## Overview, continut prezentare

## Contextul proiectului

## Scop si obiective

Scop

Realizarea unui sistem ce ofera suport de antrenament jucatorilor de sah, sistem ce respecta noile tehnologii si abordari din domeniul aplicatiilor WEB.

Obiectiv principal

Posibilitatea jucatorului virtual de a juca un meci la un anumit nivel de dificultate, cu timp de raspuns ce se incadreaza in limita specificata in cadrul cerintelor nefunctionale.

Definirea proiectarea si construirea unui sistem capabil sa ofere suport de antrenament jucatorilor de sah prin expunerea de strategii si probleme de sah.

Mod de relaxare pentru jucatorii mai putin experimentati.

## Studiu bibliografic – solutii existente

## Solutia propusa

Jucator virtual – posibilitatea de a juca sah cu un jucator virtual cu diferite nivele de dificultate

Strategii de sah – posibilitatea analizarii strategiilor de sah si descrierii mutarilor

Probleme de sah – dezvoltarea capacitatii de a juca sah

## Tipuri de utilizatori

Jucator neautentificat

* Este un jucator ocazional

Jucator autentificat

* Are acces la functionalitati aditionale precum salvarea meciurilor, vizualizare de statistici

Administratorul

* Administreaza utilizatorii

## Cerinte functionale

## Cerinte functionale – use case

## Cerinte nefunctionale

Utilizabilitatea

* Gradul in care sistemul poate fi folosit de catre utilizatorii specificati pentru a atinge obiectivele dorite cu eficienta si satisfactie.

Performanta

* Timpul de raspuns la paginile ce nu implica actiunile jucatorului virtual < 1 secunda
* Timpul de raspuns la paginile ce implica jucatorul virtual < 15 secunde

Suportabilitatea

* Facilitatea dezvoltarii ulterioare a sistemului

Disponibilitatea

* Timpul in care aplicatia este disponibila utilizatorilor: 99%.

Numarul minim de utilizatori concurenti suportat

* Pentru o arhitectura cu un singur nod de procesare trebuie, sistemul trebuie sa suporte minim 20 de utilizatori concurenti.

## Optiuni tehnologice

Tomcat

* Container-ul WEB al aplicatiei

MySQL

* Server-ul dedicat bazei de date

Maven

* Suportul pentru construirea proiectului si management-ul bibliotecilor utilizate

Java

* Limbajul de dezvoltare pe partea de server

Spring

* Managementul obiectelor Java folosite

Spring MVC

* Organizarea interfetei grafice a aplicatiei sub forma arhitecturii Model-View-Controller

Hibernate

* Maparea obiectelor java la tabele din baza de date

HTML, CSS, Java script, Jquery, Ajax

* Interfata grafica si comunicarea cu serverul

## Arhitectura conceptuala

Diagrama cu Arhitectura aplicatiei

## Arhitectura software

Arhitectura pe server

Arhitecutra jucatorului virtual

## Detalii de implementare

Comunicarea intre client si server se realizeaza utilizand protocolul HTTP (imagine)

Evitarea introducerii datelor eronate pe server se face folosind validare in layer-ul de Controller

## Detalii de implementare jucator virtual

Algoritmul folosit pentru generarea mutarii urmatoare: MINIMAX (imagine cu flow-ul generarii urmatoarei mutari)

Functia de evaluare se bazeaza pe urmatoarele criterii:

Reprezentarea interna a unei pozitii de sah:

Generarea mutarilor posibile pentru o anumtia piesa: s-au folosit pozitii precalculate pentru a evita calcule redundante la runtime

## Dificultati – rezolvari

Deployment-ul aplicatiei in container-ul web a fost o provocare pana la solutionarea problemei: obtinerea unei licente gratuite pentru IDE-ul IntelliJ IDEA ce ofera suport pentru deployment

Timpul de raspuns al jucatorului virtual – folosirea bitboard-urilor precomputate pentru generarea mutarii urmatoare.

Functia de evaluare – stabilirea criteriilor de evaluare a unei pozitii de sah in cadrul jucatorului virtual

Testarea utilizabilitatii sistemului – sondaj facut pentru jucatorii experimentati dar si pentru jucatorii ocazionali, neexperimentati. (imagine cu rezultatele sondajului)

## Elemente de testare

Proces de dezvoltare iterativ -> testare manuala la finalizarea implementarii unei componente

## Elemente de evaluare

## Interfata grafica

Print screen-uri cu interfata grafica

## Concluzii

Sistemul reuseste sa isi atinga scopul, de a putea juca un meci de sah pe diferite nivele de dificultate cu timp de raspuns ce se incadreaza in cel stabilit in cadrul cerintelor nefunctionale

Sistemul ofera suport pentru de antrenament jucatorilor de sah prin expunerea de strategii de joc si probleme.

## Dezvoltari viitoare

Extragerea jucatorului virtual intr-un serviciu WEB separat

Folosirea arhitecturii cluster pentru jucatorul virtual – instalarea serviciului WEB cu jucatorul virtual pe mai multe servere si folosirea unui load balancer pentru a obtine performanta dorita la un numar ridicat de jucatori ce utilizeaza aplicatia simultan.

Adaugarea unui dictionar de mutari pentru jucatorul virtual

## Referinte bibliografice