

**Politechnika Krakowska  
im. Tadeusza Kościuszki**

Wydział Informatyki i Telekomunikacji

**Kraków 2020**

**Daniel Pawłowski**

Numer albumu: 132484

**Przeglądarkowa gra sieciowa czasu rzeczywistego oparta na serwerze aplikacji Spring Boot**

**A Web-based real time network game,**

**based on a Spring Boot server application**

**Praca magisterska  
 na kierunku informatyka**

Praca wykonana pod kierunkiem:  
**dr inż. Jerzego Białasa**

**Uzgodniona ocena:**......................................

...........................................................................podpisy promotora i recenzenta

Spis treści:

1. **Wstęp**
   1. Cele i zadania pracy
   2. Przegląd podobnych rozwiązań
   3. Zawartość pracy
2. **Ogólny zarys projektu**
3. **Wykorzystane technologie**
   1. Technologie programistyczne
      1. Java
      2. Spring Framework
      3. JavaScript
      4. Angular
   2. Technologie frontendowe
      1. HTML
      2. CSS
   3. Technologie bazodanowe
      1. mySQL
      2. JPA
   4. Technologie bezpieczeństwa
      1. JWT
      2. Auth0
   5. Technologie testowania
      1. jUnit
   6. Technologie komunikacji
      1. http
      2. json
      3. REST
   7. Pozostałe
      1. Maven
   8. Biblioteki zewnętrzne
      1. JPA
      2. Jackson
4. **Architektura aplikacji**
   1. Aplikacja frontendowa
   2. Aplikacja backendowa
   3. Architektura bazy danych
   4. Konfiguracja bezpieczeństwa
5. **Opis gry**

Podsumowanie pracy

Bibliografia

Wykaz rysunków i listingów

**1.Wstęp**

Rozwój techniki w całym XX wieku sprawił, że pojęcie rozrywki nabrało innego znaczenia, spora jej część przeniosła się czy to do telewizji czy też przed komputer osobisty. Jeśli w przypadku telewizji użytkownik jest tylko biernym odbiorcą przekazu, tak w przypadku rozrywki z komputerem lub telefonem ma już ogromny wpływ na to, co zobaczy. Jednym ze sposobów na spędzanie czasu są gry. Wraz z rozwojem technologii rozgrywka zmieniała swoje oblicze: od prostej gry Pong wydanej w 1972 roku, przez gry na automatach, później konsolach, a skończywszy na grach na komputery osobiste i na urządzenia mobilne. Według szacunków firmy Newzoo przychody branży gier wideo w 2020 roku wyniosą około 160 miliardów dolarów [1], co jeszcze kilkadziesiąt lat temu wydawało się niemożliwe. W świecie wirtualnej rozrywki każdy znajdzie coś dla siebie, ulubiona drużyna przegrała mecz? Możesz podjąć tego samego przeciwnika na konsoli i zwyciężyć. Chcesz zawalczyć na frontach drugiej wojny światowej? Istnieje bardzo wiele symulatorów pozwalających wcielić się w żołnierza. Poza tym liczba gatunków gier jest niezliczona, między innymi: strategie, symulatory, gry platformowe, gry zręcznościowe czy gry logiczne. Innym ważnym czynnikiem uatrakcyjniającym rozgrywkę było powstanie Internetu, który pozwala na rozgrywkę z dowolną liczbą osób podłączonych do sieci. Na początku lat 90 powstał protokół http, pozwalał on na przesyłanie tak zwanego hipertekstu. Tekst taki jest następnie przetwarzany przez przeglądarkę internetową, a przetworzona treść wyświetlana jest na ekranie dla użytkownika. W poniższej pracy opisana zostanie gra przeniesiona do przeglądarki Internetowej.

**1.1 Cel i zadania pracy**

Celem pracy jest opis projektu, który jest przeglądarkową grą czasu rzeczywistego. Gracz rozpoczyna rozgrywkę w momencie rejestracji i trwa ona nieprzerwanie aż do momentu, w którym gracz postanowi usunąć swoje konto lub zostanie ono usunięte z innego powodu. Większość obliczeń jest wykonywana na głównym serwerze aplikacji, który działa nieprzerwanie, akcje wykonywane przez użytkownika w przeglądarce pozwalają na przesyłanie zapytań przy użyciu protokołu http do serwera, który następnie zapisuje te dane do bazy danych, a także wysyła do użytkownika informacje zwrotne. Realizacja projektu składa się z:

* Oprogramowanie silnika gry
* Stworzenie interfejsu użytkownika (jako projekt Angular),
* Konfiguracja zewnętrznego serwera uwierzytelniania i autoryzacji (z wykorzystaniem strony auth0.com),
* Utworzenie struktury bazy danych do przechowywania informacji o grze,
* Utworzenie testów jednostkowych zarówno dla serwera aplikacji jak i aplikacji Angular,
* Stworzenie serwera aplikacji, który będzie:
  + obsługiwał zapytania użytkowników,
  + wykonywał obliczenia związane bezpośrednio z rozgrywką,
  + (coś z bezpieczeństwem, sprawdzanie bezpieczeństwa)
  + Komunikował się z bazą danych

Gra stworzona na potrzeby tej pracy jest grą strategiczną umieszczoną w świecie fantasy, gdzie rozgrywka toczy się nieprzerwanie, posiada ona aspekty ekonomiczne i wojenne. Została ona zaprojektowana w taki sposób aby jej dalszy rozwój był prosty i nie wymagał zmiany głównych części projektu.

**1.2 Przegląd podobnych rozwiązań**

W sieci istnieje wiele podobnych gier jednak każda z nich będzie posiadać cechy, które będą rozróżniać je od innych. Do takich cech należeć będą:

* Ekonomia rozgrywki,
* System walki,
* Uniwersum, w którym odbywa się rozgrywka
* I wiele innych

W rozdziale tym zostanie omówione kilka najbardziej rozpoznawalnych gier przeglądarkowych

**1.2.1 OGame**



Rys. 1.1 Logo gry OGame

Gra ta została stworzona przez niemiecką firmę Gameforge w 2002 roku, w roku 2005 została ona przetłumaczona na język polski i wtedy też powstał pierwszy polski serwer. Rozgrywka toczy się w przestrzeni kosmicznej, każdy gracz po założeniu konta otrzymuje jedną planetę i niewielką ilość zasobów na dalszy rozwój. W grze co jakiś czas powstają nowe serwery, nazywane uniwersami. W każdym uniwersum istnieje 9 galaktyk, a w każdej galaktyce 500 układów słonecznych, w których jest miejsce dla 15 planet. Każda z planet może mieć pozycję od 1 do 15 w danym układzie słonecznym, im wyższy numer pozycji planety, tym na planecie występuje niższa średnia temperatura. Każdy gracz może zasiedlać nowe planety (tworzyć kolonie), a ich limit ograniczony jest poziomem techniki astrofizyki. W rozgrywce występują trzy główne surowce: metal, kryształ i deuter. Mogą one zostać wykorzystane do rozbudowy budynków, przeprowadzania badań, tworzenia jednostek obronnych lub statków kosmicznych, a deuter jest też wykorzystywany jako paliwo dla floty. Oprócz trzech wspomnianych powyżej surowców każda planeta posiada też energię, którą można pozyskać za pomocą budowy elektrowni lub umieszczania satelitów słonecznych na orbicie planety. Energia służy do zasilania kopalni, może także pomóc w badaniach. Gracze wchodzą w interakcje między innymi poprzez handel lub ataki. W grze istnieją także sojusze, które mogą toczyć ze sobą wojny.



Rys. 1.2 Ekran główny gry OGame

**1.2.2 Plemiona**



Rys. 1.3 Logo gry Plemiona

Plemiona (zwane też w oryginale jako Tribal Wars) jest grą czasu rzeczywistego, której akcja toczy się w średniowieczu. Twórcami są ludzie ze studia InnoGames, powstała ona w 2003 roku. Polska wersja gry została uruchomiona w 2006 roku. Istnieje wiele serwerów zwanych też światami. Każdy nowy gracz rozpoczyna rozgrywkę z jedną wioską. W grze istnieje trzy rodzaje surowców: drewno, glina i żelazo pozyskiwane odpowiednio z tartaku, cegielni i huty żelaza. Ilość surowców jaka otrzymywana jest w określonym okresie czasu zależna jest od poziomu powyższych budynków. Zasoby te przechowywane są w spichlerzu i należy pamiętać o rozbudowywaniu go, gdyż przekroczenie jego pojemności powoduje utratę surowców. Do tworzenia jednostek bojowych należy zadbać o odpowiednią liczbę ludności, którą można zwiększyć poprzez podnoszenie poziomu zagrody. Rozbudowa rynku pozwala na handlowanie surowcami wraz z najbliższymi sąsiadami. W zależności od poziomów różnych budynków dostępne są różne rodzaje wojsk. W trakcie rozgrywki istnieje możliwość dołączenia do plemienia, współplemieńcy mogą pomóc w rozbudowie wioski lub też wspomóc swoimi jednostkami militarnymi w przypadku ataku innych graczy. Wraz ze startem serwera administratorzy gry określają warunki, po spełnieniu których gra zostaje wygrana.



Rys 1.4 Pogląd wioski w Plemionach

**1.2.3 Shakes & Fidget**



Rys 1.5 grafika z gry Shakes & Fidget

**1.3 Zawartość pracy**

W rozdziale drugim opisany zostanie ogólny zarys projektu – przedstawione zostaną główne elementy, jakie role będzie posiadał każdy z nich i w jaki sposób odbywa się komunikacja między nimi.

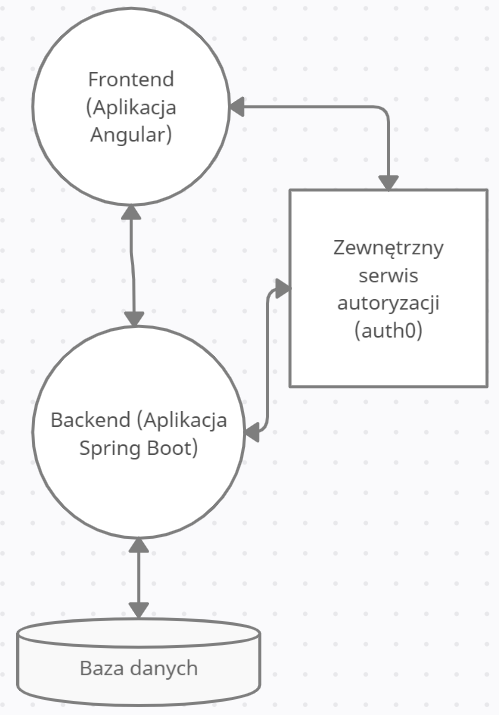
Rozdział trzeci zostanie poświęcony przedstawieniu wszystkich wykorzystanych technologiach – krótkich historiach ich powstania, opisie, zastosowaniach a także zastosowaniach w projekcie.

W kolejnym, czwartym rozdziale każdy z elementów aplikacji zostanie dogłębnie omówiony, zostaną wskazane najważniejsze komponenty, a także przedstawione zostaną konfiguracje niezbędne do prawidłowego działania.

Piąty rozdział opisywał będzie właściwą rozgrywkę – jakie akcje może wykonać użytkownik, a także zasady działania wszystkich elementów rozgrywki.

Rozdziały szósty i siódmy skupią się na podsumowaniu projektu, a także na wskazaniu materiałów źródłowych, na podstawie których powstała ta praca.

**2. Ogólny zarys projektu**

****

Rys 2.1 Ogólna architektura rozwiązania

Pierwszym elementem aplikacji jest komponent frontendowy. Został on napisany w technologii Angular. Jego głównym zadaniem jest wyświetlanie treści użytkownikowi. Komunikuje się on bezpośrednio z serwerem backendowym, a także z zewnętrznym serwisem autoryzacyjnym. W komponencie tym użyte zostały technologie takie jak:

* Html – wykorzystywna do tworzenia treści i struktury wyświetlanych stron,
* Css – odpowiedzialny jest za wygląd i wyświetlanie się elementów html
* JavaScript – dzięki niemu możliwe jest dodanie dynamicznych elementów pozwalających na interakcję z użytkownikiem,
* JSON – format danych wykorzystywany w przesyle informacji pomiędzy komponentami

Autoryzacja użytkownika możliwa jest poprzez wykorzystanie zewnętrznego serwisu auth0. Na potrzeby projektu została utworzona w serwisie tym aplikacja. Dzięki odpowiedniej konfiguracji pozwala ona na rejestrację oraz logowanie w bezpieczny sposób.

Aplikacja backendowa spina wszystkie komponenty. W tym miejscu odbywają się wszystkie obliczenia dotyczące bezpośrednio rozgrywki, wysyłania danych niezbędnych do wyświetlenia w aplikacji frontendowej, danych potrzebnych do zapisania w bazie danych i decyzji do jakich zasobów powinien mieć dostęp dany użytkownik. Najważniejsze technologie użyte w tym komponencie:

* Java,
* Spring,
* Maven

Baza danych – jak wskazuje nazwa, służy do przechowywania danych o kontach użytkowników. Wykorzystuje ona model tak zwanej relacyjnej bazy danych, a systemem zarządzania jest MySQL.

Komunikacja między aplikacją frontendową a serwisem auth0 odbywa się za pomocą protokołu http. Aby możliwa była jakakolwiek akcja podjęta przez użytkownika potrzebuje on tak zwanego tokenu autoryzacyjnego. Jeśli takowy nie jest posiadany – musi odbyć się autoryzacja – polega ona na tym, że korzystający z aplikacji zostaje przekierowany na specjalną stronę logowania, gdzie po wprowadzeniu swoich danych i udanej autoryzacji otrzymuje on token. Nie będzie on widoczny dla użytkownika ale zostanie zapisany przez aplikację i do czasu jego wygaśnięcie będzie służył do autoryzacji wszystkich podjętych akcji.

Komunikacja aplikacji angular i spring boot podobnie jak w poprzednim przypadku odbywa się przy wykorzystaniu protokołu http. Wszelkie aktywności, które wykonuje użytkownik zostają upakowane w zapytanie http wykorzystujące format danych JSON jako ciało zapytania i odpowiednie nagłówki (w tym nagłówek autoryzacyjny). Skutki podjętych działań są następnie przetwarzane przez aplikację backendową i zostaje wysłana informacja zwrotna.

Aplikacja spring boot komunikuje się z serwisem auth0 aby sprawdzić poprawność tokena podawanego w zapytaniach użytkownika i aby uzyskać dodatkowe informacje o tymże użytkowniku.

Ostatnim komponentem istniejącym w aplikacji jest baza danych, do której dostęp ma komponent backendowy. Komunikacja odbywa się przez interfejs JDBC.

**3. Wykorzystane technologie**

**3.1 Technologie programistyczne**

**3.1.1 Java**



**Rys 3.1 Logo języka programowania Java**

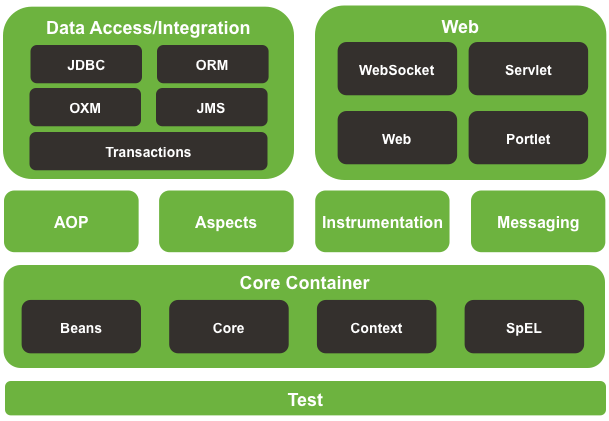
Java jest obiektowym językiem programowania opartym o klasy. Główną domeną języka Java jest „write once, run anywhere” (napisz raz, uruchom wszędzie), oznaczająca że raz skompilowany kod może działać na dowolnej maszynie wspierającej ten język programowania. Kod Javy kompilowany jest na tak zwany kod bajtowy, który może zostać uruchomiony przez maszynę wirtualną Javy (JVM) niezależnie od systemu na którym działa. Składnia tego języka jest podobna do języka C/C++ jednak nie posiada ona kilku niskopoziomowych funkcjonalności (takich jak na przykład wskaźniki), a posiada na przykład wbudowany mechanizm zarządzania pamięcią (tak zwany garbage collector) czy też mechanizm refleksji, który nie występuje w tradycyjnych językach programowania. Głównym założeniem Javy jest obiektowość. Oznacza to, że procedury i obiekty są ze sobą ściśle powiązane. Java jest też językiem silnie typowanym, co rozumieć należy przez to, że każde wyrażenie będzie mieć przypisany typ danych i próba wykorzystania operacji dla typu do tego nie przeznaczonego będzie kończyć się niepowodzeniem. Pierwsza wersja Javy oznaczona jako 1.0 została wydana w 1996 roku przez firmę Sun Microsystems. Wykorzystywana była ona głównie przez przeglądarki podczas uruchamiania tak zwanych apletów na stronach internetowych. Od tego czasu postało wiele kolejnych wersji Javy. Aktualnie najnowszą wersją jest Java 15 wydana w dniu 15 września 2020 roku. Java jest jednym z najpopularniejszych języków programowania. Według tak zwanego wskaźnika TIOBE index, wskazującego popularność różnych języków programowania tworzonego przez firmę TIOBE, za grudzień 2020 roku, z wynikiem 12,53% Java jest drugim najpopularniejszym językiem programowania i ustępuje tylko językowi C [przypis].

**3.1.2 Spring Framework**



**Rys 3.2 Logo Spring Framework**

Spring Framework jest wszechstronnym narzędziem programistycznym dla nowoczesnych aplikacji biznesowych tworzonych w języku Java. Narzędzie to działa na licencji open-source co oznacza, że kod jest ogólnie dostępny i nie jest wymagane uiszczanie żadnych opłat ze używanie go. Pierwsza wersja została napisana przez Roda Johnsona w 2003 roku i opublikowana na licencji Apache 2.0. Głównym założeniem Spring Framework jest ułatwienie tworzenia rozbudowanych aplikacji w języku Java. Jednym z głównych mechanizmów wykorzystywanych przez to narzędzie jest wstrzykiwanie zależności (dependency injection). Spring Framework posiada tak zwany „kontener” (inversion of control container). Kontener ten odpowiedzialny jest za zarządzaniem obiektami: tworzy je, uruchamia metody inicjalizujące, a także konfiguruje je tworząc odpowiednie powiązania. Obiekty stworzone przez to narzędzie nazywane są „beanami”. Konfiguracja takiego kontenera może być zapisana w pliku .xml lub zostać umieszczona w plikach klas Javy z użyciem adnotacji. Platforma Spring posiada wiele modułów ułatwiających tworzenie aplikacji, ogólny schemat został pokazany na rysunku 3.3.



**Rys 3.3 Schemat modułów Spring Framework**

Spring Data – głównym zadaniem tego modułu jest zapewnienie przyjaznego i zwięzłego modelu dostępu do bazy danych. Pozwala w prosty sposób połączyć się z relacyjnymi i nierelacyjnymi bazami danych, a także usługami sieciowymi zapewniającymi dostęp do danych. Projekt ten jest zbiorem pomniejszych modułów zapewniających dostęp do różnych implementacji bazodanowych takich jak JDBC, JPA czy MongoDB. Spring Data ułatwia także konstruowanie i wysyłanie zapytań bazodanowych na przykład poprzez tworzenie zapytania na podstawie nazwy metody.

Spring Web MVC – moduł służący do budowy aplikacji przeglądarkowych opartych o architekturę Model-View-Controller. Komponenty te odpowiedzialne są za:

* Model – definiuje i przechowuje dane używane w aplikacji,
* View – odpowiedzialny jest za wyświetlanie danych zawartych w modelu, zazwyczaj używa kodu html przetwarzanego następnie przez przeglądarkę,
* Controller – służy do przetwarzania zapytań użytkownika, budowę odpowiedniego modelu i przekazywanie go do „view” aby ten mógł wyświetlić odpowiednie dane.

Spring Web MVC oparty jest o tak zwany DispatcherServlet, którego schemat działania został przedstawiony na rys 3.4. Element ten zarządza wszystkimi zapytaniami http, zarówno przychodzącymi jak i wychodzącymi.



**Rys 3.4 Diagram działania DispatcherServleta.**

Przychodzące zapytanie http zostaje przetwarzane w następujących krokach:

1. Odebrane zapytanie zostaje odczytane przez HandlerMapper, który następnie uruchamia odpowiedni Controller.
2. Controller odbiera zapytanie i wywołuje odpowiednią metodę w serwisie w zależności od wywołanej usługi i metody http (POST, GET itd.). Zapytanie zostaje przetworzone według logiki biznesowej zawartej w wywołanej metodzie, a następnie ustawiony zostaje model, który zostaje wysłany z powrotem do DispatcherServleta.
3. Kolejnym krokiem jest pobranie z View Resolvera odpowiedniego widoku.
4. Ostatecznie widok zostaje wysłany do przeglądarki gdzie będzie wyświetlony.

Wszystkie wspomniane powyżej elementy (Handler Mapper, Controller, View Resolver) są częściami kontekstu webowego (WebApplicationContext).

Spring AOP – jest to moduł służący jako element pośredniczący między różnymi komponentami. Moduł ten pozwala na programowanie zorientowane aspektowo. Oznacza to, że tworzony kod jest bardziej modularny co z kolei oznacza możliwość na przykład oddzielenia warstwy technicznej od logiki biznesowej.

Spring Security – moduł służący do zapewnienia bezpieczeństwa aplikacji. Posiada mechanizmy wspierające autoryzację i dostęp do zasobów dla aplikacji bazujących na frameworku spring. Moduł ten pozwala ochronić aplikację przed atakami takimi jak: przechwytywanie sesji, clickjacking czy cross site request forgery.

Spring Boot –pozwala w prosty sposób uruchomić aplikację opartą o spring. Głównymi cechami tego rozwiązania są:

* Proste uruchomienie – w przeciwieństwie do „podstawowego” springa, Spring Boot posiada wbudowany serwer aplikacji, a także wszystkie pozostałe elementy niezbędne do uruchomienia aplikacji,
* Automatyczna konfiguracja – aby uruchomić aplikację nie jest potrzebna żadna dodatkowa konfiguracja.
* Szybkość – wykorzystanie Spring Boota pozwala na dużo prostsze tworzenie aplikacji co pozwala na szybszy i tańszy proces developmentu.

**3.1.3 JavaScript**



**Rys 3.5 Logo JavaSript**

JavaScript jest językiem skryptowym programowania wysokiego poziomu. Wraz z CSS i HTML jest podstawą interaktywnych stron internetowych. Oprócz tego JavaScript posiada szersze zastosowanie, wykorzystuje się go w aplikacjach desktopowych, dokumentach PDF czy też aplikacjach internetowych po stronie serwera. JS jest językiem interpretowalnym co oznacza, że program napisany w takim języku nie jest kompilowany (nie jest zamieniany na postać bitową), a przechowywany jest jako kod źródłowy. Następnie podczas uruchomienia takiego programu jest on czytany, interpretowany, a na koniec interpreter tego języka wykonuje go. Język ten posiada wiele paradygmatów, można w nim programować imperatywnie, obiektowo i funkcyjnie.

JavaScript posiada następujące cechy:

* Kod wykonywany jest po stronie klienta co pozwala na odciążenie serwera,
* Skrypty mogą zostać wykonanie na dowolnej stronie internetowej niezależnie jakie posiada ona rozszerzenie
* JavaScript jest niezależny od systemu operacyjnego
* Składnia języka jest relatywnie prosta
* Do zmiennej można przypisać funkcję
* Wykorzystywane jest typowanie dynamiczne co oznacza że typ zmiennej nadawany jest podczas działania programu i może się zmieniać.
* Typowanie słabe – jeśli zachodzi taka potrzeba, typ danych zmiennej zostaje zmieniony

**3.1.4 Angular**