

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SYSTEMS

ELEKTRONICKÁ SPRÁVA ZÁVODŮ AGILITY

BAKALÁŘSKÁ PRÁCE

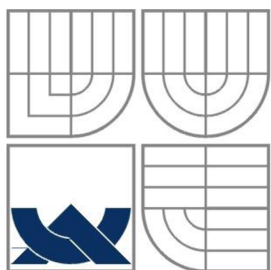
BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

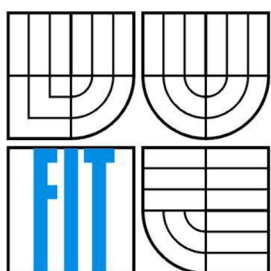
PAVLA VORLOVÁ

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SYSTEMS

ELEKTRONICKÁ SPRÁVA ZÁVODŮ AGILITY

ELECTRONIC MANAGEMENT OF AGILITY RACE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVLA VORLOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

ING. JAN KAŠTIL

BRNO 2010

Abstrakt

Bakalářská práce popisuje implementaci systému pro správu závodů agility, který je tvořen webovým portálem a aplikací na zpracování výsledků v místě závodu. Jsou zde popsány technologie JSP, Java Beans a MySQL. Zvláštní důraz je kladen na způsob přenosu dat mezi databází webového rozhraní a databází klientské aplikace.

Abstract

The work describes implementation of system for agility competition management. The system contains web interface and program which evaluates results in competition location. This thesis describes usage of JSP, Java Beans and MySQL. The focus was on transferring data between web database and client application database.

Klíčová slova

Webový portál, klientská aplikace, databáze, Java, JSP, Java Bean, MySQL, kopírování mezi databázemi.

Keywords

Web portal, klient application, database, Java, JSP, Java Bean, MySQL, data transferring between databases.

Citace

Vorlová Pavla: Elektronická správa závodů agility, bakalářská práce, Brno, FIT VUT v Brně, rok 2010

ELEKTRONICKÁ SPRÁVA ZÁVODŮ AGILITY

Prohlášení

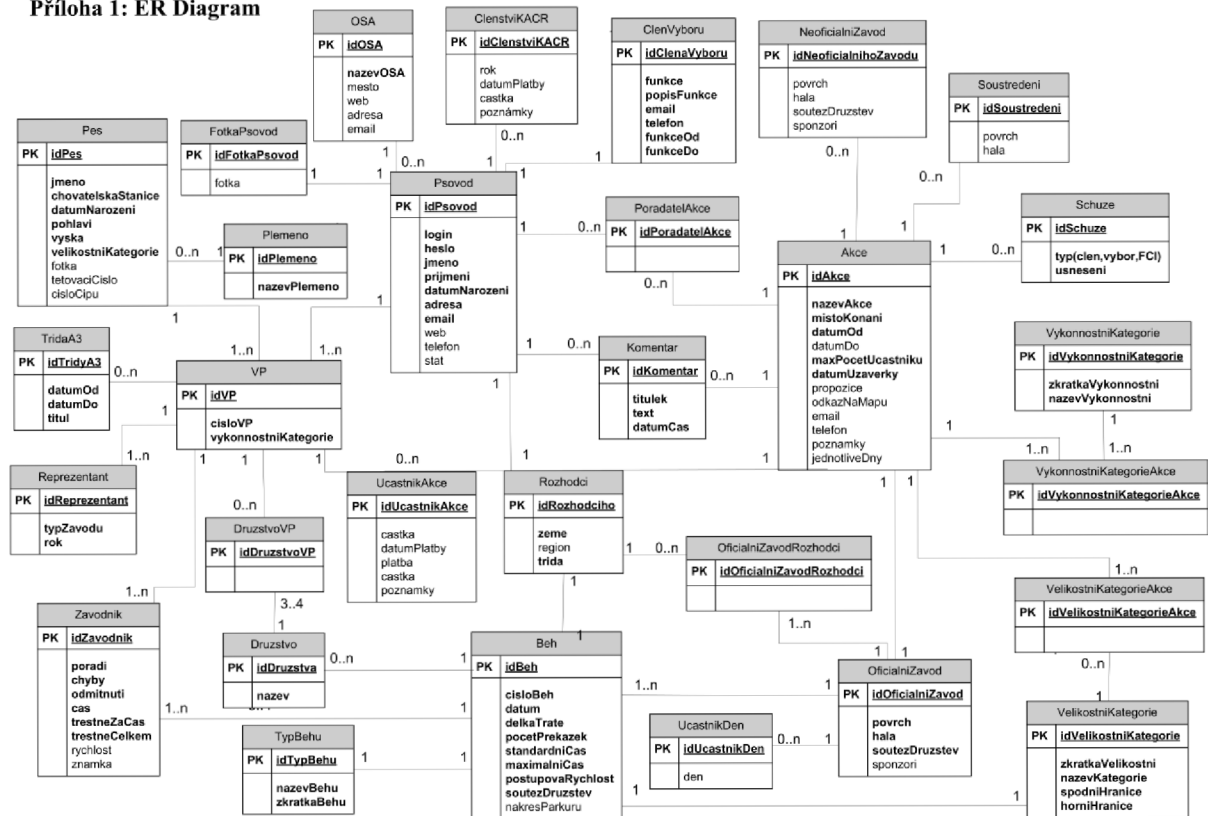
Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením Ing. Jana Kaštila. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....
Pavla Vorlová
19. 5. 2010

Poděkování

Chtěla bych poděkovat svému vedoucímu Ing. Janu Kaštilovi, za jeho odbornou pomoc a konzultace, které mi poskytoval během tvorby této práce.

Příloha 1: ER Diagram



© Pavla Vorlová, 2010

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod.....	3
1.1	Úvod do agility a jejich závodů.....	3
1.2	Motivace.....	3
2	Specifikace	4
2.1	Webové rozhraní	4
2.2	Aplikace na zpracování závodů.....	5
3	Volba technologií a jejich použití	6
3.1	Volba implementačního jazyka	6
3.2	Volba databáze	7
3.3	Databáze ve webovém rozhraní	7
3.4	Databáze v aplikaci	7
3.5	HTML a kaskádové styly	8
3.6	JavaServer Pages a servlety.....	8
3.7	Java Beans	9
4	Návrh databáze	10
4.1	ER diagram.....	10
4.2	Use Case diagram.....	10
5	Implementace webového rozhraní.....	11
5.1	Struktura webové stránky	11
5.2	Formuláře	11
5.3	Zobrazení dat z databáze	12
5.4	Databázové tabulky a jejich role v systému	12
6	Komunikace webu s aplikací.....	14
6.1	Vytvoření Java Web Start	14
6.2	Kopírování dat mezi databázemi	14
7	Implementace a funkce aplikace	16
7.1	Tabulky a další grafické komponenty	16
7.2	Tiskové sestavy	17
8	Testování programu.....	19
9	Závěr	20
10	Možnosti rozšíření.....	20
	Literatura	21
	Seznam příloh.....	22
	Příloha 1: ER diagram databáze webového portálu	23

Příloha 2: ER diagram databáze aplikace.....	24
Příloha 3: Use Case diagram	25
Příloha 4: Screenshoty webového rozhraní	26
Příloha 5: Screenshoty klientské aplikace	27

1 Úvod

1.1 Úvod do agility a jejich závodů

Agility je odvětví kynologického sportu, které vzniklo z koňských parkurových závodů. Oproti závodům koňským, psi obsahují více druhů překážek a psovod běží společně se psem. Parkúr agility vždy obsahuje 12-20 překážek a záleží na fantazii rozhodčího, jak bude celá trať vypadat. Vítězí tým, který má nejméně chyb a odmítnutí, pokud je takových týmů více, tak rozhoduje nejrychlejší čas. Závodní týmy se dělí do tří velikostních kategorií Small, Medium a Large, podle kterých mají nastavenou velikost některých překážek. Týmy se také člení do tříd podle výkonnosti A1, A2, A3.

Závody agility se skládají z několika běhů za den a někdy i z několika dní. Existují tři typy běhů: open agility, jumping a zkoušky, kde spolu soutěží týmy pouze stejné výkonnostní kategorie, oproti openu a jumpingu, ve kterých všechny kategorie závodí dohromady. Vše podstatné kolem závodu určuje pořadatel (program závodu, z kterých běhů se bude skládat, závodní povrch, apod.).

Zkoušky agility jsou vždy hodnoceny známkou pro daný tým, která se odvíjí od celkového počtu trestných bodů získaných na trati. Týmy začínají závodit ve výkonnostní kategorii A1, ze které postupují do vyšší po splnění kritérií (3 zkoušky se známkou Výborně). Speciální je elitní třída A3, kterou týmy musí opustit, pokud ročně nesplní daný limit. Naopak pokud se v ní udrží 3 roky po sobě, získávají titul A3 Champion.

1.2 Motivace

Závody agility se konají každý víkend a momentálně neexistuje žádný software, který by celou správu závodu usnadňoval. Pořadatelé tak všechny údaje o přihlášených týmech zpracovávají manuálně. To znamená několikrát opisovat údaje o všech závodnících, kontrolovat, zda jsou přihlášky vyplněny správně, chybějící údaje dohledávat a nejdůležitější informace (členství v Klubu agility, číslo průkazu závodníka) kontrolovat se seznamy klubu. Zpracování výsledků se často liší závod od závodu. Řád agility nespécifikuje způsob prezentace výsledků ze závodu a tak neexistuje jednotná forma záznamů těchto dat, proto každý pořadatel používá svůj software.

Dle mého mínění by bylo vhodné všechna data sjednotit a uložit do centrální databáze. K této databázi by se přistupovalo přes webové rozhraní, které by umožňovalo vkládat nová data o závodnících, výboru klubu, reprezentaci a informace o akcích agility. Každý uživatel by měl vlastní profil, přes který by se mohl přihlašovat na všechny akce bez nutnosti vyplňování přihlášky na každý závod. Registrovaný uživatel by měl možnost zakládat nové akce a evidovat platby účastníků. Před závodem by byly ke stažení údaje o závodě a startovní listina účastníků do aplikace na zpracování výsledků. Po vyhodnocení závodů by byly výsledky nahrávány přes webové rozhraní do databáze a tak bychom získali přehled nad všemi výkony závodníků.

2 Specifikace

Základ bude tvořit databáze obsahující všechny podstatné informace o závodních týmech, rozhodčích, výboru klubu, připravovaných i proběhnutých závodech a jiných akcích, jako jsou výborové schůze, neoficiální závody anebo soustředění. Ty budou do databáze přidávány a měněny prostřednictvím webového rozhraní, kde také budou data prezentována. Aby mohly být do databáze zařazovány i výsledky závodů, bude součástí práce i klientská aplikace na zpracování výsledků v průběhu závodu.

Aplikace by umožňovala vybrat složení běhů v závodě, zpracovávat výsledky závodů, počítat odvody Klubu agility ČR a exportovat výsledky do různých formátů jako jsou tiskové sestavy, html stránky i export do databáze webového portálu.

2.1 Webové rozhraní

Webové rozhraní obsahuje kalendář všech akcí agility, jak budoucích, tak proběhlých, informace o závodních týmech, rozhodčích a výboru klubu. Jeho součástí jsou také kolekce parkůrů od jednotlivých rozhodčích a výsledky závodů pořádaných v ČR. Kdokoliv se může registrovat jako uživatel. Registrovaný uživatel může přidávat své psy, se kterými závodí, editovat data o sobě i o nich. Systém také umožňuje jednodušší přihlašování na závody, kdy psovodovi stačí si jen vybrat psa, se kterým bude v závodě startovat, oproti nynějšímu systému, kdy na každou akci musí závodník vyplňovat všechny údaje o sobě i psovi znovu a znovu. Tím, že přihlašování na závody bude probíhat jen přes tento systém, má uživatel k dispozici vlastní kalendář, kde je uvedeno, na které akce je přihlášen, a zda jej pořadatel už registroval, popř. se v něm může z akcí odhlašovat.

Kterýkoliv přihlášený uživatel má možnost vložit nový závod nebo soustředění, kde udává, kteří uživatelé také patří mezi pořadatele a budou moci editovat informace o této akci. Pořadatelé evidují platby přihlášených účastníků a mají přístup ke všem údajům o startujících v daném závodě. Přihlašování končí po naplnění kapacity závodu nebo v den uzávěrky. Před závodem si pořadatelé stahují z webu klientskou aplikaci a soubor obsahující informace o závodě a přihlášených účastnících, což využijí ke zpracování výsledků. Po závodě se data z aplikace nahrávají do databáze, aby k nim měli všichni uživatelé přístup. V systému pak posuzující rozhodčí přidávají nákresy jednotlivých běhů a další uživatelé vkládají komentáře k závodě, popř. odkazy na fotky a videa.

Další funkce jsou určeny členům výboru klubu, kteří mají přístup ke všem datům oproti běžným uživatelům, kteří například nevidí adresy ostatních lidí a další osobní údaje. Pokladní zadává ke každému členovi, do kterého data má zaplacené členské příspěvky klubu a zapisuje přijaté platby za odvody ze závodů.

2.2 Aplikace na zpracování závodů

Aplikace je zaměřena zejména na jednoduché, efektivní a rychlé vyhodnocení jednotlivých běhů a jejich součtů. Při startu aplikace se program nejdříve snaží najít databázi. Pokud je databáze nalezena, program se k ní připojí a načte informace o posledním zobrazeném závodě. Uživateli je tak umožněno spustit poslední editovaný závod. Pokud databáze neexistuje, inicializuje se nová a provede se připojení. V tomto případě uživatel spustí závod tak, že nahraje soubor, který stáhnul z webového rozhraní. Poslední možností je založení svého vlastního závodu s tím, že jsou účastníci přidáváni přes přihlašovací formulář. Uživatel může také nahrát závod již vyhodnocený, který si v minulosti z programu uložil. Nahrát data z webového rozhraní lze vložením zazipovaného souboru, který uživatel stáhne z webu. Po kontrole souboru aplikace vytvoří strukturu databáze a naplní ji daty z rozzipovaného souboru.

Pořadatel při prezenci určuje, které týmy jsou přítomny, a vybírá, jaké typy běhů bude závod obsahovat. Podle toho se přítomné týmy dle velikostních kategorií rozdělí do jednotlivých běhů, kde uživatel bude mít možnost měnit pořadí startu závodníků nebo zadat, že závodník nestartuje. Pořadatel má k dispozici startovní listiny na každý běh, které program umožňuje tisknout a tabulku k zapisování a vyhodnocování výsledků. Tabulku je také možné vytisknout a zapisovat výsledky nejprve na papír, v případě, že není možné mít počítač přímo u parkúru, a poté je přepsat do programu.

Každý závodní běh má vlastní parametry, jako je kategorie, pro kterou je určen, délka trati, standardní a maximální čas, za který musí závodníci parkúr překonat, počet překážek a nákras trati. U každého závodníka se hodnotí počet chyb, což jsou například shozené tyčky, a odmítnutí, kdy pes vyhne překážku v daném pořadí. Důležitý je také čas, za který tým běh absolvoval, protože pokud je čas delší než standardní čas určený rozhodčím, dostává tým trestné body za čas. Z trestných bodů za čas, chyb a odmítnutí se počítají celkové trestné body a z délky trati a času týmu postupová rychlost. Podle těchto proměnných se určuje výsledné pořadí, vyhrává tým s minimálním počtem trestných bodů a co nejrychlejším časem.

Po zapsání výsledků může pořadatel tisknout výsledkovou listinu, převést daný běh na webovou stránku a tisknout výsledky do výkonnostních průkazů. Aplikace také umí vytvářet součty, kdy se týmům sečtou výsledky ze dvou běhů a podle nich se určí nové pořadí. Po vyhodnocení všech běhů budou všechny výsledky exportovány do souboru a pořadatel má povinnost je nahrát přes webové rozhraní k danému závodě.

3 Volba technologií a jejich použití

Nejdříve se budeme zabývat tím, co se od aplikace očekává, a jaké by měly být důležité vlastnosti jazyka, ve kterém bude aplikace naprogramována. Pole uživatelů bude velmi různorodé, jak z hlediska zkušeností uživatele s užíváním PC, tak z hlediska operačního systému, na kterém může aplikace běžet. Tím pádem klademe důraz na jednoduchost, intuitivnost, a rychlost ovládání aplikace, a zároveň i co nejjednodušší a nejkratší instalaci programu i databáze, která musí s aplikací spolupracovat. Dalším faktorem při výběru je fakt, že aplikace i databáze v ní musí běžet offline, protože v prostředí závodů je internetové připojení jen velmi zřídka. Pro přehlednost, udržitelnost a čistotu kódu se dobrou volbou jeví objektově orientované jazyky. Jedny z nejpobulárnějších objektově orientovaných jazyků současnosti jsou C++, Java a C#.

3.1 Volba implementačního jazyka

Jazyk C++ nabízí přenositelnost na rozličné platformy, avšak distribuce programu na ně je celkem náročná. Jeho kód je kompilovaný přímo do jazyka symbolických instrukcí (assembler) a to zajišťuje rychlejší interpretaci programu. Jako diskutabilní výhoda se jeví podpora vícenásobné dědičnosti a přetížení operátorů. Nevýhodou je manuální správa paměti, která může být pro programátora náročnější, a také předávání hodnot pomocí ukazatelů, které mohou vést k zápisu mimo požadovanou oblast. [7]

Dalším kandidátem je **Java**. Oproti C++ je překládána do vlastního mezikódu společného pro všechny platformy. Tento tzv. „byte-code“ je následně interpretován ve vlastním prostředí JRE (Java Runtime Environment). Díky tomuto faktu je interpretace pomalejší než u jazyka C++. Mezi další nevýhody je možno uvést vyšší paměťovou náročnost JRE oproti nativně kompilovanému kódu. Na druhou stranu je Java vyšší objektově orientovaný jazyk a nabízí velké množství předprogramovaných funkcí a knihoven, což znatelně urychluje vývoj aplikací. Velkou výhodou je i automatická správa paměti, o kterou se stará garbage collector, a programátor tak nemusí řešit zbytečné „memory leaky“. Při tvorbě GUI jsou velmi užitečné nativní knihovny Swing a Awt, které prostřednictvím předpřipravených komponent umožňují snadnou stavbu grafického uživatelského rozhraní. Jazyk Java je přímo podporován při tvorbě webových aplikací serverovou technologií JSP (java server pages). Tak je možné kód napsaný pro desktopovou aplikaci použít i ve webovém prostředí. [2]

Posledním porovnávaným jazykem je nejmladší **C# společnosti Microsoft**. C# je stejně jako Java kompilován do mezikódu (IL - assembly), který je následně přeložen do strojového kódu pomocí Common Language Runtime .NET Frameworku. Rychlost zpracování je tedy o něco pomalejší než u jazyka C++, ale srovnatelná s Javou. Jako vyšší objektově orientovaný jazyk spravuje paměť automaticky a to pomocí garbage collectoru. Také díky .NET Frameworku můžeme přistupovat k velkému množství užitečných knihoven. Může se tak vytvářet i rozsáhlá hierarchie výjimek v jednom hlídaném bloku.

Nevýhodou je nemožnost nebo komplikovanost nasazení aplikace na jiné operační systémy než MS Windows. [5]

Pro tento projekt byla vybrána Java. To, že se jedná o vyšší objektově orientovaný jazyk, s velkou podporou komunity a s multiplatformním nasazením předčilo výhody ostatních jazyků. Velkou výhodou je také použití jednoho jazyka, jak pro aplikaci, tak pro webové rozhraní.

3.2 Volba databáze

Jako databázový systém byl zvolen **MySQL**, jelikož je jedním z nejrozšířenějších, a tak i dobře dostupným. Zároveň je multiplatformní (Windows, Linux, Mac OS, Unix) a může být k dispozici i pod bezplatnou licenci. Jeho výhodou je zejména rychlost a stabilita a aplikace s MySQL je možné psát v rozmanitých programovacích jazycích. Důležité je i to, že dokumentace k MySQL je dostupná, jak na internetu, tak prostřednictvím mnoha publikací a to zdařilou formou.

Vlastnosti MySQL, které jsou výhodné při tvorbě našeho projektu je architektura typu klient/server, kdy server tvoří MySQL a klienty jsou různé aplikační programy komunikující se serverem. Důležité je, že současná verze MySQL podporuje různá kódování a ukládání dat na fyzický disk. Pro lepší návrh databáze upřednostníme transakční úložiště dat (InnoDB), které mimo jiné podporuje i cizí klíče a vztahy mezi tabulkami. Klíčovou roli v tomto projektu při exportu a importu dat mezi databázemi hraje podpora příkazů „SELECT INTO OUTFILE“ a „LOAD DATA INFILE“. [1]

3.3 Databáze ve webovém rozhraní

K připojení databáze poskytuje Java rozhraní **Java Database Connectivity (JDBC)**, což je kolekce tříd pro programování databázových aplikací. Pomocí rozhraní tohoto API zprostředkovává komunikaci s daným databázovým systémem. Programování je tak odstíněno od specifického API databáze a jednotné rozhraní JDBC se může použít pro přístup ke kterémukoliv databázovému systému. Rozdílem je pouze použití odlišného ovladače. Pro JDBC existují čtyři typy ovladačů. Typ 4 je celý čistě napsaný v Javě, výhodou je to, že není nutné instalovat žádné lokální ovladače a nemusí být nakonfigurovaný. Čtvrtého typu je i ovladač pro MySQL nazvaný **MySQL Connector/J**, který je volně přístupný. [1]

3.4 Databáze v aplikaci

V aplikaci jsem využila Java Utility balíček s názvem **MySQL Connector/MXJ**, který zprostředkovává nasazení a správu databáze MySQL. Uživatel tak nemusí k programu zvlášť instalovat žádnou externí databázi. Stačí pouze stáhnout aplikaci a o zpřístupnění databázového systému se postará tento konektor.

Při inicializaci databáze MySQL se pro každou platformu nakonfigurují režijní data, která jsou uložena na předdefinované místo. Databáze se vytváří jako javové složky, ve kterých se ukládají všechny informace o databázi, její struktuře a obsahu. Tento balíček může pracovat na platformách Linux, FreeBSD, Windows, Solaris Sparc a Mac OS. Perzistence dat je zajištěna uložením dat v příslušných složkách v klientském počítači. [8]

3.5 HTML a kaskádové styly

HyperText Markup Language (HTML) je standardní značkovací jazyk sloužící pro tvorbu webových stránek. Umožňuje vytváření strukturovaných dokumentů. Je charakterizován množstvím značek (tagů), které určují sémantiku textu, jako jsou nadpisy, odstavce, seznamy, citace atd. Značky obsahují atributy popisující jejich vlastnosti. Umožňuje také vkládat do stránek obrázky a jiné objekty.

Kaskádové styly (CSS) definují vzhled jednotlivých elementů ve webové stránce. Jednou z nejdůležitějších vlastností CSS je možnost oddělení struktury a obsahu dokumentu od jeho vzhledu. Díky tomu je možné definovat jednotný vzhled stejných elementů pro celý webový portál a je také docíleno přehlednějšího kódu. Kaskádové styly se mapují na HTML stránku pomocí selektorů, které vyberou danou oblast, na kterou pak aplikují příslušné styly. Jednoduše se tak sjednocuje vzhled elementů stejného typu. Dědičnost v CSS určuje, že element, který nemá nastaveny vlastnosti, je dědí od nadřazeného elementu. [4]

3.6 JavaServer Pages a servlety

Java Server Pages (JSP) jsou webové stránky tvořené HTML a dynamickým obsahem. Dynamický obsah je definován pomocí speciálních značek, tzv. tagů. JSP se používají zejména v případech, kdy většinu stránky tvoří statický obsah a dynamický jej jen doplňuje.

Oproti tomu **servlety** generují jen dynamické části stránky. Jsou to programy psané v Javě, které mohou reagovat na http požadavky. Generovaným obsahem bývají nejčastěji části stránek HTML, ale mohou vytvářet také soubory typu XML. Velkou výhodou servletů je vysoký výkon a snadné použití. Mezi další výhody servletů patří možnost zpracovávat data přijatá z formuláře a dekodovat je příslušným způsobem, což je při tvorbě webu velmi výhodné, protože formuláře tvoří majoritní část informačních systémů. Dalším pozitivem je množství již napsaných tříd, metod a struktur pro práci s HTML. JSP a servlety mohou běžet na více platformách a je možné jejich části použít i v aplikaci, protože jsou rovněž psané v Javě.

V souvislosti se servlety se nabízí elegantní řešení údržby kontextu v informačním systému. Jelikož je protokol http bezstavový a nedokáže tak udržovat žádné informace o uživateli, musíme tento problém řešit. To se udržuje buď pomocí cookies, připisováním informací do URL nebo skrytými formulářovými poli. Servlety tento problém obcházejí pomocí třídy **HttpSession**. Pokud má uživatel cookies zakázány použije přepis do URL.

Udržování kontextu je také nutné k uchování informací o přihlášeném uživateli, které určují jeho oprávnění. [6]

3.7 Java Beans

Java Beans jsou speciální třídy v jazyce Java uložené na serveru, ke kterým se přistupuje pomocí referencí z každé webové stránky napsané v JSP. Předností je, že je tak oddělen dynamický obsah od statického. Java Beans jsou mezi komunitou velmi populární a proto existuje velké množství hotových řešení. Aby mohly být použity, je nutné dodržovat určité konvence. Třída, kterou používáme jako Java Bean, musí mít prázdný konstruktor, který se volá při referencování třídy z JSP. Jelikož všechny vlastnosti třídy jsou soukromé, můžeme k nim přistupovat pouze přes metody, které jednotlivé vlastnosti nastavují anebo vracejí jejich hodnotu.

Za výbornou vlastnost Java Beans považuji to, že když při nastavování vlastností je uvedena „*“ místo nějakého konkrétního atributu, naplní se všechny hodnoty ve třídě, které jsou odeslány z formuláře. Java Beans jsou v projektu používány při odesílání většiny formulářů, při jejich zpracování a zároveň při získávání informací z databáze. [6]

4 Návrh databáze

Správný návrh databáze je velmi důležitý, protože ovlivňuje flexibilitu, rozšiřitelnost i správu a údržbu celého projektu. Chyba v návrhu má za následky složité opravy v průběhu psaní programu. Důležité je zejména správně navrhnout strukturu tabulek, aby nedocházelo k redundanci dat, správně zvolit datové typy a jejich potřebnou alokaci, abychom zbytečně neplýtvali místem v paměti. Pro čistotu návrhu je také důležité správně zvolit všechny číselníky. Podstatné jsou vazby mezi tabulkami, které by měly umožňovat co nejjednodušší skládání dotazů.

4.1 ER diagram

Pro znázornění struktury databáze jak ve webovém rozhraní, tak v aplikaci byl využit **Entity Relationship diagram**. ERD databáze webového rozhraní tvoří 30 tabulek, rozdělených do dvou částí. První část tvoří tabulky vztahující se k hlavní tabulce, a tou je registrovaný uživatel neboli *Psovod*, na kterou navazují všechny jeho funkce a přidružené objekty. Druhou důležitou částí jsou *Akce*, které se dělí na oficiální závody, neoficiální závody a soustředění. K nim se pak vážou podrobnější informace. Nákres ERD je umístěn v příloze číslo 1. ERD databáze v aplikaci je složený jen z 11 tabulek je k nalezení v příloze 2.

4.2 Use Case diagram

Use case diagram se využívá ke znázornění funkční struktury systému z pohledu uživatele. Zejména definuje chování systému bez odhalení vnitřní struktury tak, že zachycuje případy užití, aktéry a jejich vztahy. V našem případě máme aktéry několika rolí, jako je běžný návštěvník webu, registrovaný uživatel, člen výboru Klubu agility ČR, pokladní klubu, rozhodčí, pořadatel akce a čas.

Běžný návštěvník může prohlížet všechny veřejné údaje o uživatelích, psech, výboru klubu, reprezentaci, kalendář akcí, a další, ale nemůže nic editovat. Oproti tomu registrovaný uživatel může měnit data o sobě a svých psech, psy přidávat, přihlašovat se na všechny akce. Také může akci založit a určit tak její pořadatele.

Pořadatel akce může akci editovat, spravovat platby na akci, akceptovat účastníky, mít přístup ke všem informacím o daném účastníkovi, stahovat program na vyhodnocení závodů a startovní listinu. Po skončení závodu musí nahrát výsledky závodu do hlavní databáze přes webové rozhraní. Dalším aktérem je výbor klubu, který má přístup ke všem údajům o uživatelích, může přidávat další členy výboru, nové rozhodčí a měnit kategorii rozhodčích. Pokladní klubu eviduje platby členských příspěvků a odvody ze závodů. Poslední rolí je čas, který určuje zařazení akcí mezi proběhlé nebo budoucí a podle toho se buď zobrazuje startovní listina, nebo výsledková. Také může vypršet členství v klubu v průběhu času. Nákres Use Case diagramu je v příloze číslo 3.

5 Implementace webového rozhraní

Pro implementaci webového rozhraní je použito HTML, JSP, JavaScript a kaskádové styly. Hlavní část tvoří formuláře, jejichž prostřednictvím se vkládají a editují data v databázi. Další podstatnou složkou jsou různé seznamy reprezentující uložená data.

5.1 Struktura webové stránky

Pro tvorbu GUI byly využity kaskádové styly, které popisují způsob zobrazení webových stránek. Webové rozhraní je tvořeno nadpisem, horním menu, patičkou a dvousloupcovým layoutem, který se skládá se z levého sloupce zobrazujícího nabídku, a hlavního obsahu stránky. Horní menu je výraznější a obsahuje funkce, které bude uživatel nejčastěji využívat (kalendář akcí, profil a přihlášení). Druhé menu je umístěno v levém sloupci a obsahuje seznam stránek, které slouží pro zprostředkování dalších služeb webového portálu. Toto menu se mění podle oprávnění přihlášeného uživatele. Stránka také obsahuje oblast určenou k vyhledávání. Umožňuje vyhledat psovoda, jak podle jména, příjmení, tak i podle loginu. Psy vyhledává podle jména a chovné stanice a akce, podle názvu a místa.

5.2 Formuláře

Formuláře jsou významnou částí informačních systémů. Jsou tvořeny několika prvky, jako jsou textová pole, zatržítka, přepínače, rolovací menu a tlačítka. Uživatel je povinen vyplnit určitá pole, a po vyplnění údajů formulář odesílá. Každý formulář odkazuje na JSP stránku, která jej zpracuje pomocí příslušné Java Bean třídy, která načte hodnoty polí do svých vlastností. Hodnoty vlastností jsou poté zkontrolovány, a pokud některá data nejsou korektní, zůstává formulář uživateli zobrazený.

Vložení nového psovoda

Login:

Heslo:

Heslo znovu:

Prosím vyplňte heslo znovu

Hesla se neshodovaly!

Jméno:

Příjmení:

Datum narození:

Rodné číslo:

Adresa:

Email:

Telefon:

OSA: *

Web:

Stát:

Obrázek 1: Ukázka webového formuláře

Chybné údaje jsou v něm zvýrazněny a uživatel je upozorněn, jak má formát špatně zadaného pole vypadat. V momentě, kdy jsou všechny hodnoty korektní, se daný objekt ukládá do databáze a uživatel je přesměrován na další stránku.

Registrovaný uživatel se může do systému přihlásit zadáním jména a hesla v přihlašovacím formuláři. Tento formulář ověřuje, jestli zadaný login v databázi existuje a kontroluje, zda uživatel zadal správné heslo. Při neúspěšném přihlášení je uživatel upozorněn na konkrétní chybu, v opačném případě se uloží login uživatele do session, kde je přístupný k ověření identity a přiřazení odpovídajících oprávnění.

5.3 Zobrazení dat z databáze

V informačním systému je důležité zvolit správnou prezentaci dat uložených v databázi. Zobrazení údajů je realizováno přímo javovým kódem, vloženým do webové stránky pomocí JSP tagů. Při vyhledání dat je zajištěno připojení k databázi a pomocí knihovny java.sql jsou posílány konkrétní SQL dotazy. Výsledky dotazů jsou přímo v JSP stránce zpracovány a vypsány.

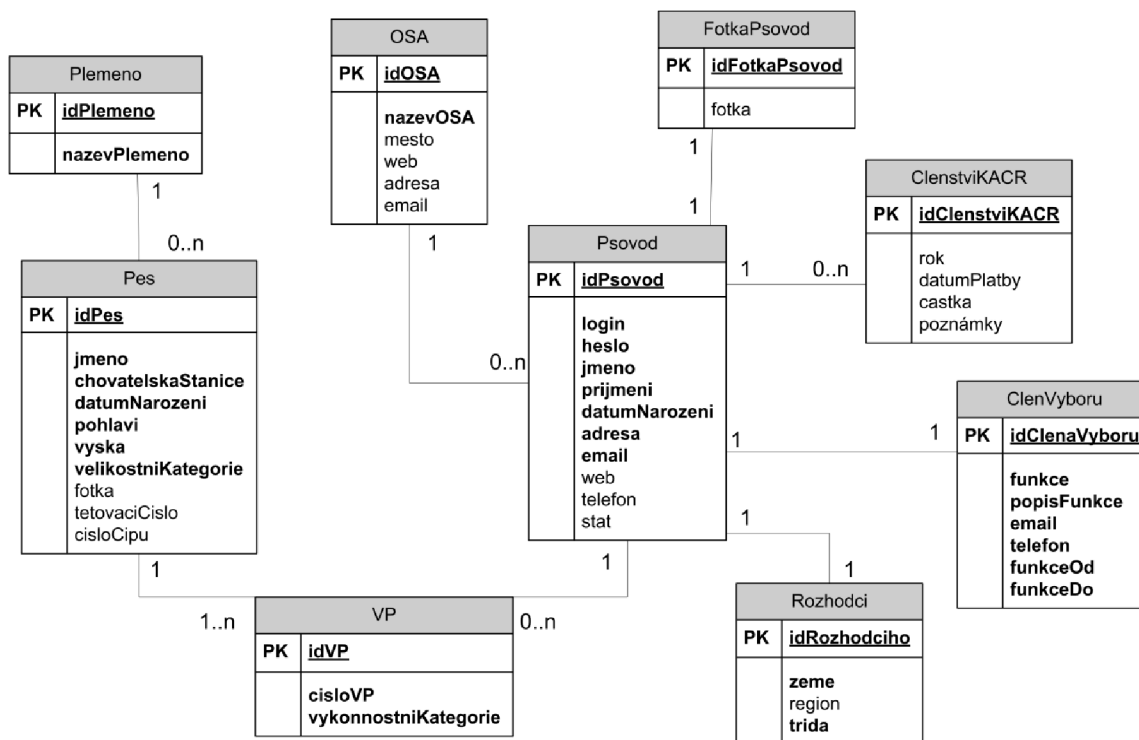
Dalším užitým způsobem je vyhledávání informací pomocí Java Beans. Ve webovém rozhraní je definována třída pro práci s databází, která zabezpečuje připojení a volání několika typů dotazů (executeUpdate, executeQuery). Výsledky vyhledání jsou přijímány prostřednictvím rozhraní ResultSet, které má vlastní kurzor ukazující na aktuální řádek. Třída volající dotaz pak jednotlivé řádky zpracovává. Jednou z možností je uložení dat z ResultSetu do hashovací tabulky. V JSP stránce se pak určí, jak bude zobrazena.

5.4 Databázové tabulky a jejich role v systému

Databáze je tvořena několika tabulkami propojenými přes cizí klíče s integritními omezeními. Hlavní formulář s názvem „*Vložit psovoda*“ slouží k registraci uživatelů do systému. Údaje z něj se zpracovávají ve třídě Psovod, kde se kontroluje, zda v databázi neexistuje stejný login. Ověřená data se nahrávají do databázových tabulek *Psovod* a *OSA* (oblastní skupina agility), kde jsou k dispozici ostatním částem systému. Tabulka *OSA* představuje číselník všech skupin a může být plněna při zadávání osobních údajů uživatelů. Data uložená v těchto tabulkách jsou prezentována na webové stránce „*Seznam psovodů*“ a výsledná množina získaná z databáze je řazena podle požadavků uživatele. *ClenVýboru* a *Rozhodci* určují role uživatelů a rozšiřují tabulku *Psovod* o potřebné detaily. Pro založení rozhodčího nebo člena výboru existují speciální formuláře („*Přidání rozhodčího*“, „*Přidání člena výboru*“), které může vyplňovat nebo editovat pouze člen výboru nebo administrátor.

Tabulka *Pes* obsahuje všechny informace o psovi a váže se na tabulku *Plemeno* (číselník ras). Psi mohou být přidáváni pouze z profilu uživatele formulářem „*Vložit psa*“. Z tohoto formuláře se naplní i tabulka *VP* (výkonnostní průkaz), která určuje, jaký pes patří k danému psovodovi. Běžně nastává situace, kdy jeden pes patří více psovodům, a i psovod může závodit s více psy. Formulář plní pouze tabulku *VP* vytváří spojení mezi psem, který

již v databázi existuje a přihlášeným uživatelem. *ClenstviKACR* reprezentuje informace o uživatelích - zda jsou členy klubu, jakou přispívají částkou a kdy ji platili.



Obrázek 2: Část struktury databáze ve webovém portálu

Všechny akce klubu mají společnou tabulku *Akce*, ve které jsou základní údaje. Tuto tabulku rozšiřují tabulky, podle daného typu akce (oficiální závod, neoficiální závod, soustředění). Každý typ akce tak má vlastní formulář. Prostřednictvím tabulky *PoradatelAkce*, jsou ke každé akci přiřazeni pořadatelé, kteří jsou reprezentováni uživateli uloženými v tabulce *Psovod*. Stejným způsobem jsou přidáni i účastníci akce tabulkou *UcastnikAkce* a rozhodčí *RozhodciOficialniZavod*.

6 Komunikace webu s aplikací

Vzhledem k širokému poli uživatelů bylo cílem, aby zavedení aplikace bylo co nejjednodušší. Java nabízí technologii **Java Web Start (JWS)**, která umožňuje spouštět aplikace jediným kliknutím na odkaz umístěný na webové stránce. Je tak možné spouštět program na jakékoli platformě bez nutné instalace. Program je stažen na klientský počítač a může být kdykoliv spuštěn i v offline režimu. Programy se spustí i na počítačích, na kterých doposud není nainstalována Java, protože JWS zajišťuje stažení potřebné verze JRE.

6.1 Vytvoření Java Web Start

Když se vytváří JWS, je nutné mít správně nakonfigurovaný webový server, aby dokázal zpracovat soubory typu **Java Network Launching Protocol (JNLP)**. Tento soubor se vytvoří zabalením všech tříd programu do jednoho nebo více souborů **Java Archive (JAR)**. Pokud aplikace při běhu potřebuje vyšší oprávnění, která jsou mimo rozsah práv appletů, musíme JAR digitálně podepsat. I když jsou programy spouštěny mimo webový prohlížeč, stále zůstávají v tzv. „restricted containeru“ (zóna omezené funkčnosti programu). K tomuto dochází zejména v případě přístupu k disku nebo síťovým zdrojům. Po podepsání archivu je ještě nutné vytvořit soubor, který bude definovat, jak má být daný program spuštěn. Tento soubor je ve formátu XML a musí podléhat standardu Java Network Launching Protocol & API Specification. Soubor obsahuje cesty k Java archivům, kódování, informace o autorovi a popis aplikace. Prostřednictvím tohoto XML souboru se nastavuje bezpečnost přístupu ke zdrojům.

6.2 Kopírování dat mezi databázemi

Jelikož se před každým závodem část databáze z webu kopíruje do databáze aplikace, důležitým aspektem je přenášení minimálního množství dat mezi databázemi. Před závodem se vyberou informace o závodnících, kteří se budou daného závodu účastnit a údaje o závodě. Tyto data jsou uloženy do textových souborů podle struktury databáze v klientské aplikaci. Textové soubory jsou zazipovány pomocí knihovny `java.util.zip`, uživatel je stáhne z webu a nahraje do programu, který soubor rozbalí a naplní těmito daty interní databázi. Struktura databáze v klientské aplikaci je odlišná, protože do ní ukládáme pouze data potřebná pro zpracování výsledků během závodu. Nahrání výsledků na web probíhá po vyhodnocení závodu tak, že aplikace vyexportuje zip archiv, ve kterém je soubor obsahující všechny informace ohledně běhů daného závodu (z tabulky *Beh*) a soubor zahrnující výsledky účastníků (z tabulky *Zavodnik*). Tento archiv nahraje pořadatel na web, kde se zip archiv rozbalí a naplní tabulky databáze webového portálu.

Kopírování údajů z webové databáze je realizováno výběrem dat pomocí příkazu ‘SELECT’ a uložení výsledku dotazu do souboru. Tímto způsobem lze vybrat určité sloupce

v tabulkách, tabulky vzájemně spojovat a výsledné množiny i řadit. Bez sebemenších problémů je tak dosaženo synchronizace mezi odlišnými strukturami databáze na webu a v klientské aplikaci. Díky tomu, že import a export dat probíhá přímo pomocí dotazovacího jazyka SQL, je možné klauzulí 'WHERE' efektivně vybrat data, která jsou opravdu potřeba. Ukládání do textového souboru je uživatelem čitelné a tak je možné snadno zkontrolovat vybranou část databáze. Textové soubory jsou také dobře komprimovatelné a tak se snižuje velikost přenášených dat.

Kopírování dat pomocí příkazu 'SELECT INTO OUTFILE' má i své nevýhody. Soubory jsou přímo z databáze ukládány na server, na kterém běží MySQL. To znamená, že jak databáze, tak webové stránky musí běžet na stejném stroji. Další vlastností je nemožnost přepisu již existujícího souboru s totožným jménem. Jelikož v jednom časovém okamžiku může přijít více požadavků ke stažení různých archivů, je každý z nich uložen v osobní složce daného uživatele.

Import dat do databáze je realizován pomocí příkazu 'LOAD DATA INFILE', což je velmi rychlé a není potřeba řešit vkládání jednotlivých záznamů. Jedinou podmínkou je stejná struktura a typy dat, jak v souboru, tak v databázi.

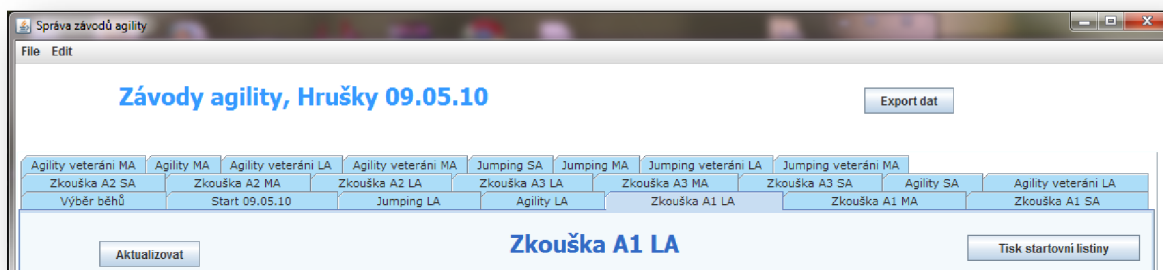
7 Implementace a funkce aplikace

Uživatelské rozhraní aplikace je zaměřeno na jednoduchou a intuitivní práci s aplikací tak, aby i příležitostní uživatelé počítače neměli s řízením programu problémy. Důležitou vlastností navrženého GUI je přehlednost a rychlá navigace.

Grafické uživatelské rozhraní v Javě zprostředkovávají dvě grafické knihovny, kterými jsou starší **Abstract Windowing Toolkit** (AWT) a novější **Java Foundation Classes** (JFC) známá spíše pod jménem jeho součásti **Swing**. Knihovna Swing zahrnuje velké množství komponent jako různá tlačítka, popisky, formulářové prvky, ovládací prvky tabulek, seznamy a další. Také poskytuje různé textové sady a rozličné grafické motivy. Užitečné jsou i funkce uživatelského rozhraní jako obsluha událostí, přizpůsobené vykreslování a správa oken. Při tvorbě aplikace pracující se složkami v počítači jsou nedocenitelné dialogy a prohlížeče souborů. Pokud programátorovi ani toto nestačí, může si nadefinovat komponenty vlastní. Vzhled oken aplikace je možné nastavit na nativní grafickou formu dané platformy nebo se nastavuje vlastní vzhled pomocí tzv. Synth balíčků. Knihovna Swing samozřejmě podporuje i přenos dat pomocí kopírování a vkládání mezi různými komponentami. [2]

7.1 Tabulky a další grafické komponenty

Hlavní okno aplikace tvoří zejména záložkový panel (**Tabbed Pane**), který obsahuje stránku pro výběr běhů, startovní listiny na jednotlivé dny a individuální běhy. Záložkový panel je přehledný, intuitivní a nabízí rychlou navigaci mezi stránkami. V panelech jsou použity comboboxy např. pro výběr typu běhu, nebo výběr rozhodčího. Další podobnou komponentou je listbox, kde je možné vybrat i více položek najednou, což se hodí pro výběr velikostních kategorií daného běhu. Parametry parkuru uživatel zadává do textfields, které umožňují kontrolu zadaných hodnot. Pokud uživatel může v programu něco odstranit nebo smazat, je tázán prostřednictvím kontrolního formuláře **JOptionPane**, kterým uživatel schválí své rozhodnutí, popř. volbu zruší.



Obrázek 3: Část GUI klientské aplikace (záložkový panel)

V aplikaci jsou nejvíce využity tabulky, jak k zobrazení seznamu běhů a k seznamu startujících, tak i na zapisování výsledků. Swing má pro reprezentaci tabulek speciální třídu `JTable`, která je určena pro zobrazení tabulárních dat. Tyto data lze editovat, mazat a formátovat. Za chodu aplikace je také umožněno měnit pořadí sloupců a jejich šířku. Sloupec v tabulce je reprezentován třídou `ColumnModel`, která určuje minimální, maximální a preferovanou šířku sloupců, popř. i formát dat v buňkách. K formátování buněk slouží `CellRenderery`, které dokážou změnit barvu obsahu buňky, pozadí apod. podle příslušných kritérií. V aplikaci tak můžeme zvýraznit požadované buňky (např. průběžné pořadí prvních třech závodníků). Podobně pracují i `CellEditors`, které určují, jak je možné obsah dané buňky změnit. Filtrují vstupní hodnoty, stanovují, jak bude buňka vypadat v průběhu editace a mohou být prezentovány jako některé grafické komponenty (např. tlačítko, checkbox atd.).

Vlastní obsah tabulky je tvořen třídou `AbstractTableModel`, která může být navázána například na seznam objektů. Programátor pak určuje, které vlastnosti objektu budou v daném sloupci zobrazeny. Z vlastností objektu jsou vytvořeny třídy komparátorů, pomocí kterých je možné řadit data v tabulce. Hlavní tabulka zobrazující výsledky je načtena ze seznamu přítomných závodníků. Ke každému řádku (závodníkovi) je možné zapsat počet chyb, počet odmítnutí a čas. Pomocí tabulky a vnořených tlačítek je možné specifické řazení startujících a označení závodníka jakožto nestrartujícího či diskvalifikovaného. Zbytek údajů (trestné body za čas, trestné body celkem, postupová rychlost, známka a průběžné pořadí) se vypočítávají ze zadaných údajů. Barevně se vyznačují diskvalifikované týmy a týmy na průběžných prvních třech místech.

Číslo	Psovod	Pes	Plemeno	Chyby	Odmítnutí	Čas	Trest body za čas	Trest body celkem	Známka	Postup. rychlost	Pořadí	DIS	Nahoru	Dolů	Nestrartuje
1	Lucie Burgetová	Jump Jet	BOC	0	0	29.75	0.00	0.00	V	4.97	1	N...	▲	▼	Startuje
2	Lukáš Blažek	Baldrick Nick	BOT	0	0	32.41	0.00	0.00	V	4.57	2	N...	▲	▼	Startuje
3	Barbora Valentinová	Deborah	flat coated retriev	0	0	36.46	0.00	0.00	V	4.06	3	N...	▲	▼	Startuje
4	Daniela Nováková	Jašika	MOK	0	0	37.64	0.00	0.00	V	3.93	4	N...	▲	▼	Startuje
5	Ludmila Bučková	Ben	kříženec	0	0	40.03	0.00	0.00	V	3.70	5	N...	▲	▼	Startuje
6	Simona Švestková	Mark Twain	briard	0	0	43.59	0.00	0.00	V	3.40	6	N...	▲	▼	Startuje
7	Petra Palešková	Charlie	MOK	1	0	34.06	0.00	5.00	V	4.35	7	N...	▲	▼	Startuje
8	Barbora Valentinová	All for One	BOC	2	0	33.56	0.00	10.00	VD	4.41	8	N...	▲	▼	Startuje
9	Radka Bilečková	Afon	AJO	1	1	38.65	0.00	10.00	VD	3.83	9	N...	▲	▼	Startuje
10	Maritina Píkrá	Casta	BOC	3	1	33.52	0.00	20.00	D	4.42	10	N...	▲	▼	Startuje
11	Barbora Klobásová	Frodo	irský tenér	5	1	43.28	0.00	30.00	BC	3.42	11	N...	▲	▼	Startuje
12	Václav Hůzl	Cooper	zlatý retriv	5	2	41.53	0.00	35.00	BC	3.56	12	N...	▲	▼	Startuje
13	Dana Straková	Crazy	BOT	2	1	0.0	0.00	15.00	DIS	∞	DIS	Výsledky	▲	▼	Startuje
14	Eva Langová	Albert	ESP	0	0	0.0	0.00	0.00	DIS	∞	DIS	DIS	▲	▼	Startuje

Obrázek 4: Tabulka z aplikace vytvořená pomocí `AbstractTableModel`

7.2 Tiskové sestavy

Tisk v Javě zajišťuje **Java Print Service API**, které umožňuje tisk na všech platformách. Vyhledává všechny přístupné tiskárny, určí formát tisknutých dat a odesílá dokument do vybrané tiskárny. Toto API stanovuje okraje papíru a vymezuje tisknutelnou oblast. Dokument k tisku je jakákoliv grafická komponenta implementující rozhraní `Printable`. Toto rozhraní definuje metodu `print()`, která se stará o nastavování příznaků, obsluhu výjimek a o převod dat do tisknutelné podoby. Parametry metody `print` určují, zda bude stránka otočená na výšku nebo na šířku, obsah hlavičky, patičku a číslo stránky. Poté se zobrazí klasický dialog pro komunikaci s tiskárnou, kde je možné vybrat kvalitu tisku, určit počet kopií apod.

V klientské aplikaci se nejvíce tisknou tabulky. Tabulka se při tisku může rozdělit na více stránek, přičemž každá stránka obsahuje její záhlaví. V aplikaci je nutné mít nad každou tabulkou rozsáhlejší hlavičku, která obsahuje několik řádků, popř. i jiné grafické komponenty. Proto je důležité metodu print překrýt a definovat si tisk vlastní, kdy se tabulka tiskne jen do spodní části stránky a horní část zaujímá již zmíněná hlavička.

Výsledková listina

Jumping LA

Závodů agility 09.05.10

Délka tratě: 143.0 m
Počet překážek: 18
Standardní čas: 30.0 s
Maximální čas: 60.0 s
Post. rychlost: 4,77 m/s

Datum běhu: 09.05.10
Typ běhu: Jumping
Rozhodčí: Jaroslava Podmolová

Pořadí	Jméno psovoda	Jméno psa	Plateno	OSA	Chyby	Odměnění	Čas	Trž. ze čas	Trž. celkem	Rychlost	Známa
1	Ivo Bartek	Dakota	BOC	KK Žetor Brno	0	0	24,50	0,00	0,00	5,84	V
2	Yeronika Vargová	Good Luck	BOC	KKHodonín	0	0	25,91	0,00	0,00	5,52	V
3	Zuzana Jašková	Connie	AJO	ROSA Zlín	0	0	27,41	0,00	0,00	5,22	V
4	Eva Langová	Megy	kříženec	Agility Hrušky	0	0	27,47	0,00	0,00	5,21	V
5	Ludmila Bučková	Ben	kříženec	KKHodonín	0	0	28,53	0,00	0,00	5,01	V
6	Eva Homolková	Amarethy	chodský pes	KKHodonín	0	0	28,91	0,00	0,00	4,95	V
7	Evžen Tikovský	Áron	kříženec	PŠ Na Ostrově	0	0	29,53	0,00	0,00	4,84	V
8	Katka Benešová	Porsche	ESP	agility rodina Brno	0	0	29,57	0,00	0,00	4,84	V
9	Lukáš Blažek	Baldrick Nick	BOT	Agility Hrušky	0	0	31,60	1,60	1,60	4,52	V
10	Dušan Buček	Dasty	kříženec	KKHodonín	1	0	25,37	0,00	5,00	5,64	V
11	Darja Caletková	First	KK	PŠ Na Ostrově	1	0	27,15	0,00	5,00	5,27	V
12	Lucie Brettšnajdrová	Emir	AJO	Ag Zlechov	1	0	28,97	0,00	5,00	4,94	V
13	Radka Bilavčíková	Afron	AJO	Ag Zlechov	1	0	30,15	0,15	5,15	4,74	V
14	Kateřina Novotná	Lambra	RR	PŠ Na Ostrově	1	0	30,63	0,63	5,63	4,67	V
15	Soňa Haluzová	Agáta	MOK	ROSA Zlín	0	0	36,25	6,25	6,25	3,94	VD
16	Barbora Valentínová	Deborah	flat coatedretrivr	Hrbolná	0	1	33,91	3,91	8,91	4,22	VD
17	Martina Konečná	Dost Dobrá	KK	agility camaro	2	0	24,22	0,00	10,00	5,90	VD
18	Petra Marečková	Acid Apple	BOC	PŠ Na Ostrově	2	0	25,31	0,00	10,00	5,65	VD
19	Pavla Vorlová	Kara Nica	BOT	Agility Hrušky	2	0	25,65	0,00	10,00	5,58	VD
20	Yeronika Otýpková	Daria	gordonsetr	KK Pedigree VFU...	2	0	26,72	0,00	10,00	5,35	VD
21	Martina Škovroňová	Abra	dlouhosrstá kolie	KK Pedigree VFU...	2	0	28,06	0,00	10,00	5,10	VD
22	Eva Langová	Albert	ESP	Agility Hrušky	1	1	28,53	0,00	10,00	5,01	VD
23	Petra Palečková	Charlie	MOK	Amonra Čebín	1	1	29,90	0,00	10,00	4,78	VD
24	Simona Švestková	Mark Twain	briard	KK Jihlava	1	0	35,78	5,78	10,78	4,00	VD
25	Barbora Klobásová	Frodo	irský teriér	KK Šlapanice	2	0	31,31	1,31	11,31	4,57	VD
26	Monika Prokopová	First	BOT	Ranč Loučka	3	0	24,46	0,00	15,00	5,85	VD

- 1 -

Obrázek 5: Vytisknutá výsledková listina

8 Testování programu

Testování se skládalo z ověřování správné funkčnosti webového portálu, kontroly spuštění aplikace z webového rozhraní a testování klientské aplikace. Byly zjištěny problémy s opětovným připojením webového portálu k databázi MySQL, kdy po několika hodinách vypršel časový limit daného spojení a server neumožňoval připojení webových aplikací. Problém byl vyřešen povolením znovu připojení v přípojovacím řetězci před JDBC. Stažení klientské aplikace z webu bylo rychlé a nevyskytly se žádné problémy. Jako nepříznivé se jevilo první spuštění staženého programu, které trvalo velmi dlouhou dobu díky inicializaci knihoven vytvářejících vestavěnou databázi MySQL. Další spuštění klientské aplikace byly již téměř okamžitá.

Při pořádání závodů v Hruškách 9. 5. 2010 byla možnost otestovat celý systém v praxi. Závodu se účastnilo 92 týmů rozdělených do 11 kategorií. Bohužel webové rozhraní bylo spuštěno až po uzavěrce přihlášek, a tak byla data vkládána pořadatelé. Proto bylo testování práce s webovým portálem omezeno jen na malé množství uživatelů. Testování klientské aplikace probíhalo přímo v prostředí závodů, kde se všechny záznamy psaly rovnou do programu. Znatelné urychlení bylo vidět při změnách startovního pořadí, které se v předešlém programu muselo dělat manuálně a docházelo tak k různým chybám předdefinovaných funkcí. Také tisknutí výsledkových a startovních listin bylo rychlejší a program umožňoval kvalitnější tisk. Užitečná byla kontrola správnosti vyplněných údajů, nedocházelo tak k překlepům, protože uživatel byl okamžitě upozorněn na chybu. Při testování programu v terénu jsme přišli na drobná vylepšení, která usnadní a zpříjemní práci s klientskou aplikací. Při zadávání naměřených hodnot do hlavní tabulky programu je vhodné vymazat výchozí hodnoty buněk. Tato úprava vede k mnohem rychlejšímu zadávání nových hodnot. Užitečné se jeví i podbarvení buněk, které určují průběžné pořadí závodníků nebo jejich diskvalifikaci. Tyto údaje je nutné mít přímo na očích a nedovolit jejich přehlédnutí nebo zaměnění s ostatními. Aplikace se během celého závodu chovala korektně a dle očekávání. Všechna data, která byla získána prostřednictvím klientské aplikace, byla bez problémů importována do webového portálu.

9 Závěr

Cílem této práce bylo seznámit se s organizačním zajištěním závodů agility a navrhnout vhodný program pro zpracování výsledků v průběhu závodu. Práce také zahrnuje popis vytvoření webové rozhraní, které je součástí agendy na pořádání závodů. Tato práce měla za úkol představit použité technologie při tvorbě webové rozhraní, kde byly využity zejména JSP, servlety, Java Beans a CSS. Práce se v několika kapitolách věnuje právě MySQL a přibližuje jeho propojení jak s webovým rozhraním, tak s klientskou aplikací. Je popisována implementace aplikace za použití knihoven jazyka Java, především při vytváření grafického uživatelského rozhraní a při programování tiskových sestav. Také představuje balíčky používané k nasazení vestavěné databáze MySQL. Systém se podařilo implementovat i s klientskou aplikací a experimentálně ho otestovat přímo v prostředí závodů. Testování proběhlo úspěšně, program se po celou dobu závodu choval korektně včetně bezproblémových tiskových výstupů.

10 Možnosti rozšíření

Jelikož se českých závodů účastní i týmy ze zahraničí bylo by vhodné celé webové rozhraní vypracovat ve více jazycích. Samozřejmostí by byla angličtina, němčina a francouzština, ale příhodné by byly i jazyky blízkých zemí jako polština a maďarština. Při soutěžích agility se výjimečně konají soutěže družstev. Každé družstvo tvoří 3 až 4 členové, kteří se účastní daného běhu. Výsledek každého družstva je tvořen součtem tří nejlepších výsledků závodníků. Webové rozhraní již umožňuje přihlašování i do soutěží družstev a klientská aplikace by měla být v budoucnu rozšířena o toto vyhodnocování. Dalším možným rozšířením je platba závodů přímo ve webovém portále přes platební brány. Uživatel by také mohl mít možnost zvolit si vlastní typ vzhledu.

Literatura

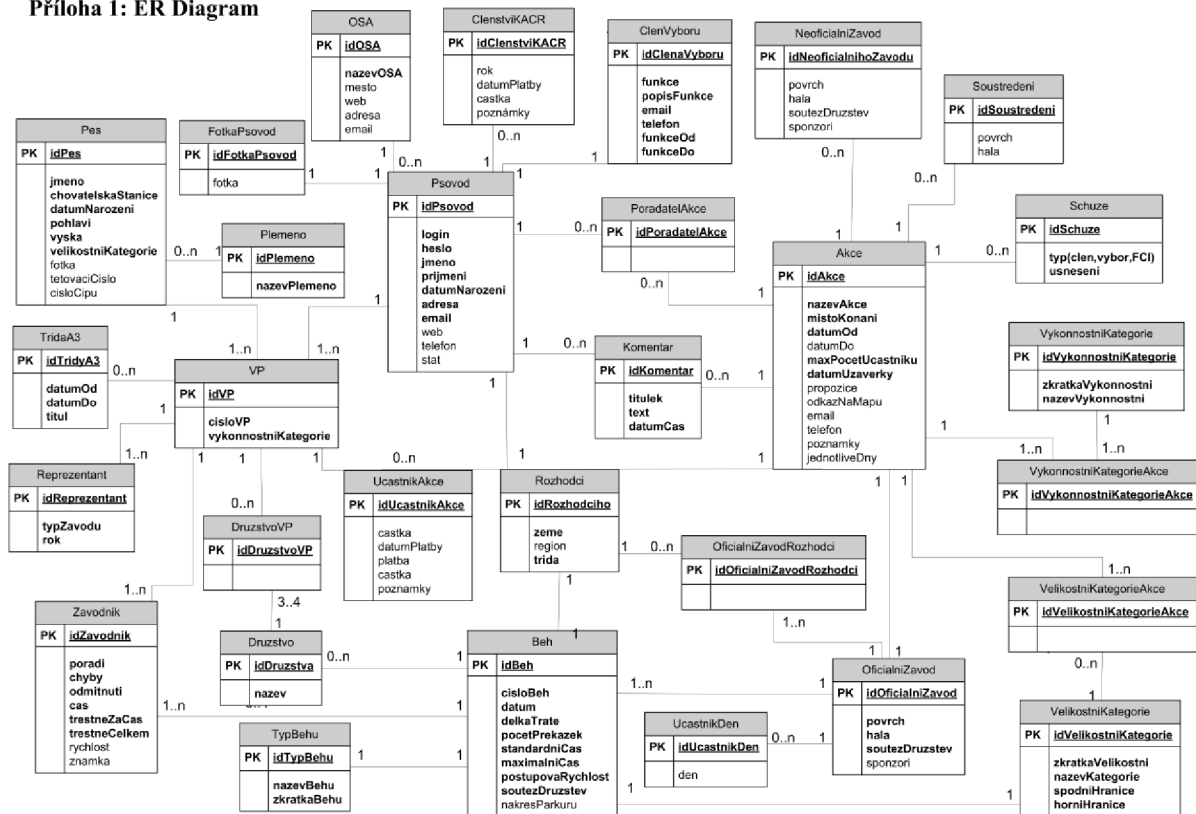
- [1] KOFLER, Michael. Mistrovství v MySQL 5. Computer Press, a.s., 2007.
- [2] ZAKHOUR, Sharon a kolektiv. Java 6: Výukový kurz. Computer Press, a.s., 2007.
- [3] KISZKA, Bogdan. 1001 tipů a triků pro jazyk Java. Computer Press, a.s., 2009.
- [4] CROFT, Jeff a kolektiv. Mistrovství v CSS. Computer Press, a.s., 2007.
- [5] KENT, Jeff. Visual C# 2005: Bez předchozích znalostí. Průvodce pro samouky. Computer Press, a.s., 2007.
- [6] BOLLINGER, Gary, NATARAJAN, Bharathi. JSP: Java Server Pages, podrobný průvodce začínajícího tvůrce. Grada publishing, a.s., 2003.
- [7] KENT, Jeff. C++ bez předchozích znalostí. Průvodce pro samouky. Computer Press, a.s., 2009.
- [8] MySQL AB. MySQL Connector/MXJ Abstract. Sun Microsystems, Inc. 2010.

Seznam příloh

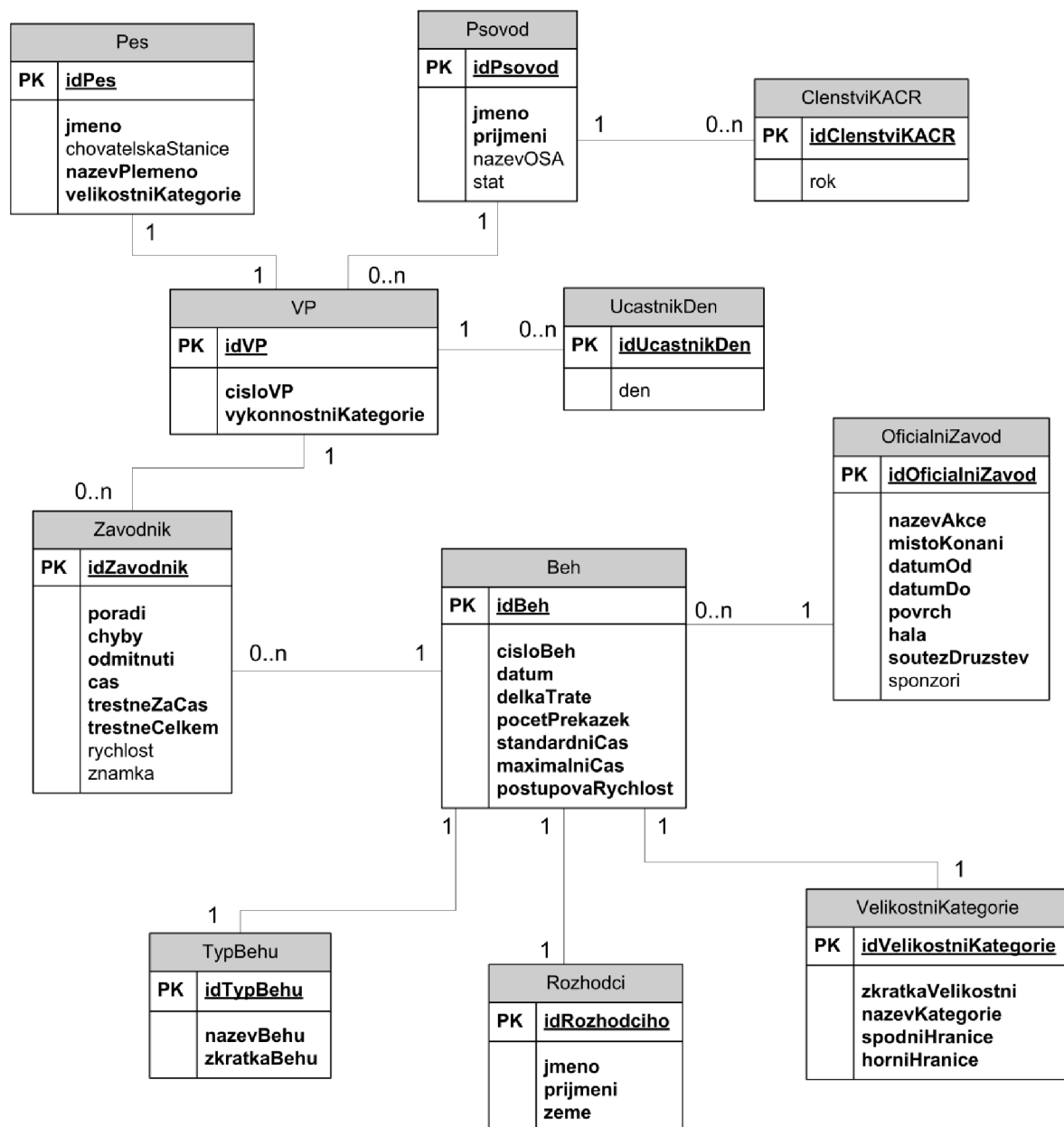
1. ER diagram databáze webového portálu
2. ER diagram databáze v aplikaci
3. Use Case Diagram
4. Screenshoty webu
5. Screenshoty aplikace
6. DVD se zdrojovými kódy

Příloha 1: ER Diagram databáze webového portálu

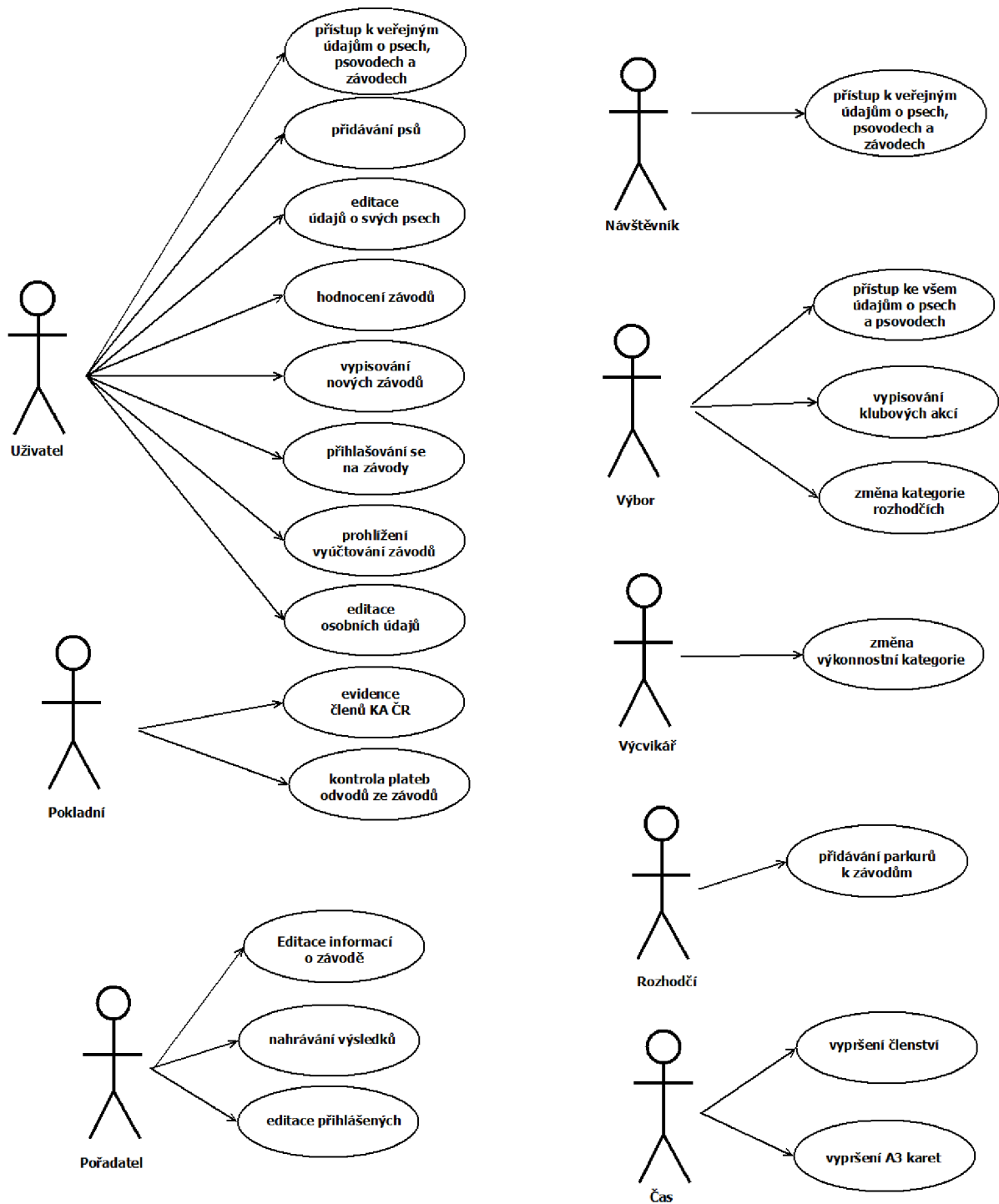
Příloha 1: ER Diagram



Příloha 2: ER Diagram databáze v aplikaci



Příloha 3: Use Case diagram



Příloha 4: Screenshoty webového rozhraní

EVIDENCE AGILITY

Kalendář akcí Registrovat se Můj profil Odhlásit

*** Seznamy ***
Seznam psovodů
Seznam psů
Seznam členů KA

*** Složky klubu ***
Rozhodčí
Výbor klubu
Reprezentace

*** Akce ***
Budoucí akce
Proběhlé akce
Výběr akcí
Oficiální závody
Neoficiálky
Soustředění

*** Vložení ***
Vložit psovoda
Vložit závod

*** Mapa webu ***

Pavla Vorlová
Login: PavlaVorlova
Email: pavla@mapnet.cz
OSA: Agility Hrušky
Editovat vlastní údaje

Vložení dalšího psa:

Kara Nica Z Kovárny
BOT
LA2
Výška: 56.00
Pohlaví: fena
Datum narození: 12.04.07
Editovat údaje o psovi

Proběhlé akce
09.05.10 Závody agility, Hrušky Kara Nica Z Kovárny

Pořádané akce budoucí
20.06.10 | Agility, Hrušky
Změnit Platby Evidovat zaplacené Údaje o závodnících Stáhnout startovku

Pořádané akce proběhlé
09.05.10 Závody agility Hrušky Vložit výsledky

Příloha 5: Screenshoty klientské aplikace

Správa závodů agility

File Edit

Závody agility, Hrušky 09.05.10

Export dat

Agility veteráni MA Agility MA Agility veteráni LA Agility veteráni MA Jumping SA Jumping MA Jumping veteráni LA Jumping veteráni MA

Zkouška A2 SA Zkouška A2 MA Zkouška A2 LA Zkouška A3 LA Zkouška A3 MA Zkouška A3 SA Agility SA Agility veteráni LA

Výběr běhů Start 09.05.10 Jumping LA Agility LA Zkouška A1 LA Zkouška A1 MA Zkouška A1 SA

Výběr běhů

Přidat nový běh

Datum běhu:

Typ běhu:

Velikostní kat.:

Součty běhů

Součet 2 běhů

Název součtu:

Datum běhu	Typ běhu	Velikostní kategorie	Odebrání
09.05.10	Jumping	LA	Odebrat
09.05.10	Agility	LA	Odebrat
09.05.10	Zkouška A1	LA	Odebrat
09.05.10	Zkouška A1	MA	Odebrat
09.05.10	Zkouška A1	SA	Odebrat
09.05.10	Zkouška A2	SA	Odebrat
09.05.10	Zkouška A2	MA	Odebrat
09.05.10	Zkouška A2	LA	Odebrat
09.05.10	Zkouška A3	LA	Odebrat
09.05.10	Zkouška A3	MA	Odebrat
09.05.10	Zkouška A3	SA	Odebrat
09.05.10	Agility	SA	Odebrat
09.05.10	Agility veteráni	LA	Odebrat
09.05.10	Agility veteráni	MA	Odebrat
09.05.10	Agility	LA	Odebrat
09.05.10	Agility veteráni	MA	Odebrat
09.05.10	Agility veteráni	SA	Odebrat
09.05.10	Jumping	MA	Odebrat
09.05.10	Jumping veteráni	LA	Odebrat
09.05.10	Jumping veteráni	MA	Odebrat

Správa závodů agility

File Edit

Závody agility, Hrušky 09.05.10

Export dat

Agility veteráni MA Agility MA Agility veteráni LA Agility veteráni MA Jumping SA Jumping MA Jumping veteráni LA Jumping veteráni MA

Zkouška A2 SA Zkouška A2 MA Zkouška A2 LA Zkouška A3 LA Zkouška A3 MA Zkouška A3 SA Agility SA Agility veteráni LA

Výběr běhů Start 09.05.10 Jumping LA Agility LA Zkouška A1 LA Zkouška A1 MA Zkouška A1 SA

Zkouška A1 LA

Typ běhu: Zkouška A1

Datum běhu: 09.05.10

Rozhodčí:

Délka tratě: metrů

Počet překážek:

Standardní čas: sekund

Maximální čas: sekund

Postup. rychlost: 2,96 m/s

Číslo	Pavod	Pes	Plemeno	Chyby	Odměnou	Čas	Trest. body za čas	Trest. body oeklem	Známka	Postup rychlost	Pořadí	DIS	Nahoru	DoLů	Nestartuje
1	Lucie Burgetová	Jump Jet	BOC	0	0	29.75	0.00	0.00	V	4,97	1	N...	▲	v	Startuje
2	Lukáš Blažek	Baldrick Nick	BOT	0	0	32.41	0.00	0.00	V	4,57	2	N...	▲	v	Startuje
3	Barbora Valentínová	Deborah	flat coated retrivr	0	0	36.46	0.00	0.00	V	4,06	3	N...	▲	v	Startuje
4	Daniela Nováková	Jašíka	MOK	0	0	37.64	0.00	0.00	V	3,93	4	N...	▲	v	Startuje
5	Ludmila Bučková	Ben	kříženec	0	0	40.03	0.00	0.00	V	3,70	5	N...	▲	v	Startuje
6	Simona Švestková	Mark Twain	biрад	0	0	43.59	0.00	0.00	V	3,40	6	N...	▲	v	Startuje
7	Petra Palečková	Charlie	MOK	1	0	34.06	0.00	5.00	V	4,35	7	N...	▲	v	Startuje
8	Barbora Valentínová	All for One	BOC	2	0	33.56	0.00	10.00	VD	4,41	8	N...	▲	v	Startuje
9	Radka Bilavčíková	Airon	AUO	1	1	38.65	0.00	10.00	VD	3,83	9	N...	▲	v	Startuje
10	Maritna Příkrá	Cesta	BOC	3	1	33.52	0.00	20.00	D	4,42	10	N...	▲	v	Startuje
11	Barbora Klobasová	Frodo	irský teriér	5	1	43.28	0.00	30.00	BO	3,42	11	N...	▲	v	Startuje
12	Václav Hůžl	Cooper	zlatý retrivr	5	2	41.53	0.00	35.00	BO	3,56	12	N...	▲	v	Startuje
13	Dana Straková	Crazy	BOT	2	1	0.0	0.00	15.00	DIS	∞	DIS	Výsledky	▲	v	Startuje
14	Eva Langová	Albert	ESP	0	0	0.0	0.00	0.00	DIS	∞	DIS	▲	v	Startuje	
15	Iveta Dupalová	Altaban	BOC	0	0	0.0	0.00	0.00	DIS	∞	DIS	▲	v	Startuje	
16	Sandra Letnerová	Genesis	BOC	0	0	0.0	0.00	0.00	DIS	∞	DIS	▲	v	Startuje	
1000	Zuzana Jašková	Connie	AUO	0	0	0.0	0.00	0.00	DIS	∞	NEST	N...	▲	v	Nestartuje