

Overlord - Software Architecture Document

Jakub Gołębiowski Adam Kawa Piotr Krewski Tomasz Weksej

5 czerwca 2006

Spis treści

1	Wprowadzenie	3
1.1	Cel	3
1.2	Zakres	3
1.3	Definicje	3
2	Założenia i zależności	3
3	Prezentacja architektury systemu	3
4	Przegląd przypadków użycia	4
4.1	Graj w grę - planowanie ataku	5
4.2	Graj w grę - budowa fabryki	8
5	Dekompozycja logiczna systemu	9
6	Instalacja Systemu	11
7	Implementacja Systemu	11
7.1	Pakiety modelu	11
7.1.1	com.overlord.dziedzina	12
7.1.2	com.overlord.zdarzenia	13
7.1.3	com.overlord.obslugaGry	14
7.1.4	com.overlord.obslugaSystemu	15
8	Przechowywane dane	15
8.1	Omówienie	15
8.2	Dane nieprzechowywane w bazie	16
8.3	Gracz	16
8.4	Państwo	16
8.5	Kontynent	17

8.6	Terytorium	17
8.7	Budynek	18
8.8	Sojusz	18
8.9	Armia	18
8.10	Generał	18
8.11	JednostkaBojowa	19
8.12	Wojna	19
8.13	PosiadaneTechnologie	20
9	Jakość	20
9.1	Rozszerzalność	20
9.2	Przenośność	20
9.3	Łatwość obsługi	20
9.4	Bezpieczeństwo	20
9.5	Niezawodność	21

1 Wprowadzenie

1.1 Cel

Niniejszy dokument ma na celu opisanie budowy systemu „Overlord”, przedstawienie najważniejszych technologii wykorzystanych podczas jego realizacji oraz omówienie najistotniejszych jego elementów.

1.2 Zakres

Dokument porusza kwestie związane z wyborem zastosowanej technologii, sposobem realizacji przypadków użycia, dekompozycją systemu na warstwy i pakiety. Zawiera także opis przechowywanych danych, przyjęte założenia dotyczące jakości systemu.

1.3 Definicje

Struts - zestaw bibliotek napisanych w języku Java służących do rozdzielania warstwy prezentacji aplikacji od logiki biznesowej w architekturze MVC (model - widok - kontroler).

2 Założenia i zależności

- System zostanie napisany w obiektowym języku Java, przy wykorzystaniu darmowych i ogólnodostępnych narzędzi oraz bibliotek towarzyszących.
- System będzie przenośny.
- System w przyszłości może ulegać zmianom. W celu ułatwienia ewentualnych modyfikacji, rozszerzeń wykorzystano obiektowe techniki modelowania systemu.

3 Prezentacja architektury systemu

- **Aplikacja w pełni internetowa** Podczas projektowania systemu zrezygnowano z architektury klient-serwer, decydując się na aplikację w pełni internetową. Głównym argumentem przemawiającym na korzyść tego rozwiązania jest brak jakichkolwiek przygotowań komputera po stronie klienta do użytkowania systemu. Wystarczy tutaj jedynie wpisanie w przeglądarce obsługującej Javę, właściwego adresu systemu. Z kolei wadą architektury klient-serwer są także trudności w rozpowszechnianiu aktualizacji systemu oraz konieczność przechowywania części danych po stronie klienta.
- **Obiektowe techniki modelowania** System zostanie zaimplementowany w obiektowym języku Java. Bardzo wygodne i efektywne wydaje się być wykorzystanie obiektowych technik modelowania: podział na klasy i obiekty, polimorfizm, czy dziedziczenie. Dzięki

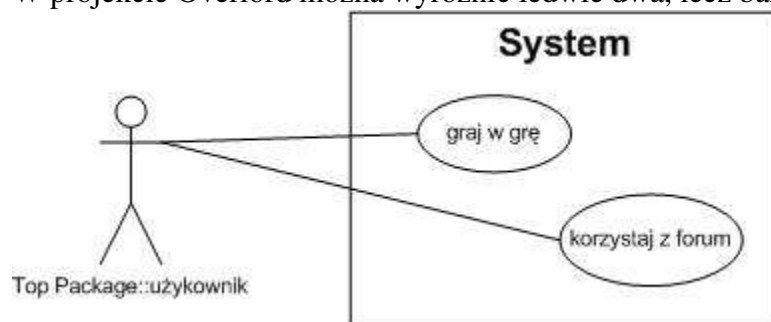
temu kod stanie się bardziej przejrzysty i łatwiej modyfikowalny. Głównym atutem przemawiającym za Java jest duża liczba darmowych bibliotek i narzędzi z nią związanych.

- **JDBC** Pakiet JDBC (Java DataBase Connectivity) to podstawowe narzędzie umożliwiające aplikacjom napisanym w Javie komunikować się z bazami danych za pomocą języka SQL. Dla twórców systemu wykorzystanie tego rozwiązania jest bardzo korzystne, ponieważ do stworzenia efektywnej komunikacji z dowolną bazą danych (o ile JDBC posiada odpowiednie sterowniki) wystarczy tylko znajomość języka SQL. Innym alternatywnym pomysłem jest wykorzystanie techniki odwzorowania obiektowo-relacyjnego, czyli odwzorowanie obiektowego modelu klas i obiektów na relacyjną bazę danych w sposób zautomatyzowany. Mimo zalet tego rozwiązania, nie zostanie ono wykorzystane podczas realizacji systemu ze względu na słabą znajomość technologii przez twórców Overlord.
- **Architektura MVC** Koncepcja MVC (Model - View - Controller) zakłada podział komponentów na trzy kategorie:
 1. **Model** (komponenty reprezentujące dane oraz operacje wykonywane na nich)
 2. **Widok** (komponenty odpowiedzialne za prezentację dla użytkownika danych znajdujących się w Modelu, dostarczające interfejs użytkownika)
 3. **Kontroler** (komponenty sterujące, odpowiedzialne za przechwytywanie żądań użytkowników oraz ich interpretacje, czyli wywoływanie odpowiednich operacji z Modelu i wyświetlenie właściwego widoku jako odpowiedź).

Taka dekompozycja systemu na trzy różne funkcjonalnie kategorie, powoduje że duża aplikacja staje się bardziej przejrzysta, łatwiej testowana i modyfikowalna.

4 Przegląd przypadków użycia

W projekcie Overlord można wyróżnić ledwie dwa, lecz bardzo pojemne przypadki użycia.

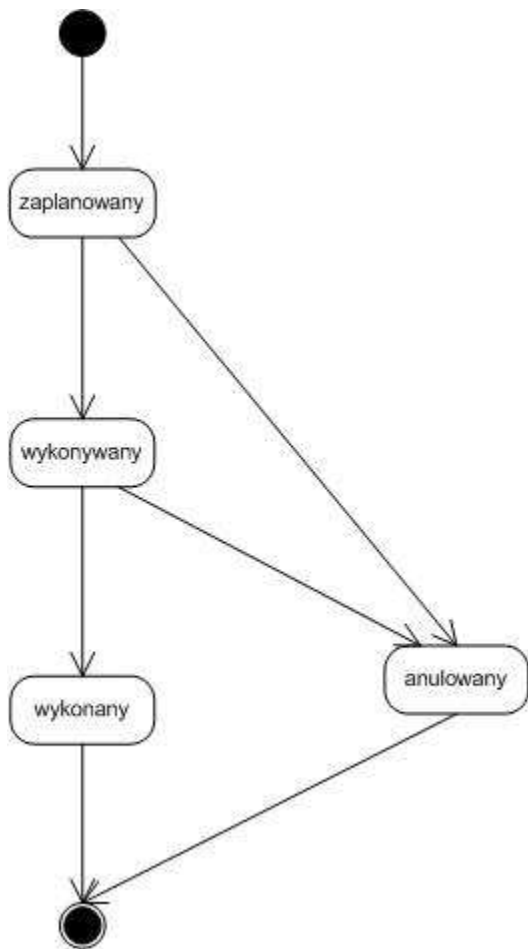


Korzystanie z forum będzie realizowane przez zewnętrzną aplikację (np. phpBB), przyjrzymy się więc przykładom realizowania przypadku użycia "Graj w grę".

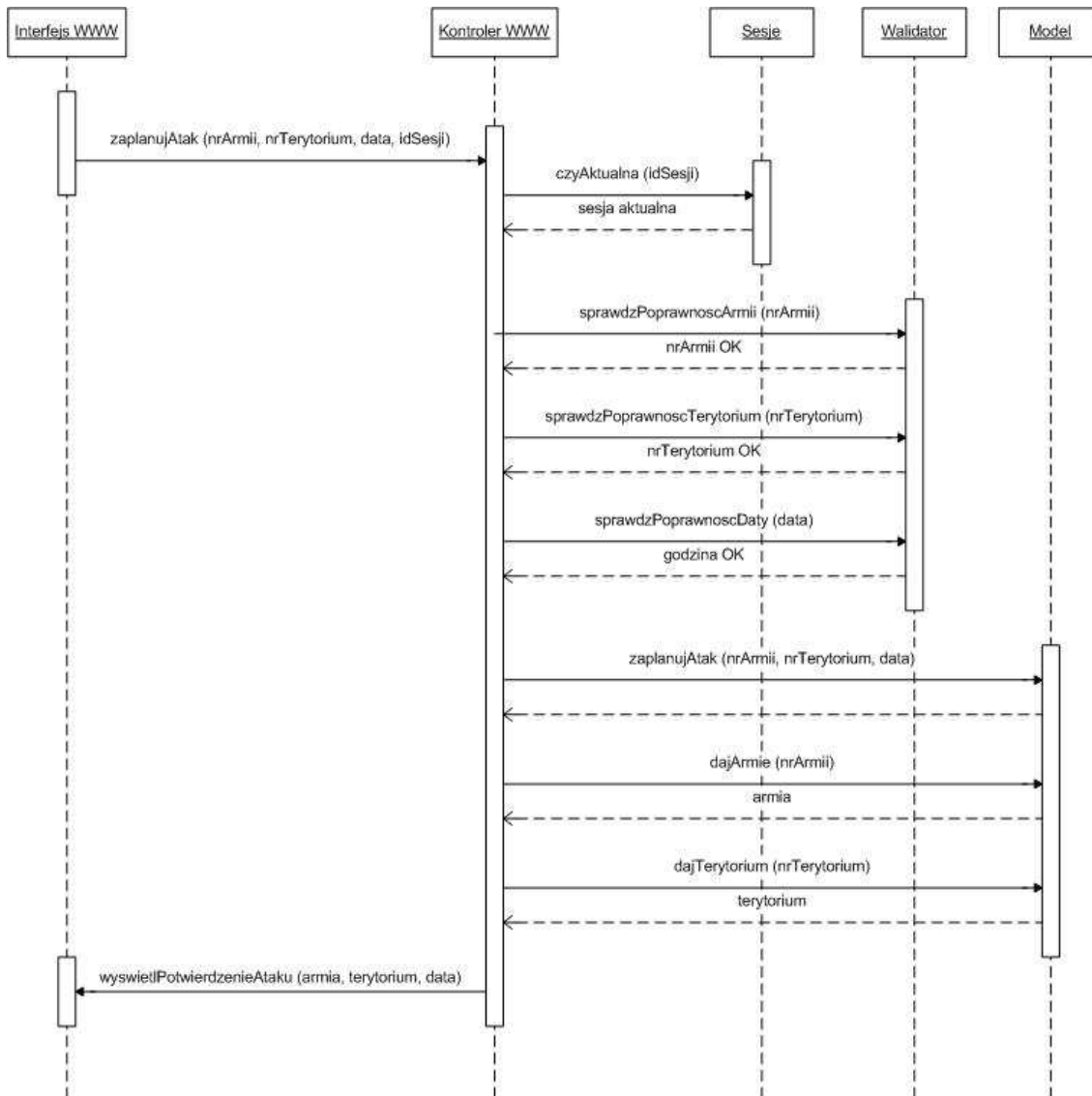
4.1 Graj w grę - planowanie ataku



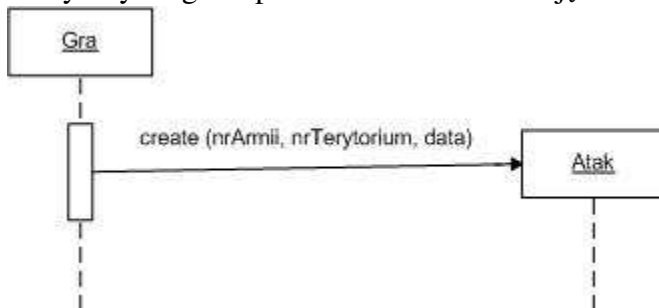
Planowanie ataku polega na wybraniu armii atakującej, terytorium atakowanego i daty rozpoczęcia ataku.



Atak może być zaplanowany(jeszcze nie realizowany), wykonywany (w trakcie realizacji) lub wykonany. Istnieje też możliwość odwołania ataku.

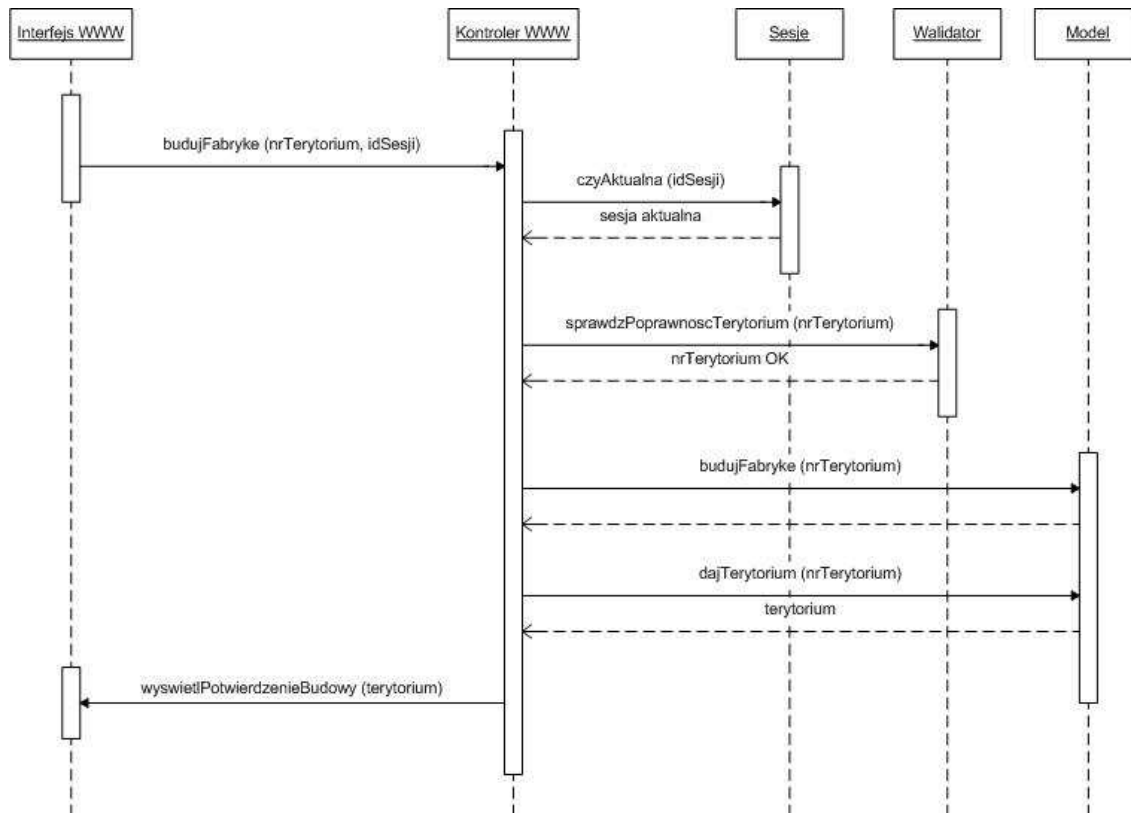


Powyższy diagram przedstawia komunikację interfejsu z modelem.

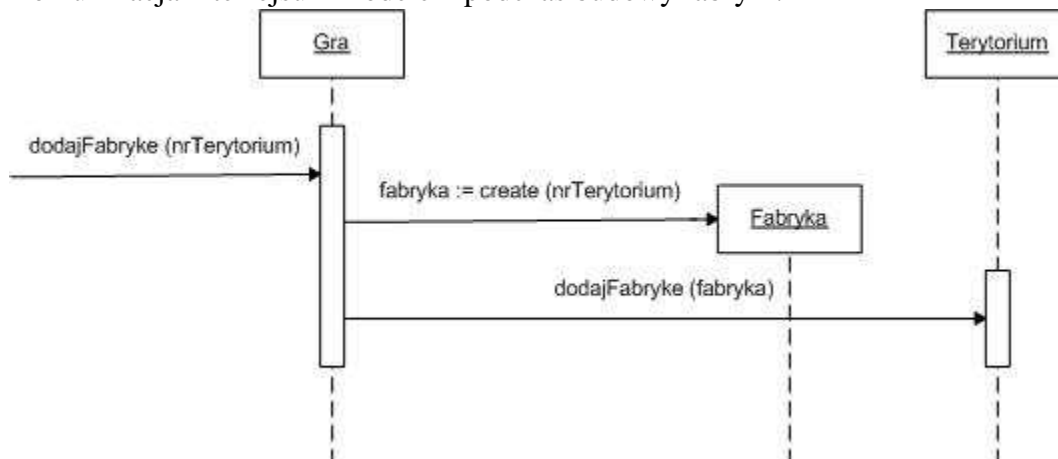


A tutaj już sama sytuacja w modelu. W tym przypadku jest ona bardzo prosta.

4.2 Graj w grę - budowa fabryki



Komunikacja interfejsu z modelem podczas budowy fabryki.

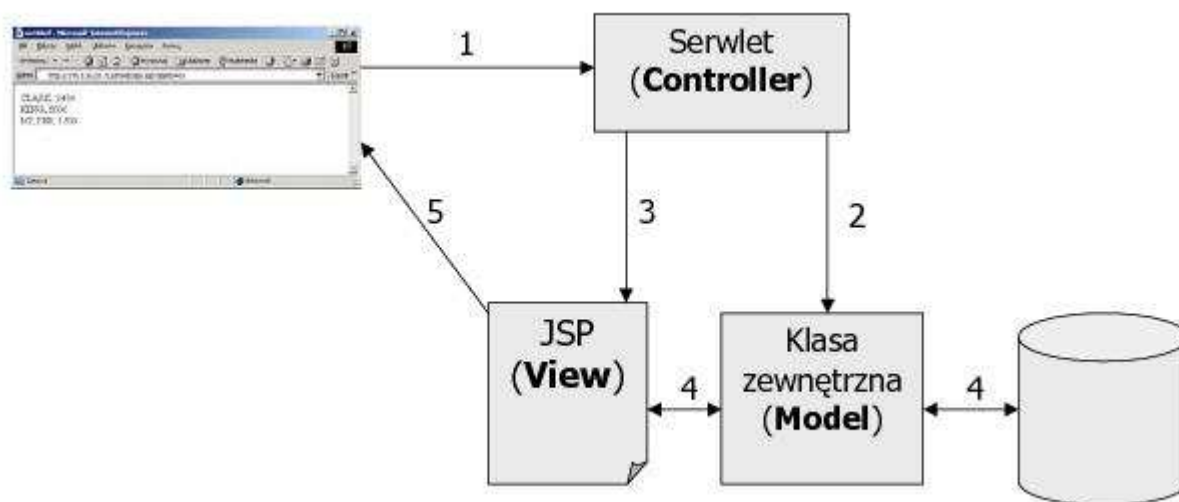


Budowa fabryki - jak to wygląda w modelu.

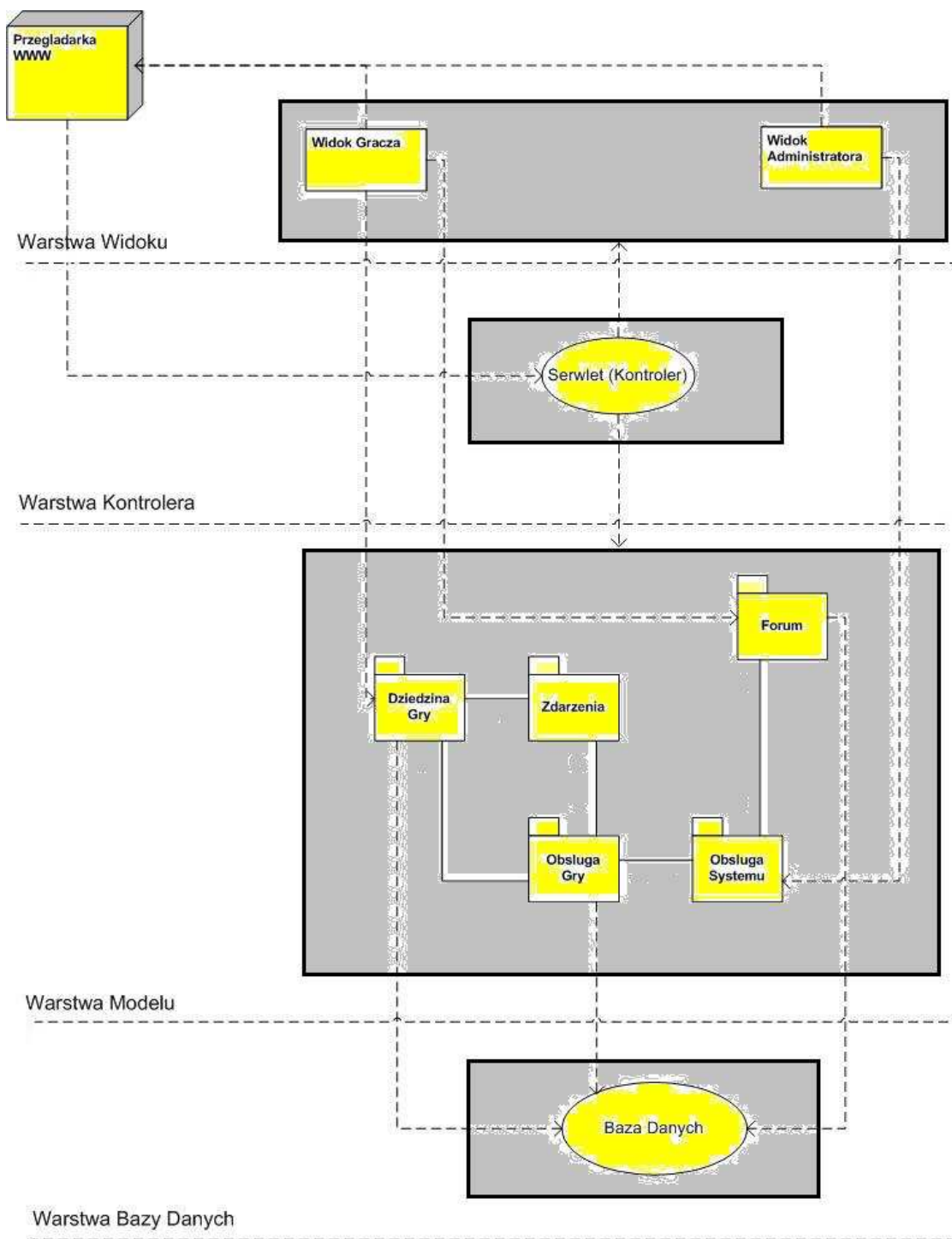
Pozostałe diagramy postanowiłem pominąć, ze względu na ich podobieństwo do poprzednich.

5 Dekompozycja logiczna systemu

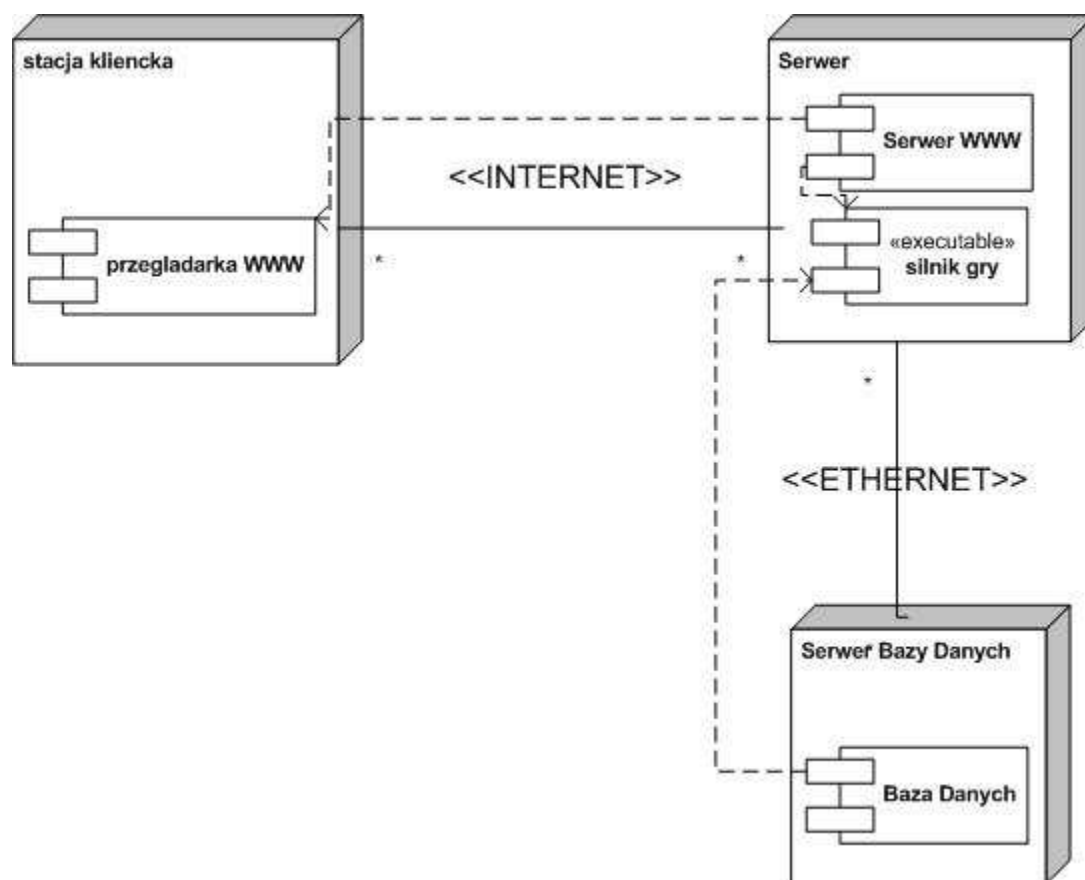
Projekt został wykonany przy wykorzystaniu architektury MVC. Wyróżniono zatem trzy warstwy: Warstwa Widoku, Warstwa Kontrolera, Warstwa Modelu.



- **Warstwa Widoku** - opisuje część systemu widoczną z zewnątrz. Należą do niej komponenty prezentujące dane, dostarczające graficzny interfejs użytkownika. W przypadku systemu Overlord będą to dynamiczne strony JSP (Java Server Pages), przy ich konstrukcji wykorzystane będą znaczniki dostarczane przez bibliotekę Struts. bibliotekę
- **Warstwa Modelu** - obiekty języka Java odwzorowujące całą logikę biznesową systemu, zapewniające obsługę bazy danych.
- **Warstwa Kontrolera** - funkcje kontrolera pełni wyspecjalizowany serwlet będący częścią biblioteki Struts. Konfiguracja Struts odbywa się poprzez pliki XML.



6 Instalacja Systemu



7 Implementacja Systemu

7.1 Pakiety modelu

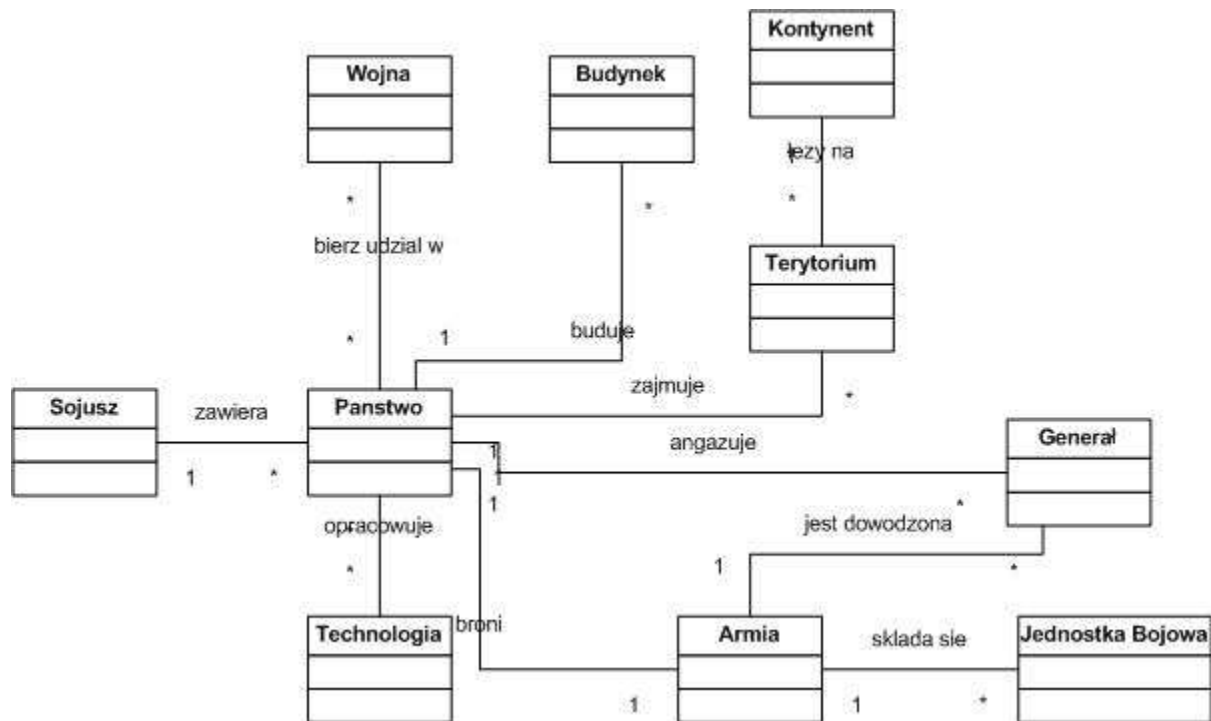
W logice biznesowej wyróżnione zostały następujące pakiety:

- `com.overlord.dziedzina` - klasy reprezentujące obiekty składające się na dziedzinę gry, obsługujące zapisy i odczyty z bazy danych,
- `com.overlord.zdarzenia` - klasy obsługujące zdarzenia mogące zachodzić w grze zlecane przez gracza,
- `com.overlord.obslugaGry` - klasy egzekwujące zasady gry i przetwarzające wszelkie niezbędne dla funkcjonowania gry współczynniki
- `com.overlord.obslugaSystemu` - klasy nie mające nic wspólnego z dziedziną gry Overlord, obsługujące np. logowanie, rejestracje i czynności administracyjne,

- `com.overlord.forum` - klasy obsługujące forum

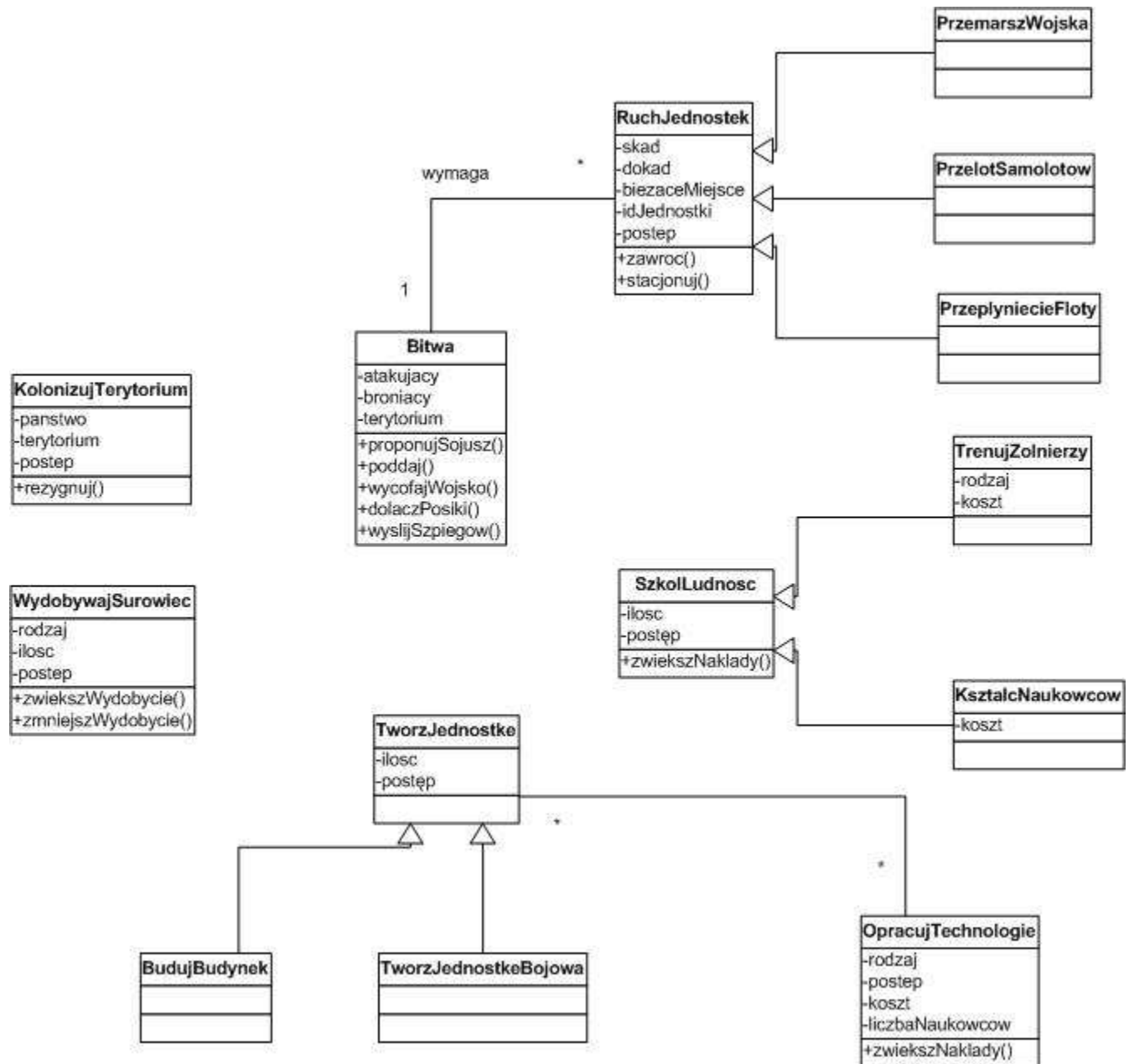
Poniżej znajduje się bardziej szczegółowy opis wymienionych pakietów. Opis uwzględnia główne klasy należące do danych pakietów wraz z najważniejszymi metodami tych klas.

7.1.1 `com.overlord.dziedzina`



Każda z klas tego pakietu zapewnia dostęp do analogicznej (co do nazwy) tabeli z bazy danych. Inne pakiety za ich pomocą dostają się do potrzebnych danych, obsługa bazy odbywa się za pomocą pakietu JDBC.

7.1.2 com.overlord.zdarzenia

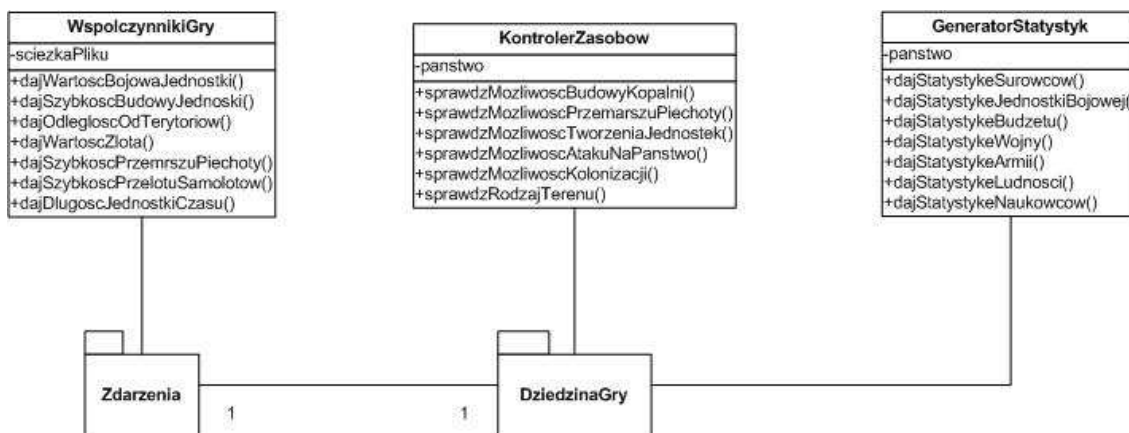


Klasy tego pakietu są wątkami odpowiedzialnymi zleconą przez gracza akcją. Każda klasa dziedziczy z nadklasy (nie zaznaczono nadklasy na rysunku ze względu na zachowanie jego czytelności) dziedziczy metody umożliwiające rozpoczęcie, wstrzymanie i zakończenie akcji. Długość ich życia jest uwarunkowana współczynnikami czasowymi zawartymi w plikach konfiguracyjnych gry. Lista najważniejszych zdarzeń:

1. `OpracujTechnologie` - wątek ten nadzoruje opracowywanie danej technologii, zleca na bieżąco aktualizację bazy danych i zmienia widok gracza,

2. wydobywajSurowiec - wątek ten odpowiedzialny jest za wydobywanie określonego surowca
3. szkolLudnosc klasa abstrakcyjna, klasy potomne:
 - trenujZolnierzy - ten wątek realizuje akcje szkolenia żołnierzy z ludności zamieszkującej państwo
 - ksztalcNaukowcow - ten wątek realizuje akcje kształcenia naukowców z ludności zamieszkującej państwo
4. kolonizujTerytorium - wątek ten odpowiedzialny jest za pozyskiwanie przez gracza niezasiedlonych terytoriów
5. bitwa - wątek realizujący bitwę między państwami
6. ruchJednostek - wątek realizujący przemieszczanie się jednostek bojowych stworzonych przez gracza. jest to klasa abstrakcyjna, jej klasy potomne:
 - przelotSamolotow
 - przemarszWojska
 - przeplyniecieFloty
7. tworzMJednostke - wątek odpowiedzialny za budowanie nowych jednostek/budynków w państwie gracza. Jest to klasa abstrakcyjna, jej klasy potomne:
 - budujBudynek
 - tworzMJednostkeBojowa

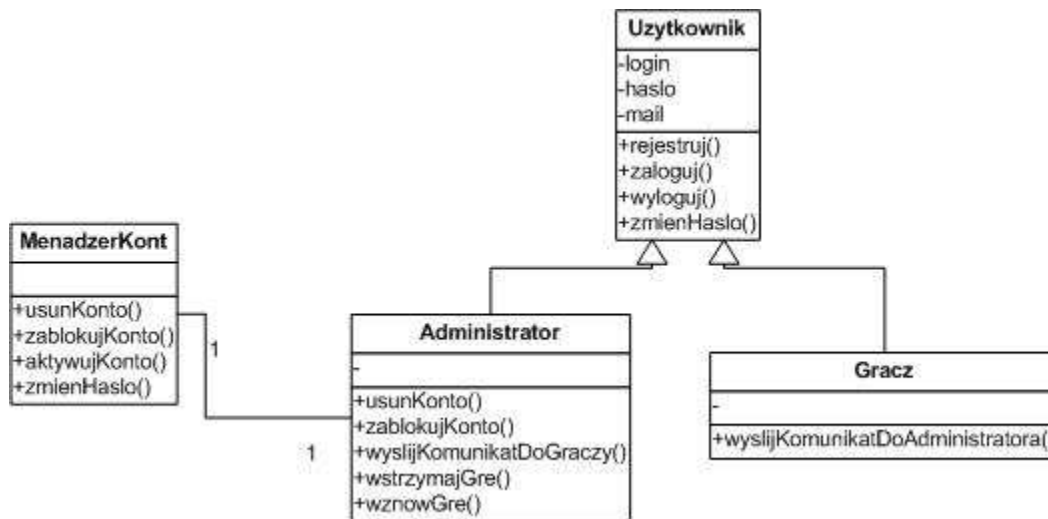
7.1.3 com.overlord.obslugaGry



Klasy tego pakietu zajmują się obsługą technicznych aspektów gry Overlord. Najważniejsze klasy to:

1. `WspolczynnikGry` - klasa ta zajmuje się przetwarzaniem pliku konfiguracyjnego zawierającego informacje o rozmaitych współczynnikach definiujących np. szybkość budowania jednostek, wartość bojową armii. Współczynniki te są udostępniane pakietowi `com.overlord.zdarzenia`.
2. `KontrolerZasobow` - klasa ta zajmuje się oceną aktualnej sytuacji w jakiej znajduje się państwo gracza i komunikując się z widokiem decyduje jakie opcje sterowania państwem mają być w danym momencie dostępne dla gracza.
3. `GeneratorStatystyk` - klasa zajmująca się dostarczaniem rozmaitych statystyk niezbędnych do prowadzenia rozgrywki, na żądanie gracza.

7.1.4 `com.overlord.obsługaSystemu`



Klasy tego pakietu zajmują się obsługą systemu realizującego grę Overlord. Najważniejsze klasy to:

1. `MenadzerKont` - klasa udostępniająca administratorowi interfejs do zarządzania kontami graczy.
2. `Uzytkownik` - abstrakcyjna klasa reprezentująca użytkownika systemu
 - `Administrator` - klasa reprezentująca administratora systemu, pozwalająca mu wykonywać czynności administracyjne
 - `Gracz` - klasa reprezentująca Gracza, zawiera zestaw metod do zarządzania pojedynczym kontem gracza.

8 Przechowywane dane

8.1 Omówienie

W systemie Overlord dane przechowywane są na dwa sposoby:

1. na serwerze jako pliki konfiguracyjne - opisane w punkcie 8.2
2. W relacyjnej bazie danych PostgreSQL lub MySQL. Dane zostaną zorganizowane w postaci tabel o ustalonym schemacie. W poniższym opisie uwzględniono najważniejsze tabele w systemie wraz z krótko opisanymi atrybutami. W opisie danych przedstawiono w szczególności:
 - klucz główny tabeli
 - klucze obce
 - najważniejsze pola tabel wraz z krótkim opisem przechowywanych danych

8.2 Dane nieprzechowywane w bazie

Dane przechowywane poza główną bazą danych:

Rodzaj	Uzasadnienie
pliki konfiguracyjne	potrzebne szybko, zawierają współczynniki i przeliczniki niezbędne to prowadzenia rozgrywki, niedużych rozmiarów
grafika	grafikę wygodniej jest przechowywać w postaci plików

8.3 Gracz

Tabela przechowująca informacje o zarejestrowanych graczach

nazwa pola	opis	typ
LOGIN	login użytkownika w systemie	ciąg znaków klucz główny
MAIL	adres mailowy użytkownika	ciąg znaków zawierający znak '@'
LOGDATE	data ostatniego logowania	format daty

8.4 Państwo

W tej tabeli przechowywane są informacje o państwach w świecie Overlord

nazwa pola	opis	typ
WLASCICIEL	gracz władający danym państwem	odwołanie do tabeli Gracz klucz główny
NAZWA	nazwa państwa	ciąg znaków klucz główny

nazwa pola	opis	typ
LUDNOSC	liczba ludności zamieszkująca państwo	liczba naturalna
DATAZAL	data utworzenia państwa	format daty
SOJUSZ	sojusz w którym jest państwo	odwołanie do tabeli Sojusz

8.5 Kontynent

Tabela zawierająca informacje o kontynentach, na których mieszczą się państwa

nazwa pola	opis	typ
NAZWA	nazwa kontynentu	ciąg znaków klucz główny
WIELKOSC	wielkość kontynentu (ilość jednostkowych pól)	liczba naturalna

8.6 Terytorium

Tabela zawierająca terytoria. Pojedyncze terytorium jest jednostkowym kwadratem na mapie, jednoznacznie identyfikowalnym poprzez swoje współrzędne. Państwo może składać się z wielu terytoriów na różnych kontynentach

nazwa pola	opis	typ
PANSTWO	państwo, do którego terytorium należy	odwołanie do tabeli Państwa
WSPX	współrzędna pozioma położenia	liczba naturalna klucz główny
WSPY	współrzędna pionowa położenia	liczba naturalna klucz główny
KONTYMENT	kontynent, na którym terytorium leży	odwołanie do tabeli Kontynent
GORZYSTOSC	współczynnik procentowy górzystości terenu	liczba naturalna <0,100>
ROPA	zasobność w ropę terytorium (wartość w baryłkach)	liczba naturalna
ZLOTO	zasobność w rudę złota terytorium (wielkość w kilogramach)	liczba naturalna
WEGIEL	zasobność w węgiel kamienny terytorium (wielkość w tonach)	liczba naturalna
ZELAZO	zasobność w rudę żelaza terytorium (wielkość w tonach)	liczba naturalna

8.7 Budynek

Tabela przechowująca dane o budynkach. Jako klucz główny wprowadzono sztuczny identyfikator.

nazwa pola	opis	typ
ID	unikalny identyfikator budynku	liczba naturalna klucz główny
PAŃSTWO	państwo, w którym jest budynek	odwołanie do tabeli Państwo
TERYTORIUM	terytorium, na którym stoi dany budynek	współrzędne terytorium
RODZAJ	rodzaj budynku	ciąg znaków
UKOŃCZONY	wielkość procentowa - stopień ukończenia budowy	liczba naturalna <0,100>

8.8 Sojusz

Tabela zawiera informacje o sojuszach zawartych przez graczy.

nazwa pola	opis	typ
NAZWA	nazwa sojuszu	ciąg znaków klucz główny
ZAŁOŹYCIEL	gracz, który dany sojusz założył	odwołanie do tabeli Gracz

8.9 Armia

Tabela zawierająca informacje o armiach poszczególnych państw. Armia skupia jednostki bojowe oraz zawiera pewną ilość żołnierzy.

nazwa pola	opis	typ
PANSTWO	państwo, do którego armia należy	odwołanie do tabeli Państwo klucz główny
GENERAŁ	generał dowodzący armią	odwołanie do tabeli Generał klucz główny
ZOLNIERZE	liczba żołnierzy w danej armii	liczba naturalna

8.10 Generał

Tabela przechowuje informacje o generałach. Generał dowodzi jakąś armią, charakteryzuje go kilka istotnych współczynników.

nazwa pola	opis	typ
PAŃSTWO	państwo, dla którego służy generał	odwołanie do tabeli Państwo klucz główny
ID	unikalny numer porządkowy generała	liczba naturalna klucz główny
MORALE	morale generała, im większe tym lepiej	liczba naturalna <0,100>
DOSW	doświadczenie generała nabywane w czasie wojen	liczba naturalna
UMIEJ	umiejętności generała	liczba naturalna
WIEK	wiek generała	liczba naturalna
SPEC	określa czy generał dowodzi jednostkami naziemnymi, powietrznymi czy wodnymi	napis

8.11 JednostkaBojowa

Tabela przechowuje informacje o jednostkach bojowych. Jednostki bojowe wchodzi w skład armii. Jeden określony rodzaj jednostki może występować w dowolnej liczbie egzemplarzy.

nazwa pola	opis	typ
ARMIA	armia, do której jednostka należy	odwołanie do tabeli Armia klucz główny
RODZAJ	rodzaj jednostki (czołg, samolot itp)	napis klucz główny
TYP	szczegółowy typ jednostki	ciąg znaków
ILOSC	ilość egzemplarzy danej jednostki w armii	liczba naturalna
UKONCZENIE	procentowa wielkość mówiąca o stanie ukończenia produkcji jednostki	liczba naturalna <0,100>

8.12 Wojna

Tabela przechowująca informacje o konfliktach w grze.

nazwa pola	opis	typ
PANSTWO1	pierwsze z państw prowadzących wojnę	odwołanie do tabeli Państwo klucz główny
PANSTWO2	drugie z państw prowadzących wojnę	odwołanie do tabeli Państwo klucz główny
DATAWYB	data wypowiedzenia wojny	format daty

8.13 Posiadane Technologie

Tabela przechowująca dane o technologiach posiadanych przez poszczególne państwa

nazwa pola	opis	typ
PAŃSTWO	państwo mające do dyspozycji daną technologię	odwołanie do tabeli Państwo klucz główny
TECH	nazwa danej technologii	napis klucz główny
POZIOM	poziom zaawansowania w danej technologii	liczba naturalna
NAUKOWCY	ilość naukowców pracująca nad daną technologią	liczba naturalna
UKOŃCZENIE	procentowa wielkość ukazująca stopień zaawansowania prac nad daną technologią	<0,100>

9 Jakość

9.1 Rozszerzalność

Przejrzysty podział systemu na klasy i obiekty oraz wykorzystane techniki obiektowe zagwarantują, że system będzie łatwo rozszerzalny o nowe funkcjonalności.

9.2 Przenośność

Wykorzystanie do implementacji systemu języka Java zapewnia jego przenośność. Z kolei niezależność od bazy danych zapewni pakiet JDBC dostarczający interfejs programowania pozwalający na komunikację z wieloma bazami danych.

9.3 Łatwość obsługi

Dołożono wielkich starań, aby interfejs użytkownika był wystarczająco prosty oraz na tyle intuicyjny, aby nawet niedoświadczony użytkownik bez problemu mógł korzystać z systemu. Zdecydowano się na aplikację w pełni internetową, tak aby użytkowanie systemu sprowadzało się do wpisania odpowiedniego adresu internetowego systemu oraz korzystanie z niego, jak ze strony WWW.

9.4 Bezpieczeństwo

Overlord jest systemem, gdzie bezpieczeństwo danych odgrywa kluczową rolę w jego funkcjonowaniu. Do przechowywania danych zostanie wykorzystana więc sprawdzona i ciesząca się powszechnym uznaniem baza PostgreSQL.

9.5 Niezawodność

Na niezawodność systemu zostanie położony szczególny nacisk. Celem jest napisanie systemu, który będzie mógł pracować nieprzerwanie oraz niezawodnie, niemniej mogą zaistnieć sytuacje niezależne od sposobu implementacji, mogące spowodować przerwę w działaniu systemu (np. przerwa w dostawie prądu). Rozwiązania wprowadzone w celu zapobiegania utracie danych to m.in. zapasowa kopia bazy danych oraz prowadzenie dziennika zmian.