## Tehnologii pe partea de server

### Java

#### Descriere

Java este un limbaj de programare obiect orientat, independent de platforma pe care ruleaza. Codul Java este compilat intr-un limbaj intermediar, numit bytecode, ce este rulat mai departe de un Java Virtual Machine. Acest Java Virtual Machine converteste bytecode-ul in codul masina. Java este unul dintre cele mai folosite limbaje pentru aplicatiile ce folosesc arhitectura client-server.

#### Motivatie

Am ales acest limbaj de programare, deoarece exista multe tool-uri ce faciliteaza dezvoltarea codului, exista o vasta documentatie pe internet pentru Java si nu in ultimul rand, exista multe framework-uri si librarii ce pot folosite in sprijinul dezvoltarii unei aplicatii de tip web. Exemple de astfel de framework-uri si librarii sunt urmatoarele: Spring, Hibernate.

### Tomcat

#### Descriere

Este o implementare a specificatiei Java servlet. Tomcat este un server web dezvoltat de Apache Software Foundation care executa servlet-uri de java si randeaza pagini web care include JSP.

#### Motivatie

Am ales acest server WEB, deoarece ofera suport pentru deploy-ul aplicatiilor scrise in java si arhivate in structuri de tip war.

### Spring

#### Descriere

Este un framework de dezvoltare a aplicatiilor pentru enterprise java. Spring ofera suport pentru ciclul de viata al obiectelor, injectie de dependinte, functionalitati ce ajuta programatorul sa se focuseze mai mult pe business-ul aplicatiei.

Cu ajutorul injectiei de dependinte oferit de Spring, programatorul are posibilitatea sa asambleze componentele intre ele.

##### Facilitati oferite de Spring

###### Injectia de dependinte

Orice aplicatie bazata pe limbajul de programare java are obiecte care lucreaza impreuna pentru a oferi o functionalitate. Pentru aplicatiile complexe este important ca obiectele sa fie cat mai independente cu putinta pentru a crea posibilitatea refolosirii lor. Injectia de dependinte oferita de Spring ofera suport in acest sens.

Exista doua tipuri de injectie de dependinte:

* Injectie de dependinte prin constructor (atunci cand container-ul invoca un constructor de clasa cu un numar de argumente, fiecare reprezentand o dependinta catre o alta clasa)
* Injectie de dependinte prin settere (atunci cand container-ul invoca un setter unui obiect avand ca argument o dependinta catre o alta clasa.

###### Inversion of Control (IoC)

Suportul oferit de Spring pentru inversion of control se aplica la management-ul obiectelor java si al ciclului de viata al acestora. Obiectele java ce sunt instantiate de containerul IoC se numesc bean-uri. Container-ul de IoC forteaza modelul de injectie a dependintelor pentru obiectele java, rezultand un sistem decuplat.

#### Motivatie

Am ales acest framework deoarece aveam nevoie de suport pentru organizarea si legarea componentelor intre ele. Fiind o arhitectura de tip layered, fiecare layer trebuie legat de layerul precedent. Spring m-a ajutat sa simplific arhitectura aplicatiei.

### Spring MVC

#### Descriere

Spring MVC este componenta WEB a framework-lui. Spring MVC ofera o functionalitate bogata pentru crearea aplicatiilor WEB robuste. Aceasta componenta este proiectata pentru a putea fi configurata cat mai usor. Spring MVC este construit pe baza arhitecturii Model-View-Controller, ceea ce reprezinta o repartizare a claselor java in functie de responsabilitatea acestora.

Spring MVC ofera suport pentru procesarea request-urilor HTTP ce vin de la client, prin manipularea formularelor transmise si afisarea informatiilor prezente pe partea de server intr-un mod organizat.

#### Motivatie

Am folosit Spring MVC deoarece ofera o separare clara a responsabilitatilor intre controllere, modele si view-uri, controllerele pot fi configurate folosing inversion of control, la fel ca si celelalte componente folosite in aplicatie. Astfel, in aplicatia de fata controllerele proceseaza request-urile HTTP, view-urile randeaza interfata grafica a aplicatiei, iar modelul este responsabil cu transmiterea informatiei

### Spring security

#### Descriere

Spring security este un framework ce ofera suport pentru management-ul utilizatorilor: autentificarea si autorizarea acestora pentru a vizualiza anumite pagini ale aplicatiei.

Autentificarea unui utilizator este mecanismul prin care sistemul identifica utilizatorii.

Autorizarea este un mecanism prin care sistemul determina rolul utilizatorului si nivelul de acces la anumite pagini ale aplicatiei.

#### Motivatie

Am folosit acest framework deoarece ofera suport pentru management-ul utilizatorilor si se integreaza cu framework-ul principal folosit in aplicatie, si anume Spring.

### MySQL

#### Descriere

Conform Wikipedia, *“****MySQL****este un*[*sistem de gestiune a bazelor de date*](https://ro.wikipedia.org/wiki/Sistem_de_gestiune_a_bazelor_de_date)[*relaționale*](https://ro.wikipedia.org/w/index.php?title=Baze_de_date_rela%C8%9Bionale&action=edit&redlink=1)*, produs de compania suedeza*[*MySQL AB*](https://ro.wikipedia.org/wiki/MySQL_AB)*și distribuit sub*[*Licența Publică Generală GNU*](https://ro.wikipedia.org/wiki/Licen%C8%9Ba_Public%C4%83_General%C4%83_GNU)*. Este cel mai popular SGBD*[*open-source*](https://ro.wikipedia.org/wiki/Open-source)*la ora actuala”.*

*MySQL este un RDBMS (relational database management system) folosit pentru dezolvtare aplicatiilor web.*

*Bazele de date relationale sunt folosite pentru a stoca si organiza volume meri de date. Termenul de baza de date relationala vine de la faptul ca datele sunt stocate in tabele diferite intre case sunt stabilite relatii utilizand chei primare si chei straine.*

#### Motivatie

Am folosit acest sistem de gestiune a bazelor de date, deoarece:

* Este usor de utilizat, fiind necesar un numar minim de pasi pentru instalarea si utilizarea acestui sistem de gestiune
* Este securizat: exista mai multe tipuri de utilizatori cu diferite roluri, avand atribuite diferite drepturi.
* Este rapid
* Este scalabil

### Hibernate

#### Descriere

Hibernate este un serviciu de persistenta obiect-relational, un ORM (object relational mapping) cu responsabilitatea principala de a mapa clasele java la tabelele din baza de date.

Object Relational Mapping este o tehnica de programare pentru convertirea datelor intre bazele de date relationale si limbajele de programare obiect orintate.

Principalul avantaj al ORM-urilor este acela ca ofera o interfata de nivel mai inalt bazei de date decat JDBC, ascunzand detaliile interogarilor SQL.

Un alt avantaj important este ca ORM-urile separa aplicatia de baza de date, ceea ce ofera posibilitatea schimbarii sistemului de gestiune a bazei de date cu usurinta.

#### Motivatie

Am ales sa folosesc Hibernate in primul rand deoarece este independent de baza de date folosita si ofera o interfata de nivel inalt asupra bazei de date.

### Junit

#### Descriere

Junit este un framework utilizat pentru testarea aplicatiilor scrise in limbajul de programare java.

Principalul avantaj al Junit este ca testele pot fi rulate automat. Un alt avantaj este ca testele pot fi organizate atat sub forma de suite de test,

#### Motivatie

Am folosit Junit pentru testarea corectitudinii algoritmilor utilizati in aplicatie.

### Jackson

#### Descriere

Jackson este o biblioteca ce permite serializarea si deserializarea obiectelor java in JSON. Principalul avantaj acestei librarii este ca nu e nevoie de o mapare manuala intre obiectul java si json. Un alt avantaj consta in faptul ca aceasta librarie nu necesita dependinte suplimentare.

#### Motivatie

Am folosit Jackson deoarece permite serializarea si deserializarea obiectelor Java intr-un mod transparent si usor de folosit.

### Maven

#### Descriere

Maven este un sistem de build si management al proiectelor, scris in Java. Functionalitatile sale principale sunt descrierea procesului de build al softwareului si descrierea dependentelor acestuia. Maven descarca automa bibliotecila neceare dintr-un repository online.

#### Motivatie

Am folosit acest sistem deoarece ofera suport pentru managementul dependintelor, descarcand automat bibliotecile necesare. Astfel am obtinut un mai bun control asupra versiunilor bibliotecilor si dependintelor dintre modulele aplciatiei.

### JSP

#### Descriere

JavaServer Pages este o tehnologie de programare pe partea de server ce permite crearea de pagini WEB dinamice si independente de platforma.

#### Motivatie

Am folosit JSP deoarece este tehnologia recomandata de randare a informatiilor ce vin din controllerele create cu sprijinul framework-ului Spring MVC.

## Tehnologii pe partea de client

### HTML

#### Descriere

HTML este un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afisate intr-un browser. Scopul HTML este acela de a prezenta informatiile prin organizarea lor in paragrafe, tabele, liste sau link-uri.

#### Motivatie

Am folosit HTML pentru organizarea informatiilor in paginile interfetei grafice.

### CSS

#### Descriere

Este un standard pentru formatarea elementelor unui document HTML. Se pot atasa diferite stiluri tag-urilor de HTML, astfel oferind o experienta mai placuta utilizatorului.

#### Motivatie

Am folosit CSS pentru a formata informatiile din paginile interfetei grafice intr-un mod intuitiv, cu focalizare asupra informatiilor importante.

### JavaScript

#### Descriere

Este un limbaj de programare obiect orientat bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor functionalitati in paginile web, codul javascript din aceste pagini fiind rulat de catre browser.

Utilizarea Javascriptului este dupa cum spun cei de la Wikipedia:

*“Cea mai des întâlnită utilizare a JavaScript este în scriptarea*[*paginilor web*](https://ro.wikipedia.org/wiki/Pagin%C4%83_web)*. Programatorii web pot îngloba în paginile HTML script-uri pentru diverse activități cum ar fi verificarea datelor introduse de utilizatori sau crearea de meniuri și alte efecte animate.*

*Browserele rețin în memorie o reprezentare a unei pagini web sub forma unui*[*arbore*](https://ro.wikipedia.org/wiki/Arbore)*de*[*obiecte*](https://ro.wikipedia.org/wiki/Obiect_(informatic%C4%83))*și pun la dispoziție aceste obiecte script-urilor JavaScript, care le pot citi și manipula. Arborele de obiecte poartă numele de*[*Document Object Model*](https://ro.wikipedia.org/w/index.php?title=Document_Object_Model&action=edit&redlink=1)*sau*[*DOM*](https://ro.wikipedia.org/w/index.php?title=DOM&action=edit&redlink=1)*. Există un standard*[*W3C*](https://ro.wikipedia.org/wiki/W3C)*pentru DOM-ul pe care trebuie să îl pună la dispoziție un browser, ceea ce oferă premiza scrierii de script-uri*[*portabile*](https://ro.wikipedia.org/w/index.php?title=Portabilitate&action=edit&redlink=1)*, care să funcționeze pe toate browserele. În practică, însă, standardul W3C pentru DOM este incomplet implementat. Deși tendința browserelor este de a se alinia standardului W3C, unele din acestea încă prezintă incompatibilități majore, cum este cazul*[*Internet Explorer*](https://ro.wikipedia.org/wiki/Internet_Explorer)*.”*

#### Motivatie

Principalul motiv pentru care am folosit Javascript este pentru a putea manipula tabla de sah.

### Jquery

#### Descriere

Jquery este o platforma de dezvoltare bazata pe Javascript, conceputa pentru a usura si imbunatati procese precum modificare continutului paginii in mod dinamic, selectarea informatiei din pagina sau animatii grafice.

#### Motivatie

Am folosit Jquery pentru navigarea intre mutarile unei liste de strategii de sah.

### Ajax

#### Descriere

Este prescurtarea de la Asynchronous Javascript and XML, este o tehnica de programare pentru crearea de aplicatii web interactive ce permite transmiterea de request-uri HTTP asyncrone.

#### Motivatie

Am folosit Ajax pentru a genera urmatoarea mutare in cadrul unul meci de sah.

### ChessBoard.js

#### Descriere

Este un API ce randeaza tabla de sah.

#### Motivatie

Am folosit acest API pentru randarea tablei de sah.

## Tehnologii pentru comunicarea intre client si server

### HTTP

#### Descriere

Este un protocol de comunicare bazat pe TCP/IP care este folosit pentru a transmite date in format text, HTML, Json. Dintre avantajele protocolului de comunicare HTTP se remarca urmatoarele:

* HTTP nu necesita o conexiune: clientul HTTP initiaza un request HTTP iar dupa ce request-ul este trimis, clientul se deconecteaza de la server si asteapta raspunsul HTTP. Server-ul proceseaza request-ul si reinitiaza conextiunea cu clientul, trimitand raspunsul HTTP inapoi.
* Media-independent: orice tip de date poate fi transferat utilizand HTTP, atat timp cat si clientul si serverul sunt capabili sa utilizeze acel tip de date.
* HTTP nu are stare

#### Motivatie

Am folosit HTTP deoarece acesta este protocolul implicit de comunicare in aplicatiile WEB bazate pe Spring MVC.

### JSON

#### Descriere

Este un format de prezentare si interschimb de date intre aplicatii. Principalul avantaj al JSON-ului este acela ca nu depinde de limbajul in care e scrisa aplicatia ce trimite sau cea ce primeste informatia.

#### Motivatie

Am folosit JSON deoarece necesita mai putine metadate decat XML.