, baza kont i baza plików nie są klasami w systemie w systemie lecz stanowią one fizyczną bazę danych.

2.1. Opis głównych struktur systemu.

Głównymi strukturami systemu ASS.8 są :

• klient – program instalowany na komputerze użytkownika. Program ten ma za zadanie ułatwiać organizację plików, określanie ich praw dostępu dla innych użytkowników systemu oraz wyszukiwanie innych użytkowników systemu.

• serwer – który łączy się z bazą danych oraz jest pośrednikiem między klientem a bazą danych.

• baza danych – gdzie przechowywane są pliki i informacje.

• serwis WWW – strony na której dany użytkownik systemu może się zalogować, dzięki czemu uzyskuje dostęp do swojego profilu widocznego dla innych użytkowników. Osoba niezalogowana może przeglądać profile zarejestrowanych użytkowników i ma dostęp do plików oznaczonych jako publiczne.

2.1.1. Klient.

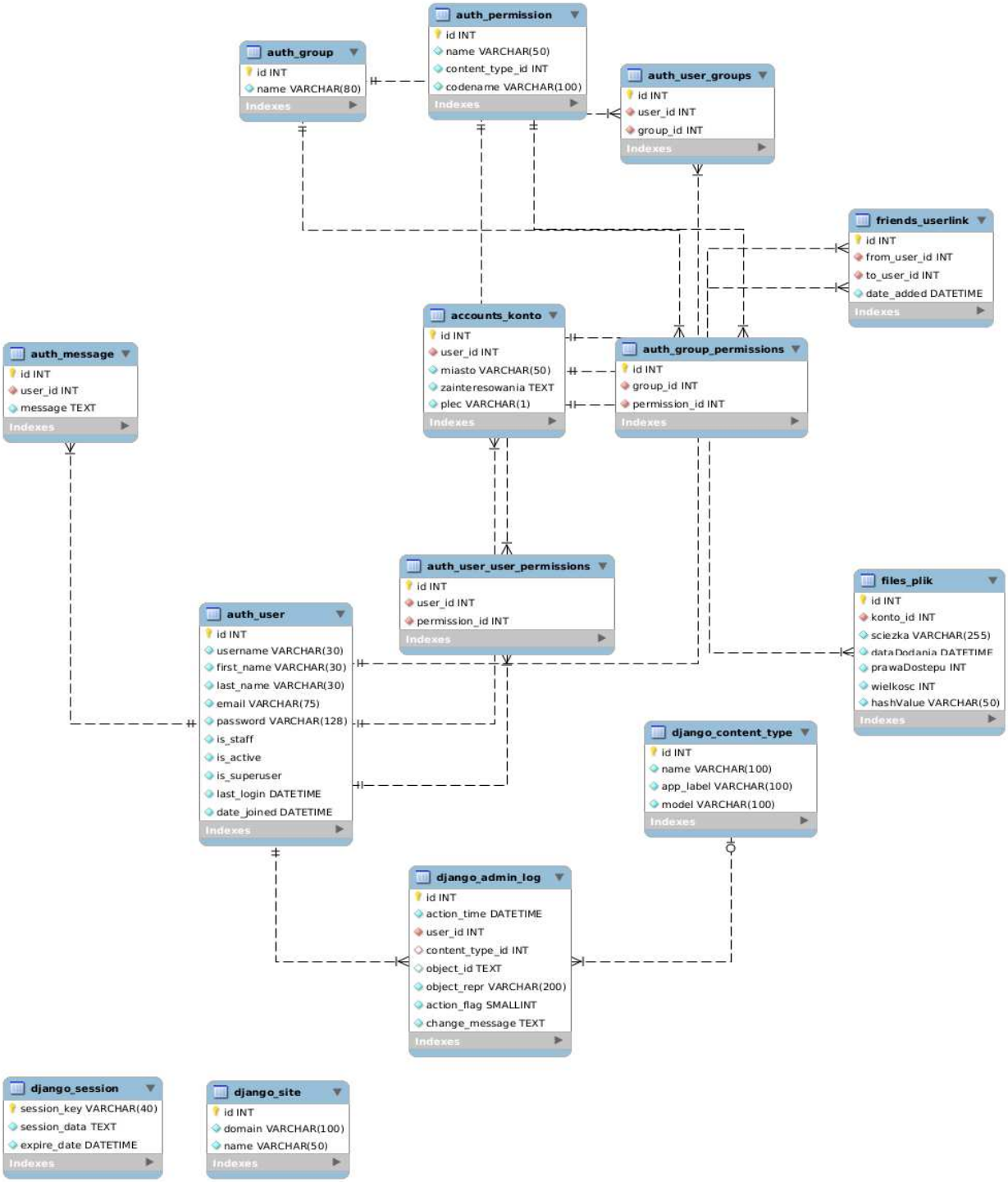
Klient po wpisaniu loginu i hasla wysyła je do serwera i czeka na odpowiedz i poprawną weryfikację. Jeżeli dane są poprawne, klient zapisuje otrzymany od serwera id sesji, który jest wysyłany przy każdej operacji. Po zalogowaniu okno programu minimalizuje się i uruchamiany jest watek, który co ustalony czas sprawdza pliki na serwerze i na dysku. Klient najpierw pobiera listę plików z serwera, potem porównuje to z lista plików zapisanych w lokalnym pliku xml i z plikami rzeczywiście na dysku, w zależnosci od różnic wykonuje wysyłanie plików na serwer, sciąganie i usuwanie plików z serwera i z katalogu lokalnego.

2.1.2. Serwer.

Serwer napisany w C++ jest obsługiwany przez system Linux. Ma strukturę monolityczna i jest synchroniczny. Potrafi obsłużyć wielu klientów jednocześnie i działa na wątkach. W protokole przesyłania danych wykorzystywany jest XML.

2.1.3. Baza danych.

Poniższy diagram przedstawia strukturę bazy danych. Część z tabel bazy została wygenerowana automatycznie dzięki frameworkowi django. Tabele te maja za zadanie usprawnić prace z frameworkiem oraz przechowuja informacje takie jak lista stron i szablonów. Ich dokładna dokumentacje można znaleźć na: http://docs.djangoproject.com/en/dev/.
Tabela accounts_konto zawiera wszystkie informacje o użytkowniku oraz klucz obcy do tabeli auth_users, gdzie przechowywane są szczegółowe informacje dotyczące konta użytkownika.
Tabela files_pliki, przechowywane informacje o plikach, zaś tabela friends_userlink informacje dotyczące znajomych użytkownika oraz o dane dotyczące udostępnianych plików.



2.1.4. Portal WWW.

Portal WWW został wygenerowany za pomocą frameworka django. Składa się on z 3 aplikacji.

- Aplikacji accounts odpowiedzialnej za zarządzanie kontami użytkowników.
- Aplikacji friends odpowiedzialnej za zarządzanie znajomymi.
- Aplikacją files odpowiedzialną za zarządzanie plikami.

Każda z aplikacji związana jest z osobną tablicą w bazie danych i korzysta ze wspólnej bazy jednak każda działa samodzielnie i korzysta z własnej logiki.

Przewidywana funkcjonalność aplikacji:
Dzięki aplikacji accounts można dodawać i usuwać konta użytkowników, logowanie i wylosowywanie, autoryzację użytkownika oraz wyszukiwanie kont innych użytkowników a również edycje swoich danych i oglądanie profilu.
Aplikacja friends pozwala nam na dodawanie i usuwanie użytkowników do listy znajomych oraz wyświetlanie listy znajomych.
Aplikacja files umożliwia usuwanie plików, wyświetlanie listy plików oraz zmianę praw dostępu do plików.

3. Literatura i źródła.

„Szkoła programowania Język C++” Stephen Prata – Książka traktująca o programowaniu w C++.
„Visual C# 2008. Projektowanie aplikacji. Pierwsze starcie” Jacek Matulewski. –Książka traktująca o programowaniu w C#.
http://www.tibik.nazwa.pl/index.php?option=com_frontpage&Itemid=1 – Strona o programowaniu w C#.
http://pl.docs.pld-linux.org/uslugi_bazydanych__mysql.html - strona o MYSQL
http://webmade.org/kursy-online/kurs-mysql.php - strona o MYSQL
http://www.python.rk.edu.pl/ - Biblioteka Pytona -Dokumentacja i przewodniki w języku polskim.
http://www.djangoproject.com/ - Oficjalna strona Django.
http://www.djangobook.com/ - - Darmowa książka o Django.
http://www.django.pl/ - - Polska Społeczność Django.
http://pl.wikipedia.org/wiki/Django - Strona Wikipedi poświęcona Django.

4. Załączniki

4.1. Przebieg testów programu.

Testy portalu:
Podczas testowania pojawiły się następujące błędy:
- Brak dodawania kont do bazy w momencie rejestracji.
- Możliwość usuwania kont bez odpowiednich praw dostępu.

Błędy zostały naprawione.

Testy klienta:
Podczas testowania pojawiły się następujące błędy:
- Brak możliwości połączenia z serwerem
- Problemy z parsowaniem XML.
- Wiele pomniejszych.

Błędy zostały naprawione.

Testy serwera:
Podczas testowania pojawiły się następujące błędy:
- Problem z dodawaniem plików do bazy danych
- Wiele pomniejszych.

Błędy zostały naprawione.

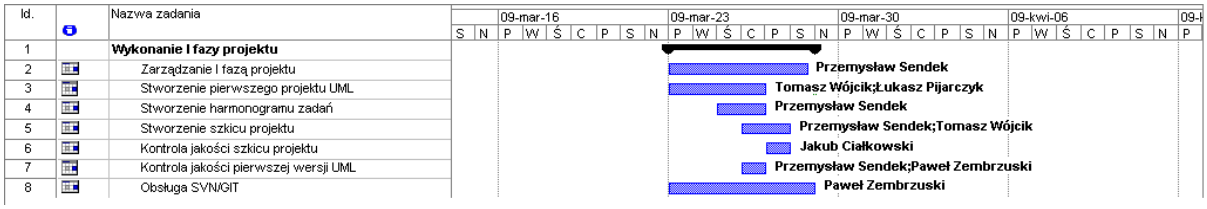
4.2. Changelog Serwera

```
     08 maj 2009
   released version 0.3.17 of Ass8-server

     Change log:
        -Fixed: wysylanie listy plikow
        -Fixed: odbieranie plikow
        -Fixed: dodawanie plikow do bazy (ponownie)
        -Fixed: wiele pomniejszych

08 maj 2009
   released version 0.3.16 of Ass8-server

     Change log:
        -Applied: Proteza na odbieranie pliku (przez telnet
działa)
        -Added: Kolejne informacje uzyteczne przy odrobaczaniu
        -Fixed: Parę małych poprawek

08 maj 2009
   released version 0.1.10 of Ass8-server

     Change log:
```

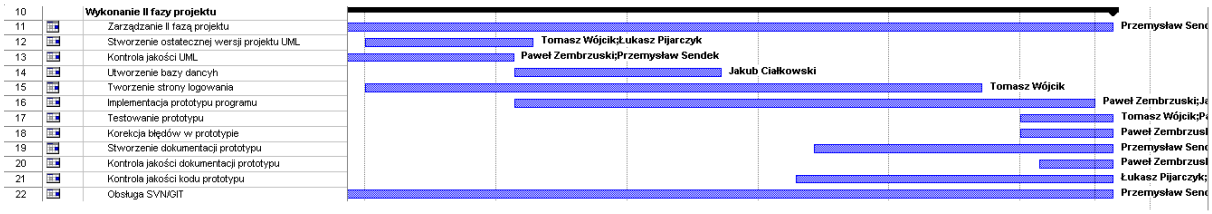
```
        -Fixed: pobieranie listy plikow z bazy
        -Fixed: pobieranie id z bazy
        -Fixed: usuwanie socketu gdy klient sie rozlacza
        -Fixed: wiele pomniejszych bledow
        -Removed: przywitanie serwera

07 maj 2009
   released version 0.0.9 of Ass8-server

     Change log:
        -Added: Wysylanie pliku na serwer
        -Added: Umieszczanie pliku w bazie
        -Added: więcej informacji przy debugowaniu

07 maj 2009
   released version 0.0.7 of Ass8-server

     Change log:
        -Updated: Parsowanie
        -Added: Wysylanie plikow
        -Added: Wysylanie listy plikow
        -Added: Logowanie
        -Added: Wiele wiecej

02 maj 2009
   released version 0.0.6 of Ass8-server

     Change log:
        -Added: odbieranie, kolejkowanie klientow
        -Added: fork()
        -Added: watek usuwający zombie

02 maj 2009
   released version 0.0.1 of Ass8-server

     Change log:
        -Added: Zaczątki
```

4.3. Terminarz.

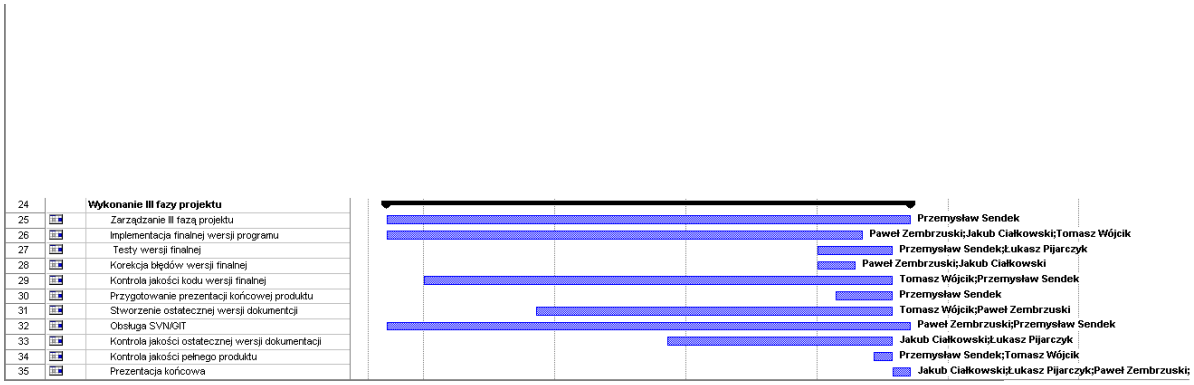
Wykres prezentujący I fazę projektu:



Wykres prezentujący II fazę projektu:



Wykres prezentujący III fazę projektu:



Zadania

Nazwa zasobu	Nazwa zadania	Cz. trw.	Rozp.	Zak.
	Wykonanie I fazy projektu	6 dn	Pon, 09-03-23	Sob, 09-03-28
Przemysław Sendek	Zarządzanie I fazą projektu	6 dn	Pon, 09-03-23	Sob, 09-03-28
Tomasz Wójcik;Łukasz Pijarczyk	Stworzenie pierwszego projektu UML	4 dn	Pon, 09-03-23	Czw, 09-03-26
Przemysław Sendek	Stworzenie harmonogramu zadań	2 dn	Śro, 09-03-25	Czw, 09-03-26
Przemysław Sendek;Tomasz Wójcik	Stworzenie szkicu projektu	2 dn	Czw, 09-03-26	Pią, 09-03-27
Jakub Ciałkowski	Kontrola jakości szkicu projektu	1 dzień	Pią, 09-03-27	Pią, 09-03-27
Przemysław Sendek;Paweł Zembrzuski	Kontrola jakości pierwszej wersji UML	1 dzień	Czw, 09-03-26	Czw, 09-03-26

Paweł Zembrzuski	Obsługa SVN/GIT	6 dn	Pon, 09-03-23	Sob, 09-03-28
	Wykonanie II fazy projektu	41 dn	Nie, 09-03-29	Pią, 09-05-08
Przemysław Sendek	Zarządzanie II fazą projektu	41 dn	Nie, 09-03-29	Pią, 09-05-08
Tomasz Wójcik;Łukasz Pijarczyk	Stworzenie ostatecznej wersji projektu UML	9 dn	Pon, 09-03-30	Wto, 09-04-07
Paweł Zembrzuski;Przemysław Sendek	Kontrola jakości UML	9 dn	Nie, 09-03-29	Pon, 09-04-06
Jakub Ciałkowski	Utworzenie bazy dancyh	11 dn	Wto, 09-04-07	Pią, 09-04-17
Tomasz Wójcik	Tworzenie strony logowania	33 dn	Pon, 09-03-30	Pią, 09-05-01
Paweł Zembrzuski;Jakub Ciałkowski	Implementacja prototypu programu	31 dn	Wto, 09-04-07	Czw, 09-05-07
Tomasz Wójcik;Paweł Zembrzuski	Testowanie prototypu	5 dn	Pon, 09-05-04	Pią, 09-05-08
Paweł Zembrzuski	Korekcja błędów w prototypie	5 dn	Pon, 09-05-04	Pią, 09-05-08
Przemysław Sendek;Łukasz Pijarczyk	Stworzenie dokumentacji prototypu	16 dn	Czw, 09-04-23	Pią, 09-05-08
Paweł Zembrzuski;Tomasz Wójcik	Kontrola jakości dokumentacji prototypu	4 dn	Wto, 09-05-05	Pią, 09-05-08
Łukasz Pijarczyk;Przemysław Sendek	Kontrola jakości kodu prototypu	17 dn	Śro, 09-04-22	Pią, 09-05-08
Przemysław Sendek;Paweł Zembrzuski	Obsługa SVN/GIT	41 dn	Nie, 09-03-29	Pią, 09-05-08
	Wykonanie III fazy projektu	28 dn	Sob, 09-05-09	Pią, 09-06-05
Przemysław Sendek	Zarządzanie III fazą projektu	28 dn	Sob, 09-05-09	Pią, 09-06-05
Paweł Zembrzuski;Jakub Ciałkowski;Tomasz Wójcik	Implementacja finalnej wersji programu	25 dn	Sob, 09-05-09	Śro, 09-06-03
Przemysław Sendek;Łukasz Pijarczyk	Testy wersji finalnej	4 dn	Pon, 09-06-01	Czw, 09-06-04
Paweł Zembrzuski;Jakub Ciałkowski	Korekcja błędów wersji finalnej	2 dn	Pon, 09-06-01	Wto, 09-06-02
Tomasz Wójcik;Przemysław Sendek	Kontrola jakości kodu wersji finalnej	25 dn	Pon, 09-05-11	Czw, 09-06-04
Przemysław Sendek	Przygotowanie prezentacji końcowej produktu	3 dn	Wto, 09-06-02	Czw, 09-06-04

Tomasz Wójcik;Paweł Zembrzuski	Stworzenie ostatecznej wersji dokumentcji	19 dn	Nie, 09-05-17	Czw, 09-06-04
Paweł Zembrzuski;Przemysław Sendek	Obsługa SVN/GIT	28 dn	Sob, 09-05-09	Pią, 09-06-05
Jakub Ciałkowski;Łukasz Pijarczyk	Kontrola jakości ostatecznej wersji dokumentacji	12 dn	Nie, 09-05-24	Czw, 09-06-04
Przemysław Sendek;Tomasz Wójcik	Kontrola jakości pełnego produktu	1 dzień	Czw, 09-06-04	Czw, 09-06-04
Jakub Ciałkowski;Łukasz Pijarczyk;Paweł Zembrzuski;Przemysław Sendek;Tomasz Wójcik	Prezentacja końcowa	1 dzień	Pią, 09-06-05	Pią, 09-06-05

Grupa zasobów	**Nazwa zasobu**
Programista, Bazodanowiec, Kontrola jakości	Jakub Ciałkowski
Drugi projektant, Programista, Kontrola jakości, Tester	Łukasz Pijarczyk
Główny programista, Kontrola jakości, Obsługa SVN	Paweł Zembrzuski
Menadżer, Kontrola Jakości, Obsługa SVN, Tester, Programista, Projektant prezentacji końcowej, Twórca dokumentacji	Przemysław Sendek
Główny Projektant, Programista, Kontrola jakości, Twórca dokumentacji, Tester	Tomasz Wójcik

Zasoby i przydziały	**Rozp.**	**Zak.**
Jakub Ciałkowski	Pią, 09-03-27	Pią, 09-06-05
Kontrola jakości szkicu projektu	Pią, 09-03-27	Pią, 09-03-27
Utworzenie bazy dancyh	Wto, 09-04-07	Pią, 09-04-17
Implementacja prototypu programu	Wto, 09-04-07	Czw, 09-05-07
Implementacja finalnej wersji programu	Sob, 09-05-09	Śro, 09-06-03
Korekcja błędów wersji finalnej	Pon, 09-06-01	Wto, 09-06-02
Kontrola jakości ostatecznej wersji dokumentacji	Nie, 09-05-24	Czw, 09-06-04
Prezentacja końcowa	Pią, 09-06-05	Pią, 09-06-05
Łukasz Pijarczyk	Pon, 09-03-23	Pią, 09-06-05
Stworzenie pierwszego projektu UML	Pon, 09-03-23	Czw, 09-03-26
Stworzenie ostatecznej wersji projektu UML	Pon, 09-03-30	Wto, 09-04-07
Stworzenie dokumentacji prototypu	Czw, 09-04-23	Pią, 09-05-08
Kontrola jakości kodu prototypu	Śro, 09-04-22	Pią, 09-05-08
Testy wersji finalnej	Pon, 09-06-01	Czw, 09-06-04
Kontrola jakości ostatecznej wersji dokumentacji	Nie, 09-05-24	Czw, 09-06-04
Prezentacja końcowa	Pią, 09-06-05	Pią, 09-06-05
Paweł Zembrzuski	Pon, 09-03-23	Pią, 09-06-05

Kontrola jakości pierwszej wersji UML	Czw, 09-03-26	Czw, 09-03-26
Obsługa SVN/GIT	Pon, 09-03-23	Sob, 09-03-28
Kontrola jakości UML	Nie, 09-03-29	Pon, 09-04-06
Implementacja prototypu programu	Wto, 09-04-07	Czw, 09-05-07
Testowanie prototypu	Pon, 09-05-04	Pią, 09-05-08
Korekcja błędów w prototypie	Pon, 09-05-04	Pią, 09-05-08
Kontrola jakości dokumentacji prototypu	Wto, 09-05-05	Pią, 09-05-08
Obsługa SVN/GIT	Nie, 09-03-29	Pią, 09-05-08
Implementacja finalnej wersji programu	Sob, 09-05-09	Śro, 09-06-03
Korekcja błędów wersji finalnej	Pon, 09-06-01	Wto, 09-06-02
Stworzenie ostatecznej wersji dokumentcji	Nie, 09-05-17	Czw, 09-06-04
Obsługa SVN/GIT	Sob, 09-05-09	Pią, 09-06-05
Prezentacja końcowa	Pią, 09-06-05	Pią, 09-06-05
Przemysław Sendek	Pon, 09-03-23	Pią, 09-06-05
Zarządzanie I fazą projektu	Pon, 09-03-23	Sob, 09-03-28
Stworzenie harmonogramu zadań	Śro, 09-03-25	Czw, 09-03-26
Stworzenie szkicu projektu	Czw, 09-03-26	Pią, 09-03-27
Kontrola jakości pierwszej wersji UML	Czw, 09-03-26	Czw, 09-03-26
Zarządzanie II fazą projektu	Nie, 09-03-29	Pią, 09-05-08
Kontrola jakości UML	Nie, 09-03-29	Pon, 09-04-06
Stworzenie dokumentacji prototypu	Czw, 09-04-23	Pią, 09-05-08
Kontrola jakości kodu prototypu	Śro, 09-04-22	Pią, 09-05-08
Obsługa SVN/GIT	Nie, 09-03-29	Pią, 09-05-08
Zarządzanie III fazą projektu	Sob, 09-05-09	Pią, 09-06-05
Testy wersji finalnej	Pon, 09-06-01	Czw, 09-06-04
Kontrola jakości kodu wersji finalnej	Pon, 09-05-11	Czw, 09-06-04
Przygotowanie prezentacji końcowej produktu	Wto, 09-06-02	Czw, 09-06-04
Obsługa SVN/GIT	Sob, 09-05-09	Pią, 09-06-05
Kontrola jakości pełnego produktu	Czw, 09-06-04	Czw, 09-06-04
Prezentacja końcowa	Pią, 09-06-05	Pią, 09-06-05
Tomasz Wójcik	Pon, 09-03-23	Pią, 09-06-05
Stworzenie pierwszego projektu UML	Pon, 09-03-23	Czw, 09-03-26
Stworzenie szkicu projektu	Czw, 09-03-26	Pią, 09-03-27
Stworzenie ostatecznej wersji projektu UML	Pon, 09-03-30	Wto, 09-04-07
Tworzenie strony logowania	Pon, 09-03-30	Pią, 09-05-01
Testowanie prototypu	Pon, 09-05-04	Pią, 09-05-08
Kontrola jakości dokumentacji prototypu	Wto, 09-05-05	Pią, 09-05-08
Implementacja finalnej wersji programu	Sob, 09-05-09	Śro, 09-06-03
Kontrola jakości kodu wersji finalnej	Pon, 09-05-11	Czw, 09-06-04
Stworzenie ostatecznej wersji dokumentcji	Nie, 09-05-17	Czw, 09-06-04

Kontrola jakości pełnego produktu	Czw, 09-06-04	Czw, 09-06-04
Prezentacja końcowa	Pią, 09-06-05	Pią, 09-06-05