

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO  
KATEDRA INFORMATIKY

## BAKALÁŘSKÁ PRÁCE

Herní databáze pro občanské sdružení Atmosféra



## **Anotace**

*Databáze her je programové vybavení, které má využití při zážitko-pedagogických programech. Základem je databáze her, ke které lze přistupovat pomocí počítačové aplikace a webových stránek. Obě rozhraní mají možnost vyhledávání v databázi, přihlášení k databázi, tvorbu a úpravu záznamů databáze.*

Děkuji RNDr. Miroslavu Kolaříkovi, Ph.D. za jeho vedení při zpracování bakalářské práce a Jiřímu Šindelářovi, který byl ideálním konzultantem z řad členů Atmosféra o.s..

# Obsah

<b>1. Úvod</b>	<b>7</b>
1.1. Cíle práce . . . . .	7
<b>2. Použité technologie a vývojová prostředí</b>	<b>8</b>
2.1. Databáze . . . . .	8
2.2. Počítačová aplikace . . . . .	8
2.3. Webové stránky . . . . .	9
<b>3. Teorie charakterizace her</b>	<b>10</b>
<b>4. Databáze</b>	<b>13</b>
<b>5. Počítačová aplikace</b>	<b>17</b>
5.1. Uživatelská příručka . . . . .	17
5.1.1. Hlavní okno . . . . .	17
5.1.2. Vyhledávání . . . . .	19
5.1.3. Registrace . . . . .	20
5.1.4. Přihlášení . . . . .	21
5.1.5. Moje údaje . . . . .	22
5.1.6. Vytvoření hry . . . . .	23
5.1.7. Moje hry . . . . .	24
5.1.8. Nastavení připojení do databáze . . . . .	25
5.2. Programátorská příručka . . . . .	26
<b>6. Webové stránky</b>	<b>28</b>
6.1. Uživatelská část . . . . .	28
6.2. Programátorská část . . . . .	33
6.2.1. Index a grafická část . . . . .	33
6.2.2. Obsahová část . . . . .	34
<b>Závěr</b>	<b>36</b>
<b>Reference</b>	<b>37</b>
<b>A. Podrobná programátorská příručka k počítačové aplikaci</b>	<b>38</b>
A.1. Main . . . . .	38
A.2. Main Window . . . . .	38
A.3. Editace hry . . . . .	43
A.4. Moje údaje . . . . .	46
A.5. Nastavení . . . . .	47
A.6. Podrobný výpis . . . . .	48
A.7. Přihlášení . . . . .	49

A.8. Registrace . . . . .	51
A.9. Text koule . . . . .	52
A.10.Vyhledávání . . . . .	52
A.11.Výpis položka . . . . .	53
A.12.Výpis seznam . . . . .	54
<b>B. Obsah přiloženého CD</b>	<b>56</b>

## Seznam obrázků

1.	Hlavní okno programu . . . . .	17
2.	Formulář jednoduchého vyhledávání . . . . .	19
3.	Formulář rozšířeného vyhledávání . . . . .	19
4.	Registrační formulář . . . . .	20
5.	Formulář přihlášení uživatele . . . . .	21
6.	Formulář informující o údajích uživatele . . . . .	22
7.	Formulář pro vytváření záznamů v databázi . . . . .	23
8.	Seznam her vytvořených přihlášeným uživatelem . . . . .	24
9.	Formulář pro nastavení přihlašovacích údajů do databáze . . . . .	25
10.	Úvodní stránka webu – jednoduché vyhledávání . . . . .	28
11.	Rozšířené vyhledávání na webu . . . . .	28
12.	Registrace na webu . . . . .	29
13.	Přihlášení na webu . . . . .	29
14.	Moje údaje na webu – prohlížení . . . . .	30
15.	Moje údaje na webu – úprava hodnot . . . . .	30
16.	Formulář editace hry na webu . . . . .	31
17.	Výpis her přihlášeného uživatele na webu . . . . .	32

# 1. Úvod

Občanské sdružení Atmosféra je sdružení zajímající se o zdravý životní styl, výchovu přírodou a zážitkovou pedagogikou. Hlavním způsobem jak naplňuje své zaměření je prostřednictvím realizace seznamovacích a adaptačních kurzů, školních výletů a letních či zimních zážitkových kurzů a programů. Při této činnosti využívají instruktoři občanského sdružení nepřeberné množství her a programů. V současné době je katalog těchto aktivit v *papírové*, dalo by se říci analogové, formě. Veškeré hry a programy jsou řazeny abecedně v šanonech velikosti A4. Bohužel udržet pořádek v dnes asi pěti šanonech není vůbec jednoduché, obzvláště když je využívá více než dvacet lidí. Už několik let se ve sdružení hovoří o tvorbě digitální databáze, a proto jsem se letos v rámci své bakalářské práce rozhodl takovouto databázi vytvořit.

Logickým krokem bylo inspirovat se knihami Zlatý fond her I – III, ze kterých instruktoři sdružení čerpají některé hry. V nich jsou shrnuty dlouholeté poznatky instruktorů Prázdninové školy Lipnice, která na našem území působí v oblasti zážitkové pedagogiky již více než 20 let.

Základem databázového systému je vlastní databáze umístěná na SQL serveru a dvě přístupová rozhraní. Počítačová aplikace instalovatelná pod operačním systémem Windows a webové stránky umístěné při webových stránkách občanského sdružení Atmosféra na [hry.os-atmosfera.net](http://hry.os-atmosfera.net).

## 1.1. Cíle práce

- Vytvořit databázi vhodnou pro uchovávání her používaných občanským sdružením Atmosféra. Databáze bude umístěna na webovém serveru. Uživatelé nebudou mít přímý přístup k databázi. Přes webové rozhraní (phpMyAdmin) bude mít přístup jen omezený počet osob, přibližně 2 – 3 osoby.
- Vytvořit funkční počítačovou aplikaci pro přístup do databáze. Aplikace bude umožňovat registraci do databáze a autorizovaný přístup do databáze. Přihlášený uživatel bude moci vytvářet nové záznamy v databázi a editovat záznamy jím vytvořené. Vyhledávání a rozšířené vyhledávání bude přístupné i bez přihlášení.
- Vytvořit funkční webové stránky pro přístup do databáze. Funkcionalita tohoto rozhraní bude stejná jako funkcionalita počítačové aplikace. Vzhled formulářů by měl být co nejpodobnější vzhledu formulářů v aplikaci.
- Umožnit fulltextové vyhledávání a vyhledávání podle klíčových slov.

## 2. Použité technologie a vývojová prostředí

### 2.1. Databáze

- jazyk: **Sql**  
Strukturovaný dotazovací jazyk (což je doslovný překlad zkratky sql – structured query language), který je používán pro práci s daty v databázích, které jsou založené na relačním databázovém modelu. S ohledem na dostupnost jsem vybral databázový systém **MySQL**.
- prostředí: **phpMyAdmin**  
phpMyAdmin je volný softwarový nástroj, zamýšlený pro správu administrace MySQL přes www rozhraní. Podporuje širokou škálu MySQL operací. Ty nejvíce používané operace jsou podporovány přímo uživatelským prostředím. Samozřejmostí je schopnost vykonávat jakýkoli Sql dotaz [1].

### 2.2. Počítačová aplikace

- jazyk: **C++**  
Programovací jazyk, který má otevřenou ISO-standardizaci. Je to kompilovaný jazyk silně a nebezpečně typovaný. Podporuje statickou i dynamickou kontrolu typů. Nabízí několik možností paradigmat programování (nejčastěji procedurální a objektově orientované). C++ je přenositelné mezi různými překladači jazyka, které běží na různých platformách. Má širokou škálu knihovní podpory [2].
- grafická knihovna: **Qt**  
Qt je multiplatformní aplikace a UI rámeček. Obsahuje multiplatformní knihovny tříd, integrované vývojové prostředí a multiplatformní IDE [3].
- prostředí: **QtCreator, QtDesigner**  
QtCreator je multiplatformní integrované vývojové prostředí šité na míru vývojářům Qt. QtDesigner je silný multiplatformní nástroj pro tvorbu formulářů a vzhledů oken [4].



## 2.3. Webové stránky

- jazyk: **html, css, php**

Html (hypertext markup language) a css (cascading style sheet) jsou dvě ze základních technologií pro vytváření webových stránek. Html poskytuje způsob tvorby struktury webové stránky, css potom upravuje vizuální a sluchové uspořádání [5].

Php je široce používaný univerzální skriptovací jazyk, který je zvláště vhodný pro tvorbu webových aplikací a může být vkládán do html kódu [6].

- prostředí: **PSPad, Google Chrome**

PSPad je volně šiřitelný univerzální textový editor, který využijí i tvůrci webových stránek. Editor podporuje práci s php i html soubory a je schopen zvýrazňovat syntaxi těchto jazyků [7].

Google Chrome je webový prohlížeč od společnosti Google, který byl primárně určený pro testování webových stránek.

### 3. Teorie charakterizace her

Hru lze charakterizovat pomocí mnoha kritérií. Za jeden z hlavních důvodů, proč katalogizovat takové množství dat k jednomu záznamu, považují jednoduchý a rychlý výběr hry po stanovení dramaturgických cílů kursu nebo akce. Dále bych uvedl, že podle [8] „špičkový instruktor přijíždí na tábor a ve správný čas vytahuje z krabice vše co mohl připravit doma“. Aby mohl takovou přípravu instruktor provést je potřeba mít o hrách přehledné záznamy.

Při výběru jednotlivých charakteristik, které by měl záznam o hře obsahovat, jsem vyšel z třídění her podle Prázdninové školy lipnice a téměř všechny jsme převzali do tvorby naší databáze. Upravili jsme jejich pořadí a u některých použili pro nás názornější názvy. Výsledkem jsou následující kritéria.

#### **Název hry**

Pojmenování hry jejími autory. Může ukazovat na charakter hry nebo na námět, který inspiroval její vznik.

#### **Hru vložil**

Tento atribut určuje uživatele, který hru vložil do databáze.

#### **Cíl hry**

Jedná se o výchovný cíl, to znamená na jaké složky osobnosti má hra působit a rozvíjet. Typicky hovoříme o hrách na rozvoj intelektu, tvořivosti, sociálních dovedností, motoriky a pohybových dovedností, vůle, sebepojetí, hry kombinované a speciální hry.

#### **Typ hry**

Zařazení hry do typu dává obecnější pohled na hru a zlepšuje tak představu o jejich charakteristických rysech. Jednotlivých typů her je mnoho, proto zde uvedeme jen jejich výčet.

**Typy her:** IceBreaker, Dynamics, Iniciativní, Simulační, Inscenační, Dramatická, Psychodrama, Sociodrama, Cvičení, Test, Závod, Utkání, Turnaj, Soutěž, Pořad, Dílna, Ateliér, Miniškola, Minikurs, Výstava, Představení, Přednáška, Diskuze, Seminář, Putování, Relaxace, Rozbor.

#### **Fyzická zátěž**

Určuje kolik fyzické síly budou muset účastníci dané hry vydat (1-žádná, 2-malá, 3-střední, 4-velká, 5-vyčerpávající).

#### **Psychická zátěž**

Určuje jak moc daná aktivita působí (je náročná) na psychiku hráčů (1-žádná, 2-malá, 3-střední, 4-velká, 5-vyčerpávající).

### **Čas na přípravu**

Je myšlen čas, který stráví instruktor přípravou hry na místě jejího konání. Není zde započítán čas, který je stráven přípravou materiálu.

### **Čas na hru**

Je čas, který průměrně stráví účastníci hraním hry samotné, při dodržení pravidel. Není započítán čas potřebný k vysvětlení pravidel.

### **Počet hráčů**

Určuje ideální počet hráčů, který je vhodný pro hraní hry. V databázi je možnost udat číselné rozpětí.

### **Věková kategorie hráčů**

Určuje pro jakou věkovou skupinu je hra určena. V databázi je možnost udat číselné rozpětí.

### **Instruktorů na přípravu**

Určuje počet instruktorů nutný k dodržení času na přípravu hry.

### **Instruktorů na hru**

Určuje počet instruktorů, který je nutný pro zvládnutí hladkého průběhu hry.

### **Prostředí**

Určuje prostředí, které je vhodné pro realizaci hry (místnost, louka, les, hřiště, tělocvična, skály, voda). V databázi je navíc položka *poznámka k prostředí*, ve které lze lépe charakterizovat dané prostředí.

### **Denní doba**

Určuje v jaké části dne je hra vhodné realizovat (dopoledne, odpoledne, večer, noc, svítání, kdykoliv).

### **Roční období**

Určuje, ve kterém ročním období by měla být hra uvedena, aby neztratila svoji kvalitu. (jaro, léto, podzim, zima, kdykoliv) **Charakteristika**

Stručný popis hry. Shrnuje nejdůležitější fakta, aby si instruktor mohl udělat rychle představu o hře samotné.

### **Motivace**

Popisuje způsob jakým můžou instruktoři vtáhnout hráče do hry a pomoci jim se ztotožnit s hrou.

### **Pravidla**

Přesný popis pravidel a průběhu hry. Detailní poznámky k řešení problémových situací a role rozhodčího při hře.

**Metodika**

Zaznamenané zkušenosti a postřehy z dřívějšího uvádění hry.

**Materiál**

Co možná nejúplnější seznam materiálu, který je potřebný k realizaci hry.

## 4. Databáze

Pro tvorbu databáze byl použitý jazyk MySQL verze 5.5. V testovacím prostředí byl využit databázový server Apache verze 2.2.21. Databáze je v současné době umístěna na studentském serveru katedry informatiky tux.inf.upol.cz. Je to z důvodu omezení práv uživatele na hostingu, který využívá Atmosféra o.s.. Na něm není možné se připojit z libovolné IP adresy na MySQL server a tím pádem jej nelze použít spolu s počítačovou aplikací.

Databáze se skládá ze dvou tabulek – *uzivatele* a *hry*. Tabulka *uzivatele* má typ úložiště **InnoDB**. Tabulka *hry* má typ úložiště **MyISAM**, kvůli fulltextovému vyhledávání, které první uvedený typ nepodporuje.

Tabulka *uzivatele* obsahuje informace o uživateli herní databáze. Jejimi atributy jsou:

- email – typ **varchar**, **unikátní klíč** (obsahuje email uživatele)
- jmeno – typ **varchar** (obsahuje křestní jméno uživatele)
- prijmeni – typ **varchar** (obsahuje příjmení uživatele)
- opraveni – typ **int** (určuje oprávnění uživatele, jak může zasahovat do databáze)
- prezdivka\_logIn – typ **varchar**, **primární klíč** (obsahuje přihlašovací jméno)
- heslo – typ **varchar** (obsahuje heslo)

Údaje v této tabulce slouží pro uchování informací o jednotlivých uživateli databáze. Pomocí ní je řešeno přihlašování uživatelů, kdy dochází k porovnávání hesla a přihlašovacího jména. Ostatní údaje jsou potřebné pro přesnější identifikaci jednotlivých uživatelů. Email je zařazen kvůli komunikaci s uživateli a také jako unikátní klíč, aby nevznikali duplicitní záznamy se stejným emailem a různými přihlašovacími jmény. SQL příkazy, které jsou prováděny nad touto tabulkou se týkají přihlášení uživatele, registrace uživatele a úpravy údajů uživatele.

- Přihlášení uživatele do databáze  
„SELECT \* FROM uzivatele WHERE prezdivka\_logIn = [hodnota přihlašovací jméno] AND heslo = [hodnota heslo]“
- Registrace uživatele do databáze  
„INSERT INTO uzivatele (email, jmeno, prijmeni, prezdivka\_logIn, heslo) VALUES ([hodnota email], [hodnota jméno], [hodnota příjmení], [hodnota přihlašovací jméno], [hodnota heslo])“

- Úprava údajů uživatele  
`„UPDATE uzivatele SET email =[hodnota email], jmeno = [hodnota jméno], prijmeni = [hodnota příjmení], prezdivka_logIn = [hodnota přihlašovací jméno], heslo = [hodnota heslo] WHERE prezdivka_logIn = [hodnota přihlašovací jméno]“`

Druhá tabulka *hry* je hlavní složkou celé databáze. V ní jsou uchovávána všechna data. Její jednotlivé atributy se shodují s formuláři pro vytváření a úpravu her. Atributy této tabulky jsou:

- ID\_hry – typ **timestamp**, **primární klíč** (určuje datum vzniku záznamu hry)
- nazev\_hry – typ **varchar**
- hru\_vlozil – typ **varchar**
- cil\_hry – typ **text**
- typ\_hry – typ **varchar**
- fyzicka\_zatez – typ **int**
- psychicka\_zatez – typ **int**
- cas\_priprava\_h – typ **int**
- cas\_priprava\_min – typ **int**
- cas\_hra\_h – typ **int**
- cas\_hra\_min – typ **int**
- pocet\_hracu\_od – typ **int**
- pocet\_hracu\_do – typ **int**
- vek\_hracu\_od – typ **int**
- vek\_hracu\_do – typ **int**
- instruktor\_priprava – typ **int**
- instruktor\_hra – typ **int**
- prostredi – typ **varchar**
- poznamka\_prostredi – typ **text**
- denni\_doba – typ **varchar**
- rocni\_obdobi – typ **varchar**
- charakteristika – typ **text**
- motivace – typ **text**
- pravidla – typ **text**
- metodika – typ **text**
- material – typ **text**

Sql dotazy probíhající nad touto tabulkou se týkají hlavně vyhledávání a zobrazení výsledků vyhledávání. Dále pak výpisu her přihlášeného uživatele a vytváření, úpravy záznamů her.

- Vyhledávání jednoduché – fulltextové  
`„SELECT * FROM hry WHERE MATCH (nazev_hry, hru_vlozil, cil_hry, typ_hry, prostredi, poznamka_prostredi, denni_doba, rocni_obdobi, charakteristika, motivace, pravidla, metodika, material) AGAINST ([řetězec pro porovnávání] IN BOOLEAN MODE)“`

- Vyhledávání rozšířené – fulltext + klíčová slova a hodnoty  
`„SELECT * FROM hry WHERE MATCH (nazev_hry, hru_vlozil, cil_hry, typ_hry, prostredi, poznamka_prostredi, denni_doba, rocni_obdobi, charakteristika, motivace, pravidla, metodika, material) AGAINST ([řetězec pro porovnávání] IN BOOLEAN MODE) AND MATCH (prostredi, denni_doba, rocni_obdobi) AGAINST ([řetězec pro porovnávání] IN BOOLEAN MODE) AND typ_hry = [hodnota] AND fyzicka_zatez = [hodnota] AND psychicka_zatez = [hodnota] AND cas_prip_rava_h <= [hodnota] AND cas_prip_rava_min <= [hodnota] AND cas_hra_h <= [hodnota] AND cas_hra_min <= [hodnota] AND [hodnota počet hráčů od] BETWEEN pocet_hracu_od AND pocet_hracu_do AND [hodnota počet hráčů do] BETWEEN pocet_hracu_od AND pocet_hracu_do AND [hodnota věk hráčů od] BETWEEN vek_hracu_od AND vek_hracu_do AND [věk hráčů do] BETWEEN vek_hracu_od AND vek_hracu_do AND instruktor_prip_rava <= [hodnota] AND instruktor_hra <= [hodnota]“`
- Podrobný výpis hry  
`„SELECT * FROM hry WHERE ID_hry = [hodnota id hry]“`
- Výpis her přihlášeného uživatele  
`„SELECT * FROM hry WHERE hru_vlozil = [aktuální přihlašovací jméno]“`
- Vytvoření záznamu hry  
`„INSERT INTO hry (nazev_hry, hru_vlozil, cil_hry, typ_hry, fyzicka_zatez, psychicka_zatez, cas_prip_rava_h, cas_prip_rava_min, cas_hra_h, cas_hra_min, pocet_hracu_od, pocet_hracu_do, vek_hracu_od, vek_hracu_do, instruktor_prip_rava, instruktor_hra, prostredi, poznamka_prostredi, denni_doba, rocni_obdobi, charakteristika, motivace, pravidla, material, metodika) VALUES ([hodnota název hry], [hodnota hru vložil], [hodnota cíl hry], [hodnota typ hry], [hodnota fyzická zátěž], [hodnota psychická zátěž], [hodnota čas H příprava], [hodnota čas Min příprava], [hodnota čas H hra], [hodnota čas Min hra], [hodnota počet hráčů od], [hodnota počet hráčů do], [hodnota věk hráčů od], [hodnota věk hráčů do], [hodnota instruktor příprava], [hodnota instruktor hra], [hodnota prostředí], [hodnota poznámka k prostředí], [hodnota denní doba], [hodnota roční období], [hodnota charakteristika], [hodnota motivace], [hodnota pravidla], [hodnota materiál], [hodnota metodika])“`

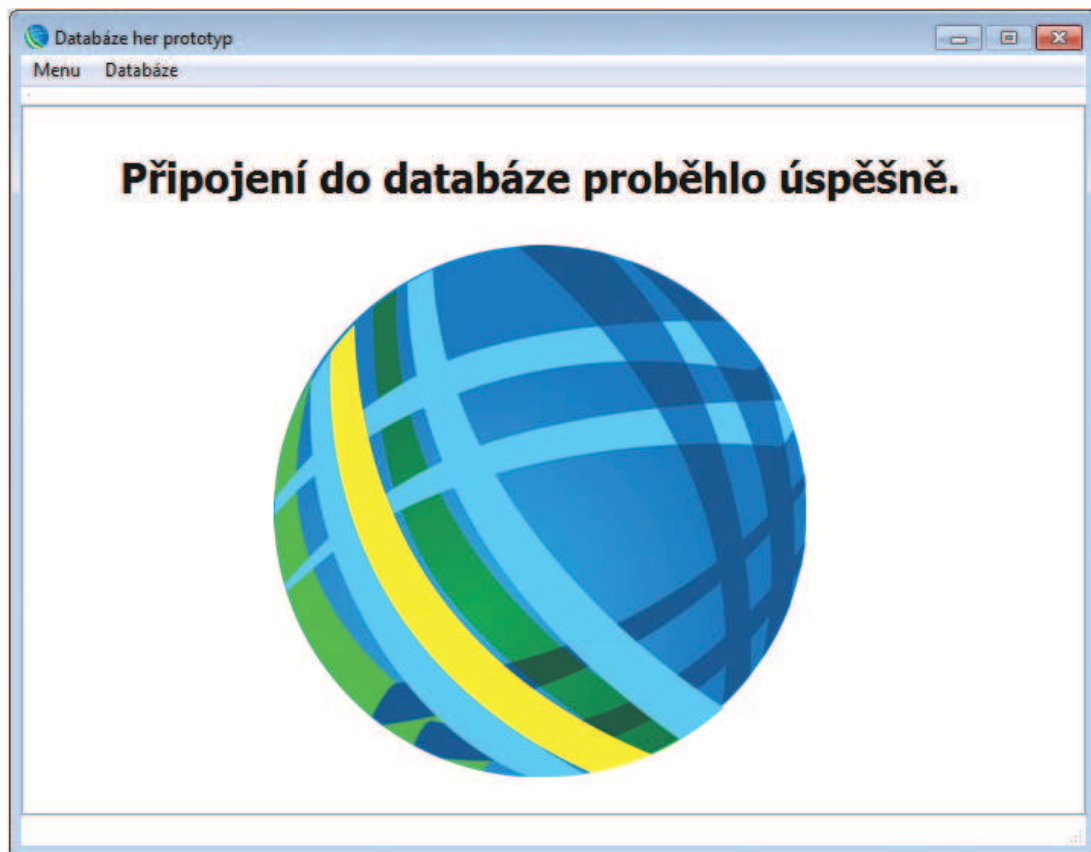
- Upravení záznamu hry  
 „UPDATE hry SET nazev\_hry = [hodnota název hry], hru\_vlozil = [hodnota hru vložil], cil\_hry = [hodnota cíl hry]', typ\_hry = [hodnota typ hry], fyzicka\_zatez = [hodnota fyzická zátěž], psychicka\_zatez = [hodnota psychická zátěž], cas\_pripava\_h = [hodnota čas H příprava], cas\_pripava\_min = [hodnota čas Min příprava], cas\_hra\_h = [hodnota čas H hra], cas\_hra\_min = [hodnota čas Min hra], pocet\_hracu\_od = [hodnota počet hráčů od], pocet\_hracu\_do = [hodnota počet hráčů do], vek\_hracu\_od = [hodnota věk hráčů od], vek\_hracu\_do = [hodnota věk hráčů do], instruktor\_pripava = [hodnota instruktor příprava], instruktor\_hra = [hodnota instruktor hra], prostredi = [hodnota prostředí], poznamka\_prostredi = [hodnota poznámka k prostředí], denni\_doba = [hodnota denní doba], rocni\_obdobi = [hodnota roční období], charakteristika = [hodnota charakteristika], motivace = [hodnota motivace], pravidla = [hodnota pravidla], material = [hodnota materiál], metodika = [hodnota metodika] WHERE ID\_hry = [hodnota id hry]“



## 5. Počítačová aplikace

### 5.1. Uživatelská příručka

#### 5.1.1. Hlavní okno



Obrázek 1. Hlavní okno programu

Hlavnímu oknu programu vévodí logo občanského sdružení Atmosféra. Nad ním se v průběhu práce s programem objevují informace o úspěšně či neúspěšně provedených akcích uživatele. Lišta menu nabízí dvě rozbalovací nabídky – Menu a Databáze.

Rozbalovací nabídka *Menu* obsahuje možnosti:

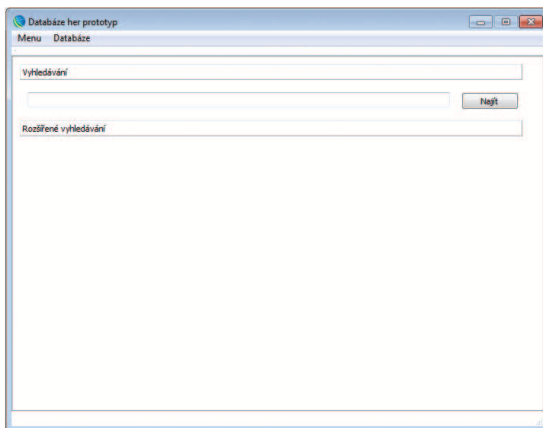
- Vyhledávání – otevře vyhledávací formulář
- Moje údaje\* – otevře údaje o přihlášeném uživateli a umožní jejich změnu
- Moje hry\* – vypíše seznam her náležících k přihlašovacímu jménu uživatele
- Vytvořit hru\* – otevře formulář pro tvorbu záznamu hry do databáze
- Přihlásit se – otevře formulář pro přihlášení uživatele do databáze
- Odhlásit se – odhlásí uživatele z databáze
- Registrace – otevře formulář pro registraci nového uživatele
- Konec – ukončí aplikaci

Položky označené \* jsou přístupné až po přihlášení uživatele do databáze.

Rozbalovací nabídka *Databáze* obsahuje možnosti:

- on-line databáze – zaškrtačací políčko, signalizuje použití on-line databáze
- off-line databáze – zaškrtačací políčko, neaktivní, připravené pro použití s off-line databází
- Nastavení připojení – otevře formulář, který umožní uživateli nastavit připojení do databáze

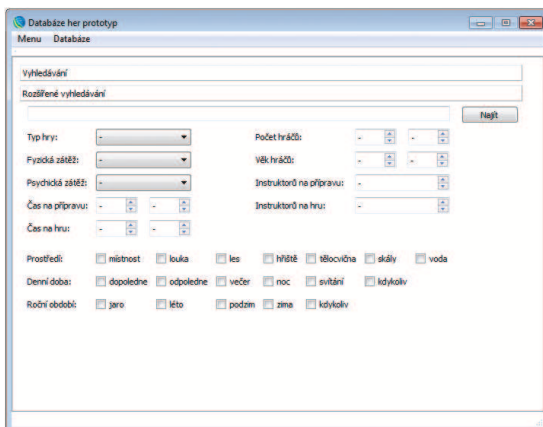
### 5.1.2. Vyhledávání



Obrázek 2. Formulář jednoduchého vyhledávání

Vyhledávání je jednou z hlavních funkcionalit, kterou přístupové rozhraní k databázi přináší uživateli. Do editovatelného řádku se píše hledané výrazy. Pokud bude více výrazů napsáno jen s mezerami budou vyhledány záznamy, které obsahují všechny slova (např. vejce místnost). V případě, že jednotlivá slova oddělíme *čárkou*, budou vyhledány záznamy, které obsahují alespoň jedno slovo (např. běhání, louka, voda).

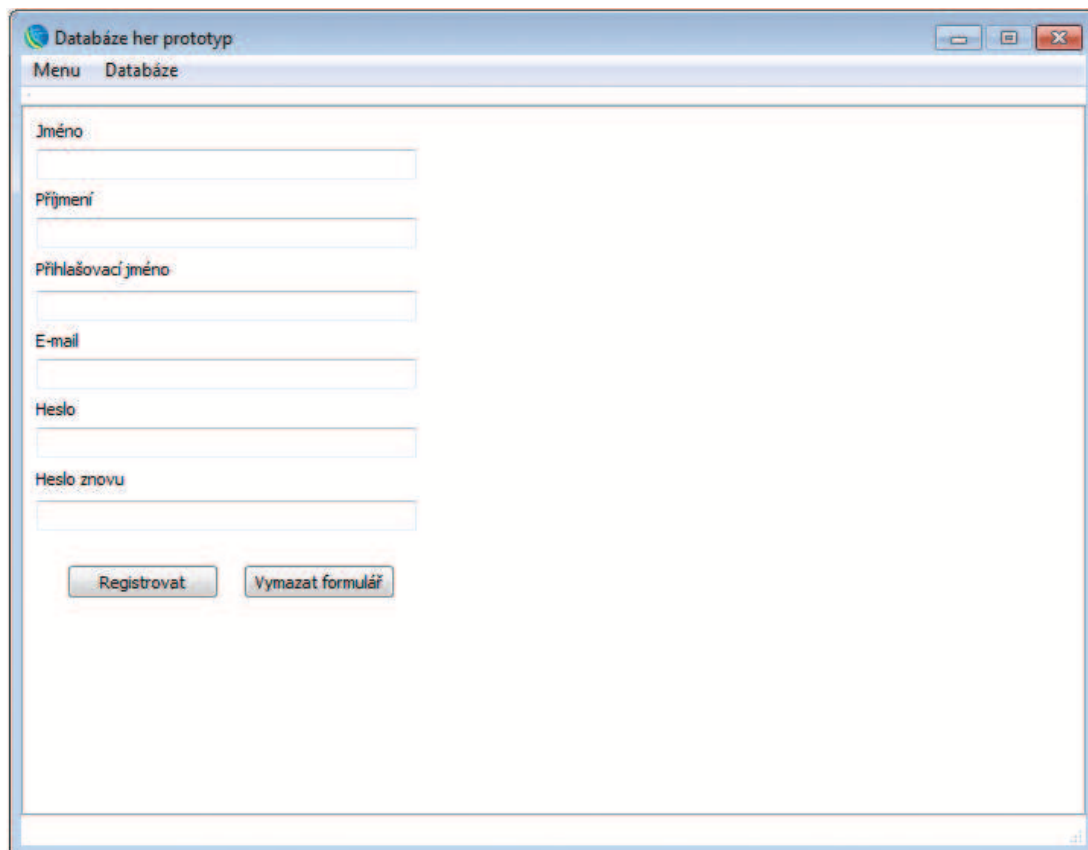
#### Rozšířené vyhledávání



Obrázek 3. Formulář rozšířeného vyhledávání

Rozšířené vyhledávání nabízí kromě editovatelného řádku ještě možnost nastavení různých kritérií, která by měl záznam splňovat. Všechna kritéria odpovídají jednotlivým atributům v databázi.

### 5.1.3. Registrace



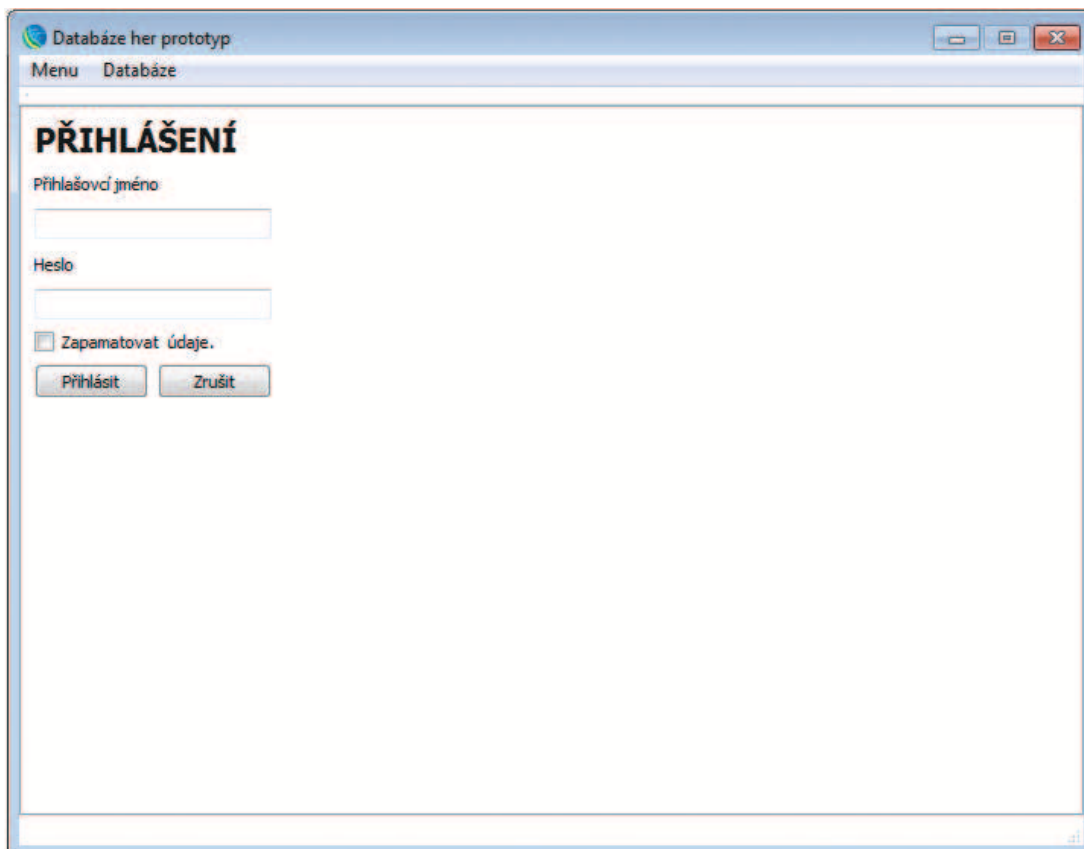
The image shows a screenshot of a web application window titled "Databáze her prototyp". The window has a menu bar with "Menu" and "Databáze". The main content area contains a registration form with the following fields and buttons:

- Jméno:
- Příjmení:
- Přihlašovací jméno:
- E-mail:
- Heslo:
- Heslo znovu:
- Buttons: "Registrovat" and "Vymazat formulář"

Obrázek 4. Registrační formulář

Registrační formulář nabízí šest položek k vyplnění, všechny jsou povinné. Heslo musí být dlouhé minimálně pět znaků. Po registraci musí uživatel vyčkat než mu administrátor jeho přístup do databáze autorizuje a přidělí mu příslušné oprávnění přístupu.

#### 5.1.4. Přihlášení



The image shows a screenshot of a web application window titled "Databáze her prototyp". The window has a menu bar with "Menu" and "Databáze". The main content area is titled "PŘIHLÁŠENÍ" in large, bold, black letters. Below the title, there are two text input fields: the first is labeled "Přihlašovací jméno" and the second is labeled "Heslo". Below these fields is a checkbox labeled "Zapamatovat údaje." which is currently unchecked. At the bottom of the form are two buttons: "Přihlásit" and "Zrušit".

Obrázek 5. Formulář přihlášení uživatele

Přihlašovací formulář vyžaduje po uživateli vyplnění jména a hesla, které se musí shodovat s údaji zadanými při registraci. Je zde možnost zapamatování přihlašovacích údajů, které se uloží do textového souboru a při opětovném spuštění se znovu načtou.

### 5.1.5. Moje údaje

The image shows two overlapping windows from a web application titled "Databáze her prototyp". The main window, titled "Menu Databáze", displays the following information:

- Jméno:
- Příjmení:
- Přihlašovací jméno:
- E-mail:
- Upravit údaje

The inset window, also titled "Menu Databáze", displays the "Změnit heslo" (Change password) form with the following fields and buttons:

- Jméno:
- Příjmení:
- Přihlašovací jméno:
- E-mail:
- Změnit heslo
- Staré heslo:
- Nové heslo:
- Nové heslo znovu:
- Ok
- Zrušit

Obrázek 6. Formulář informující o údajích uživatele

Nabídka z *Menu Moje údaje* v první řadě zobrazí základní informace o přihlášeném uživateli. Dále nabízí možnost tlačítkem **Upravit údaje** přejít do formuláře (jak je vidět v levé spodní části obrázku), kde má uživatel možnost své údaje upravit a změnit si i přístupové heslo.

### 5.1.6. Vytvoření hry

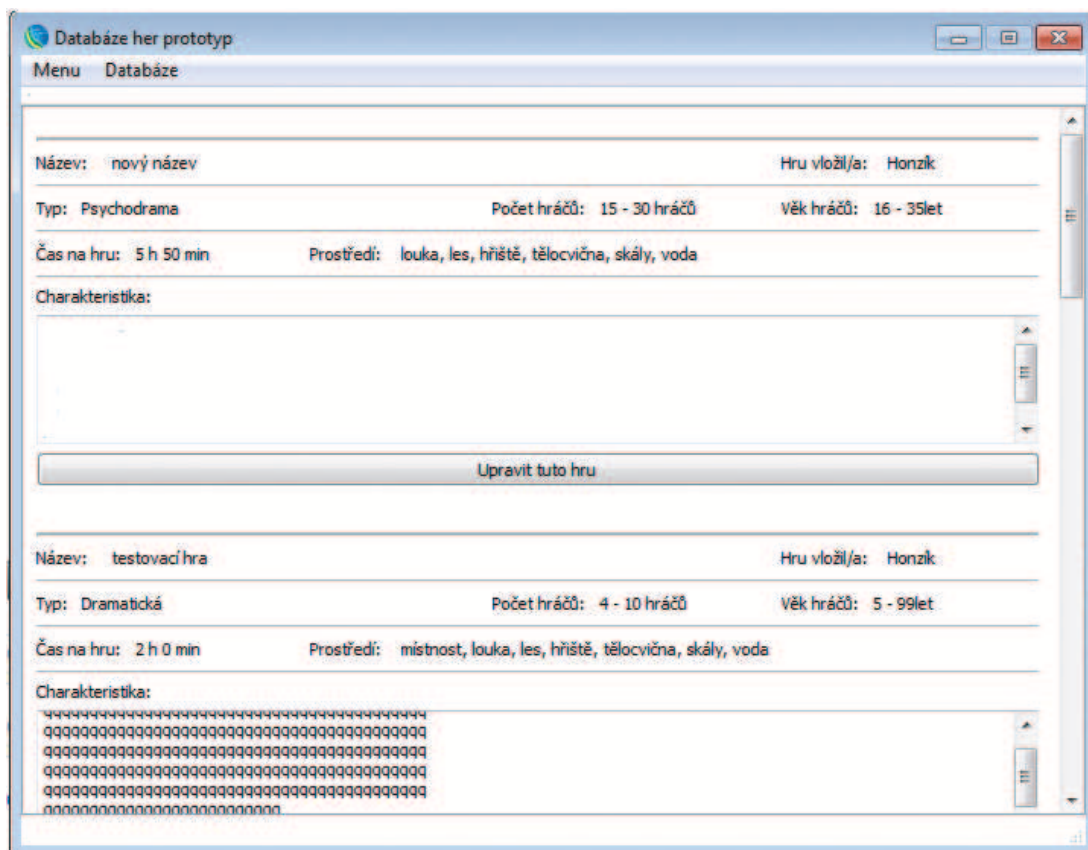
The screenshot shows a web application window titled "Databáze her prototyp". The main form is titled "Odeslat hru" and contains the following fields and controls:

- Název hry: [text input]
- Hru vložil: Honzik [text input]
- Cíl hry: [text input]
- Typ hry: IceBreaker [dropdown menu]
- Fyzická zátěž: 1 - žádná [dropdown menu]
- Psychická zátěž: 1 - žádná [dropdown menu]
- Čas na přípravu: 0 h [spin] 0 min [spin]
- Čas na hru: 0 h [spin] 0 min [spin]
- Počet hráčů: od 0 [spin] do 0 [spin]
- Věk hráčů: od 0 [spin] do 0 [spin]
- Instruktorů na přípravu: 0 instruktorů [spin]
- Instruktorů na hru: 0 instruktorů [spin]
- Prostředí:  místnost  louka  les  hřiště  tělocvična  skály  voda
- Poznámka k prostředí: [text input]
- Denní doba:  dopoledne  odpoledne  večer  noc  svítání  kdykoliv
- Roční období:  jaro  léto  podzim  zima  kdykoliv
- Charakteristika: [text area]
- Motivace: [text area]
- Pravidla: [text area]
- Materiál: [text area]
- Metodika: [text area]
- Odeslat hru [button]

Obrázek 7. Formulář pro vytváření záznamů v databázi

Formulář pro vytváření záznamu v databázi je nejrozsáhlejší formulář, se kterým se uživatel v aplikaci setká. Pomocí tohoto formuláře může uživatel vytvářet nové záznamy v databázi. Podrobný popis jednotlivých atributů naleznete v 3. kapitole. Dopředu je vyplněna jen položka *Hru vložil*, která je i dále editovatelná. Vyplněná pole formuláře se po odeslání zapíše do databáze. Odeslání se provádí pomocí jednoho ze dvou tlačítek **Odeslat hru**.

### 5.1.7. Moje hry

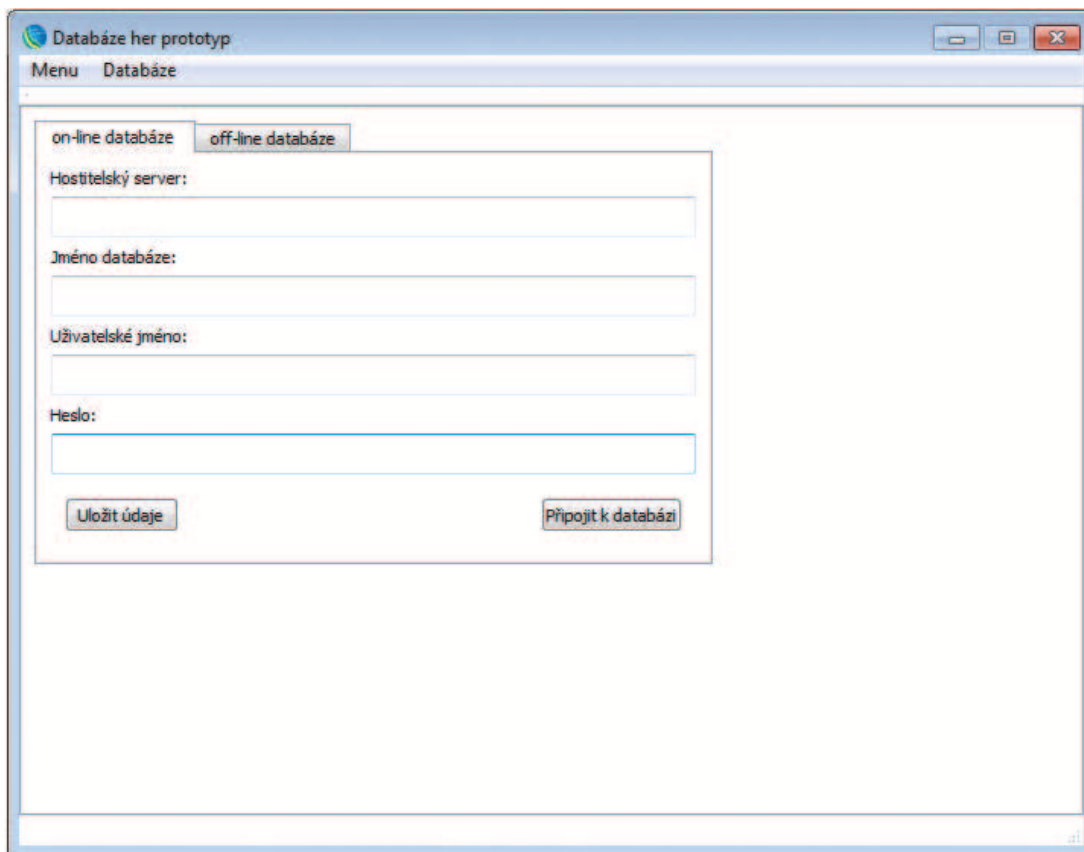


Obrázek 8. Seznam her vytvořených přihlášeným uživatelem

Nabídka z *Menu Moje hry* nabídne uživateli seznam her, které se vztahují k jeho přihlašovacímu jménu. Nabízí základní charakteristiky hry a tlačítko **Upravit tuto hru**, pomocí kterého se uživatel dostane na formulář *Úprava hry*. **Úprava hry** Formulář úpravy hry vypadá úplně stejně jako formulář pro její vytvoření. Jediným rozdílem je, že jednotlivá pole formuláře jsou dopředu vyplněna daty načtenými z databáze.



### 5.1.8. Nastavení připojení do databáze



The screenshot shows a window titled "Databáze her prototyp" with a menu bar containing "Menu" and "Databáze". The main area has two tabs: "on-line databáze" (active) and "off-line databáze". The "on-line" tab contains a form with four text input fields: "Hostitelský server:", "Jméno databáze:", "Uživatelské jméno:", and "Heslo:". Below the fields are two buttons: "Uložit údaje" and "Připojit k databázi".

Obrázek 9. Formulář pro nastavení přihlašovacích údajů do databáze

Nastavení připojení do databáze nabízí dvě záložky **on-line databáze** a **off-line databáze**. Záložka pro on-line připojení do databáze nabízí čtyři pole k vyplnění. Všechna pole jsou povinná pro úspěšné přihlášení do databáze. Dále je zde možnost vyplněné údaje uložit do textového souboru pomocí tlačítka **Uložit údaje**. Pro připojení do databáze potom slouží tlačítko **Připojit k databázi**. Záložka off-line databáze je sice aktivní, ale její obsah zatím ne. Je před chystána pro možnost vyhledávání v databázi bez nutnosti připojení do Internetu.

## 5.2. Programátorská příručka

Zde uvádím jen přehled hlavních částí programu. Podrobná programátorská příručka je součástí přílohy A.

- **Main**

Soubory: main.cpp

Tento soubor obsahuje jedinou funkci **main**. V ní je vykonán příkaz *exec* nad proměnnou **app** pro spuštění aplikace.

- **MainWindow**

Soubory: mainwindow.cpp, mainwindow.h, mainwindow.ui

**MainWindow** je hlavní částí programu. Deklaruje proměnné pro uchování instancí jednotlivých tříd, proměnnou pro práci s databází, proměnnou pro uchování adresáře, ve kterém je aplikace nainstalována. Načítá počáteční data pro přístup do databáze a provádí připojení do databáze. MainWindow zobrazuje jednotlivé formuláře v oblasti s posuvníky a realizuje většinu Sql dotazů nad databází.

- **editaceHry**

Soubory: editaceHry.cpp, editaceHry.h, editaceHry.ui

Objekt **editaceHry** realizuje formulář pro vytváření nových a úpravu stávajících záznamů databáze. Odesílá data z formuláře k dalšímu zpracování v Sql příkazu INSERT nebo UPDATE. Umožňuje nastavení hodnot z databáze do formuláře pro jejich úpravu.

- **mojeUdaje**

Soubory: mojeUdaje.cpp, mojeUdaje.h, mojeUdaje.ui

V objektu **mojeUdaje** jsou uloženy osobní údaje aktuálně přihlášeného uživatele. Umožňuje prohlížení těchto údajů a jejich úpravu. Umožňuje změnu hesla uživatele.

- **nastaveni**

Soubory: nastaveni.cpp, nastaveni.h, nastaveni.ui

Formulář **Nastavení** má dvě záložky – on-line a off-line databáze. První záložka realizuje zadání přístupových údajů pro připojení do databáze na serveru a umožňuje uložení údajů do textového souboru. Druhá záložka (*zatím není využita*) realizuje zadání přístupové cesty k souboru s off-line databází.

- **podrobnyVypis**

Soubory: podrobnyVypis.cpp, podrobnyVypis.h, podrobnyVypis.ui

Objekt **podrobnyVypis** vytváří a zobrazuje podrobný výpis jednoho záznamu z databáze. Dále provádí přípravu a umožňuje tvorbu pdf dokumentu s tímto záznamem.

- **prihlaseni**  
Soubory: prihlaseni.cpp, prihlaseni.h, prihlaseni.ui  
Objekt **prihlaseni** v sobě nese proměnnou pro záznam, zda-li je uživatel přihlášený. Zároveň si uchovává data aktuálně přihlášeného uživatele. Provádí přihlášení uživatele k databázi. Pomocí Sql příkazu porovná zadaná data uživatelem s daty, které jsou uloženy v databázi. A o výsledky informuje uživatele zprávou.
- **registrace**  
Soubory: registrace.cpp, registrace.h, registrace.ui  
Objekt **registrace** prochází všechna pole formuláře a pokud jsou správně vyplněna odešle registraci pomocí signálu. O problémech informuje uživatele zprávou.
- **textKoule**  
Soubory: textKoule.cpp, textKoule.h, textKoule.ui  
Objekt **textKoule** slouží jako informační okno. Zobrazuje text zpráv, které jiné objekty zasílají pro zobrazení uživateli.
- **vyhledavani**  
Soubory: vyhledavani.cpp, vyhledavani.h, vyhledavani.ui  
Objekt **vyhledavani** realizuje dva typy formulářů. První formulář pro jednoduché vyhledávání, který odesílá vstupní text k zpracování v Sql dotazu a druhý formulář, který nabízí rozšířené vyhledávání. Formulář rozšířeného vyhledávání umožňuje zadat téměř všechny atributy, které má záznam v databázi her a takto vyplněná data odesílá k dalšímu zpracování v Sql dotazu.
- **vypisPolozka**  
Soubory: vypisPolozka.cpp, vypisPolozka.h, vypisPolozka.ui  
Objekt **vypisPolozka** je jednotlivým dílkem celkového výpisu nalezených záznamů. Upravuje data přijatá z databáze do podoby, ve které je uvidí uživatel. Umožňuje přechod na podrobný výpis položky nebo její editaci.
- **vypisSeznam**  
Soubory: vypisSeznam.cpp, vypisSeznam.h, vypisSeznam.ui  
Nejdůležitější proměnná tohoto objektu je **QList<VypisPolozka\*> seznamPolozek**, která uchovává seznam jednotlivých instancí třídy **vypisPolozka**. Pomocí této proměnné vytváří objekt samotný seznam výpisu položek.

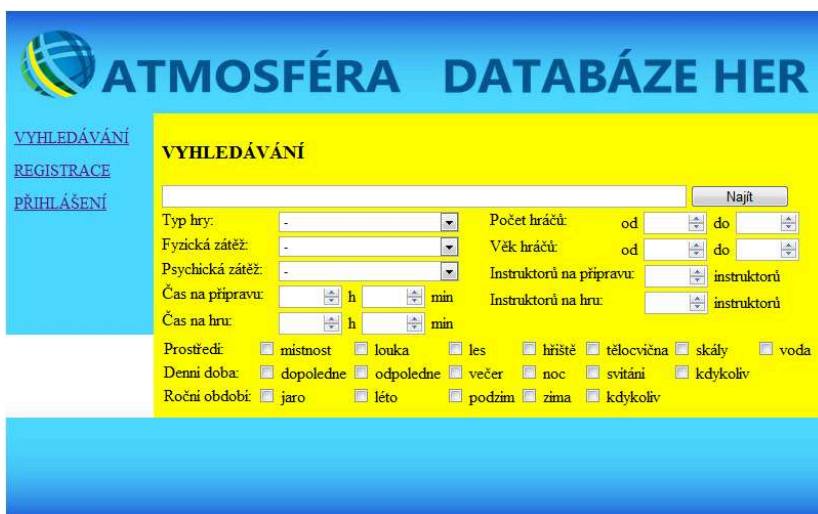
## 6. Webové stránky

### 6.1. Uživatelská část

Na následujících stranách je přehled jednotlivých webových stránek, se kterými se může uživatel setkat. Jejich funkcionalita je stejná jako u jednotlivých oken programu, které jsou popsány v 5. kapitole. Testování při vývoji bylo prováděno v prohlížeči Google Chrome, dále jsou plně podporovány prohlížeče Opera a Safari. Prohlížeče Internet Explorer a Firefox nepodporují některé html značky, které jsou použity ve formuláři rozšířeného vyhledávání. Proto v těchto prohlížečích tento formulář nevypadá tak jak má.



Obrázek 10. Úvodní stránka webu – jednoduché vyhledávání



Obrázek 11. Rozšířené vyhledávání na webu

**ATMOSFÉRA DATABÁZE HER**

[VYHLEDÁVÁNÍ](#)  
[REGISTRACE](#)  
[PŘIHLÁŠENÍ](#)

**REGISTRACE**

Jméno

Příjmení

Přihlašovací jméno

E-mail  
@

Heslo

Heslo znovu

Obrázek 12. Registrace na webu

**ATMOSFÉRA DATABÁZE HER**

[VYHLEDÁVÁNÍ](#)  
[REGISTRACE](#)  
[PŘIHLÁŠENÍ](#)

**PŘIHLÁŠENÍ**

Přihlašovací jméno

Heslo


Obrázek 13. Přihlášení na webu



Obrázek 14. Moje údaje na webu – prohlížení



Obrázek 15. Moje údaje na webu – úprava hodnot



# ATMOSFÉRA DATABÁZE HER

[VYHLEDÁVÁNÍ](#)  
[MOJE ÚDAJE](#)  
[MOJE HRY](#)  
[VYTVOŘIT HRU](#)  
[REGISTRACE](#)  
[ODHLÁŠENÍ](#)

## VYTVOŘIT HRU

Název hry:  Hru vložil:

Cíl hry:  Typ hry:

Fyzická zátěž:  Počet hráčů: od  do

Psychická zátěž:  Věk hráčů: od  do

Čas na přípravu:  h  min Instruktorů na přípravu:  instruktorů

Čas na hru:  h  min Instruktorů na hru:  instruktorů

Prostředí:  místnost  louka  les  hřiště  tělocvična  skály  voda

Poznámka k prostředí:

Denní doba:  dopoledne  odpoledne  večer  noc  svitání  kdykoliv

Roční období:  jaro  léto  podzim  zima  kdykoliv

Charakteristika:

Motivace:

Pravidla:

Materiál:

Metodika:

Obrázek 16. Formulář editace hry na webu

The screenshot shows a web application interface. At the top, there is a blue header with a logo on the left and the text "ATMOSFÉRA DATABÁZE HER" in white. Below the header, on the left side, there is a vertical menu with blue links: "VYHLEDÁVÁNÍ", "MOJE ÚDAJE", "MOJE HRY", "VYTVOŘIT HRU", "REGISTRACE", and "ODHLÁŠENÍ". The main content area has a yellow background and is titled "MOJE HRY". It contains a list of seven game entries, each with a text label and a "Upravit hru" button to its right. The entries are: "Název hry: nový název 1.hry", "Název hry: testovací hra", "Název hry: Zatím poslední testovací záznam", "Název hry: poslední test 14.7.", "Název hry: testovací hra kvůli webu", "Název hry: hra z webu", and "Název hry: test common".

MOJE HRY	
Název hry: nový název 1.hry	<input type="button" value="Upravit hru"/>
Název hry: testovací hra	<input type="button" value="Upravit hru"/>
Název hry: Zatím poslední testovací záznam	<input type="button" value="Upravit hru"/>
Název hry: poslední test 14.7.	<input type="button" value="Upravit hru"/>
Název hry: testovací hra kvůli webu	<input type="button" value="Upravit hru"/>
Název hry: hra z webu	<input type="button" value="Upravit hru"/>
Název hry: test common	<input type="button" value="Upravit hru"/>

Obrázek 17. Výpis her přihlášeného uživatele na webu



## 6.2. Programátorská část

Strukturu webových stránek můžeme rozdělit na grafickou a obsahovou část. Obě tyto části spojuje soubor **index.php**, který nahrává grafické části, a kterému je pomocí proměnné předávána obsahová část stránky. Obě tyto části využívají **css** záznamy ze souboru **screen.css**.

### 6.2.1. Index a grafická část

V souboru `index.php` nahráváme pomocí příkazu *include* [9] skripty, které vytváří hlavní rámec stránek. Díky tomu máme stále stejný vzhled stránek. Drobnou výjimku tvoří část **menu**, která se mění podle toho, zda-li je uživatel přihlášený nebo ne. V souboru `index.php` je proměnná **\$filename**, pomocí které se zobrazuje obsah jednotlivých stránek.

#### Seznam nahrávaných skriptů:

- `common.php`  
Skript, který obsahuje funkce používané na webových stránkách.
- `start_session.php`  
Skript, který zahajuje uživatelskou relaci (session). Ta umožňuje odlišit jednotlivé uživatele a slouží k zaznamenání informací o přihlášení. [10]
- `hlavicka.php`  
Skript bez php kódu. Reprezentuje html hlavičku.
- `zahlavi.php`  
Skript bez php kódu. Nahrává grafické části záhlaví stránek.
- `connect.php`  
Skript, který realizuje připojení do databáze.
- `logout.php`  
Skript, který realizuje odhlášení uživatele z databáze.
- `login.php`  
Skript, který realizuje přihlášení uživatele k databázi.
- `menu.php`  
Skript, který zobrazuje v levé části stránek grafický pás s **menu**. Menu obsahuje položky **Vyhledávání**, **Registrace** a **Přihlášení**. Pokud je uživatel přihlášený rozšíří se menu o položky **Moje údaje**, **Moje hry**, **Vytvořit hru** a položka **Přihlášení** se změní na **Odhlášení**.
- `zapati.php`  
Skript bez php kódu, který nahrává grafické části zápatí stránek.

### 6.2.2. Obsahová část

Obsah stránky vždy vytváří série skriptů – *jmeno.php*, který ukládá název skriptu *jmeno\_obsah.php* do proměnné **\$filename**, pomocí které dojde k zobrazení obsahu požadované stránky. Na některých stránkách může uživatel provádět nějakou akci. Formulář ho z takové stránky odešle na skript *jmeno\_action.php*, který ukládá název dalšího skriptu *jmeno\_action\_obsah.php* do proměnné **\$filename**, pomocí které dojde k zobrazení výsledku dané akce.

#### Seznam skriptů jednotlivých stránek:

- *moje\_hry.php*, *moje\_hry\_obsah.php*  
Skripty realizují zobrazení **her** aktuálně přihlášeného uživatele.
- *moje\_udaje.php*, *moje\_udaje\_obsah.php*  
Skripty realizují zobrazení **údajů** aktuálně přihlášeného uživatele.
- *odhlaseni.php*, *odhlaseni\_obsah.php*  
Skripty zobrazí zprávu o odhlášení uživatele.
- *podrobny\_vypis.php*, *podrobny\_vypis\_obsah.php*  
Skripty realizují podrobný výpis položky z databáze po přijetí odpovědi na Sql dotaz.
- *prihlaseni.php*, *prihlaseni\_obsah.php*, *prihlaseni\_action.php*,  
*prihlaseni\_action\_obsah.php*  
Skripty realizují přihlašovací formulář do databáze her a zobrazení zprávy v závislosti na úspěchu provedené akce.
- *registrace.php*, *registrace\_obsah.php*, *registrace\_action.php*,  
*registrace\_action\_obsah.php*  
Skripty realizují registraci nového uživatele do databáze her pomocí Sql dotazu.
- *ulozit\_pdf.php*, *ulozit\_pdf\_obsah.php*  
Skripty realizují tvorbu pdf souboru s podrobným výpisem položky databáze. Pro tvorbu pdf souboru je využívána php třída mPDF, která je šiřitelná pod GPL licencí.
- *upravit\_hru.php*, *upravit\_hru\_obsah.php*, *upravit\_hru\_action.php*,  
*upravit\_hru\_action\_obsah.php*  
Skripty zobrazí formulář s údaji z databáze o konkrétním záznamu databáze a umožní provést změnu záznamu pomocí Sql dotazu.
- *upravit\_udaje.php*, *upravit\_udaje\_obsah.php*, *upravit\_udaje\_action.php*,  
*upravit\_udaje\_action\_obsah.php*  
Skripty zobrazí údaje aktuálně přihlášeného uživatele a umožní jejich úpravu.

- vyhledavani.php, vyhledavani\_obsah.php, vyhledavani\_action.php, vyhledavani\_action\_obsah.php  
Skripty zobrazí formulář pro jednoduché vyhledávání a po provedení Sql dotazu zobrazí nalezené záznamy.
- vyhledavaniR.php, vyhledavaniR\_obsah.php, vyhledavaniR\_action.php, vyhledavaniR\_action\_obsah.php  
Skripty zobrazí formulář pro rozšířené vyhledávání a po provedení Sql dotazu zobrazí nalezené záznamy.
- vytvorit\_hru.php, vytvorit\_hru\_obsah.php, vytvorit\_hru\_action.php, vytvorit\_hru\_action\_obsah.php  
Skripty zobrazí prázdný formulář pro tvorbu nového záznamu v databázi a pomocí Sql dotazu provedou jeho vložení.

## Závěr

Výsledkem této práce je databáze her se dvěma přístupovými rozhraními, počítačovou aplikací a webovými stránkami. Počítačovou aplikaci je možné instalovat pomocí instalačního souboru. Podařilo se mi realizovat následující funkcionality – jednoduché a rozšířené vyhledávání záznamů v databázi, registrace nového uživatele do databáze, přihlášení uživatele do databáze, tvorba nových a úprava stávajících záznamů databáze, zobrazení a úprava osobních údajů přihlášeného uživatele. Stejně funkcionality nabízí i webové stránky.

Proti původnímu zadání jsem nerealizoval možnost off-line vyhledávání v databázi. Pro tuto funkcionality jsem chtěl využít databázový formát SQLite, který není přímo kompatibilní s MySQL. Také neexistuje 100% funkční skript, který by realizoval převod mezi těmito systémy. Do budoucna chci tento problém vyřešit i z toho důvodu, že počítačová aplikace je na tuto možnost prohlížení databáze připravena. Dále jsem nerealizoval možnost stažení databáze do programu, což souvisí s výše zmíněným problémem a také zpětné nahraní databáze na server. Myslím si, že webové rozhraní phpMyAdmin je naprosto vyhovující pro správu databáze a provádění její zálohy a obnovy.

Grafický vzhled obou přístupových rozhraní je nyní, dle mého názoru, vyhovující. Jeho případné změny budu provádět až po delším zavedení systému do praxe. Webové stránky jsou plně funkční na webových prohlížečích Google Chrome, Opera, Safari. Prohlížeče Internet Explorer a Firefox nepodporují některé části formuláře pro rozšířené vyhledávání. Mimo toto omezení v nich stránky fungují.

Přínosem pro mou osobu byla práce s Sql dotazy, kterou jsem nikdy v takovémto rozsahu nedělal. Při tvorbě počítačové aplikace se mi podařilo realizovat funkční oblast okna s posuvníky, což mi dříve dělalo problémy. Tvorba webových stránek byl v podstatě můj první pokus o psaní stránek za pomoci php s přístupem k databázi.

Pevně věřím, že se tato databáze v občanském sdružení Atmosféra uchytlí a bude využívána a rozšiřována. V budoucnu si dokáží představit i mobilní aplikaci pro chytré telefony, která bude mít minimálně možnost vyhledávání v databázi. Vyhledávání by mohlo být upraveno ze současného přímého rozhodování na fuzzy Sql.

## Reference

- [1] phpMyAdmin devel team. (2012). *About*.  
[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php).
- [2] cplusplus.com. (2012). *A brief description – C++ Information*.  
<http://www.cplusplus.com/info/description/>.
- [3] Nokia Corporation. (2012). *Qt – Cross-platform application and UI framework*.  
<http://qt.nokia.com/>.
- [4] Nokia Corporation. (2012). *Qt Creator IDE and tools*.  
<http://qt.nokia.com/products/developer-tools>.
- [5] W3C. (2012). *HTML CSS – W3C*.  
<http://www.w3.org/standards/webdesign/htmlcss>.
- [6] The PHP group. (2012). *PHP: Hypertext Preprocessor*.  
<http://cz.php.net/>.
- [7] Fiala, Jan. (2012). *PSPad – free unicode vývojářský editor určený pro HTML, PHP, XHTML, JavaScript, ASP, Perl, C a mnoho dalších jazyků, obsahuje HEXa editor*.  
<http://www.pspad.com/cz/>.
- [8] Hrkal, Jan & Hanuš, Radek. *Zlatý fond her II*. Portál, Praha, 2007.
- [9] Kosek, Jiří. *PHP – tvorba interaktivních internetových aplikací*. Grada Publishing, spol. s r.o.. Praha. 1999.
- [10] Naramore, Elizabeth. Gerner, Jason. Scouarnec, Yann Le. Stolz, Jeremy. Glass, Michael K. *Vytváříme webové aplikace v PHP5, MySQL a Apache*. Computer Press, a.s.. Brno. 2006.

# A. Podrobná programátorská příručka k počítačové aplikaci

## A.1. Main

**Soubory:** main.cpp

**Include:** mainwindow.h, QApplication

**Proměnné:**

- QApplication app – proměnná pro spuštění a řízení celé aplikace
- QString c – proměnná pro uchování adresáře, ve kterém je aplikace nainstalována
- MainWindow mw – proměnná pro instanci hlavního okna

**Funkce:**

Tento soubor obsahuje jedinou funkci **main**. Její návratová hodnota je typu **int**. V ní jsou deklarovány výše zmíněné proměnné a je vykonán příkaz *exec* nad proměnnou **app** pro spuštění aplikace.

## A.2. Main Window

**Soubory:** mainwindow.cpp, mainwindow.h, mainwindow.ui

**Include:** iostream, stdio.h, QtGui, QMainWindow, QVBoxLayout, QSql, QList, QScrollArea, QLabel, QFile, QString, QTextStream, QLabel, registrace.h, prihlaseni.h, editaceHry.h, vyhledavani.h, vypisPolozka.h, vypisSeznam.h, mojeUdaje.h, nastaveni.h, textKoule.h, podrobnyVypis.h

**Proměnné:**

- Ui::MainWindow \*ui, Registrace \*registrace, Prihlaseni \*prihlaseni, EditaceHry \*editaceHry, Vyhledavani \*vyhledavani, vyhledavani, VypisSeznam\*vypisSeznam, MojeUdaje \*mojeUdaje, Nastaveni \*nastaveni, TextKoule \*textKoule, PodrobnyVypis \*podrobnyVypis, QScrollArea \*scrollArea, QLabel \*koule – proměnné pro uchování instancí jednotlivých tříd
- QSqlDatabase db – proměnná pro práci s databází
- QString cestaSouboru – proměnná pro uchování adresáře, ve kterém je aplikace nainstalována

## Funkce:

- `MainWindow(QString cesta, QWidget *parent = 0)` – konstruktor třídy `MainWindow`  
V konstruktoru je nastavena proměnná `ui` pro správu grafických objektů. Dále je nastavena proměnná `cestaSouboru` z hlavičky funkce. Lokálně je vytvořen `mainWidget` (`QWidget`), který je nastaven jako `CentralWidget` v `MainWindow`. Rozložení `mainWidgetu` `layout1` (`QVBoxLayout`) a `scrollArea` (`QScrollArea`), která má za rodiče `mainWidget` a je tedy jeho hlavní výplň. Do této `scrollArea` jsou potom vkládány aktivní widgety k zobrazení v hlavním okně. Jsou zde inicializovány proměnné instancí jednotlivých tříd a jsou všechny skryty příkazem `hide`. Probíhá volání funkce `nacteniPocatecnichDat`, napojení signálů a slotů, otevření databáze pomocí proměnné `db` a nastavení `textKoule` jako aktuální výstup do hlavního okna.
- `~MainWindow()` – destruktory třídy `MainWindow`
- `void nacteniPocatecnichDat()` – funkce realizující načtení hodnot pro přihlášení do databáze a přihlášení uživatele

## Signály:

- `void mojeUdajeHide()`  
Vyvolá spuštění slotu `restoreSlot` z `mojeUdaje`.
- `void mojeHry_sig(QSqlQuery, int)`  
Odešle výsledek `Sql` dotazu pro vyhledávání do slotu `vytvorSeznam` z `vypisSeznam` a hodnotu jaký typ seznamu se má vytvořit (seznam moje hry).
- `void logovaciData(QString, QString)`  
Odešle přihlašovací údaje uživatele do slotu `prijem_logData` z `prihlaseni`.
- `void nastavitHodnoty_editaceHry(QSqlQuery query)`  
Odešle záznam z `Sql` dotazu do slotu `nastavHodnoty` z `editaceHry`.
- `void nastavitHodnoty_podrobnyVypis(QSqlQuery query)`  
Odešle záznam z `Sql` dotazu do slotu `nastavHodnoty` z `podrobnyVypis`.
- `void vyhledavani_sig(QSqlQuery, int)`  
Odešle výsledek `Sql` dotazu pro vyhledávání do slotu `vytvorSeznam` z `vypisSeznam` a hodnotu jaký typ seznamu se má vytvořit (seznam nalezených záznamů her po vyhledávání).

- void posliText\_sig(QString msg)  
Odešle text zprávy do slotu **zadejText** z **textKoule**.
- void connectDB\_data(QString HostName, QString JmenoDB, QString UzivatelskeJmeno, QString HesloDB)  
Odešle údaje pro přihlášení do databáze do slotu **connectDB** z **mainwindow** a do slotu **prijem\_connectDB\_data** z **nastaveni**.
- void pripojeniInner\_sig()  
Vyvolá spuštění slotu **pripojeniInner** z **nastaveni**.
- void pripojitDB\_clicked\_sig()  
Vyvolá spuštění slotu **pripojitDB\_clicked** z **nastaveni**.
- void restoreMojeUdaje()  
Vyvolá spuštění slotu **restoreSlot** z **mojeUdaje**.

#### Sloty:

- void connectDB(QString HostName, QString JmenoDB, QString UzivatelskeJmeno, QString HesloDB)  
Realizuje připojení aplikace k databázi.
- void actionRegistrace\_slot()  
Nastaví widget **registrace**, jako widget, který se zobrazí v hlavním okně.
- void actionPrihlasit\_se\_slot()  
Nastaví widget **prihlaseni**, jako widget, který se zobrazí v hlavním okně a upraví jeho pozici.
- void actionVyhledavani\_slot()  
Nastaví widget **vyhledavani**, jako widget, který se zobrazí v hlavním okně.
- void actionVytvoritHru\_slot()  
Nastaví widget **editaceHry**, jako widget, který se zobrazí v hlavním okně.
- void actionMojeUdaje\_slot()  
Nastaví widget **mojeUdaje**, jako widget, který se zobrazí v hlavním okně.
- void actionNastaveni\_slot()  
Nastaví widget **nastaveni**, jako widget, který se zobrazí v hlavním okně.
- void actionMojeHry\_slot()  
Nastaví widget **vypisSeznam**, jako widget, který se zobrazí v hlavním okně. A pošle signál **sqlMojeHry**.



- void actionOdhlasit\_se\_slot()  
Pošle zprávu na hlavní okno a nastaví proměnnou **loggedIn** v **prihlaseni** na *false*.
- void uloz\_prihlasovaciUdaje(QString login, QString heslo)  
Uloží přihlašovací údaje uživatele do textového souboru.
- void uloz\_udajeDB(QString HostName, QString JmenoDB, QString UzivatskeJmeno, QString HesloDB)  
Uloží přihlašovací údaje pro přihlášení do databáze do textového souboru.
- void sqlPosliUzivatel(QString email, QString jmeno, QString prijmeni, QString prezdivka\_logIn, QString heslo)  
Provede Sql příkaz pro odeslání údajů nového uživatele do databáze.
- void sqlPosliHru(QString nazevHry, QString hruVlozil, QString cilHry, QString typHry, int fyzZatez, int psychZatez, int casH\_Priprava, int casMin\_Priprava, int casH\_Hra, int casMin\_Hra, int pocetHracu\_od, int pocetHracu\_do, int vekHracu\_od, int vekHracu\_do, int insPriprava, int insHra, QString prostredi, QString poznamkaProstredi, QString deniDoba, QString rocniObdobi, QString charakteristika, QString motivace, QString pravidla, QString metodika, QString material)  
Provede Sql příkaz pro odeslání údajů nové hry do databáze. O úspěchu či neúspěchu operace informuje uživatele pomocí signálu **posliText\_sig**.
- void sqlMojeHry(int typPolozky)  
Provede Sql příkaz pro načtení her z databáze, které odpovídají přihlašovacímu jménu přihlášeného uživatele. V případě úspěchu vyšle signál **mojeHry\_sig**.
- void sqlPodrobnyVytah(int typ, QString id\_hry)  
Provede Sql dotaz pro načtení jednoho konkrétního záznamu z databáze. V případě úspěchu vyšle signál **nastavitHodnoty\_podrobnyVypis**, ve kterém výsledek odešle k dalšímu zpracování.
- void sqlUpdateHry(QString id\_hry, QString nazevHry, QString hruVlozil, QString cilHry, QString typHry, int fyzZatez, int psychZatez, int casH\_Priprava, int casMin\_Priprava, int casH\_Hra, int casMin\_Hra, int pocetHracu\_od, int pocetHracu\_do, int vekHracu\_od, int vekHracu\_do, int insPriprava, int insHra, QString prostredi, QString poznamkaProstredi, QString deniDoba, QString rocniObdobi, QString charakteristika, QString motivace, QString pravidla, QString metodika, QString material)  
Provede Sql příkaz pro odeslání upravených údajů hry do databáze. O úspěchu či neúspěchu operace informuje uživatele pomocí signálu **posliText\_sig**.

- void sqlVyhledavaniJednoduche(QString str)  
Provede Sql příkaz pro fulltextové vyhledávání v databázi. Při neúspěchu o tom informuje uživatele pomocí signálu **posliText\_sig**. V opačném případě nastaví widget **vypisSeznam**, jako widget, který se zobrazí v hlavním okně. A dále pomocí signálu **vyhledavani\_sig** odešle výsledek hledání k dalšímu zpracování.
- void sqlVyhledavaniRozsirene(QString hlavniRetezec, QString typHry, int fyzZatez, int psychZatez, int casH\_Priprava, int casMin\_Priprava, int casH\_Hra, int casMin\_Hra, int pocetHracu\_od, int pocetHracu\_do, int vekHracu\_od, int vekHracu\_do, int insPriprava, int insHra, QString pdr)  
Provede Sql příkaz pro rozšířené vyhledávání. Tento se skládá ze tří hlavních částí. V první se zpracovává stejný řetězec jako při jednoduchém vyhledávání. Ve druhé části se fulltextově prohledávají atributy *prostredi*, *denni\_doba*, *rocni\_obdobi*. A v poslední části se přímo porovnávají hodnoty z formuláře s hodnotami atributů v databázi. Kterákoli z těchto částí může v příkazu chybět pokud dané pole není vyplněno.
- void showPodrobnyVypis\_slot()  
Nastaví widget **podrobnyVypis**, jako widget, který se zobrazí v hlavním okně.
- void show\_textKoule()  
Nastaví widget **textKoule**, jako widget, který se zobrazí v hlavním okně.
- void act\_offline\_dbs(bool bl) Podle proměnné **bl** zaškrtně nebo odškrtně pole **on-line databáze** v roletce *Databáze*, a vyšle signál **pripojeniInner\_sig**.
- void act\_online\_dbs(bool bl)  
Podle proměnné **bl** zaškrtně nebo odškrtně pole **off-line databáze** v roletce *Databáze*, a vyšle signál **pripojitDB\_clicked\_sig**.

### A.3. Editace hry

**Soubory:** eidtaceHry.cpp, editaceHry.h, editaceHry.ui

**Include:** QWidget, QSql, QList, QStringList

**Proměnné:**

- Ui::EditaceHry \*ui – proměnná pro přístup ke grafickým objektům
- bool updatePolozky – proměnná typu bool, pro rozhodování, jaký signál odeslat
- QString ID – proměnná pro uchování **ID\_hry** z výsledku Sql dotazu
- QList<QString> seznamTypu – proměnná pro uchování seznamu typů her
- QList<QString> seznamProstredi – proměnná pro uchování seznamu typů prostředí
- QList<QString> seznamDenniDoba – proměnná pro uchování seznamu typů denní doby
- QList<QString> seznamRocniObdobi – proměnná pro uchování seznamu typů ročního období

**Funkce:**

- EditaceHry(QWidget \*parent = 0) – konstruktortřídy EditaceHry  
V konstruktoru je nastavena proměnná **ui** pro správu grafických objektů. Dále jsou volány funkce *vytvor\_seznamTypu*, *vytvor\_seznamProstredi*, *vytvor\_seznamDenniDoba*, *vytvor\_seznamRocniObdobi*. Je nastavena proměnná **updatePolozky** na *false* a je provedeno napojení místních signálů a slotů.
- ~EditaceHry() – destruktork třídy EditaceHry
- void vytvor\_seznamTypu() – funkce, která vytvoří seznam v proměnné **seznamTypu**
- void vytvor\_seznamProstredi() – funkce, která vytvoří seznam v proměnné **seznamProstredi**
- void vytvor\_seznamDenniDoba() – funkce, která vytvoří seznam v proměnné **seznamDenniDoba**
- void vytvor\_seznamRocniObdobi() – funkce, která vytvoří seznam v proměnné **seznamRocniObdobi**

## Signály:

- void odeslatHru\_sig(QString nazevHry, QString hruVlozil, QString cilHry, QString typHry, int fyzZatez, int psychZatez, int casH\_Priprava, int casMin\_Priprava, int casH\_Hra, int casMin\_Hra, int pocetHracu\_od, int pocetHracu\_do, int vekHracu\_od, int vekHracu\_do, int insPriprava, int insHra, QString prostredi, QString poznamkaProstredi, QString deniDoba, QString rocniObdobi, QString charakteristika, QString motivace, QString pravidla, QString metodika, QString material)  
Odešle data z formuláře pro tvorbu záznamu hry k dalšímu zpracování v Sql příkazu INSERT do slotu **sqlPosliHru** v **mainwindow**.
- void updateHry\_sig(QString id\_hry, QString nazevHry, QString hruVlozil, QString cilHry, QString typHry, int fyzZatez, int psychZatez, int casH\_Priprava, int casMin\_Priprava, int casH\_Hra, int casMin\_Hra, int pocetHracu\_od, int pocetHracu\_do, int vekHracu\_od, int vekHracu\_do, int insPriprava, int insHra, QString prostredi, QString poznamkaProstredi, QString deniDoba, QString rocniObdobi, QString charakteristika, QString motivace, QString pravidla, QString metodika, QString material)  
Odešle data z formuláře pro tvorbu záznamu hry k dalšímu zpracování v Sql příkazu UPDATE do slotu **sqlUpdateHry** v **mainwindow**.
- void show\_upravaHry()  
Vyvolá spuštění slotu **actionVytvoritHru\_slot** z **mainwindow**.

## Sloty:

- void OdeslatHru()  
Provede odeslání dat z formuláře na správný Sql příkaz INSERT v případě vkládání nového záznamu nebo na příkaz UPDATE v případě úpravy záznamu, který již existuje. Rozhodující je proměnná **updatePolozky**. Podle ní se vyšle signál **odeslatHru\_sig** nebo signál **updateHry\_sig**.
- QString prostrediToQstr()  
Vytvoří řetězec QString podle zaškrtnutých políček ve formuláři, které odpovídají prostředí a vrátí jej.
- QString deniDobaToQstr()  
Vytvoří řetězec QString podle zaškrtnutých políček ve formuláři, které odpovídají denní době a vrátí jej.
- QString rocniObdobiToQstr()  
Vytvoří řetězec QString podle zaškrtnutých políček ve formuláři, které odpovídají ročnímu období a vrátí jej.

- void nastavAutora(QString autor)  
Nastaví přihlašovací jméno aktuálně přihlášeného hráče do formuláře editace hry.
- void nastavHodnoty(QSqlQuery query)  
Nastaví proměnnou **updatePolozky** na *true*. Nastaví všechny hodnoty ve formuláři podle doručeného záznamu z databáze pro následnou úpravu záznamu hry.
- void nastavProstredi(QString str)  
Pomocná funkce, která z řetězce nastaví hodnoty prostředí ve formuláři.
- void nastavDenniDobu(QString str)  
Pomocná funkce, která z řetězce nastaví hodnoty denní doby ve formuláři.
- void nastavRocniObdobi(QString str)  
Pomocná funkce, která z řetězce nastaví hodnoty ročních období ve formuláři.
- void vycistitFormular()  
Nastaví veškeré hodnoty ve formuláři na výchozí.
- int vratIndexTypu(QString str)  
Funkce vrací jako svou návratovou hodnotu pozici daného typu hry v proměnné **str** v **seznamTypu**.
- int vratIndexProstredi(QString str)  
Funkce vrací jako svou návratovou hodnotu pozici daného prostředí v proměnné **str** v **seznamProstedi**.
- int vratIndexDenniDoby(QString str)  
Funkce vrací jako svou návratovou hodnotu pozici dané denní doby v proměnné **str** v **seznamDenniDoba**.
- int vratIndexRocnihoObdobi(QString str)  
Funkce vrací jako svou návratovou hodnotu pozici daného ročního období v proměnné **str** v **seznamRocniObdobi**.

## A.4. Moje údaje

**Soubory:** mojeUdaje.cpp, mojeUdaje.h, mojeUdaje.ui

**Include:** QWidget, QSql

**Proměnné:**

- Ui::MojeUdaje \*ui – proměnná pro přístup ke grafickým objektům
- QString actJmeno – proměnná pro uložení jména, aktuálně přihlášeného uživatele
- QString actPrijmeni – proměnná pro uložení příjmení, aktuálně přihlášeného uživatele
- QString actEmail – proměnná pro uložení e-mailu, aktuálně přihlášeného uživatele
- QString actLogin – proměnná pro uložení přihlašovacího jména, aktuálně přihlášeného uživatele
- QString actHeslo – proměnná pro uložení hesla, aktuálně přihlášeného uživatele

**Funkce:**

- MojeUdaje(QWidget \*parent = 0) – konstruktor třídy MojeUdaje. Napojení signálů a slotů dané třídy. Skrytí editovatelných částí formuláře.
- QString getActLogin() – vrátí přihlašovací jméno aktuálně přihlášeného uživatele.

**Signály:**

- void restore()  
Vyvolá spuštění slotu **restoreSlot** z **mojeUdaje**.
- void posliText\_sig(QString msg)  
Odešle text zprávy do slotu **zadejText** z **textKoule**.

**Sloty:**

- void zmenaHesla\_t()  
Provede zviditelnění editovatelných částí pro změnu hesla ve formuláři.
- void upravitClicked()  
Provede zviditelnění editovatelných částí osobních údajů přihlášeného uživatele ve formuláři.

- void zrusitClicked()  
Vymaže jakákoli vyplněná pole formuláře a skryje editovatelné části.
- void okClicked()  
Provede z vyplněných údajů ve formuláři Sql příkaz UPDATE a podle úspěchu či neúspěchu operace vyšle patřičný signál se zprávou pro uživatele.
- void restoreSlot()  
Zavolá slot **zrusitClicked**.
- void actData\_slot(QString actJ, QString actP, QString actE, QString actL, QString actH);  
Nastaví *act* proměnné a zavolá slot **nastavFormular**.
- void nastavFormular()  
Nastaví osobní údaje přihlášeného uživatele z *act* proměnných do formuláře.

## A.5. Nastavení

**Soubory:** nastaveni.cpp, nastaveni.h, nastaveni.ui

**Include:** QWidget, QString, QFileDialog

**Proměnné:**

- Ui::Nastaveni \*ui – proměnná pro přístup ke grafickým objektům
- QString cestaSouboru – proměnná pro uchování adresáře, ve kterém je aplikace nainstalována

**Funkce:**

- Nastaveni(QString cs, QWidget \*parent = 0) – konstruktor třídy Nastaveni  
V konstruktoru je nastavena proměnná **ui** pro správu grafických objektů. Je nastavena proměnná **cestaSouboru** a jsou napojeny místní signály a sloty.
- ~Nastaveni() – destruktork třídy MainWindow

**Signály:**

- void ulozUdajeDB\_sig(QString HostName, QString JmenoDB, QString UzivatelskeJmeno, QString HesloDB)  
Vyvolá spuštění slotu **uloz\_udajeDB** z **mainwindow**.
- void pripojitDB\_sig(QString HostName = "", QString JmenoDB = "", QString UzivatelskeJmeno = "", QString HesloDB = "")  
Vyvolá spuštění slotu **connectDB** z **mainwindow**.

### Sloty:

- void cestaDB()  
Provede vyvolání dialogu pro otevření souboru a umožní tak nastavení souboru pro off-line databázi.  
*Zatím nevyužito.*
- void příjem\_connectDB\_data(QString HostName, QString JmenoDB, QString UzivatelskeJmeno, QString HesloDB)  
Provede vyplnění formuláře, údaji z hlavičky, pro připojení do online databáze.
- void ulozUdajeDB\_clicked()  
Vyšle signál pro uložení dat z formuláře pro připojení do online databáze.
- void pripojitDB\_clicked()  
Vyšle signál s údaji pro připojení do online databáze.
- void pripojeniInner()  
Vyšle signál pro připojení k off-line databázi. Pokud není vyplněna cesta souboru vyvolá také dialogové okno pro otevření souboru pomocí **cestaDB**.

## A.6. Podrobný výpis

**Soubory:** podrobnyVypis.cpp, podrobnyVypis.h, podrobnyVypis.ui

**Include:** QWidget, QSql, QStringList, QTextDocument, QPrinter, QFileDialog

**Proměnné:**

- Ui::PodrobnyVypis \*ui – proměnná pro přístup ke grafickým objektům
- QStringList seznamZatezi – seznam řetězců textu reprezentující zátěž aktivity
- QString cestaSouboru – proměnná pro uchování adresáře, ve kterém je aplikace nainstalována

**Funkce:**

- PodrobnyVypis(QString cesta, QWidget \*parent = 0) – konstruktor třídy PodrobnyVypis  
V konstruktoru je nastavena proměnná **ui** pro správu grafických objektů. Je nastavena proměnná **cestaSouboru**, je volána funkce **vytvor\_seznamZatezi** a jsou napojeny místní signály a sloty.
- ~PodrobnyVypis() – destruktork třídy PodrobnyVypis



- void vytvor\_seznamZatezi()  
Naplní proměnnou **seznamZatezi** příslušnými řetězci.
- QString pripravHTML()  
Vrací řetězec, který obsahuje hodnoty z formuláře a html značky, který je vhodný pro výstup do pdf souboru.
- QString vratMezery(int kolik)  
Pomocná funkce která vrací řetězec nezlomitelných html mezer.

#### Signály:

- void showPodrobnyVypis()  
Vyvolá spuštění slotu **showPodrobnyVypis\_slot** z **mainwindow**.

#### Sloty:

- void nastavHodnoty(QSqlQuery query)  
Provede nastavení celého formuláře z odpovědí na Sql dotaz.
- void tiskPDF()  
Provede vytvoření pdf souboru z dat ve formuláři.

## A.7. Přihlášení

**Soubory:** prihlaseni.cpp, prihlaseni.h, prihlaseni.ui

**Include:** QWidget, QSql

**Proměnné:**

- Ui::Prihlaseni \*ui – proměnná pro přístup ke grafickým objektům
- QString actJmeno – proměnná pro uložení jména, aktuálně přihlášeného uživatele
- QString actPrijmeni – proměnná pro uložení příjmení, aktuálně přihlášeného uživatele
- QString actEmail – proměnná pro uložení e-mailu, aktuálně přihlášeného uživatele
- QString actLogin – proměnná pro uložení přihlašovacího jména, aktuálně přihlášeného uživatele
- QString actHeslo – proměnná pro uložení hesla, aktuálně přihlášeného uživatele
- bool loggedIn – proměnná, která zaznamenává, zda-li je uživatel přihlášený

### Funkce:

- `Prihlaseni(QWidget *parent = 0)` – konstruktor třídy `Prihlaseni`  
V konstruktoru je nastavena proměnná **ui** pro správu grafických objektů.  
Je nastavena proměnná **loggedIn** na *false* a jsou napojeny místní signály a sloty.

### Signály:

- `void actData(QString, QString, QString, QString, QString)`  
Odešle přihlašovací údaje uživatele do slotu **actData\_slot** a do stejnojmenného slotu v **mojeUdaje**.
- `void posliAutora(QString)`  
Odešle přihlašovací jméno uživatele do slotu **nastavAutora** v **editaceHry**.
- `void logIn(bool)`  
Vyšle signál a podle nastavení proměnné **loggedIn** dojde k zviditelnění nebo zmizení položek v menu (Odhlásit se, Moje hry, Moje údaje, Vytvořit hru a Přihlásit se).
- `void pamatujPrihlaseni(QString, QString)`  
Pošle přihlašovací údaje do slotu **uloz\_prihlasovaciUdaje** z **mainwindow**.
- `void posliText_sig(QString msg)`  
Odešle text zprávy do slotu **zadejText** z **textKoule**.

### Sloty:

- `void prihlaseniClicked()`  
Provede přihlášení uživatele k databázi. Pomocí `Sql` příkazu porovná zadaná data uživatelem s daty, které jsou obsaženy v databázi. A o výsledky informuje uživatele zprávou.
- `void zrusitClicked()`  
Vyčistí data ve formuláři.
- `void actData_slot(QString actJ, QString actP, QString actE, QString actL, QString actH)`  
Nastaví proměnné **actJmeno**, **actPrijmeni**, **actEmail**, **actLogin**, **actHeslo**.
- `void prijem_logData(QString login, QString heslo)`  
Nastaví ve formuláři přihlašovací jméno a heslo z přijatých dat.

## A.8. Registrace

**Soubory:** registrace.cpp, registrace.h, registrace.ui

**Include:** QWidget, QLabel, QDebug

**Proměnné:**

- Ui::Registrace \*ui – proměnná pro přístup ke grafickým objektům

**Funkce:**

- Registrace(QWidget \*parent = 0) – konstruktor třídy Registrace  
V konstruktoru je nastavena proměnná **ui** pro správu grafických objektů a jsou napojeny místní signály a sloty.

**Signály:**

- void posliText\_sig(QString msg)  
Odešle text zprávy do slotu **zadejText** z **textKoule**.
- void sig\_sqlPosliUzivatel(QString email, QString jmeno, QString prijmeni, QString prezdivka\_logIn, QString heslo)  
Odešle data potřebná pro registraci uživatele do slotu **sqlPosliUzivatel** z **mainwindow**.

**Sloty:**

- void prubehRegistrace()  
Tento slot je vyvolán po stisknutí tlačítka k odeslání registrace nového uživatele. Projde všechny pole formuláře a pokud jsou správně vyplněna odešle registraci pomocí signálu **sig\_sqlPosliUzivatele**. O problémech informuje uživatele zprávou.
- void vymazatFormularReg()  
Vymaže všechny pole formuláře.
- void dokonceniRegistrace(bool answ)  
Odešle zprávu uživateli o neúspěchu registrace a vymaže formulář.

## A.9. Text koule

**Soubory:** textKoule.cpp, textKoule.h, textKoule.ui

**Include:** QWidget, QString, QLabel

**Proměnné:**

- ui – proměnná pro přístup ke grafickým objektům

**Funkce:**

- Ui::TextKoule \*TextKoule(QWidget \*parent = 0) – konstruktor třídy textKoule  
V konstruktoru je nastavena proměnná **ui** pro správu grafických objektů a jsou napojeny místní signály a sloty.

**Signály:**

- void posliText\_sig(QString msg)  
Odešle text zprávy do slotu **zadejText** z **textKoule**.
- void showTextKoule()  
Vyvolá spuštění slotu **show\_textKoule** v **mainwindow**.

**Sloty:**

- void zadejText(QString msg)  
Nastaví text zprávy pro zobrazení uživateli a provede její zobrazení.

## A.10. Vyhledávání

**Soubory:** vyhledavani.cpp, vyhledavani.h, vyhledavani.ui

**Include:** QWidget, QString, QtGui

**Proměnné:**

- Ui::Vyhledavani \*ui – proměnná pro přístup ke grafickým objektům

**Funkce:**

- Vyhledavani(QWidget \*parent = 0) – konstruktor třídy Vyhledavani.  
V konstruktoru je nastavena proměnná **ui** pro správu grafických objektů a jsou napojeny místní signály a sloty.
- ~Vyhledavani() – destruktork třídy Vyhledavani.

**Signály:**

- void vyhledavaniJednoducheRetezec\_sig(QString str)  
Odešle vyhledávaný text do slotu **sqlVyhledavaniJednoduche** v **mainwindow**.

- void vyhledavaniRozsirene\_sig(QString hlavniRetezec, QString typHry, int fyzZatez, int psychZatez, int casH\_Priprava, int casMin\_Priprava, int casH\_Hra, int casMin\_Hra, int pocetHracu\_od, int pocetHracu\_do, int vekHracu\_od, int vekHracu\_do, int insPriprava, int insHra, QString pdr)  
Odešle data pro rozšířené vyhledávání do slotu **sqlVyhledavaniRozsirene** v **mainwindow**.

### Sloty:

- void actionVyhledavani\_sl(int index)  
Podle proměnné **index** provede změnu velikosti zobrazované oblasti formuláře pro vyhledávání.
- void vyhledavaniJednoduche\_clicked()  
Zavolá signál **vyhledavaniJednoducheRetezec\_sig** a vymaže obsah pole pro vyhledávání.
- void vyhledavaniRozsirene\_clicked()  
Odešle data z formuláře pro rozšířené vyhledávání pomocí signálu **vyhledavaniRozsirene\_sig** a volá slot **vycistiForm\_vyhledavaniRozsirene**.
- QString vytvorPDR\_retezec() Vrátí řetězec, který bude obsahovat *prostředí, denní dobu a roční období* podle zaškrtnutých políček.
- void vycistiForm\_vyhledavaniRozsirene()  
Vymaže veškerá vyplněná data ve formuláři rozšířeného vyhledávání.

## A.11. Výpis položka

**Soubory:** vypisPolozka.cpp, vypisPolozka.h, vypisPolozka.ui

**Include:** QWidget, QString, QList, QDebug

**Proměnné:**

- Ui::VypisPolozka \*ui – proměnná pro přístup ke grafickým objektům
- QList<QString> seznamTypu – proměnná pro uchování seznamu typů her
- QList<QString> seznamProstredi – proměnná pro uchování seznamu typů prostředí
- int typ – proměnná pro informaci, jaký typ položky se má vygenerovat
- QString ID – proměnná pro uchování **ID\_hry**.

### Funkce:

- `VypisPolozka(QWidget *parent = 0, int typ_prom = 0)` – konstruktor třídy `VypisPolzka`  
V konstruktoru je nastavena proměnná **ui** pro správu grafických objektů. Jsou volány funkce **vytvor\_seznamTypu** a **vytvor\_seznamProstredi**. Podle požadovaného typu je nastavena proměnná **typ** a jsou napojeny místní signály a sloty.
- `~VypisPolozka()` – destruktor třídy `vypisPolozka`
- `QString uprav_pocetHracu(QString str)`  
Vrací upravený řetězec **str** v podobě pro zobrazení uživateli.
- `QString uprav_vekHracu(QString str)`  
Vrací upravený řetězec **str** v podobě pro zobrazení uživateli.
- `QString uprav_casNaHru(QString str)`  
Vrací upravený řetězec **str** v podobě pro zobrazení uživateli.

### Signály:

- `void prechodNaPodrobny_sig(int typPolozky, QString id_hry)`  
Pomocí tohoto signálu přechází zobrazení seznamu položek na podrobný výpis jediné položky. Vyvolá spuštění slotu **prechodNaPodrobny** v **vypisSeznam**.

### Sloty:

- `void prechodNaPodrobny()`  
Tento slot je vyvolán kliknutím na tlačítko **Úprava hry** a vyvolá signál **prechodNaPodrobny\_sig**.

## A.12. Výpis seznam

**Soubory:** `vypisSeznam.cpp`, `vypisSeznam.h`, `vypisSeznam.ui`

**Include:** `QWidget`, `QList`, `QtSql`, `QDebug`, `vypisPolozka.h`

**Proměnné:**

- `Ui::VypisSeznam *ui` – proměnná pro přístup ke grafickým objektům
- `QList<VypisPolozka*> seznamPolozek` – proměnná, která uchovává seznam jednotlivých instancí třídy **vypisPolozka**.

### Funkce:

- `VypisSeznam(QWidget *parent = 0)` – konstruktor třídy `VypisSeznam`  
V konstruktoru je nastavena proměnná **ui** pro správu grafických objektů.

- `~VypisSeznam()` – destruktor třídy `VypisSeznam`
- `void vycistiSeznamPolozek()`  
Zavolá destruktor na každou jednotlivou položku seznamu `seznamPolozek` a vymaže tento seznam voláním funkce `clear`.

### Signály:

- `void prechodNaPodrobny_sig(int typPolozky, QString id_hry)`  
Umožňuje přechod na podrobný výpis zvolené položky seznamu. Vyvolává slot `showPodrobnyVypis_slot` v `mainwindow`.

### Sloty:

- `void vytvorSeznam(QSqlQuery query, int typPolozky)`  
Hlavní slot, který vytváří samotný seznam výpisu položek. Převádí záznam z `Sql` odpovědí do jednotlivých instancí třídy `vypisPolozka`.
- `void prechodNaPodrobny(int typPolozky, QString id_hry)` Vyvolává signál této třídy `prechodNaPodrobny_sig`.

## B. Obsah příloženého CD

### `bin_pc/`

Instalátor `DH_SETUP.EXE` programu spustitelný přímo z CD.

### `bin_web/`

Kompletní adresářová struktura webových stránek DATABÁZE HER pro zkopírování na webový server. Adresář obsahuje i všechny potřebné knihovny a další soubory pro bezproblémový provoz na webovém serveru.

### `doc/`

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PŘF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace, tj. zdrojový text dokumentace, vložené obrázky, apod.

### `src_pc/`

Kompletní zdrojové texty programu DATABÁZE HER se všemi potřebnými (převzatými) zdrojovými texty, knihovnami a dalšími soubory pro bezproblémové vytvoření spustitelných verzí programu.

### `readme_pc.txt`

Instrukce pro instalaci a spuštění programu DATABÁZE HER, včetně požadavků pro jeho provoz.

### `readme_web.txt`

Instrukce pro nasazení webových stránek DATABÁZE HER na webový server, včetně požadavků pro její provoz, a webová adresa, na které je aplikace nasazena pro testovací účely a pro účel obhajoby práce.