

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

# BAKALÁŘSKÁ PRÁCE



Martin Postupa

## Webová databáze šachových partií

Katedra softwarového inženýrství

Vedoucí bakalářské práce: Mgr. Pavel Ježek Ph.D.

Studijní program: Informatika

Studijní obor: Programování

Praha 2015

## Obsah

1.	Úvod.....	4
1.1.	Podobné databáze na internetu .....	5
1.2.	Cíl práce .....	7
2.	Šachová pravidla.....	8
3.	Analýza .....	19
3.1.	Jaký typ aplikace .....	19
3.2.	Rozcestník aplikace .....	20
3.3.	Obsahy stránek .....	22
3.4.	Třídy a detaily aplikace .....	26
4.	Vývojová dokumentace .....	28
4.1.	Složení aplikace.....	28
4.2.	Databázové tabulky .....	30
4.3.	Kernel .....	31
4.4.	Modul board.php .....	32
4.5.	Zápis partie .....	34
4.6.	Modul Move.php .....	35
4.7.	Modul Piece.php.....	36
4.8.	Modul Image.php .....	37
4.9.	Modul gamelist.....	38
4.10.	Modul review .....	38
4.11.	Modul MovingBoard.....	39
4.12.	Modul searchboard .....	39
4.13.	Pages.....	39
4.14.	Složka codes .....	42
4.15.	Modul elo .....	43
5.	Uživatelská dokumentace .....	44

5.1.	Požadavky před instalací: .....	44
5.2.	Instalace .....	44
5.3.	Registrace uživatele.....	45
5.4.	Přidání partie: .....	47
5.5.	Vyhledávání.....	48
5.6.	Přehrávání partie.....	49
5.7.	Změna hesla.....	50
5.8.	Změna údajů .....	51
5.9.	Kopírování partie.....	52
5.10.	Statistiky uživatelů a změna elá .....	52
6.	Závěr .....	53
7.	Použité zdroje: .....	55
8.	Seznam obrázků .....	56

# 1. Úvod

Každý šachista už asi narazil při své přípravě na nedostatek zdrojů partií, ze kterých by si vyvodil, co jeho soupeř hraje, kde má nedostatky a v jaké hře ho lze přehrát. Někteří z hráčů mají i potřebu své partie ukázat svým známým, pochlubit se třeba vydařenou koncovkou.

Existují dva typy uživatelů: a) silnější a b) slabší šachisté. Rozdíl mezi nimi je v tom, co požadují po šachové databázi. V celé této práci je databázi myšlena skupina partií a budeme to považovat za šachový termín.

Skupina uživatelů a) potřebuje databázi, kde si vyhledají postavení na šachovnici, proto budeme potřebovat vyhledávání mezi partiemi. Toto vyhledávání bude muset obsahovat šachovnici pro postavení hledané pozice a budeme požadovat, aby netrvalo příliš dlouho. Dalším požadavkem skupiny a) bude, aby si mohla zadat své partie do databáze. Uživatelé uvítají i pomoc při zadávání partie. Touto pomocí máme na mysli kopírování partie. Kopírováním budeme myslet možnost z jiné partie převzít několik nebo až všechny tahy. Převzít tahy budeme moci pouze jen ze začátku partie, protože jiné pozice se budou těžko kopírovat. Tím dáme uživatelům možnost usnadnit si zadávání partií.

Druhá skupina uživatelů b) bude méně náročná, bude požadovat jen zadávání partií a jejich uložení. Protože uživatelé skupiny b) neprovádějí přípravu na partie, tím je myšleno vyhledávání partií na soupeře, jsou jejich požadavky menší. Snahou práce bude podporovat obě dvě skupiny uživatelů a) i b), protože tím získáme více uživatelů.

Dále budeme podporovat vyhledávání. Zde musíme zajistit, aby postavení hledané pozice nebylo moc zdlouhavé. Tím máme na mysli postavení jednotlivých figur na šachovnici. Postavení pozice může trvat dlouho, například když budeme zadávat všechny tahy, které musíme udělat, než se dostaneme do konkrétní pozice. Možnost jak si postavit pozici rychleji bude třeba taková, že si klikneme na paletu figur a poté na pole kde stojí. Když se k tomu přidá možnost stavět buď na prázdnou šachovnici, anebo ze základní pozice, bude to pro uživatele rychlejší než první možnost - zadávání všech tahů do pozice.

Dále budeme potřebovat vyhledávání podle více kritérií, tím usnadníme skupině a) vyhledávání partií. Co tím myslíme: hlavně vyhledávání postavení na šachovnici a k tomu další údaj jako je třeba příjmení. Tato možnost nám například pomůže vyhledat partie velmistra, který hraje naše zahájení a můžeme se díky tomu poučit jak hrát dané zahájení.

Protože chceme dostat do databáze partie slabších hráčů, bude nutné mít možnost dostat do aplikace zadavatele těchto partií. Budeme tedy potřebovat registraci uživatelů, abychom měli možnost rozšiřovat množství uživatelů a tím i partií. Šachové partie budeme tím mít možnost mazat i upravovat abychom měli možnost spravovat své partie. Proto prozkoumáme existující databáze a budeme je porovnávat.

## 1.1. Podobné databáze na internetu

Po průzkumu mezi šachisty jsme zjistili, že jsou nejčastěji používány tyto tři databáze. Proto si rozebereme jejich výhody a nevýhody.

### Chessbase

Známější databáze šachových partií od společnosti Chessbase [1]. Její hlavní problém spočívá v tom, že neobsahuje mnoho partií a tím je pro uživatele nepoužitelná na případnou přípravu na soupeře. V této databázi svého soupeře s velkou pravděpodobností nenajdete, tedy pokud se nejedná o top hráče. Z pohledu vyhledávání partií je problém v zadávání více kritérií pro vyhledávání. Dalším problémem při vyhledávání je fakt, že musíme v partii zadat všechny tahy vedoucí k dané pozici. Aplikace má sice velké množství stylu figur, z hlediska potřebnosti je tato možnost dobrá, ale ne rozhodující.

### 365Chess

Další známá aplikace je 365Chess [2]. Tato databáze obsahuje údajně nejvíce partií online, z tohoto hlediska je vhodná pro někoho, kdo hledá partie na soupeře. Tato aplikace ale postrádá možnost vyhledávat pozici a zároveň údaje o hráči, takže jedna z podstatných vad v aplikaci, která sice umožňuje zadat název zahájení, ale pochybuji, že hráč, který se teprve připravuje na soupeře, ví přesně, jaké zahájení hledá. Dost často hráči potřebují najít pozici vyskytující se později než v zahájení a v této aplikaci by ji těžko hledali.

## Chess Tempo

Dále jsem analyzoval aplikaci Chess Tempo [3], umožňuje vyhledávat partie s různými údaji i s danou pozicí. Celkové množství partií je celkem velké. Ale nedostatek má například v tom, že zde nenajdeme partie, které nejsou zadané přímo organizátory turnajů a většinou volně šiřitelné. Opět zde není možnost zadávání širokou veřejností. Dalším nedostatkem je zde je nemožnost postavit pozici, například ze střední hry partie nebo z koncovky.

### Srovnání databází

Nyní se budeme zabývat srovnáním databází a tím, co jsme o nich zjistili. Porovnání provedeme pomocí tabulky. Tabulka ilustruje, jak jsou na tom jednotlivé databáze. První z kritérií je množství partií v databázi. Druhým kritériem je obsah partií slabších hráčů, kteří nehrají turnaje. Další kritérium je možnost vyhledávat postavení na šachovnici bez dalších kritérií. Posledním kritériem je možnost zadávat partie uživateli.

Kritéria	ChessBase	365 Chess	Chess Tempo
Množství partií	ne	ano	ano
Partie široké veřejnosti	ne	ne	ne
Vyhledávání partií	ano	ano	ano
Možnost zadávat partie uživateli	ne	ne	ne
Vyhledávání podle více kritérií	ne	ne	ano
Přehrávání partií	ano	ano	ano

Celkově je vidět, že databázím chybí vyhledávání podle více kritérií anebo není dokonalé, jako v případě Chess Tempo. Také mají nedostatek partií slabších hráčů. Šel by využít koncept zadávání tahů ChessBase k zadávání tahů partií.

Dále by se dalo použít třídění seznamu partií podle jednoho sloupce, na který se klikne myší. Pokud nahlédneme do databází hlouběji, uvidíme, že obsahují i třídění podle parametrů. Je vidět i možnost zadávání tahů pouze klikáním na šachovnici. Této vlastnosti bychom měli využít k zadávání tahů do naší databáze při zadávání partií, protože je rychlejší zadávat tahy klikáním myši než zadávat tahy na klávesnici pomocí algebraické šachové notace.

Existující databáze mají hlavní problémy v tom, že nepodporují vyhledávání pozice a zároveň dalších údajů jako je příjmení a to znesnadňuje hledání. Dalším problémem je najít některé hráče vůbec, to by mohl vyřešit systém uživatelů, kteří by své partie do databáze zadávali a tím rozšiřovali počet partií. Proto by bylo dobré, kdyby existoval software řešící oba problémy najednou.

Dále se zaměříme na výběr vhodného databázového systému. Budeme potřebovat takový systém, který bude levný nebo zdarma, protože pro účel nasazení to minimalizuje náklady na provozování. Proto je nejlepší volit databázový systém, který je volně šiřitelný.

## 1.2. Cíl práce

Naše budoucí databáze bude umět:

- ✓ C1) zadávání šachových partií a jejich ukládání do databázové struktury
- ✓ C2) vyhledávání partií podle postavení figur na šachovnici a zároveň dalšího parametru (například příjmení).
- ✓ C3) umožnit kopírování partií – možnost převzít část tahů partie při přehráváním partie.
- ✓ C4) třídění partií sestupně nebo vzestupně podle parametrů.
- ✓ C5) umožnit registraci nových uživatelů.
- ✓ C6) bude obsahovat volně šiřitelnou databázový systém na ukládání dat.

## 2. Šachová pravidla

Je nutné znát i šachová pravidla protože ne každý zná šachy, tahy figur a k tomu slouží tato kapitola. Na konci kapitoly je dodatek týkající se ela. Převzato z pravidel šachu Fide [4].

### Článek 1: Povaha a cíle šachové hry

**1.1.** Šachová hra se hraje mezi dvěma soupeři, kteří střídavě přemísťují kameny na čtvercové desce, jež se nazývá „šachovnice“. Hráč s kameny světlé barvy (Bílý) provede první tah, hráč s kameny tmavé barvy (Černý) provede následující tah a pak hráči provádějí tahy střídavě. Hráč je „na tahu“ poté, když jeho soupeř provedl tah.


### Článek 2: Počáteční postavení kamenů na šachovnici


**2.1** Šachovnice se skládá z 8 x 8, tj. 64 shodných čtvercových polí střídavě světlých („bílá“ pole) a tmavých („černá“ pole).

Šachovnice se položí mezi hráče tak, že nejbližší rohové pole po pravé ruce hráče je bílé.

**2.2.** Na začátku hry má Bílý 16 kamenů světlé barvy („bílé“ kameny) a Černý 16 kamenů tmavé barvy („černé“ kameny).


Tyto kameny jsou:


Bílý král, obvykle označovaný symbolem .....  K


Bílá dáma, obvykle označovaná symbolem .....  D

Dvě bílé věže, obvykle označované symbolem .....  V


Dva bílí střelci, obvykle označovaní symbolem .....  S

Dva bílí jezdci, obvykle označovaní symbolem .....  J

Osm bílých pěšců, obvykle označovaných symbolem ..... 


Černý král, obvykle označovaný symbolem .....  K

Černá dáma, obvykle označovaná symbolem .....  D

Dvě černé věže, obvykle označované symbolem .....  V

Dva černí střelci, obvykle označovaní symbolem .....  S

Dva černí jezdcí, obvykle označovaní symbolem .....  J

Osm černých pěšců, obvykle označovaných symbolem 

**2.3.** Počáteční postavení kamenů na šachovnici je následující:



**2.4.** Pole umístěná svisle pod sebou tvoří osm „sloupců“. Pole umístěná vodorovně vedle sebe tvoří osm „řad“. Přímá linie stejnobarevných polí, která probíhá z jednoho okraje šachovnice k sousednímu okraji, tvoří „diagonálu“.

Základní postavení a definice nám pomohou pochopit jak se figury pohybují.

### Článek 3: Tahy kamenů

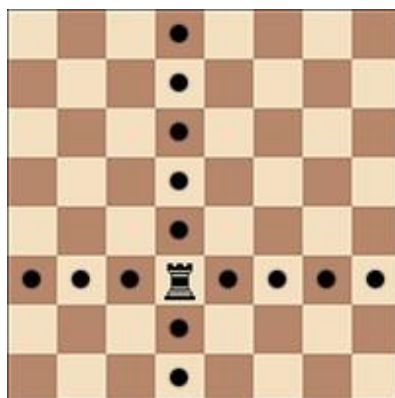
**3.1.** Žádný kámen nelze přemístit na pole obsazené kamenem stejné barvy. Pokud se kámen přemísťuje na pole obsazené soupeřovým kamenem, pak se tento soupeřův kámen bere a odstraní ze šachovnice jako součást tohoto tahu. „Kámen, napadá“ soupeřův kámen, jestliže na tomto poli může provést brání podle článku 3.2 až 3.8.

Kámen napadá pole i tehdy, když na toto pole nemůže táhnout, protože by vystavil svého krále do ohrožení nebo svého krále v ohrožení ponechal.

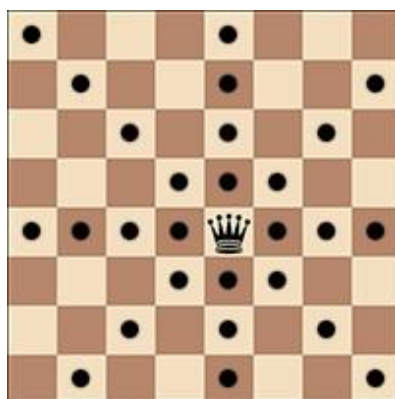
**3.2.** Střelec může táhnout na kterékoli pole po diagonálách, na kterých stojí.



**3.3.** Věž může táhnout na kterékoli pole na sloupci nebo v řadě, na kterých stojí.



**3.4.** Dáma může táhnout na kterékoli pole na sloupci, v řadě nebo diagonálách, na kterých stojí.



**3.5.** Při tazích dámy, věže nebo střelce nelze přeskočit žádný kámen, který stojí v cestě.

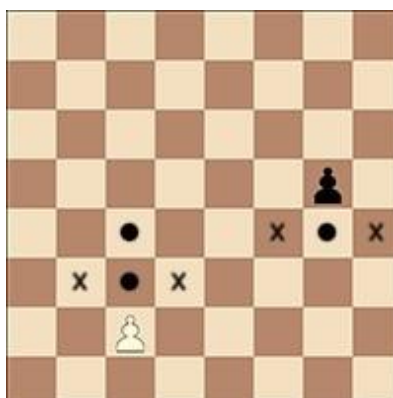
**3.6.** Jezdec může táhnout na jedno z polí nejbližších k poli, na kterém stojí, nikoliv však po sloupci, řadě nebo diagonále.



a) Pěšec může táhnout dopředu na nejbližší neobsazené pole na sloupci nebo

b) Svým prvním tahem může pěšec táhnout jako v 3.7. anebo může postoupit i o dvě pole na stejném sloupci za předpokladu, že obě pole jsou neobsazena, nebo

c) Pěšec může brát soupeřův kámen tak, že provede tah diagonálně vpřed na sousední sloupec na pole, které je obsazeno soupeřovým kamenem.



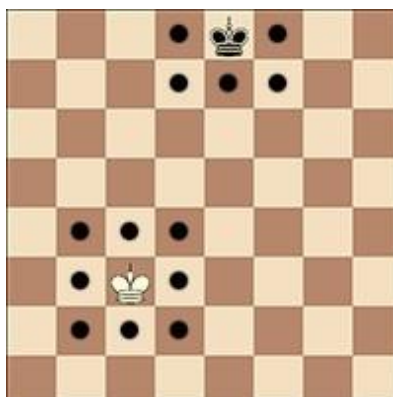
d) Pěšec ohrožující pole, které přešel soupeřův pěšec, jenž ze základního pole postoupil jedním tahem o dvě pole, může tohoto soupeřova pěšce brát, jako by tento pěšec postoupil pouze o jedno pole. Toto brání lze provést pouze jako bezprostřední odpověď a nazývá se „brání mimochodem“ („en passant“).



e) Pokud hráč, který je na tahu, zahraje pěšcem na nejbližší řadu od výchozího pole, musí tohoto pěšce jako součást téhož tahu zaměnit za nový kámen - dámu, věž, střelce nebo jezdce barvy pěšce na dosaženém poli proměny. Volba hráče není omezena kameny, které byly dříve vzaty. Tato záměna pěšce za jiný kámen se nazývá „proměna“ a působnost proměněného kamene je okamžitá.

### 3.8. Král může provést tah dvojím rozdílným způsobem:

a) může táhnout na kterékoli sousední pole



b) může provést „rošádu“. Rošáda je tah krále a jedné z věží téže barvy na první řadě hráče. Rošáda je pokládána za tah krále a provádí se takto: král se přemístí ze svého základního pole o dvě pole směrem k věži stojící na svém základním poli a věž se pak přemístí přes krále na pole, které král právě přešel.



Před malou rošádou bílého.

Před velkou rošádou černého.



Po malé rošádě bílého.

Po velké rošádě černého.



Před velkou rošádou bílého.

Před malou rošádou černého.



Po velké rošádě bílého.

Po malé rošádě černého.

(1) Právo provést rošádu zaniklo:

a) jestliže král již táhl, nebo b) s věží, která již táhla.

(2) Rošádu nelze dočasně provést:

a) jestliže je alespoň jedním soupeřovým kamenem napadeno pole, na kterém král stojí nebo které musí král překročit, případně které má král obsadit;

b) jestliže mezi králem a věží, s níž má být rošáda provedena, stojí jakýkoli kámen.

**3.9.** Král je v „šachu“, jestliže je napaden alespoň jedním soupeřovým kamenem, a to i tehdy, když tento kámen nemůže táhnout na pole obsazené králem, protože by vystavil

šachu vlastního krále nebo krále v šachu ponechal. Hráč nesmí provést tah, kterým vystavuje svého krále do šachu anebo ho v šachu ponechává.

**3.10.** a) Tah je nazýván přípustný, pokud jsou splněny všechny příslušné požadavky článků 3.1 – 3.9.

b) Tah je nazýván nepřípustným, pokud nejsou splněny všechny příslušné požadavky článků 3.1 – 3.9. c) Pozice je nepřípustná, pokud nemohla vzniknout na základě žádné posloupnosti přípustných tahů.

Toto jsou základní tahy figur následované popisem, kdy je král v šachu. Když je král v šachu, nemůže ho v tomto stavu nechat a provádět jiný tah. Toto zjištění se bude hodit pro kontrolu, zda je tah přípustný.

## Článek 8: Zapisování tahů

**8.1.** a) Každý hráč je povinen v průběhu partie zapisovat tah po tahu své tahy i tahy soupeře do partiáře předepsaného v soutěži, algebraickou notací (viz příloha C) co nejjasněji a nejčitelněji. Je zakázáno psát tahy předem, pokud hráč neuplatňuje remízu podle článku 9.2 nebo 9.3 nebo se jedná o přerušení partie dle přílohy E1a.

## Článek 10: Bodování

**10.1.** Pokud rozpis soutěže nestanoví jinak, za výhru své partie nebo za kontumační výhru získá hráč jeden bod (1), za prohru partie nebo kontumační prohru nezíská žádný bod (0), za remízu získá půl bodu (1/2).

Abychom věděli, jakého bodového výsledku lze v partii dosáhnout je zde článek bodování.

### Příloha - Notace šachové partie

FIDE uznává pro své turnaje a zápasy jen jeden systém notace, systém algebraický, a doporučuje používání této jednotné šachové notace i pro šachovou literaturu a časopisy. Zápisů užívajících jiného systému notace než algebraického nelze použít jako důkazu v případech, v nichž se hráčův zápis pro tento účel používá. Rozhodčí, který zjistí, že hráč používá jiný systém než algebraický, musí tohoto hráče na požadavek algebraické notace upozornit.

### Popis algebraického systému

C1. V dalším popisu se „kamenem“ rozumí libovolný kámen kromě pěšce.

C2. Každý kámen je označen zkratkou. V češtině prvním písmenem svého názvu, a to písmenem velkým.

Příklad: K = král, D = dáma, V = věž, S = střelec, J = jezdec.

C3. Každý hráč může zvolit zkratku názvu kamene podle národních zvyklostí.

Příklady: F = fou (francouzsky střelec), L = loper (holandsky střelec).

V tištěných publikacích je pro kameny doporučeno používat grafické symboly.

C4. Pěšci se svým prvním písmenem neoznačují a poznají se tím, že takové písmeno chybí.

Příklady: tahy jsou označeny e5, d4, a5, nesprávně pe5, Pd4, pa5.

C5. Osm sloupců (zleva doprava pro bílého a zprava doleva pro černého) je označeno malými písmeny: a, b, c, d, e, f, g, h. C6. Osm řad (zespodu nahoru ze strany bílého a shora dolů ze strany černého) je očíslováno: 1, 2, 3, 4, 5, 6, 7, 8. V

základním postavení jsou tedy bílé kameny a pěšci umístěni na 1. a 2. řadě, černé kameny a pěšci na 7. a 8. řadě. C7. Každé ze 64 polí je v důsledku předchozích pravidel pevně označeno jednoznačnou kombinací písmena a číslice:

a8	b8	c8	d8	e8	f8	g8	h8
a7	b7	c7	d7	e7	f7	g7	h7
a6	b6	c6	d6	e6	f6	g6	h6
a5	b5	c5	d5	e5	f5	g5	h5
a4	b4	c4	d4	e4	f4	g4	h4
a3	b3	c3	d3	e3	f3	g3	h3
a2	b2	c2	d2	e2	f2	g2	h2
a1	b1	c1	d1	e1	f1	g1	h1

C8. Každý tah kamene je označen:

a) zkratkou názvu příslušného kamene a b) názvem cílového pole, na které kámen táhne.

Mezi a) a b) není pomlčka. Příklady: Se5, Jf3, Vd1. U tahů pěšcem se označují pouze cílová pole. Příklady: e5, d4, a5.

C9. Pokud kámen provádí braní, vkládá se mezi a) (zkratka názvu příslušného kamene) a b) (cílové pole) symbol x.

Příklady: Sxe5, Jxf3, Vxd1 (viz článek C10).

Pokud braní provádí pěšec, označuje se nejen cílové pole, ale i sloupec, z něhož pěšec táhne, následovaný symbolem x. Příklady: dxe5, gxf3, axb5. V případě „braní mimochodem“ („en passant“) se jako cílové pole označí to pole, na kterém pěšec nakonec zůstane a k notaci se přidá „e.p.“. Příklad: exd6 e.p.

C10. Pokud mohou dva stejné kameny táhnout na stejné pole, označí se kamen, jímž se táhlo, následovně:

1. Jsou-li oba kameny na stejné řadě:

- a) zkratkou názvu kamene;
- b) názvem sloupce, z něhož kámen táhne;
- c) názvem cílového pole.

2. Jsou-li oba kameny na stejném sloupci:

- a) zkratkou názvu kamene;
- b) názvem řady, z níž kámen táhne;
- c) názvem cílového pole.

Jsou-li kameny na různých řadách a sloupcích, dává se přednost zápisu podle první metody. Při braní se mezi b) a c)

vkládá symbol x.

Příklady:

a) Na polích g1 a d2 stojí dva jezdci, z nichž jedním je proveden tah na pole f3: podle situace se zapíše Jgf3 nebo Jdf3. b) Na polích g5 a g1 stojí dva jezdci, z nichž jedním je proveden tah na pole f3: podle situace se zapíše J5f3 nebo J1f3. c) Na polích h2 a d4 stojí dva jezdci, z nichž jedním je proveden tah na pole f3: podle situace se zapíše Jhf3 nebo Jdf3.

d) Dojde-li na poli f3 k braní, změní se předchozí příklady vložení symbolu x. Podle situace se zapíše:

1. Jgxf3 nebo Jdxf3;

2. J5xf3 nebo J1xf3;

3. Jhxf3 nebo Jdxf3.

C11. V případě proměny pěšce se запиše tah pěšcem, za nímž bezprostředně následuje zkratka nového kamene. Příklady:

d8D, exf8J, b1S, g1V.

C12. Nabídka remízy se označuje symbolem (=).

C13. Zkratky:

0-0 = rošáda s věží h1 nebo h8 (malá rošáda)

0-0-0 = rošáda s věží a1 nebo a8 (velká rošáda)

x = brání

+ = šach

++ nebo # = mat

e.p. = brání mimochodem (en passant) Poslední čtyři zkratky jsou nepovinné.

Vzorový zápis partie: 1.e4 e5 2. Jf3 Jf6 3. d4 exd4 4. e5 Je4 5. Dxd4 d5 6. exd6 e.p. Jxd6 7. Sg5 Jc6 8.De3+ Se7 9. Jbd2 0-0 10. 0-0-0 Ve8 11. Kb1 (=)

Nebo: 1. e4 e5 2. Jf3 Jf6 3. d4 ed4 4. e5 Je4 5. Dd4 d5 6. ed6 Jd6 7. Sg5 Jc6 8. De3 Se7 9. Jbd2 0-0 10. 0-0-0 Ve8 11. Kb1 (=)

Protože je nutné znát šachovou notaci k zaznamenání tahů. Budeme používat algebraickou notaci k tahům.

Dále existuje elo, což je hodnocení šachistů. Každý rád uvitá možnost spočítat si změnu svého současného ela. K tomu slouží tento vzorec:

$En = k * (Vysl - P)$ , kde  $k$  je koeficient,  $P$  je hodnota z tabulky a  $Vysl$  je výsledek partie. Kde  $P$  zjistíme tak, že  $H$  značí předpokládaný výsledek hráče s vyšším elem, hodnota  $L$  značí předpokládaný výsledek hráče s menším elem. Hodnoty  $H$  a  $L$  můžeme vyčíst z Tabulka 1 procentuálního výsledku hráče s větším a menším elem [5].

Rozdíl	H	L	Rozdíl	H	L	Rozdíl	H	L
0 - 3	50	50	122 - 129	67	33	279 - 290	84	16
4 - 10	51	49	130 - 137	68	32	291 - 302	85	15
11 - 17	52	48	138 - 145	69	31	303 - 315	86	14
18 - 25	53	47	146 - 153	70	30	316 - 328	87	13
26 - 32	54	46	154 - 162	71	29	329 - 344	88	12
33 - 39	55	45	163 - 170	72	28	345 - 357	89	11
40 - 46	56	44	171 - 179	73	27	358 - 374	90	10
47 - 53	57	43	180 - 188	74	26	375 - 391	91	9
54 - 61	58	42	189 - 197	75	25	392 - 411	92	8
62 - 68	59	41	198 - 206	76	24	412 - 432	93	7
69 - 76	60	40	207 - 215	77	23	433 - 456	94	6
77 - 83	61	39	216 - 225	78	22	457 - 484	95	5
84 - 91	62	38	226 - 235	79	21	485 - 517	96	4
92 - 98	63	37	236 - 245	80	20	518 - 559	97	3
99 - 106	64	36	246 - 256	81	19	560 - 619	98	2
107 - 113	65	35	257 - 267	82	18	620 - 735	99	1
114 - 121	66	34	268 - 278	83	17	>735	100	0

Tabulka 1 procentuálního výsledku hráče s větším a menším elem

## 3. Analýza

V této kapitole se budeme zabývat analýzou problému a rozebereme si, jak bychom měli danou aplikaci vytvořit. Probereme alternativy vytvoření aplikace. Nejprve se rozhodneme, zda mít webovou či desktopovou aplikaci.

### 3.1. Jaký typ aplikace

V této podkapitole si rozebereme, zda máme mít webovou či desktopovou aplikaci a následně si vybereme programovací jazyk a databázi jakou budeme používat. Výběr určitě nebude složitý, máme tu spoustu zařízení, na kterých bychom chtěli naši aplikaci používat, jsou to tablety, stolní počítače, chytré telefony a notebooky. U stolních počítačů a notebooků se nám vyskytují různé operační systémy jako je Windows, Linux a Mac OS. Protože uživatelé naší aplikace patří do některé z těchto kategorií, chtěli bychom podporovat všechny tyto kategorie. To by v případě desktop aplikace znamenalo naprogramovat pro každou alternativu software, což by nebylo určitě jednoduché, ale existuje jiná alternativa a to taková: všechna tato zařízení mají jedno společné a to je internetový prohlížeč. Takže když se rozhodneme pro webovou aplikaci tak díky tomu obsáhneme všechna zařízení a budeme tak podporovat všechna tato zařízení, která jsou vyjmenovaná výše, proto jsem zvolil webovou aplikaci.

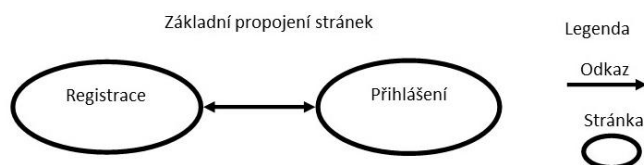
Jako další se musíme rozhodnout, jakou technologii použijeme jako programovací jazyk. Nabízí se (a) PHP [6], (b) ASP [7] nebo (c) ASP.NET [8]. Pro výběr programovacího jazyka použijeme web <http://www.hostingy.cz/> [9] a budeme srovnávat četnost užití a cenu za webhosting. Porovnávali jsme pouze placené webhostingy, protože jestli chceme zajistit kvalitní služby, tak budeme muset použít placený webhosting. Když budeme srovnávat četnost užití, tak nejvíce používané je PHP s četností 124 použití z 303, následují ASP a ASP.NET s četností 19 z celkového 303. S cenou je to podobné, nejlevnější varianta je s použitím PHP, kdy se jedná o cenu kolem 10 Kč / měsíc následuje opět ASP a ASP.NET a cenou kolem 49 Kč / měsíc. Takže nejvýhodnější bude použít PHP, protože je jak nejrozšířenější tak i nejlevnější alternativou na webhostincích. Z těchto důvodů volím možnost za a).

Jako další musíme vyřešit, jakou použijeme databázovou technologii. Podle bodu C6 chceme volně šiřitelnou databázi. Jako nejvhodnější se nabízejí databáze jako jsou (a) MySQL [10], (b) PostgreSQL [11], (c) SQLite [12]. Pro výběr se použilo kritérium nejvíce používané databáze na českých webhostinzích. Na vyhledávání mezi webhostingy se použila stránka <http://www.hostingy.cz/>, kterou jsme použili na porovnání četnosti databází na webhostinzích. Když srovnáme tyto webhostingy, tak z nich vychází nejlépe MySQL. Jejich četnost je 123 z celkového počtu 229. Druhé je PostgreSQL s 42 z 229 a třetí je SQLite s 32 z 229. Následně když začneme porovnávat podle ceny webhostingu PHP + MySQL, tak vychází jak nejlevnější varianta právě tato kombinace s cenou 10 Kč/měsíc, jako druhá je podle ceny varianta PHP + SQLite s cenou 20 Kč/měsíc a poslední skončila kombinace PHP + PostgreSQL. Takže jako nejrozumnější z hlediska ceny webhostingu a použití se jeví možnost a).

### 3.2. Rozcestník aplikace

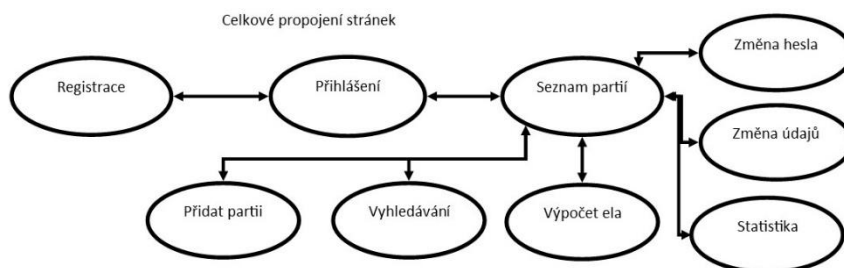
V této podkapitole si rozebereme, z čeho se má naše aplikace skládat a popíšeme všechny ovládací prvky, funkce a třídy aplikace. Nejprve začneme registrací, protože uživatel se musí nejprve zaregistrovat a poté následně může dál pokračovat v práci s naší aplikací. To je jasné i z cíle C5.

Nejprve se musíme dostat k registraci, to provedeme tak, že klikneme na odkaz registrace na přihlašovací stránce. Po ukončení registrace se dostaneme zpět tak, že klikneme na odkaz přihlášení. Základní propojení stránek registrace a přihlášení je vidět na Obrázek 1 základní propojení stránek



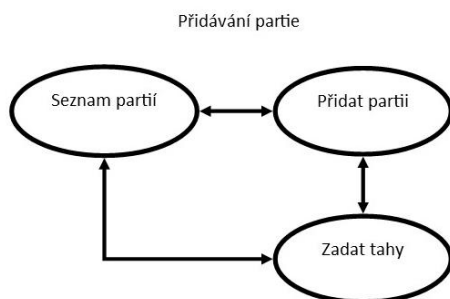
*Obrázek 1 základní propojení stránek*

Po úspěšné registraci a přihlášení uživatele do aplikace, se nám zobrazí stránka se seznamem partií. Stránka se seznamem partií slouží jako rozcestník, z něhož se dá dostat na další stránky jako je výpočet ela, zadání partie, vyhledávání, změna hesla, změna údajů a statistika. Jak to vypadá je vidět na Obrázek 2 celkové propojení stránek.



Obrázek 2 celkové propojení stránek

Pro přidání partie musíme projít více stránek, začínáme v seznamu partií, kde klikneme na zadat partií, poté se dostaneme na stránku přidat partií, následně po vyplnění údajů a kliknutí na uložit, se dostaneme na stránku zadat tahy. Celé přidávání tahů vypadá jako na Obrázek 3 přidávání partie.



Obrázek 3 přidávání partie

Další funkcí, kterou bude muset aplikace umět, je vyhledávání partií. K této funkci se dostaneme tak, že v seznamu partií klikneme na hledat, zobrazí se nám stránka s vyhledávacím formulářem a šachovnicí. Zde budeme moci definovat svůj dotaz v podobě pozice, kterou hledáme a dalšího kritéria které hledáme. Pro vyhledání postavíme nejprve pozici, kterou hledáme a to takto: klikáme postupně na figury, které chceme postavit v paletě figur a poté klikneme na šachovnici, kam chceme figuru postavit. Tím postavíme pozici, kterou hledáme. Jako další musíme definovat, co ve formuláři chceme hledat. To uděláme takto: chceme hledat partie

Ananda, které hrál bílými. Tak do inputu, kde je napsáno příjmení bílý, napíšeme Anand a klikneme na hledat. Následně se takto dostaneme do seznamu partií, které hrál Anand bílými figurami.

Jako další si můžeme nechat spočítat změnu elo po odehrané partii. K tomu se dostaneme takto: klikneme na výpočet elo , dostaneme se na stránku, kde si elo můžeme spočítat. Jak je i vidět na Obrázek 2 celkové propojení stránek, půjde se i odtud dostat nazpět na seznam partií.

Další stránka, která se bude v aplikaci vyskytovat, je stránka statistik. Na ni se dostaneme ze seznamu partií a to tak, že klikneme na odkaz statistika. Dostaneme se tak na stránku se statistikou o uživateli.

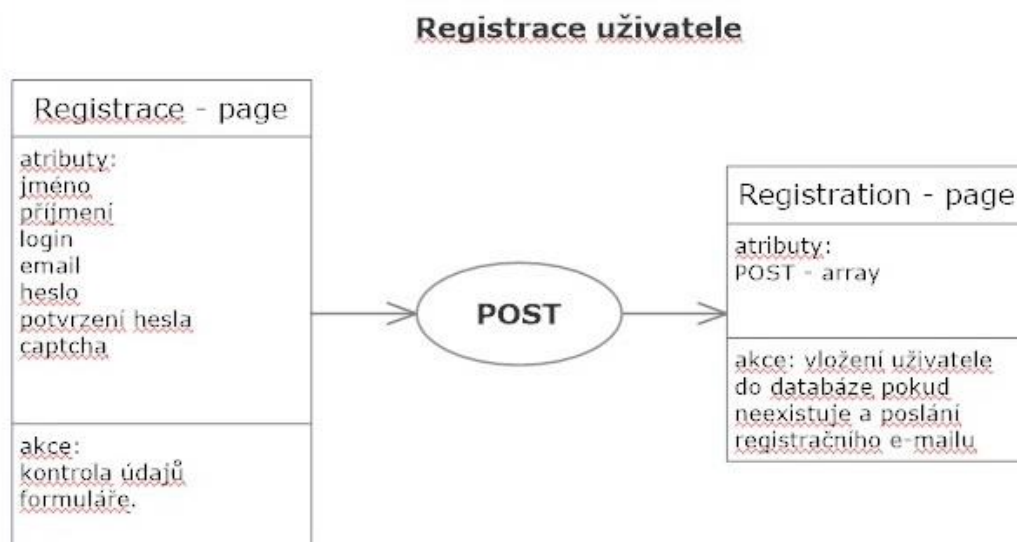
Pokud budeme chtít změnit nějaké údaje, tak jsou tu dvě stránky změna údajů a změna hesla. Na první jmenovanou se dostaneme tak, že klikneme na stránce seznam partií na odkaz změna údajů - tak se dostaneme na stránku, kde budeme moci měnit informace o sobě. Na druhou jmenovanou stránku se dostaneme po kliknutí na odkaz v seznamu partií změna hesla. Z obou stránek se budeme moci dostat nazpět na seznam partií pomocí odkazu seznam partií.

Poslední dvě věci, které budeme moci provádět v seznamu partií je, že si budeme moci nechat vypsát seznam našich partií a to po kliknutí na odkaz moje partie. Po tomto kliku se nám zobrazí seznam našich partií. Jako další si budeme moci vypsát všechny partie a to po kliknutí na vše z partií. Po kliknutí na tento odkaz se nám zobrazí všechny partie z databáze.

### 3.3. Obsahy stránek

Registrace bude obsahovat formulář s těmito údaji: jméno, příjmení, login, email, heslo, potvrzení hesla a obrázek s řetězcem malých písmen, který se bude muset opsat (captcha). Bude podobná jako je popsáno na Obrázek 4 registrace uživatele. Samozřejmě zde bude odesílací tlačítko formuláře s popiskem registrovat se. Pro registraci uživatele do aplikace budeme čekat vyplnění všech registračních údajů a zmačknutí tlačítka registrovat se. Po stisku tlačítka se ověří, že jsou opravdu údaje vyplněny a uživatelské jméno není použito u jiného uživatele. V případě neúspěchu se zobrazí chybové hlášky u jednotlivých údajů a po opravě se budeme moci pokusit znovu registrovat. Po úspěšném vyplnění všech údajů a kontrole údajů

se odešle registrační e-mail a přidáme uživatele do databázové tabulky a zobrazí se nám zpráva o tom, že jsme byli zaregistrováni. Registrační e-mail bude obsahovat: jméno, příjmení, login a heslo. Abychom se dostali na přihlašovací stránku, klikneme na odkaz přihlášení.



Obrázek 4 registrace uživatele

Přihlášení do aplikace se bude provádět přes přihlašovací stránku a ta bude obsahovat formulář s uživatelským jménem, heslem a odesílacím tlačítkem formuláře s popiskem přihlásit se. Po úspěšné registraci se můžeme pomocí údajů, které jsme použili při registraci, přihlásit do aplikace. Po úspěšném přihlášení se dostaneme na stránku se seznamem partií.

V seznamu partií se nám nejprve zobrazí menu s tlačítkem odhlásit se pro případ, že jsme skončili s prací v databázi. Další položky menu budou zadat partii, hledat, výpočet ela, moje partie, vše z partií, statistika, změna údajů a změna hesla. Pod tímto menu bude následovat tabulka se seznamem partií. Záhloví tabulky bude příjmení bílý, jméno bílý, elo bílý, vs, příjmení černý, jméno černý, elo černý, výsledek, místo, turnaj datum, přehrát, upravit smazat. Poslední dvě položky tabulky budou vyplněny jen v případě, že máme právo na úpravu dané partie. V záhlaví u položek příjmení bílý, jméno bílý, elo bílý, příjmení černý, jméno černý, elo černý, výsledek, místo, turnaj a datum budou vždy dvě šipky. Po té, co se na jednu z šipek klikne, tak se partie seřadí podle tohoto sloupce. Nakonec pod seznamem partií bude stránkování partií.

Po kliknutí na zadat partii, se nám zobrazí formulář, kde budeme moci vyplnit údaje jako je příjmení bílý, jméno bílý, elo bílý, příjmení černý, jméno černý, elo černý, výsledek, místo, turnaj datum, právo číst a právo zapisovat. Tímto splníme bod C1. Dále bude obsahovat menu s tlačítkem na seznam partií. Po vyplnění formuláře a kliknutí na odesílací tlačítko uložit se nám zobrazí šachovnice, kde pomocí kliknutí na figuru a následném kliknutí na místo, kam má figura jít, zadáváme tahy. Pomocí šipek umožníme vrácení tahu na šachovnici a tahy k partii uložíme pomocí kliknutí tlačítko s obrázkem diskety. Opět přidáme tlačítko na vrácení se k seznamu partií.

Další místo, kam se dostaneme po kliknutí na hledat v seznamu partií, je vyhledávací formulář. V tomto formuláři nalezneme šachovnici, kde si můžeme postavit jakoukoliv pozici pomocí kliknutí na paletu figur a na místo na šachovnici. Takto si postavíme pozici, kterou hledáme a plníme bod C2. Dále budeme moci ve formuláři vyplnit údaj, podle kterého ještě budeme vyhledávat. Budou to údaje příjmení bílý, jméno bílý, elo bílý od, elo bílý do, příjmení černý, jméno černý, elo černý od, elo černý do, místo a turnaj. Po kliknutí na hledat, se nám zobrazí seznam partií odpovídající tomuto dotazu. Opět zde bude i odkaz zpět na seznam partií.

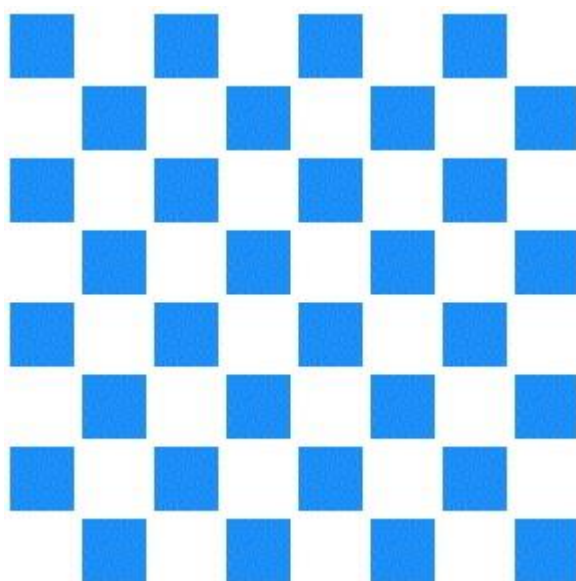
Další položkou, kterou nalezneme v menu, je výpočet ela. Po kliknutí na tento odkaz se nám zobrazí formulář na výpočet ela a bude obsahovat naše elo, soupeřovo elo, koeficient rozvoje hráče a výsledek. Koeficient rozvoje je celé číslo určující

velikost změny ela. Na odesílacím tlačítku bude nápis spočti. Bude tu i tlačítko s odkazem na seznam partií.

Na stránce se seznamem partií bude možnost tento seznam měnit tím, že si klikneme na sloupec a partie se podle tohoto sloupce seřadí. Umožníme obě možnosti řazení vzestupně či sestupně. Splníme tím i bod C4. Půjde i měnit tak, že klikneme na položku moje partie a vše z partií v menu. V tomto seznamu budou údaje, jako jsou jména, příjmení a ela soupeřů. Poté bude následovat výsledek partie, místo, kde se partie hrála, v rámci jakého turnaje se partie sehrála, datum sehraní partie. V seznamu ještě přidáme položky přehrát, smazat a upravit. Dalším způsobem jak půjde tento seznam měnit je, že budeme mezi partii vyhledávat.

.Nejprve necháme zadat uživatele data o partii, a poté necháme zadat partii tah po tahu. Tento postup bude podobný při kopírování partie, ale bude zkrácen o n tahů, které budeme kopírovat. Kopírování bude u položky přehrát partii, protože zde budeme vědět, že jsou zadané tahy korektní. Tímto splníme úkol C3.

Jako další budeme řešit, jak v jaké podobě předávat šachovnici s rozestavením na šachovnici nabízí se možnost posílat rovnou obrázek, který sestaví pomocí PHP funkcí a nechat ho poté měnit JavaScriptem. Obrázek budeme posílat jako bytestream, protože když ho budeme na webu ukládat, tak nám vznikne režie v podobě ukládání obrázku a jeho následné mazání ze serveru. Proto se vydáme cestou posílání obrázku jako bytestream. Jako další možnost se nabízí možnost posílat textový řetězec ze serveru a ten následně pomocí JavaScriptu a CSS [13] zobrazovat. Problém je v tom, že bychom nejprve museli vyřešit problém s tím, že je problematické pracovat v JavaScriptu s řetězci. Poté bychom museli nějak rozumně vyřešit i vykreslování jednotlivých figur pomocí CSS - to by šlo pomocí JavaScriptu a to změnou elementu HTML [14], ale navíc je problém s tím, že vykreslování není tak elegantní, protože kdy máme 64 obrázků pro vykreslení šachovnice, tak se nám nezobrazují vedle sebe a to je z estetického hlediska divné. Jak to vypadá při tomto způsobu je vidět na Obrázek 5 Zobrazení 64 obrázků. Z obrázků je patrné, že jsou mezi políčky mezery a to vypadá neesteticky a navíc tento způsob jsme na žádném serveru z kapitoly 1 nenašli. Jako nejlepší se nám zda možnost posílání obrázku jako bytestream a rovnou ze serveru.



Obrázek 5 Zobrazení 64 obrázků

Na zadávání tahů na šachovnici použijeme kliknutí na figuru a poté kliknutí na pole, kam se má pohybovat. Samozřejmě nepovolíme zadat nemožný tah. Proměnu pěšce budeme řešit při zadávání tak, že necháme vybrat uživatele z palety možných figur k proměně.

Jako další umožníme uživateli měnit své registrační údaje. Heslo se bude měnit samostatně a ostatní údaje jak jméno, příjmení, login a email se budou měnit zvlášť. Budeme také evidovat statistiky uživatelů a elo kalkulator. Tyto údaje vložíme na úvodní obrazovku k seznamu partií. Vytvoříme zde klikací menu, které obohatíme o odkaz na vyhledávání, zadávání partie a odhlášení z aplikace.

Vyhledávání bude mít šachovnici pro postavení hledané pozice a ještě bude obsahovat další údaje, které po vyplnění budou vyhledávány spolu s tím, co máme na šachovnici. Tyto údaje budou jména, příjmení a ela obou hráčů, místo a turnaj.

### 3.4. Třídy a detaily aplikace

Postup, jak budeme postupovat při zadávání tahu k partii, bude následující: klikneme myší na figuru a poté klikneme myší na místo, kam figura půjde. Tento proces bude nejlepší zjišťovat pomocí JavaScriptu. Šachovnici budeme vykreslovat jako obrázek, pro vytvoření obrázku budeme muset použít PHP funkci. Umožníme uživateli této databáze změnit si uživatelské údaje a heslo, které použil k registraci. Registraci a změny údajů zprostředkujeme pomocí formulářů. Vstupy z formulářů

budeme testovat pomocí PHP. Definici jednotlivých tříd provedeme také v PHP. Budeme potřebovat třídy jako je *Figura*, šachovnice a třídy stránek, které budeme zobrazovat. Budeme muset měnit i obsah - například při zadávání tahů. Změny obrázků po provedení tahu nebo po zadání tahu bude nejlepší provádět změnou HTML tagu pomocí JavaScriptu. Stejný postup použijeme i pro změnu HTML Tagu i v dalších případech, jako je změna seznamu tahu, které jsme zadali a podobně.

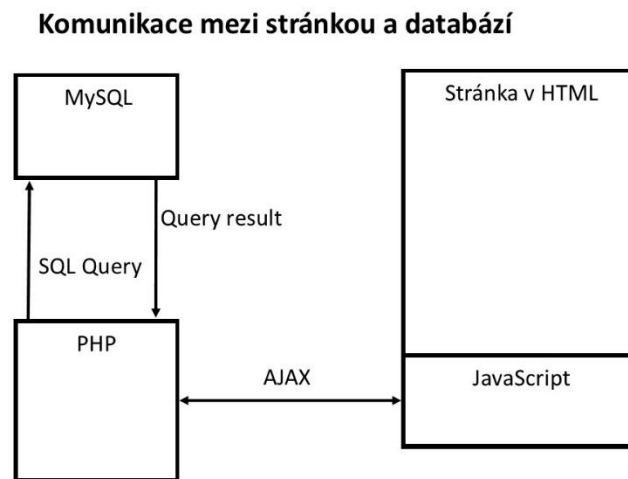
Jednotlivé stránky aplikace budeme mít jako potomky třídy *Page*, na kód stránek v HTML si vyhradíme zvláštní složku *codes*. Potomky třídy *pages* budeme ukládat do zvláštní složky *pages*. Třídy používané v potomcích třídy *page*, budeme vkládat do zvláštní složky *modules*. Budou zde třeba třídy pro vykreslování obrázků, potomci *Image*. Budou to i třídy jako jsou jednotlivé figury. Do složky *modules* bude patřit i třída *Board*, která bude představovat šachovnici. Dále zde najdeme i třídy představující jednotlivé tahy, tj. potomky třídy *Move*. Do další složky, jménem *style*, vložíme věci týkající se stylu. Do složky *image* vložíme obrázky figur a dalších potřebných obrázků. Externí knihovny vložíme do složky *library*. Když se bude něco týkat připojení do MySQL databáze, uložíme to do složky *kernel*. Budou to zejména dotazy do databáze a i další prostředky jako jsou *Managery*. *JavaScript* kód a k němu příslušné funkce. Aby aplikace měla podporu jazyků, vytvoříme i složku *text*, kde se budou vyskytovat texty aplikace.

Budeme zde mít stránky jako je registrace, ta se bude starat o registraci uživatelů. Protože budeme chtít umožnit uživatelům měnit heslo a uživatelské údaje, budeme zde potřebovat další dvě stránky na změnu uživatelských údajů a změnu hesla. Dále zde bude jedna stránka jako rozcestník a to bude stránka se seznamem partií. Zobrazení rozcestníkové stránky bude závislé na tom, jestli jsme přihlášení, takže budeme potřebovat stránku na přihlašování uživatelů. Z rozcestníků se ještě kromě změny údajů budeme moci dostat ke statistikám, výpočtu ela, vyhledávání a zadávání partie. Dále budeme řešit stránky, na které se dostaneme z rozcestníků po kliknutí na: přehrát partii, smazat partii, nebo upravit partii. K tomu nám zůstanou některé stránky skryté.

## 4. Vývojová dokumentace

### 4.1. Složení aplikace

Komunikace aplikace funguje v principu jak je vidět na Obrázek 6 . Komunikaci mezi databází MySQL a PHP vždy zajišťuje SQL dotaz. Výsledek dotazu dostaneme jako Query result. Data na HTML stránku se dostanou pomocí Ajaxu [15], který nám změní data na stránce. Aby se mohly data změnit, musí se nejprve provést nějaká akce v JavaScriptu a jako reakce ni se změní část stránky.



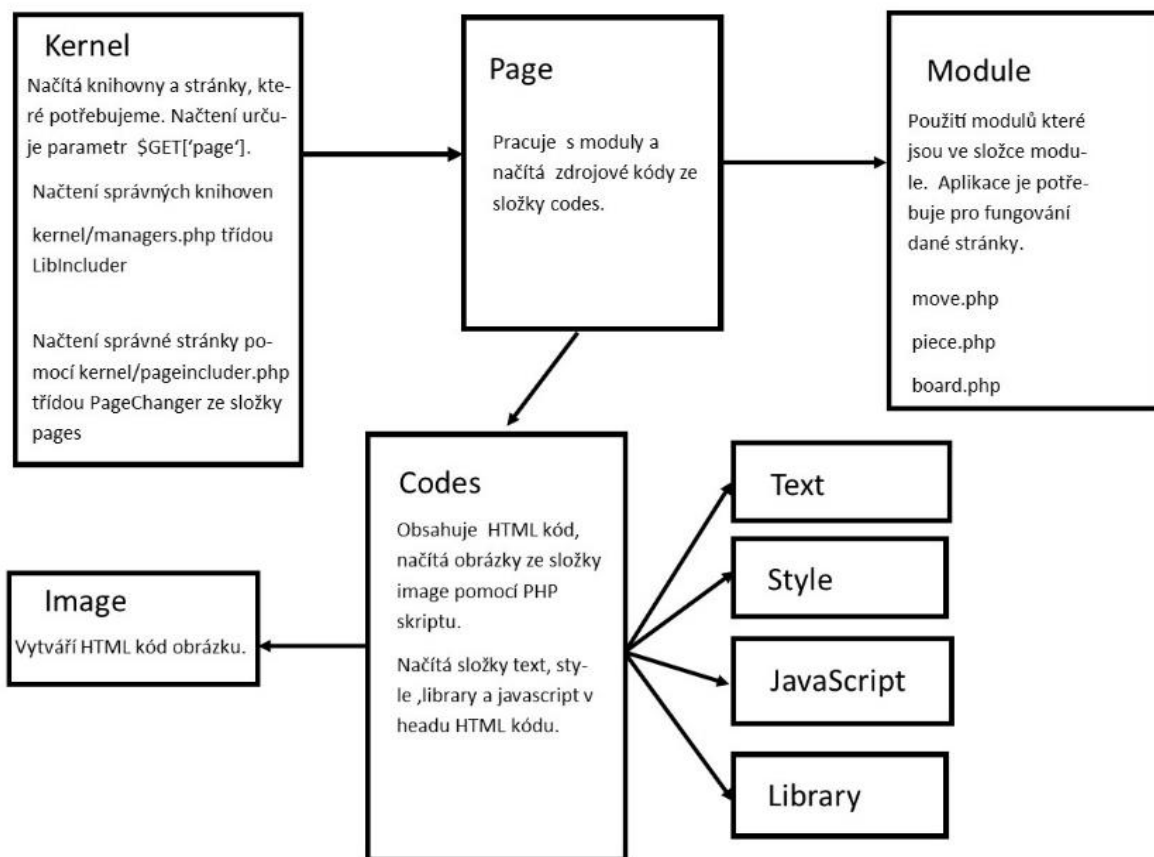
Obrázek 6 komunikace mezi stránkou a databází

Aplikace samotná se skládá ze složek

- *module* obsahuje definice PHP tříd
- *kernel* zajišťuje správné includování stránek a jejich knihoven + SQL dotazy
- *javascript* obsahuje JavaScript cele aplikace
- *codes* obsahuje zdrojové kódy HTML a PHP skripty
- *library* obsahuje externí knihovny
- *style* obsahuje CSS styly aplikace
- *pages* obsahuje PHP definice třídy page
- *image* obsahuje obrázky aplikace

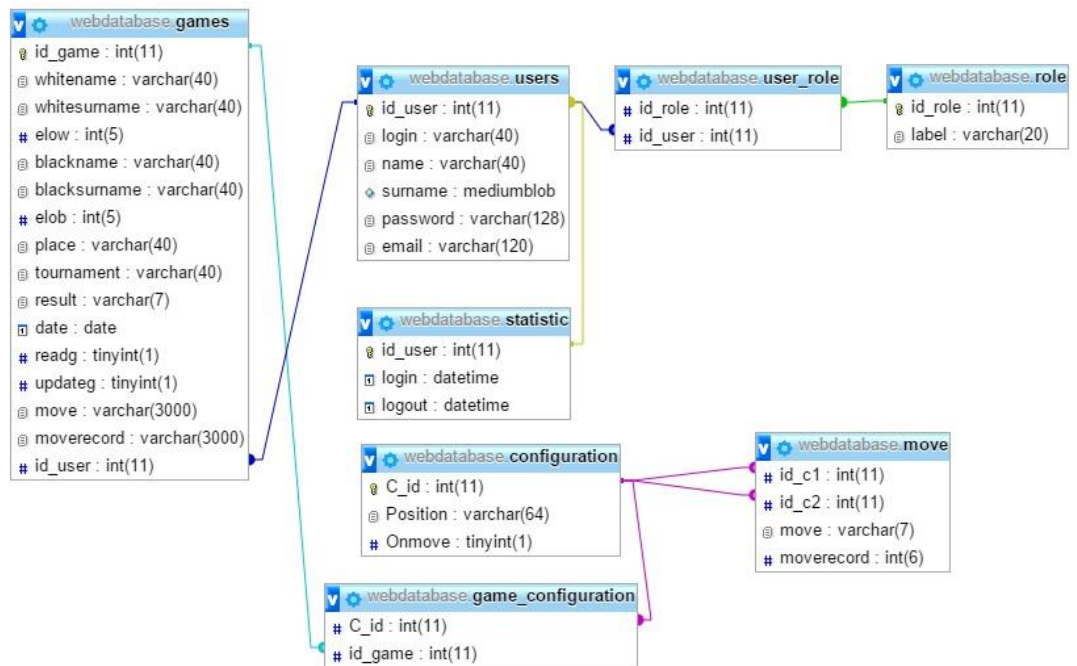
- *text* obsahuje texty aplikace

Jak se používají jednotlivé složky při načítání stránky je vidět na Obrázek 7 načítání stránek.



Obrázek 7 načítání stránek

Další součástí aplikace je databáze MySQL. Její schéma je vidět na Obrázek 8 MySQL databázové schéma. Detailní popis v sekci 4.2. Databázové tabulky.



Obrázek 8 MySQL databázové schéma

Poslední částí jsou obrázky ve složce *image* a *texty* ve složce *text*. Hlavním obsah složky *style* je stylování stránek v CSS 3. Složka *javascript* obsahuje JavaScript starající se o překreslování obrázku šachovnice a textu zápisů partie. Hlavním důvodem použití Ajax je, že se nemusí znovu načítat stránka z důvodu změny obrázku či textu. Tím snížíme zátěž na server - jen obdržíme nový obrázek a ne celou stránku. Složka *Image* obsahuje složku *board*. V této složce jsou obrázky šipek, obrázku diskety a dalších obrázků s ovládacími prvky. Složka *board* také obsahuje obrázky figur. Obsahem složky *codes* jsou zdrojové kódy stránek aplikace v HTML doplněny o PHP. Další složky obsahují zdrojové texty v PHP, tj. složky *kernel module* a *pages*. V těchto složkách jsou definice tříd a funkcí.

#### 4.2. Databázové tabulky

- Její schéma je vidět na Obrázek 8 MySQL databázové schéma
- První tabulka o obsahu aplikace je tabulka *users*, ve které je id uživatele, login, jméno, příjmení heslo a email.
- Další tabulka je *role* - obsahuje id role a jméno role.

- K propojení prvních dvou tabulek je potřeba mít tabulku *user\_role* obsahující id role a uživatele.
- Další tabulkou je *games*, obsahující údaje o partii jako je jméno bílého, jméno černého, příjmení černého, příjmení bílého, elo bílého, elo černého, turnaj, místo, datum, příznak pro upravování partie, příznak pro změnu partie, tahy partie, zápis partie a id uživatele který partii zadal.
- Tabulka pro ukládání pozic z partií se jmenuje *configuration*. Obsahuje jednobitový údaj, vyjadřující kdo byl v této pozici na tahu, id pozice a samotné postavení.
- Protože potřebujeme vědět, v jaké partii se vyskytuje daná konfigurace, máme zde tabulku *game\_configuration*, která obsahuje id partie a id konfigurace. Tím propojuje partii s danou konfigurací.
- Další tabulkou je tabulka *move* obsahující id konfigurace odkud je tah, následně obsahuje i id konfigurace kam se po provedení tahu dostaneme. Součástí údajů je i zápis tahu a jeho interpretace v číselném kódu.
- Poslední tabulka je tabulka *statictic* evidující kdy se uživatel přihlásil a odhlásil.

### 4.3. Kernel

Začneme u složky kernel - obsahuje soubor *debug.php*, jak již název napovídá, jedná se o soubor s třídou *Debugger*. Hlavní úkol této třídy je vypisovat ladící výpisy pro programátora a zajistit, aby je neviděl uživatel. Toho dosáhneme tak, že místo textu „on“ při konstrukci *Debuggeru* změním na cokoliv jiného, nejlogičtější bude změna na „off“. Dalším souborem je *inputcheck.php*. Tento soubor obsahuje třídu *Input* a její potomky. Účelem této třídy a jejích potomků je kontrolovat vstupy při kontrole formulářů. Například třída *InputLogin* používá Sql dotazy k zjištění, zda je Login uživatele používán. Soubor *manager.php* je nejobsáhlejší. Obsahuje třídu *Registrar* - účelem této třídy je registrování uživatelů do systému – uložení do databáze. Nachází se zde i třída *Checker*. Účelem této třídy je kontrola údajů. Používá k tomu například i funkce jiných tříd. Třída *Game* je zaměřena na změnu dat o partii. Tato třída umí uložit a upravit partii. Třída *Mail* obsahuje podrobnosti potřebné k poslání e-mailu. Pro změnu údajů o uživateli se používá třída *UserDataChanger*. Obdobná třída na změnu hesla je zde též a je to

třída *UserParwordChanger*. Pro práci s časem se používá třída *DateAndTimeManager*. *MailManager* posílá e-maily a používá údaje o e-mailu z třídy *Mail*. *MysqLiManager* se připojuje a odpojuje od databáze. Podobně se používá i *SessionManager* pro session. Pro nastavení jazyka je používán *LanguageManager* - nastaví aktuální jazyk. Třída starající se o přihlašování a odhlašování uživatelů je *LoginManager*. Tato třída nám i může sdělovat další podrobnosti o uživateli- jako je jeho id a zda je někdo přihlášen.

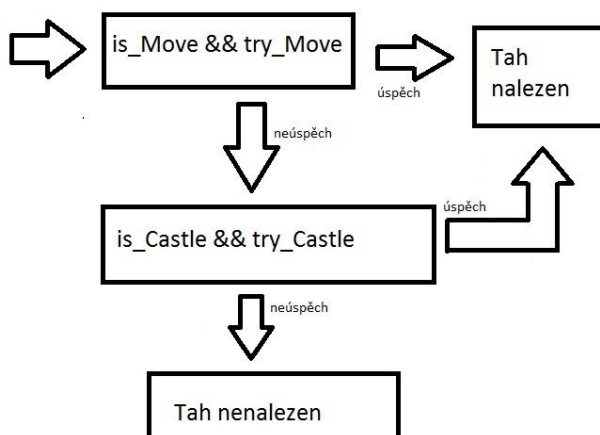
Soubor dále pokračuje třídou *LibIncluder* a zajišťuje nahrání správných knihoven pro aplikaci. Třída *ALLInputCheck* se stará o vytváření chybových hlášek pro uživatele v HTML a kontrolu všech parametrů formuláře. Podobná třída je *StatisticManager*, která se stará se o vykreslování statistik uživatelů. Tyto údaje zjišťuje z databáze MySQL. Poslední třídou v souboru *managers.php* je *DatabaseInserter* jejímž hlavním úkolem je zajistit úpravu textu ukládaného do databáze MySQL.

Posledním souborem složky kernel je soubor *pageincluder.php* v němž je jediná třída - a to *PageChanger*. Účelem této třídy je vykreslení stránky, kterou požadujeme – určujeme parametrem, který ji zašleme.

#### 4.4. Modul board.php

Složka *module* obsahuje definice různých tříd. Nejobsáhlejší je soubor *board.php* s třídou *Board*. Třída se stará o výrobu obrázku šachovnice, vytváření zápisu a vytváření tahů a jejich vracení a případné přehrání. Nejdůležitější částí je kontrola, zda je tah v souladu s pravidly [4]. Na to se používá nejprve funkce *is\_Move*. Tato funkce v závislosti na figuře překontroluje, zda se jedná o tah v souladu s pohybem figury. Funkce *is\_Move* zjistí od figury, jak se pohybuje. Následně vyhledá odpovídající pohyb a prohledává pole. Když se dostane na pole, kam má figura jít, a nenarazí po cestě k tomuto poli na jinou figuru, má napůl vyhráno. Ještě může nastat komplikace, když na pole, kam chce figura jít, stojí jiná figura. Řešení je snadné, protože každá figura má kromě pohybu i příznak zda při tomto pohybu může brát figuru. Další funkce na určení, zda je tah v souladu s pravidly, je *try\_Move*. Ta odvádí tu horší práci a tou je zjišťování, zda není král v šachu. Provádí to tak, že vytvoří tah potomka třídy *Move*. Na další práci si zavolá

funkci *delete\_Move*. Funkce *delete\_Move* následně použije tah připravený od funkce *try\_Move*, provede ho a uloží. Poté jen zjistí, jestli je král v šachu pomocí funkce *in\_Check*. Když nebude král v šachu, tak je vše v pořádku a tah je platný. V opačném případě tah vrátí a smaže ho. Jak vše probíhá je vidět na Obrázek 9 vyhodnocení tahu.



Obrázek 9 vyhodnocení tahu

Funkce *is\_Move* a *try\_Move* prověřují všechny tahy kromě rošády. O tu se starají funkce *is\_Castle* a *try\_Castle*. Rozdíl je v tom, že funkce *is\_Small\_Castle* a *is\_Big\_Castle* zjišťují jen souřadnice odkud a kam král jde. O ostatní se stará funkce *try\_Castle*. A to prověřením zda je postavení figur správné a mají odpovídající parametry. Parametry se myslí to, zda se figury provádějící rošádu nepohnuly. Následně pokračuje kontrolou, zda se král nedostane do šachu během pohybu z původní souřadnice na cílovou. To se zjistí projitím těchto souřadnic a zavoláním funkce *in\_Check*. Funkce *in\_Check* hledá figury, které ohrožují krále. Využívá k tomu třídu *CheckConstants* a její funkce. Nejprve si uloží pohyb figury a následně prochází pole, dokud nenarazí na nějakou figuru anebo se dostane mimo šachovnici a tím pádem hledá podle jiného pohybu. Když se zasekne při hledání v poli o nějakou figuru tak prověřuje, zda není v seznamu figur, které při aktuálním pohybu dávají šach. Když jí nalezne v tomto seznamu a má opačnou barvu než král, je král v šachu a končíme s hledáním. Tento algoritmus je ještě modifikován o parametr, zda hledat dál či projít jen jednu souřadnici. Tento parametr se použije například u koně, ten může jen na určitá pole a prohledávat pole dál by postrádalo smysl. Dalším problémem je jak provést proměnu pěšce - na to se

použije funkce *promote\_Accomplish*. Funkce prověří, že jsme dostali figuru, v kterou se může pěšec proměnit, a překontroluje platnost tahu. Funkce *delete\_Move* má jednu zvláštnost, dokáže smazat i více tahů. Použijeme to tehdy, když zadáme například špatně třetí tah partie. Vrátime tahy a nechceme, abychom zbylou část partie museli zadat znovu. To obstará funkce *delete\_Move*, projde zbývající tahy a smaže jen ty, co následují za prvním nemožným tahem.

Dalším úskalím se zabývá třída *Board* a to vytvářením zápisu partie. Velkou část si odvede samotný potomek třídy *Move*. Ale zjištění, zda na stejné pole nemůže jít stejná figura, musí zjistit třída *Board* s použitím funkce *get\_Variant*. Podrobněji je to popsáno v další kapitole. Další věc, kterou musí šachovnice zvládnout, je vytváření tahu i bez zadání tahu na šachovnici. K tomu slouží funkce *create\_From\_MoveList*. Vytváří tah z číselného kódu a zápisu tahu. Šachovnice dokáže vytvořit i seznam těchto kódů pro uložení do databáze. O tento proces se stará funkce *moves*. Samozřejmě je zde i funkce na kopírování jen n tahů pro účely kopírování a k tomu slouží funkce *copy\_Moves*, kde podle parametru kopírujeme jen n tahů. Obdobné funkce jsou pro ukládání zápisu partie a to pomocí funkce *moves\_Record*. Pro kopírování jen n tahů ze zápisu slouží funkce *copy\_Moves\_Report*. Dalším obsahem jsou pomocné třídy pro třídu *Board*. V těchto třídách jsou uloženy konstanty jako základní postavení věží ve třídě *BasicRook* a základní postavení krále ve třídě *BasicKing*. Další pomocnou třídou je *CheckConstants* s konstantami pro kontrolu, zda je král v šachu. Posledním obsahem tohoto souboru je *Coordinate* jak již název napovídá, jedná se o souřadnice na šachovnici - určitě využijeme i funkce této třídy *get\_Bar\_Name* a *get\_Rank\_Name* pro vytváření zápisu.

#### 4.5. Zápis partie

Pro vytvoření zápisu tahu musíme zjistit, zda na cílové pole nemůže jít ještě jiná figura stejného typu a barvy. Z toho vyplývá, že při tahu figur jako je jezdec, střelec, dáma a věž, musíme zjistit, zda na cílové pole může jít ještě jiná figura stejného značení a barvy. Prohledáváme tedy pole z cílového pole figury a to tak, že podle pohybu figury jdeme a procházíme pole a tím získáme seznam souřadnic figur stejného typu a barvy. Z počtu figur v seznamu zjistíme, zda jsme hledali dobře a případně jestli máme přidávat písmenko či číslo varianty - tím se zabývá funkce třídy

*Board get\_Variant*. Když nenalezneme žádnou figuru, která by mohla na cílové pole, tak tah není možný. Když najdeme jen jednu a má stejné souřadnice naší figury, tak jsme hledali dobře, jinak se jedná o stejný případ jako předtím. Složitější je hledání při více figurách v seznamu. Musíme najít souřadnice naší figury a hlídat, zda ostatní figury stejného značení a barvy mají stejný sloupec nebo řadu jako právě hledaná figura. Když některá z ostatních figur má stejný sloupec, tak přidáme do zápisu číslo řady naší figury, jedná-li se o stejnou řadu, přidáme písmeno sloupce. Jestliže nemá žádná z figur ze seznamu stejný sloupec nebo řadu, tak přidáme písmeno sloupce do zápisu tahu naší figury (nastává spíš u jezce). Celkový zápis vypadá následovně: přidáme nejprve písmeno figury, k tomu se využívá funkce třídy *Piece get\_Piece\_Label*:

- K – král
- D – dáma
- J – jezdec
- S – střelec
- V – věž
- u pěšce se písmenko nepíše.

U krále se písmenko K nepíše v případě, že jde o rošádu. Místo K zapíšeme:

- pouze O-O v případě malé rošády
- O-O-O v případě velké rošády

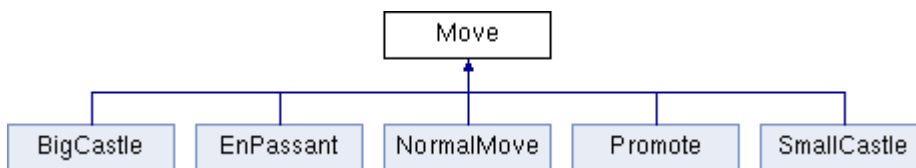
U rošády je tedy vytváření zápisu lehčí a rovnou víme, co máme vypsát podle typu rošády. Zápis pokračuje přidáním písmenka nebo varianty, pokud na políčko může víc figur stejné barvy a typu. Dále přidáme x, jestli jde o braní jiné figury, tj. stojí na cílovém poli figura jiné barvy. U pěšce při braní jiné figury přidáváme navíc před x písmenko sloupce kde stál. Stejná část pro všechny figury je přidání písmena a poté čísla cílového pole, k tomu použijeme funkce třídy *Coordinate get\_Bar\_Name* a *get\_Rank\_Name*.

#### 4.6. Modul Move.php

Na uložení tahů se používá třída *Move*. Více využijeme její potomky jako je třída *BigCastle*, *SmallCastle*, *NormalMove*, *Promote* a *EnPassant*.

*Dědičnost* třídy *Move* je vidět na Obrázek 10 dědičnost třídy *Move*. Každá třída má ke své identifikaci typ. Při konstrukci těchto potomků se zároveň vytváří i zápis partie. Díky tomu, že známe typ tahu, se nám lépe provádí tahy u třídy *Board*. Každý z těchto potomků umí vytvořit svůj kód tahu, je k tomu určena funkce *move\_Into\_Database*. Tento kód je složen z šesti čísel. První číslice je typ tahu, následně tu je i souřadnice odkud figura jde a kam jde – vždy jedna číslice řádku a jedna sloupce. Posledním číslem je většinou nula, jen v případě třídy *Promote* se zde nachází číslo udávající figuru, v kterou se proměňuje.

Samozřejmostí je i to, že tyto třídy umí vypsat i svůj zápis partie. Tomu je určena funkce *move\_Report*. V některých případech potřebujeme měnit zápis tahu, proto disponují tyto třídy i funkcí *change\_Move\_Report*. Z důvodu větší složitosti rošády je u tříd *BigCastle* a *SmallCastle* i pomocná funkce *get\_Const\_Name*. Tím zjistíme jména pomocných tříd s konstantami. Jsou to třídy *SmallCastleConst*, *BigCastleConst* a *CastleConst*. Dalším pomocnou třídou je třída *PromoteInfo*. Využívá se pro zjištění, zda se jedná o proměnu pěšce.

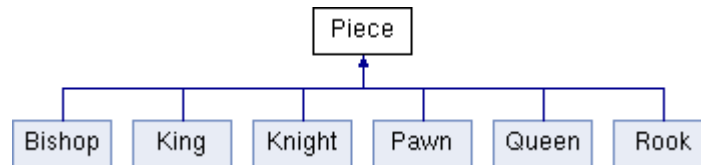


Obrázek 10 dědičnost třídy *Move*

#### 4.7. Modul *Piece.php*

Tento modul obsahuje mimo jiné definice figur. Všechny figury mají předchůdce, je jím třída *Piece*. Potomci této třídy jsou *King*, *Rook*, *Queen*, *Knight*, *Bishop* a *Pawn*. Dědičnost třídy *Piece* je vidět na Obrázek 11 dědičnost třídy *Piece*. Každá figura má i svou barvu - je jí instance třídy *Color*. Také obsahují i své další údaje o typu figury. Hlavní podstatou modulu potomků třídy *Piece* je zjistit, jak se daná figura pohybuje - k tomu jsou určeny funkce *get\_Find\_Depth*, *get\_Piece\_Move\_Count*, *get\_Piece\_Move\_Rank*, *get\_Piece\_Move\_Bar* a *get\_Exchange*. Těchto funkcí se využívá při prohledávání pole. Protože funkce *get\_Piece\_Move\_Rank* a *get\_Piece\_Move\_Bar* vrací pole, kde jsou

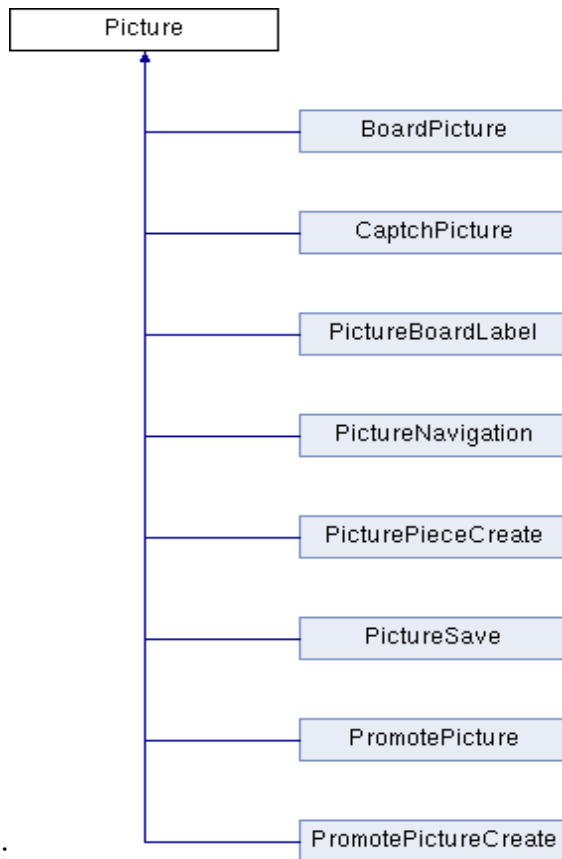
uloženy hodnoty další souřadnice. Pomocná třída, kde jsou uloženy jednotlivé pohyby figur, se nazývá *PieceMove*. Zde jsou uloženy jednotlivé tahy figur s dalšími údaji o figurách. Další třídou, kterou zde nalezneme, je *Color* - hlavním účelem je pamatovat si barvu figur a využít ji u třídy *Board* jako indikaci toho, kdo je na tahu. Funkce, kterou budeme často používat, se jmenuje *opposite\_Color*. Jak je z názvu patrné, používá se k zjištění opačné barvy k barvě, kterou funkci předložíme.



Obrázek 11 dědičnost třídy *Piece*

#### 4.8. Modul *Image.php*

V modulu je řešeno hlavně vykreslování jednotlivých obrázků a jejich vytváření. Obsahem tohoto modulu je třída *Picture* a její potomci *BoardPicture*, *CaptchPicture*, *PromotePicture*, *PromotePictureCreate*, *PicturePieceCreate*, *PictureNavigation*, *PictureSave* a *PictureBoardLabel*. Dědičnost třídy *Image* je vidět na Obrázek 12 Dědičnost třídy *Picture*. Třídy využívají k vykreslení obrázků přímé předání kódu obrázku, tím pádem se nemusí obrázek nikde ukládat a odpadne nám ukládání obrázku na serveru a poté mazání, když už nebude potřeba. Funkce se jmenuje *show\_Image*. Tato funkce přímo předává HTML kód obrázku a předává i další parametry jako je název JavaScript funkce, které používá, a případné id jméno HTML elementu.



Obrázek 12 Dědičnost třídy Picture

#### 4.9. Modul gamelist

Modulem k vygenerování seznamu partií je *gamelist.php* s třídou *GameList*. Ten zajišťuje výpis seznamu partií z databáze pomocí dotazu, který dostane. Změnit dotaz lze zasláním třídy jiného dotazu, nebo poslat pole s parametry pro vyhledávání a třída zpracuje z údajů nový dotaz. Tato třída se i stará o zpracování údajů poslaných pomocí formuláře pro vyhledávání. Dále se dokáže postarat o změnu aktuální stránky seznamu partií. Dokáže změnit i seznam partií a setřídít ho podle některého sloupce. Využíváme k tomu JavaScript funkce.

#### 4.10. Modul review

Pro přehrávání je určen modul *review*. Třída pro přehrávání je třída *Review* - té se předá zápis tahů a kódy tahů a pomocí funkcí třídy *Board* zajišťuje vše, co uživatel požaduje. Hlavním datovou strukturou je zde třída *Board*. Důležitou součástí této třídy je zajištění, aby uživatel mohl kopírovat tahy partie.

#### 4.11. Modul MovingBoard

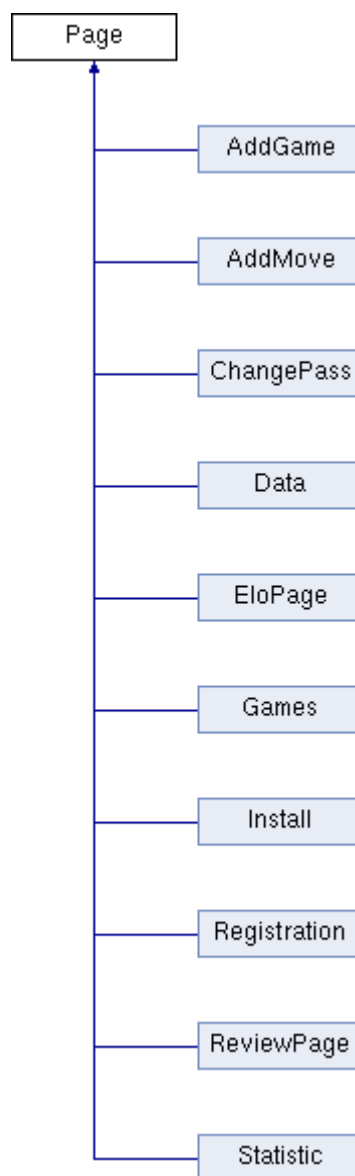
Pro přidávání tahů je určena třída *AddMoveForGame*, která se postará o zpracování, kliknutím na šachovnici dále zpracovává proměnu pěšce. Páteří této třídy je opět třída *Board*. O vytváření obrázku šachovnice se zde stará třída *BoardPicture*. Další posláním této třídy je uložení partie do databáze.

#### 4.12. Modul searchboard

Posledním modulem je *searchboard.php*. Třída *SBoard* v tomto modulu se stará o změny na šachovnici. K těmto změnám používá funkce JavaScriptu. Změny provádíme tak, že klikneme na obrázek figury, kterou chceme postavit na šachovnici, a kliknutím na šachovnici se nám překreslí figura. Hlavně tato funkce zpracovává požadavky z JavaScriptu. Mění i obrázek na základě požadavků JavaScript.

#### 4.13. Pages

Dalším obsahem aplikace je složka *pages*, v ní jsou jednotlivé definice tříd stránek, které se v aplikaci vyskytují. Dědičnost třídy *Page* je vidět na Obrázek 13  
Dědičnost třídy Page



Obrázek 13 Dědičnost třídy Page

#### 4.12.1. Stránka AddGame

Tato stránka se stará o vykreslování formuláře pro přidávání partie. Dále se nám stará o kontrolu vstupů z tohoto formuláře a úpravu dat v databázi na základě těchto údajů. Stejný formulář nám stránka zobrazí i pro úpravu a kopírování partie. Je to tím, že nechceme docílit toho, aby se pro každý z těchto úkolů generoval jiný formulář. Kontroly formuláře budou také vždy stejné, proto by bylo zvláštní je používat znovu. Kontrolování všech formulářů se provádí pomocí PHP. Je to z jednoho prostého důvodu - na některých

webech nemusí fungovat kontroly pomocí JavaScriptu. Například při přesunu aplikace z lokálního webu na webový server se vyskytly problémy s funkčností těchto kontrol. Proto jsme nechali kontroly formulářů na PHP.

#### 4.12.2. Stránka AddMove

Tato stránka nejprve vykresluje stránku na zadávání tahu na šachovnici. Dalším účelem je posílat odpovědi JavaScript funkcím, které mění obsah stránky. Také se nám tato stránka postará o uložení zadaných tahů k partii, kterou měníme, ukládáme nebo kopírujeme. Stránka pozná, o jakou činnost se jedná a vykoná příslušné akce k danému úkolu.

#### 4.12.3. Stránka Data

Tato stránka se stará o změnu údajů u právě přihlášeného uživatele. Vykresluje formulář pro změnu údajů, ale dříve než to udělá, si načte údaje o uživateli a pak jen mění tyto údaje. Formulář, stejně jako u všech formulářů, kontrolujeme po odeslání. Jestli jsou všechny údaje v pořádku, tak se postará i o změnu v databázi. Jedním důvodem proč neměnit data v databázi je například to, že login má už jiný uživatel.

#### 4.12.4. Stránka ChangePass

Tato stránka se používá pro změnu hesla uživatele. Stránka nám generuje formulář pro změnu hesla. Shodnost starého a správnost nového hesla s potvrzením tohoto hesla má za úkol kontrolovat tato stránka. O změnu hesla se postará tato stránka po odeslání formuláře se správnými údaji.

#### 4.12.5. Stránka EloPage

Tato stránka má za úkol generovat formulář pro změnu elo. Dalším účelem je po odeslání spočítat změnu elo.

#### 4.12.6. Stránka FindGame

Tato stránka generuje vyhledávací formulář a reaguje na požadavky JavaScript funkce na změnu figury na šachovnici.

#### 4.12.7. Stránka Games

Stránka generuje seznam partií a stará se o odpovědi na požadavky JavaScript funkce. Požadavky jsou nejčastěji na změnu seznamu v závislosti na jiném parametru třídění nebo stránky.

#### 4.12.8. Stránka Install

Hlavním účelem této stránky je vytvoření tabulek v *MySql* databázi.

#### 4.12.9. Stránka Page

Hlavní předek všech potomků stránek. Generuje záhlaví a zápatí všech HTML stránek kromě stránky pro elo.

#### 4.13.10. Stránka Registration

Vykresluje formulář pro registraci uživatele, údaje kontroluje opět po odeslání formuláře. Pokud jsou všechny údaje ve správném formátu, zapíše je do databáze.

#### 4.13.11. Stránka ReviewBoard

Generuje stránku pro přehrávání partie a mění stránku na základě požadavků od JavaScript funkce. Stránka též reaguje na požadavek o kopírování partie.

#### 4.13.12. Stránka Statistic

Tato stránka generuje statistiky uživatelů.

### 4.14. Složka codes

V této složce jsou obsaženy HTML kódy formulářů, stránek a všeho, co z velké části obsahuje html elementy. Tyto soubory jsou obohaceny o PHP skripty potřebné k vykreslení stránek.

#### 4.15. Modul elo

Následným modulem je *elo.php* obsahující třídu Elo. Účelem třídy Elo je vypočítat změnu *elo* po zadání koeficientu. *Elo*, které se změní, *elo* soupeře a výsledku. Výpočet obsahuje binární vyhledání, pomocí kterého a rozdílu *elo* zjistíme číslo indexu v poli s rozdíly *elo*. Poté se podíváme do pole *biggerelo*-když má *elo* ke změně větší hodnotu než *elo* soupeře, v opačném případě do pole *smallerelo*. V tomto poli se podíváme na číslo indexu, které jsme zjistili při vyhledávání. Následně od výsledku partie odečteme číslo získané z tohoto pole a použijeme koeficient, kterým vše vynásobíme. Výše popsany postup vychází z tabulky 1. *H* značí předpokládaný procentuální výsledek hráče s vyšším *elo*, hodnota *L* značí předpokládaný procentuální výsledek hráče s menším *elo*. Poté výpočet samotný je pomocí tohoto vzorce  $E_n = k * (Vysl - P)$ . Kde **k** je koeficient, **P** je hodnota z tabulky a **E<sub>n</sub>** je hodnota změny *elo* [11].

## 5. Uživatelská dokumentace

### 5.1. Požadavky před instalací:

- Nastavený SMTP [16] klient pro odesílání pošty
- Databázový server MySql ve verzi 5.6 nebo vyšší
- Povolený JavaScript v prohlížeči
- Textový editor pro úpravu souboru `config.php`
- Nainstalovaný PHP server alespoň verze 5.5.
- Mít účet administrátora a běžného uživatele pro práci s databází MySql.

### 5.2. Instalace

- ❖ Zkopírování obsahu souboru `WebDatabase.zip` instalačního CD do jedné složky na webový server.
- ❖ Konkrétně tento obsah:
  - Složka *image*
  - Složka *javascript*
  - Složka *kernel*
  - Složka *pages*
  - Složka *module*
  - Složka *text*
  - Složka *style*
  - Složka *codes*
  - Soubor *index.php*
  - Soubor *config.php*
- ❖ Nastavíme proměnnou `$CFG['database']` na údaje administrátora databáze v souboru *config.php* na webovém serveru
  - Proměnnou *to* nastavíme na adresu databázového serveru
  - Proměnnou *username* nastavíme na uživatelské jméno správce databáze
  - Proměnnou *password* nastavíme na uživatelské heslo správce databáze
  - Proměnnou *database* nastavíme na název databáze - viz Obrázek 14 nastavení databáze.

```
$CFG['DATABASE']['to'] = '127.0.0.1:3306';  
$CFG['DATABASE']['username'] = 'root1';  
$CFG['DATABASE']['password'] = '*';  
$CFG['DATABASE']['database'] = 'webdatabase';
```

Obrázek 14 nastavení databáze

- ❖ Spustíme instalační skript tabulek
  - Ve webovém prohlížeči zadáme adresu stránky odpovídající složce se soubory na webovém serveru
  - Doplníme do této adresy: `index.php?page=install` např. `http://mujweb.cz/projects/index.php?page=install`
  - Potvrdíme
  - Vytvoří se nám tabulky v databázi- viz Obrázek 15 vytvoření tabulek databáze

---

```
Tabulka USERS vytvořena !  
Tabulka GAMES vytvořena !  
Tabulka STATISTIC vytvořena !  
Tabulka CONFIGURATION vytvořena !  
Tabulka GAME_CONFIGURATION vytvořena !  
Tabulka ROLE vytvořena !  
Tabulka USER_ROLE vytvořena !  
Tabulka MOVE vytvořena !
```

---

Obrázek 15 vytvoření tabulek databáze

- Pokud jsme neuspěli, zkontrolujeme soubor `config.php` a údaje v proměnné `$CFG['database']` a vrátíme se o 4 odrážky zpět
- Smažeme soubor `install.php` ze složky pages.
- ❖ Úspěšně jsme zprovoznili databáze
- ❖ Změníme údaje v souboru `config.php` na údaje běžného uživatele databáze MySQL. Tyto změny provádíme z důvodu bezpečnosti, aby šel upravovat pouze obsah tabulek databáze.

### 5.3. Registrace uživatele

Na úvodní obrazovce klikne uživatel na položku registrace, po správném vyplnění formuláře -viz Obrázek 16 registrace uživatele do databáze, dojde k odeslání registračního e-mailu, po té dojde k dokončení registrace -viz Obrázek 17

dokončená registrace a uživatel obdrží mail s registračními údaji. Následně se může uživatel přihlásit - viz Obrázek 18 přihlášení do aplikace.

## Webová databáze šachových partií

**Registrace nového uživatele:**

Jméno:	<input type="text" value="Adam"/>
Příjmení:	<input type="text" value="Moravec"/>
Uživatelské jméno:	<input type="text" value="AdamM"/>
E-mail:	<input type="text" value="adam@seznam.cz"/>
Heslo:	<input type="password" value="....."/>
Potvrzení hesla:	<input type="password" value="....."/>
	<input type="text" value="nynmb"/>
Přihlášení:	<input type="button" value="Registrovat se"/>

Obrázek 16 registrace uživatele do databáze

## Webová databáze šachových partií

**Registrace nového uživatele:**

Jméno:	<input type="text" value="Adam"/>	v pořádku
Příjmení:	<input type="text" value="Moravec"/>	v pořádku
Uživatelské jméno:	<input type="text" value="AdamM"/>	login neexistuje
E-mail:	<input type="text" value="adam@seznam.cz"/>	v pořádku
Heslo:	<input type="password" value="....."/>	
Potvrzení hesla:	<input type="password" value="....."/>	hesla se shodují
	<input type="text" value=""/>	text z obrázku je v pořádku
Přihlášení:	<input type="button" value="Registrovat se"/>	

Registrace kompletní údaje si zkontrolujte na mailu.

Obrázek 17 dokončená registrace

## Webová databáze šachových partií

Přihlášení:	
Uživatelské jméno:	<input type="text"/>
Heslo:	<input type="password"/>
<b>Registrace</b>	<input type="button" value="Přihlásit se"/>

Obrázek 18 přihlášení do aplikace

### 5.4. Přidání partie:

Po úspěšném přihlášení klikne uživatel na položku Zadat partii. Poté vyplní formulář s údaji o partii a stiskne tlačítko uložit – viz Obrázek 19 zadávání údajů partie.

Webová databáze šachových partií	
údaje partie:	
Jméno bílý	<input type="text" value="Michail"/>
Příjmení bílý	<input type="text" value="Tal"/>
Elo bílý	<input type="text" value="2500"/>
Jméno černý	<input type="text" value="Anatoly"/>
Příjmení černý	<input type="text" value="Karpov"/>
Elo černý	<input type="text" value="2600"/>
Místo	<input type="text" value="Baku"/>
Turnaj	<input type="text" value="Baku Open"/>
Výsledek	<input checked="" type="radio"/> 1-0 <input type="radio"/> 1/2-1/2 <input type="radio"/> 0-1
Datum	<input type="text" value="09. 05. 2015"/>
Právo číst	<input type="radio"/> ano <input checked="" type="radio"/> ne
Právo upravovat	<input type="radio"/> ano <input checked="" type="radio"/> ne
Uložit	
<input type="button" value="Uložit"/>	
Webová databáze šachových partií	
autor: Martin Postupa email: martin.postupa[zavináč]seznam.cz	

Obrázek 19 zadávání údajů partie

Následně zadává tahy na šachovnici a případně si tahy vrací pomocí šipek. Po zadání všech tahů klikne na obrázek diskety a partie se uloží do databáze. Zadávání tahů je vidět na Obrázek 20 zadávání tahů k partii.



Obrázek 20 zadávání tahů k partii

## 5.5. Vyhledávání

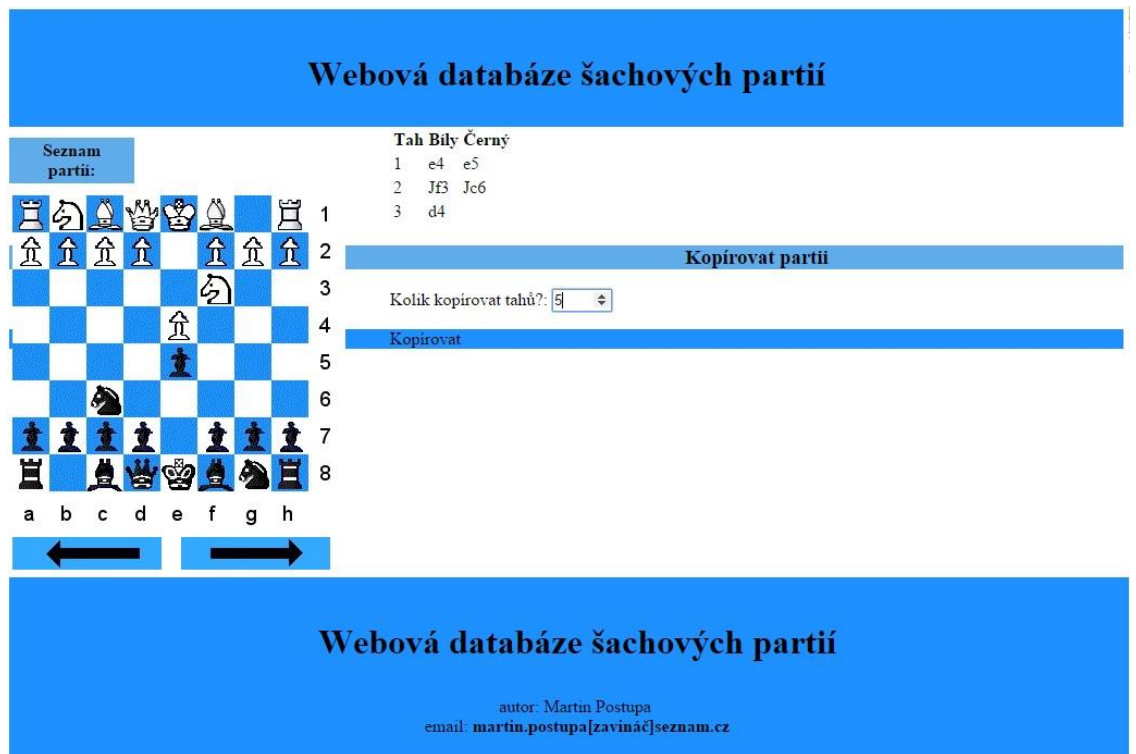
Pro vyhledávání v úvodním menu klikne uživatel na „hledat“, a poté vyplní údaje, které požaduje vyhledat, případně pozici na šachovnici – změny provádí kliknutím na paletu figur a na políčko šachovnice, nebo vymaže celé zadání pomocí tlačítka Smazat. Po vyplnění údajů klikne na tlačítko hledat. Vyhledávání je vidět na Obrázek 21 vyhledávání partie



Obrázek 21 vyhledávání partie

## 5.6. Přehrávání partie

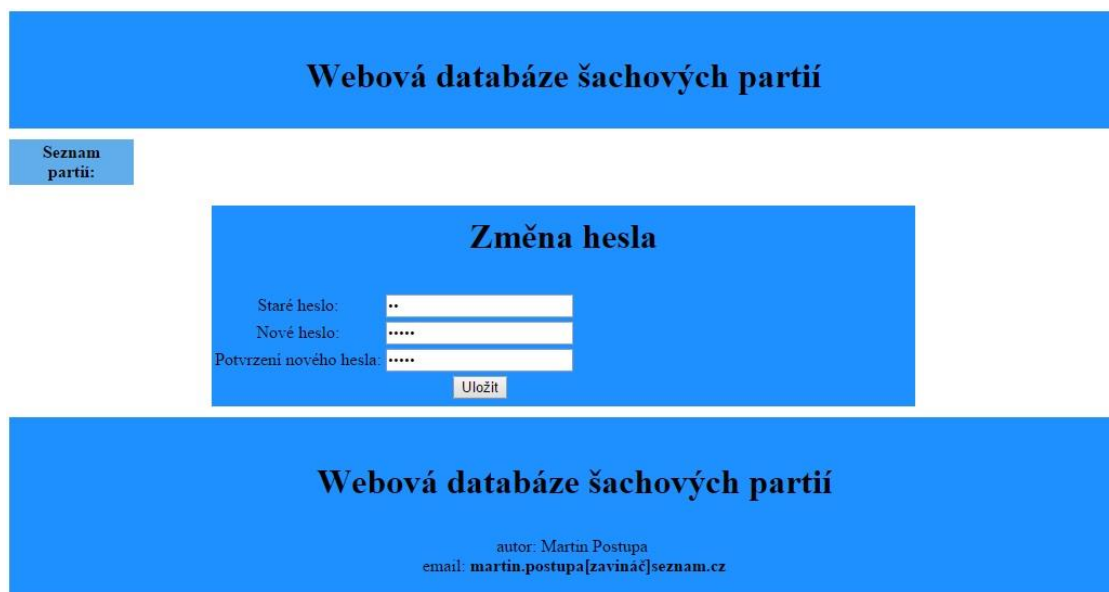
V seznamu partií klikne uživatel na tlačítko přehrát. Tahy přehrává pomocí klikání na šipky - viz Obrázek 22 přehrávání partie.



Obrázek 22 přehrávání partie

## 5.7. Změna hesla

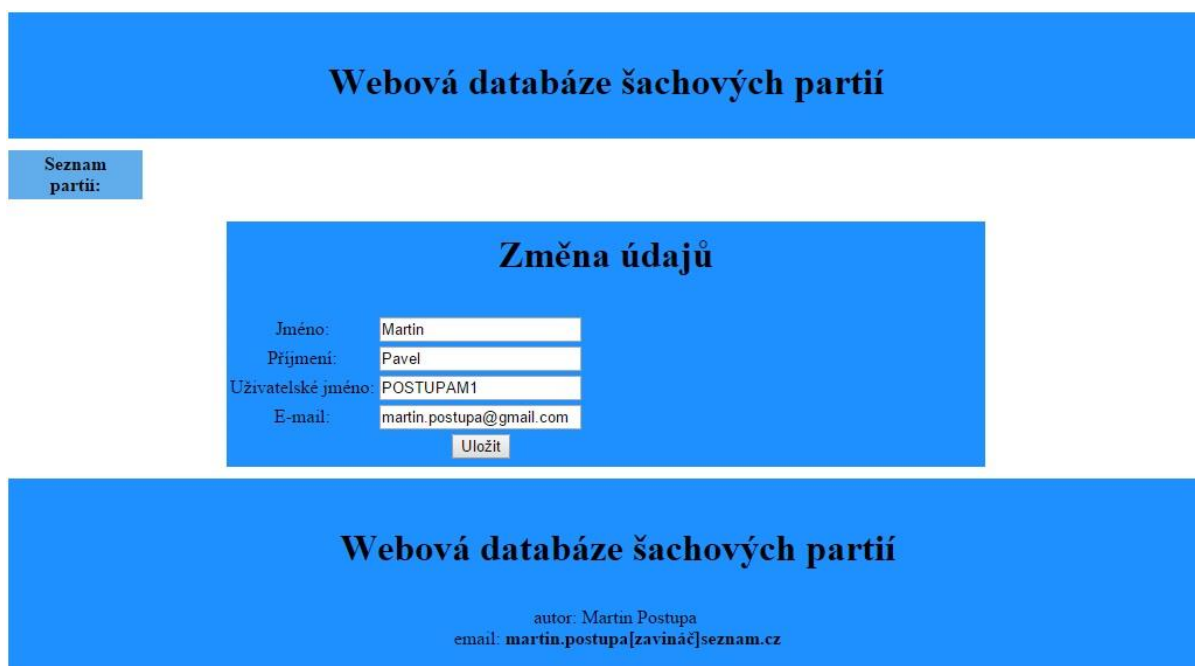
V seznamu partií klikneme na *změna hesla* a poté zadáme staré heslo a hned nové. Nové heslo potvrdíme napsáním znovu a odešleme. Musíme si být vědomi, že nové heslo obsahuje aspoň dva znaky. Změna hesla je vidět na Obrázek 23 změna hesla.



Obrázek 23 změna hesla

## 5.8. Změna údajů

V seznamu partií klikneme na změnu údajů a poté změníme údaje na správné hodnoty a uložíme. Uživatelské jméno musí mít podobně jako heslo aspoň dva znaky a nesmí ho mít žádný další uživatel. Změna údajů je vidět na Obrázek 24 změna údajů.



Obrázek 24 změna údajů

## 5.9. Kopírování partie

V seznamu partií klikneme na tlačítko přehrát. Poté navolíme počet tahů, které chceme z partie zkopírovat a klikneme na tlačítko kopírovat. Následně se pokračuje stejně jako při přidávání partie. Vyplníme formulář, kde zadáme údaje o partii. Následně upravíme tahy u partie a pokračujeme na zadávání tahů s tím rozdílem, že máme zadané tahy dopředu.

## 5.10. Statistiky uživatelů a změna ela

V seznamu partií klikneme na tlačítko statistika a to je vidět na Obrázek 25 seznam partií. Následně se nám zobrazí statistiky uživatelů. V seznamu partií klikneme na výpočet ela a zobrazí se nám formulář pro výpočet ela.

# Webová databáze šachových partií

**Seznam partií:**

[Odhlásit se](#)

[Zadat partii](#) | [Hledat](#) | [Výpočet ela](#) | [Moje partie](#) | [Vše z partií](#) | [Statistika](#) | [Změna údajů](#) | [Změna hesla](#)

**Partie 1-2**

Jméno bílý	Příjmení bílý	Elo bílý	vs.	Jméno černý	Příjmení černý	Elo černý	Výsledek	Místo	Turnaj	Datum	Přehrát	Upravit	Smazat
Martin	Postupa2	1220	-	Jan	Amos	2150	1/2-1/2	Brno	Brno Open	2015-05-03	Přehrát	Upravit	Smazat
Martina	Postupova	1900	-	Karel	Novak	2160	0-1	Praha	Open Praha	2015-05-03	Přehrát	Upravit	Smazat

[Poslední stránka](#) | [Další stránka](#)

# Webová databáze šachových partií

autor: Martin Postupa  
email: martin.postupa@zavináčseznam.cz

Obrázek 25 seznam partií

## 6. Závěr

Práce je vypracována podle stanovených cílů. Konkrétně podle cíle C1 obsahuje zadávání šachových partií, které probíhá tak, že nejprve zadáme údaje o partii a poté její tahy. Po zadání tahů a kliknutí na uložení partie, se nám uloží k partii tahy, takže jsme splnili bod C1, protože k těmto partiím existuje i databázová tabulka, kam se údaje ukládají. Navíc tato databáze umožňuje i následné přehrání partie, takže si můžeme danou partii i přehrát.

Databáze nám také umožňuje vyhledávání a to zcela tak, jak je popsáno v cíli C2. Pro tento účel nám slouží šachovnice, kde si pomocí kliknutí na obrázek figury a na kliknutí na šachovnici můžeme zadat libovolné postavení. Tento postup můžeme zkombinovat s možností zadat si další vyhledávací údaj ve formuláři. Dostaneme tak možnosti profiltrovat partie a nalézt tu, kterou hledáme. Navíc tato funkce se bude hodit všem skupinám uživatelům, protože často se stává, že potřebujeme partii, kterou jsme nezadali například z časových důvodů celou. Tak proto ji potřebujeme dožádat a k tomu se hodí právě vyhledávání.

Jak jsme si určili v cíli C3 chceme umožnit kopírovat partie. Nejlepší způsob, který jsem našli, je si v přehrávání partie navolit kolik chceme zkopírovat tahů z partie, poté klikneme na tlačítko kopírovat a tak zadáváme jakoby novou partii s tím rozdílem, že se nám tahy, které jsme zvolili, zkopírují a pouze doplníme zbývající tahy. K tomu jsme ještě nezmínili funkci, kterou obsahuje toto zadávání tahů. Pokud například se zmýlíme ve druhém tahu a zadáme místo tahu d4 tah d3 a přijdeme na to až při 30 tahu, tak pokud jsou všechny tahy zadané správné tak můžeme tento tah opravit a program nám sám ponechá pouze platné tahy.

Jako další bod jsme si zvolili třídění podle parametru. Toho jsme dosáhli jednoduše v seznamu partií. Kde si po kliknutí na sloupec např. s elem setřídíme partie bud sestupně nebo vzestupně a to podle toho, na jakou šípku nad sloupcem klikneme. Tímto způsobem jsme vyřešili bod C4.

Protože potřebujeme uživatele, potřebujeme i registraci. Proto zde máme i bod C5. Ten jsme splnili také a to tak, že jsme vytvořili registrační formulář a posíláme registrační maily. Tímto postupem získáváme uživatele do naší databáze.

Další bod cílů jsme splnili tak, že jsme použili MySQL databázi s kterou i komunikujeme pomocí SQL dotazu a máme zde uloženou řadu tabulek. Tím jsme i splnili cíl C6.

Takže jsme zjistili, že naše databáze je plně funkční, splňuje zadané úkoly a je vyzkoušená mezi šachisty. Konkrétně ji zkoušelo 10 lidí, takže je vidět, že má tato aplikace uplatnění i v praxi. Od uživatelů jsem neobdržel žádné připomínky k aplikaci. Jen si uživatelé museli zvyknout na to, jak se aplikace ovládá. Líbila se především možnost vyhledávat podle více kritérií. Moduly by šly rozšiřovat směrem k víceúčelovému šachovému serveru. Například k zadávání šachových úkolů pro děti. K tomuto cíli by bylo potřeba rozšířit počet databázových tabulek, následně vhodně rozšířit i SQL dotazy. Vyrobit zadávací mechanismus na zadávání nejprve série úkolu, poté konkrétního úkolu a vytvořit jejich přiřazení do zvolené série. K tomuto cíli by bylo potřeba ještě kontrolovat dané tahy, zda jsou korektní s pravidly, ale k tomu by stačilo použít správný modul. Ale to je námět na další práci.

## 7. Použité zdroje:

- [1] Chesbase - <http://database.chessbase.com/>
- [2] 365chess - <http://www.365chess.com/>
- [3] Chess Tempo - <http://chesstempo.com/game-database.html>
- [4] Šachová pravidla -  
<http://chess.cz/www/assets/files/informace/legislativa/Pravidla%20%C5%A1achu%20FIDE2014.pdf>
- [5] Elo vypočet –  
[http://www.chess.cz/www/informace/legislativa/zavaznedokumenty/klasifikacni\\_rad.html](http://www.chess.cz/www/informace/legislativa/zavaznedokumenty/klasifikacni_rad.html)
- [6] PHP - <http://php.net/>
- [7] ASP - <http://www.w3schools.com/asp/>
- [8] ASP.NET - <http://www.asp.net/>
- [9] Web Hostingy vyhledávání <http://www.hostingy.cz/vyber-webhostingu.html>
- [10] MySQL - <https://www.mysql.com/>
- [11] PostgreSQL - <http://www.postgresql.org/>
- [12] SQLite - <https://www.sqlite.org/>
- [13] CSS - <http://www.w3schools.com/css/>
- [14] HTML - <http://www.w3schools.com/html/>
- [15] Ajax - <http://www.w3schools.com/ajax/>
- [16] SMTP - <http://www.jakpsatweb.cz/faq/nastaveni-smtp.html>

## 8. Seznam obrázků

Obrázek 1 základní propojení stránek .....	20
Obrázek 2 celkové propojení stránek .....	21
Obrázek 3 přidávání partie.....	21
Obrázek 4 registrace uživatele.....	23
Obrázek 5 Zobrazení 64 obrázků.....	26
Obrázek 6 komunikace mezi stránkou a databází.....	28
Obrázek 7 načítání stránek.....	29
Obrázek 8 MySQL databázové schéma.....	30
Obrázek 9 vyhodnocení tahu .....	33
Obrázek 10 dědičnost třídy Move.....	36
Obrázek 11 dědičnost třídy Piece .....	37
Obrázek 12 Dědičnost třídy Picture.....	38
Obrázek 13 Dědičnost třídy Page .....	40
Obrázek 14 nastavení databáze.....	45
Obrázek 15 vytvoření tabulek databáze.....	45
Obrázek 16 registrace uživatele do databáze.....	46
Obrázek 17 dokončená registrace .....	46
Obrázek 18 přihlášení do aplikace.....	47
Obrázek 19 zadávání údajů partie.....	47
Obrázek 20 zadávání tahů k partii .....	48
Obrázek 21 vyhledávání partie .....	49
Obrázek 22 přehrávání partie.....	50
Obrázek 23 změna hesla .....	51
Obrázek 24 změna údajů .....	51
Obrázek 25 seznam partií .....	52

## **Příloha A**

Obsah přiloženého CD

K bakalářské práci je přiloženo CD s instalačními soubory, elektronickou verzí tohoto textu a dokumentací.

CD obsahuje:

- složku source se zdrojovými kódy a v ní je soubor WebDatabase.zip
- složku pdf s textem práce BcText.pdf.
- složku doc s dokumentací práce
- soubor readme.txt s popisem obsahu CD