**Marcin Kużdowicz**

**Projekt i implementacja systemu społecznościowego do gry w szachy w sieci internetowej**

1. **Uruchomienie aplikacji w systemie linux i instalacja komponentów:**

Komponenty, oprogramowanie i wersje potrzebne do uruchomienia aplikacji:

1. **Java 8 (Wraz z JDK „Java Development Kit”)**

Do sprawdzenia wersji można się posłużyć poleceniem wykonanym w terminalu:

**java -version**

1. **Eclipse IDE for Java EE Developers - MARS**

(Środowisko programistyczne dla języka Java do aplikacji webowych)

Strona z oprogramowaniem Eclipse: <https://eclipse.org/downloads/>

1. **Apache Maven:**

(program do budowania projektu i do zarządzania jego cyklem życia)

Instalacje należy wykonan z linii komend systemu operacyjnego.

(Ja pracowałem na systemie Ubuntu 15.04.)

* 1. Należy wykonać komendy**:**

**sudo apt-get install maven**

**sudo apt-get update**

To powinno wystarczyć. Opcjonalnie można wykonać komendy ustawiające zmienne środowiskowe (nie jestem dokładnie pewien dawno instalowałem maven na linuxie, sprawdzę po przeinstalowaniu systemu na świeżo):

export M2\_HOME=/usr/local/apache-maven/apache-maven-3.1.1

export M2=$M2\_HOME/bin

export MAVEN\_OPTS="-Xms256m -Xmx512m"

export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0

export PATH=$M2:$PATH

* 1. Komenda sprawdzająca czy maven jest zainstalowany:

**mvn -version**

1. **Baza danych MongoDb.**

Instalacja dla systemu Ubuntu**.** Informacje na temat instalacji:

[**https://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/**](https://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/)

* 1. Należy wykonać komendy:

**sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10**

(pobranie publicznego klucza używanego przez package management system, do pobrania oprogramowania przez Internet)

**echo "deb http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.0.list**

**sudo apt-get update**

**sudo apt-get install -y mongodb-org**

* 1. weryfikacja instalacji komendą wyświetlającą wersje oprogramowania:

**mongo -version**

* 1. uruchomienie bazy danych

Ja używałem komend:

**mongod**

(uruchomienie demona)

**mongo**

(uruchomienie powłoki dla bazy danych mongodb do wykonywania skryptów)

można tez użyć komendy:

**sudo service mongod start**

**ctr + c wyłącza proces**

* 1. Zainicjowanie bazy danych dla aplikacji:

Żeby uruchomić bazę, należy wykonać komendy:

**mongod**

**mongo**

lub samo

**mongo**

I wykonac skrypt zawarty w pliku:

**init\_mongo.js**

1. **Uruchomienie aplikacji:**
2. Otworzyć terminal linii poleceń
3. Uruchomić bazę danych
4. Przejść do głównego folderu aplikacji: **/chessApp**
5. Wykonać komendę:

**mvn clean install**

lub

**mvn clean install**

i

**mvn jetty:run**

**(jeżeli aplikacja się nie uruchomi automatycznie)**

1. Otworzyć przeglądarkę i wpisać adres: **localhost:8080**
2. Do aplikacji można się zalogować jako administrator wpisując:

login: **admin**

hasło **admin**

1. Ponadto można tworzyć nowe konta przez interfejs użytkownika
2. Żeby przetestować komunikator do gry w szachy należy uruchomić dwie różne przeglądarki i załogować się z dwóch różnych kont, wtedy można zasymulować grę między dwoma osobami.
3. **Uruchomienie aplikacji w systemie Windows i instalacja komponentów:**

Komponenty, oprogramowanie i wersje potrzebne do uruchomienia aplikacji:

* + - 1. **Java 8 (Wraz z JDK „Java Development Kit”)**

Do sprawdzenia wersji można się posłużyć poleceniem wykonanym w terminalu:

**java -version**

* + - 1. **Eclipse IDE for Java EE Developers - MARS**

(Środowisko programistyczne dla języka Java do aplikacji webowych)

Strona z oprogramowaniem Eclipse: <https://eclipse.org/downloads/>

* + - 1. **Apache Maven:**

(program do budowania projektu i do zarządzania jego cyklem życia)

Instalacja**:**

Strona z oprogramowaniem Apache Maven: <http://maven.apache.org/download.cgi>

Szczegóły na temat instalacji: <http://maven.apache.org/install.html>

trzeba ściągnąć plik binarny, w przypadku systemu Windows najlepiej plik w formacie . zip:

**apache-maven-3.3.3-bin.zip**

Następnie rozpakować plik.

Zapisać do folderu program files, co stworzy ścieżkę:

**C:\Program Files\apache-maven-3.3.3\bin**

1. Dodać ścieżkę do zmiennej środowiskowej : Path:

Należy kliknąć: Menu start, później

Computer (prawym przyciskiem myszy) > properties > Adnaced system settings > Environment variables

W System variables:

1. stworzyć zmienną środowiskową JAVA\_HOME, jeżeli nie istnieje, do której będzie zmapowana ścieżką z instalacją Javy w folderze program files:

Przykładowa ścieżka z instalacją Javy: **C:\Program Files (x86)\Java\jdk1.8.0\_60**

Istnienie zmiennej JAVA\_HOME możemy sprawdzić komendą:

**echo %JAVA\_HOME%**

Powinna się wyświetlić ścieżka z instalacją Javy

1. do zmiennej srodowiskowj Path dodać ścieżkę: **C:\Program Files\apache-maven-3.3.3\bin**

(ścieżki są separowane średnikami )

1. Następnie sprawdzamy stan instalacji maven’a komendą:

**mvn –v**

Możemy także sprawdzić stan zmiennych środowiskowych komendami:

**echo %Path%**

**echo %JAVA\_HOME%**

* + - 1. **Baza danych MongoDb.**

Instalacja**:**

Strona z oporgramowanie: <https://www.mongodb.org/downloads>

Strona o instalacji: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/?_ga=1.53835946.323062929.1441664626>

1. Należy sciągnąć pliki instalacyjne z powyższej stronyPlik jest w formacie .msi

**mongodb-win32-x86\_64-2008plus-ssl-3.0.6-signed.msi**

MognoDb zostanie domyślnie zainstalowane w folderze:

**C:\Program Files\MongoDB\Server\3.0\bin**

W przypadku instalacji customowej, można wybrać dowolną lokalizacje.

1. Następnie należy zdefiniować folder w którym mongodb będzie przechowywać dane. Wykonuje się to komendą: **md ścieżka do pliku** np.:

**md \data\db**

1. Uruchomienie bazy danych następuje po uruchomieniu pliku **mognod.exe**

Plik znajduje się w katalogu bin z instalacją

Bazę uruchamia się z linii komend.

Przykład uruchomienia:

**C:\Program Files\MongoDB\Server\3.0\bin\mongod.exe**

**Control+C** wyłącza bazę.

Uruchomienie Mongo Shell (interfejsu do wykonywania akcji na bazie danych)

Za pomocą uruchomienia pliku mongo.exe

Komenda:

**C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe**

1. **Należy wykonać komendy zapisane w pliku init\_mongo.js**

**(MongoDB dla Windows 64-bit działa tylko na systemach:**

Windows Server 2008 R2, Windows 7 64-bit i nowsze wersje)

* + - 1. **Uruchomienie aplikacji:**

1. Otworzyć terminal linii poleceń
2. Uruchomić bazę danych
3. Przejść do głównego folderu aplikacji: **/chessApp**
4. Wykonać komendę:

**mvn clean install**

lub

**mvn clean install**

i

**mvn jetty:run**

**(jeżeli aplikacja się nie uruchomi automatycznie)**

1. Otworzyć przeglądarkę i wpisać adres: **localhost:8080**
2. Do aplikacji można się zalogować jako administrator wpisując:

login: **admin**

hasło **admin**

1. Ponadto można tworzyć nowe konta przez interfejs użytkownika
2. Żeby przetestować komunikator do gry w szachy należy uruchomić dwie różne przeglądarki i załogować się z dwóch różnych kont, wtedy można zasymulować grę między dwoma osobami.
3. **Krótki opis aplikacji**
   * + 1. **Główne Technologie użyte do tworzenia programu:**

* Java
* Spring MVC
* Spring security
* JSP (Java server pages)
* Jetty
* Maven
* Ajax
* JSON
* XML
* Websocket
* JavaScript
* Jquery
* Bootstrap
* MongoDB
* Html 5
* css 3
* GIT
* REST

Jest to aplikacja Webowa uruchamiana na serwerze **Jetty**. Jetty to serwer i zarazem Java Servlet container. Więcej informacji: <http://www.eclipse.org/jetty/>

Jetty uruchamić można poprzez maven’a

**mvn clean install**

Buduje aplikacje i uruchamia Cetty jeżeli budowanie jest skonfigurowane tak, że zaraz po budowaniu ma być uruchomine Cetty. Lub komendą

**mvn jetty:run**

Gdy przeniosę program na środowisko produkcyjne to albo zostanę przy Jetty albo skorzystam z serwera **Apache Tomcat.**

1. **Architektura aplikacji.**

Program jest oparty na wzorcu architektonicznym MVC (model view controller) przy pomocy frameworku Spring MVC.

Wszystkie klasy Javowe znajdują się w pakiecie **src/main/java**

* 1. **Kontroller**.

Wszystkie requesty HTTP obsługują klasy w pakiecie **com.chessApp.controllers .**

Te klasy pełnią rolę kontrolerów.

* 1. **View**

Widok jest realizowany za pomocą technologii: Jsp (java server pages) wraz z Jtsl (JavaServer Pages Standard Tag Library), HTML 5, CSS 3, JavaScript i Jquery oraz Bootstrap (framework HTML, CSS, JavaScript). Pliki widoków znajdują się w folderze:

**/src/main/webapp**

w folderze **assets** są pliki css i javascript

w folderze **WEB-INF** znajdują się konfiguracyjne .xml

w folderze **WEB-INF/** **views** są pliki .jsp

* 1. **Model**

Model jest zmapowany do klas javovwych w pakiecie **com.chessApp.model.**

Aplikacja wykorzystuje bazę danych MonogDB, która jest bazą NoSQL, określaną też jako baza obiektową. W programie jest zastosowana koncepcja **ORM** (object relational mapping), za pomocą technologii Spring Data. ORM mapuje obiekty i kolekcje z bazy danych MongoDB do klas Javovych.

3. **System do gry w szachy.**

Interfejs do gry w szachy oraz logika silnika do gry jest napisana po stronie klienta w języku javascript. Na ten moment realizują to biblioteki

**chessboard.js** (interfejs)

<http://chessboardjs.com/>

**chess.js** (silnik gry)

<https://github.com/jhlywa/chess.js/blob/master/README.md>

Łatwiej było mi na początek zaprogramować aplikację na czymś wzorcowym, postaram się zaimplementować później swoje wersje.

Te biblioteki też wymagały programowania, pliki zawierające kod prograsmujący te biblioteki pod tą aplikacji znajdują się w katalogu:

**/src/main/webapp/assets/js/**

**4. Gra w szachy użytkownik vs użytkownik.**

Możliwość gry online wymagała zaprogramowania komunikatora.

Do tego użyłem technologii Websocket. Klasy implementujące komunikator znajdują się w pakiecie: **com.chessApp.websocket.** Implementacja komunikatora opiera się na obsłudze zdarzeń po stronie serwera w postaci **Server endpoint (oprogramowany w Javie) i** po stronie klienta w postaci Client endpoint (**oprogramowany w JavaScript**), oraz przechowywaniu informacji w sesji połaczenia realizowany przez subprotokół websocket.

Server endpoint jest zaimplementowany w pliku **WebSocketServer.java** w pakiecie **com.chessApp.websocket.**

Client ednpoint w pliku **websocketClientEndpoint.js** obecnym w katalogu pod ścieżką **/src/main/webapp/assets/js/**

Stan gry I ruchy poszczególnych użytkowników są przkazywane między nimi za pomocą notacji

**FEN:** **Notacja Forsytha-Edwardsa.** Polega ona na zapisaniu położenia wszystkich figur I do kogo należy następny ruch w postaci ciągu znaków. Przykład notacji fen:

**„r1bqkbnr/pppp1ppp/2n5/1B2p3/4P3/5N2/PPPP1PPP/RNBQK2R”**

Do pracy nad aplikacją używałem systemu do kontroli wersji GIT.