



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

APLIKACE PRO SPRÁVU DATABÁZE PLAVECKÉHO ODDÍLU

APPLICATION FOR MANAGING A SWIM CLUB DATABASE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jan Vencel

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Dydowicz, Ph.D.

BRNO 2019

Zadání diplomové práce

Ústav:	Ústav informatiky
Student:	Bc. Jan Vencel
Studijní program:	Systemové inženýrství a informatika
Studijní obor:	Informační management
Vedoucí práce:	Ing. Petr Dydowicz, Ph.D.
Akademický rok:	2018/19

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Aplikace pro správu databáze plaveckého oddílu

Charakteristika problematiky úkolu:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrh řešení, přínos práce
Závěr
Seznam použité literatury

Cíle, kterých má být dosaženo:

Vytvořit desktopovou aplikaci pro správu databáze plaveckého oddílu ASK Blansko, která by umožňovala hlavnímu trenérovi evidovat závody, styly, plavce a jejich výkony. Dát uživateli možnost vybírat, třídit uložená data a vytvářet z nich vlastní nebo použít už připravené sestavy. Poskytnout funkcionalitu pro tvorbu přihlášek na závody, import výsledků ze závodů, import vlastních dat z již existující databáze a v neposlední řadě zálohu a obnovu databáze.

Základní literární prameny:

BASL, J. a R. BLAŽÍČEK. Podnikové informační systémy. Podnik v informační společnosti. Praha: Grada, 2008. 283 s. ISBN 978-80-247-2279-5.

MOLNÁR, Z. Automatizované informační systémy. Praha: Strojní fakulta ČVUT, 2000. 126 s. ISBN 80-01-02269-2.

MOLNÁR, Z. Efektivnost informačních systémů. Praha: Grada Publishing, 2000. 142 s. ISBN 80-716-410-X.

ŘEPA, V. Analýza a návrh informačních systémů. Praha: Ekopress, 1999. 403 s. ISBN 80-86119--3-0.

SODOMKA, P. a H. KLČOVÁ. Informační systémy v podnikové praxi. Brno: Computer Press, 2010. 501 s. ISBN 978-80-251-2878-7.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně dne 28.2.2019

L. S.

doc. RNDr. Bedřich Půža, CSc.
ředitel

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Diplomová práce se zaměřuje na návrh, tvorbu a zavedení desktopové aplikace pro správu databáze plaveckého oddílu ASK Blansko. Analyzuje současný stav procesů v oddíle a srovnává je s novodobou praxí. Obsahuje popis veškerých funkcí aplikace spolu s ekonomickým zhodnocením projektu. Představuje kompletní dodávku hotového řešení z důvodu neefektivního a zastaralého způsobu vedení oddílové databáze. Zdůrazňuje úspěšné zavedení aplikace do oddílové praxe.

Klíčová slova

informační systém, aplikace, databáze, plavecký oddíl, C#, .NET, MS SQL Server, Entity Framework, ERD

Abstract

The diploma thesis focuses on the design, creation and implementation of a desktop application for managing a swimming database of the ASK Blansko swim club. It analyzes the current state of processes in the club and compares them with modern practice. It contains a description of all the functions of the application together with the economic evaluation of the project. It represents a complete solution because of the inefficient and outdated way of keeping the club database. It emphasizes the successful implementation of the application in the club practice.

Key words

information system, application, database, swim club, C#, .NET, MS SQL Server, Entity Framework, ERD

Bibliografická citace

VENCEL, Jan. *Aplikace pro správu databáze plaveckého oddílu* [online]. Brno, 2019 [cit. 2019-04-18]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/118399>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Petr Dydowicz.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 18. dubna 2019

.....
podpis autora

Poděkování

V této sekci bych chtěl poděkovat vedoucímu práce Ing. Petrovi Dydowiczovi, Ph.D. za poskytnutí odborné pomoci. Dále bych rád poděkoval vedení plaveckého oddílu ASK Blansko a jeho trenérům za poskytnutí potřebných materiálů nutných pro vypracování této diplomové práce.

OBSAH

ÚVOD.....	12
VYMEZENÍ PROBLÉMU A CÍLE PRÁCE	13
1 TEORETICKÁ VÝCHODISKA PRÁCE.....	14
1.1 DLL knihovna	14
1.2 Platforma .NET a programovací jazyk C#	15
1.2.1 Platforma .NET.....	15
1.2.2 Jazyk C#.....	17
1.3 Visual Studio	18
1.4 Databáze	18
1.4.1 Modely	19
1.4.2 Výběr databáze	20
1.4.3 Microsoft SQL Server.....	21
1.5 Microsoft SQL Server Management Studio	21
1.6 MVC architektura	22
1.7 Entity Framework.....	23
1.7.1 Přístupy k vývoji databázové aplikace.....	24
1.7.2 Konfigurace databázového modelu	24
1.8 SQL.....	25
1.9 LINQ.....	25
1.10 CSV	26
1.11 XML	27
1.11.1 XML versus CSV.....	27
1.11.2 XML Schema.....	27
1.12 Funkční modelování	28
1.12.1 Slovní popis funkčního modelu	28
1.12.2 Diagram případů užití	29
1.13 Datové modelování.....	29
1.13.1 Konceptuální model.....	30
1.13.2 Fyzický model.....	30

1.14	UML	31
1.15	Enterprise architect	32
1.16	Material Design	33
1.17	SWOT analýza.....	33
1.17.1	Silné stránky	34
1.17.2	Slabé stránky.....	35
1.17.3	Příležitosti	35
1.17.4	Hrozby	35
1.18	Řetězec data – informace – znalosti	35
1.18.1	Data	35
1.18.2	Informace	36
1.18.3	Znalosti	36
1.19	GDPR	37
2	ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE.....	38
2.1	Představení plaveckého klubu	38
2.1.1	Vznik a vývoj.....	38
2.2	Popis současného stavu	39
2.3	Organizační struktura oddílu	39
2.4	Činnost oddílu	40
2.5	Členství v ČSPS	41
2.5.1	Krajský svaz ČSPS	41
2.6	Definice procesu	41
2.6.1	Uložení výsledků	41
2.6.2	Získání požadované informace	42
2.7	Prostředky k výkonu procesu	42
2.7.1	Papírová kartotéka	42
2.7.2	MS Excel a MS Access.....	43
2.7.3	Centrální evidence	44
2.8	Analýza konkurenčních aplikací	44
2.9	Analýza současného stavu procesu	45
2.9.1	Přihlášky na závody	46
2.9.2	Výsledky ze závodů	47

2.9.3	Zaznamenávané údaje.....	48
2.9.4	Evidence výsledků	49
2.9.5	Výběr dat.....	50
2.9.6	Prezentace informací.....	51
2.9.7	SWOT analýza před začátkem projektu	51
2.9.8	Zhodnocení procesu	52
2.10	Analýza zpracovávaných osobních údajů	52
2.11	Závěr provedených analýz.....	53
2.12	Požadavky oddílu na funkcionalitu aplikace	53
3	VLASTNÍ NÁVRH ŘEŠENÍ.....	55
3.1	Volba řešení.....	55
3.1.1	Způsob užití aplikace	55
3.2	Návrh databáze	56
3.2.1	Diagram případů užití	56
3.2.2	Konceptuální návrh.....	58
3.2.3	Fyzický návrh	58
3.3	Funkční návrh.....	60
3.3.1	Hlavní okno.....	60
3.3.2	Tabulky	61
3.3.3	Sestavy	68
3.3.4	Přihlášky	69
3.3.5	Import.....	74
3.3.6	Export.....	77
3.3.7	Záloha a obnova databáze.....	79
3.3.8	Konfigurace	79
3.3.9	Uživatelská podpora	81
3.4	Grafický návrh.....	81
3.4.1	Dialogová okna	81
3.4.2	Logo aplikace.....	82
3.5	Zavedení do provozu	83
3.5.1	Testování.....	83
3.5.2	Zkušební provoz	84

3.5.3	Ostrý provoz a údržba.....	84
3.6	Zhodnocení projektu.....	85
3.6.1	Ekonomické zhodnocení.....	85
3.6.2	Přínos projektu.....	86
3.6.3	SWOT analýza po ukončení projektu.....	88
3.6.4	Budoucí rozvoj.....	89
3.6.5	Licence.....	90
ZÁVĚR		92
SEZNAM OBRÁZKŮ		98
SEZNAM TABULEK.....		100

ÚVOD

Informační systémy se v dnešní době uplatňují nejen ve firmách, ale i ve sportovních organizacích a klubech. Provozování sportovní činnosti těmito spolky si vyžaduje zacházení s velkým množstvím informací a dat. Přesto, že nemáme co dočinění s objemy dat jako například v národních pojišťovacích institucích, i v případě sportovního oddílu okresní působnosti je potřeba s daty a informacemi zacházet strukturovaným, bezchybným a bezpečným způsobem. A to tak, aby byly splněny podmínky národního svazu pro daný sport a ulehčena snaha oddílu konkurovat ostatním klubům v České republice. Jelikož dnes žijeme v ryze počítačové době, není jiného přípustného způsobu, jak vytvořit informační systém, než naprogramovat internetovou aplikaci provozovanou na serveru nebo program pro stolní nebo přenosný počítač.

Navzdory dnes mnohonásobně populárnějšímu prvnímu způsobu, je diplomová práce určena tomu druhému. V rámci práce bude tedy vytvořen program pro stolní i přenosné počítače, který bude umožňovat evidenci plaveckých dat. Pod pojmem plavecká data si lze představit údaje o plavcích v oddílu, absolvovaných závodech a zaplavaných časech. S využitím těchto dat bude oddíl mimo jiné schopen vytvářet přihlášky na závody a specializované výběry a sestavy. Program bude určen plaveckému oddílu ASK Blansko, jehož jsem dlouholetým aktivním členem.

Cílem práce je úspěšné zavedení aplikace do ostrého provozu v oddíle a spolehlivé poskytování deklarovaných funkcí. V diplomové práci se dočteme nejen o úspěšném či neúspěšném splnění stanoveného cíle, ale hlavně o tom, co mu předchází. Stěžejní strukturu práce tvoří části určené ke studiu a představení použitých technologií, analýze současné situace v oddíle i mimo něj a popisu návrhu a hodnocení vlastního řešení.

VYMEZENÍ PROBLÉMU A CÍLE PRÁCE

Primárním cílem práce je navrhnout, implementovat a zavést do provozu aplikaci pro plavecký oddíl působící ve městě Blansku. To zahrnuje důkladně analyzovat aktuální stav oddílu, definovat a rozebrat prováděné procesy, formulovat požadavky, funkcionality a vlastnosti vyvíjené aplikace. Následně provést návrh a implementovat program s využitím aktuálně dostupných technologií a nejlepších praktik v oboru.

Oddíl používá na dnešní dobu zastaralé postupy a metody pro tvorbu, správu a využití databáze plaveckých dat. Cílem zavedení této aplikace do prostředí oddílu je zrychlení, zefektivnění a zjednodušení procesů evidence výsledků ze závodů, správy databáze, vytváření přihlášek na závody, reportů a sestav. To bude znamenat následující změny, které lze interpretovat jako dílčí cíle:

- snížení nároků na trenéry pověřením jednoho hlavního trenéra obsluhou aplikace,
- zefektivnění chodu oddílu zrychlením administrativních činností,
- snížení chybovosti dat omezením lidského faktoru,
- zjednodušení práce pořadatelům závodů, na které se oddíl hlásí, vytvářením jednotného formátu přihlášek,
- příznivější prostředí pro plavce a jejich výkony díky dostupnosti nových přehledů a statistik,
- zmenšení nákladů vedení oddílu snížením trenérů potřebných v procesech správy databáze.

Vedle primárního cíle popsaného výše, má aplikace ještě cíl sekundární. Ten zahrnuje iniciativu rozšířit působnost aplikace za hranice blanenského oddílu a zavést program s případnými úpravami i do jiných plaveckých oddílů, které projeví zájem. Protože takové oddíly existují, budeme při návrhu aplikace postupovat tak, aby byla aplikace výhledově schopna splnit i sekundární cíl. Tím se již ale v práci nebudeme výrazným způsobem zabývat, neboť by mohl být přesáhnut její rozsah.

1 TEORETICKÁ VÝCHODISKA PRÁCE

V této části se zaměříme na představení technologií a metod, které budou použity k naplnění cíle diplomové práce. Seznámení se s pojmy je klíčové pro pochopení problematiky, kterou se diplomová práce zabývá.

1.1 DLL knihovna

V textu je často skloňován pojem knihovna. Pojd'me si proto ujasnit jeho význam. DLL¹ je soubor obsahující kód a data, který lze použít více než jedním programem zároveň. Koncept zavedla společnost Microsoft pro operační systém Windows.

DLL knihovny jsou v podstatě shodné se spustitelnými EXE soubory, proto je vhodné použití DLL z důvodu větší přehlednosti a modulárnosti programu. DLL soubory není možné přímo spouštět. Ke spuštění vyžadují EXE soubor.

Pro vývojáře aplikací je operační systém Windows kolekcí DLL souborů, které jsou aktualizovány. To umožňuje aplikacím vytvořeným pro jednu verzi operačního systému fungovat i na jeho novějších verzích. Knihovny DLL jsou spouštěny v rámci probíhajícího programu a sdílejí s ním paměťový prostor a přístupová práva. Tím je zajištěna nízká režie při použití. Zároveň ale pro volající program není zajištěna ochrana proti případným chybám v DLL knihovně. Pokud tedy spouštěný program nenalezne odpovídající DLL knihovnu, velice pravděpodobně se úspěšně nespustí.

Problémy s DLL knihovnami jako nemožnost spuštění některých programů kvůli chybějícím knihovnám, chybné fungování programů kvůli nekompatibilním verzím knihoven nebo hromadění nevyužívaných knihoven a jejich verzí se souhrnně nazývají DLL peklo. Jedním z řešení, které Microsoft poskytuje je zavedení platformy .NET. [1]

¹ Z angl. Dynamic-link library – dynamicky linkovaná [připojovaná] knihovna

1.2 Platforma .NET a programovací jazyk C#

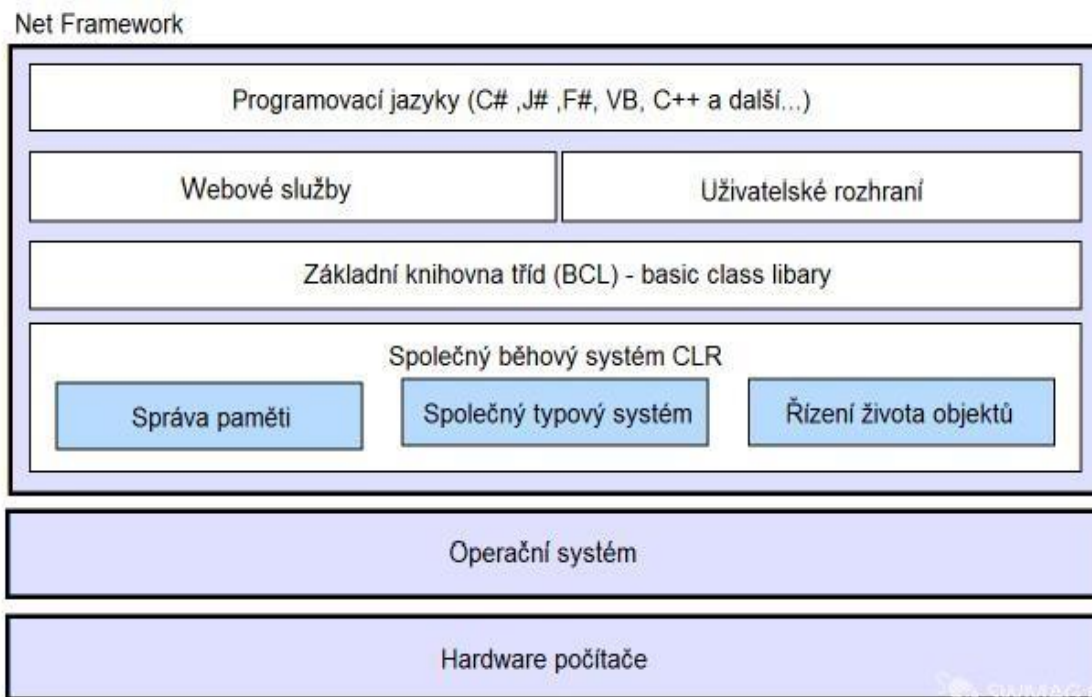
Jazyk C# byl navržen a implementován firmou Microsoft pro platformu .NET. Ze všech vyšších programovacích jazyků, v nichž lze vyvíjet programy pro tuto platformu, poskytuje největší možnosti. Důvodem je jejich souběžný vývoj. Hovoříme-li o C#, musíme mluvit i o platformě .NET, a hlavně o ní – jinak to prostě nejde. Jazyk C# totiž neobsahuje žádné vlastní knihovny. Opírá se výhradně o knihovny tohoto prostředí.

1.2.1 Platforma .NET

Platforma .NET byla odborné veřejnosti představena poprvé v roce 2001. V roce 2002 byla oficiálně uvolněna její první verze a spolu s ní i Visual Studio 2002, první verze vývojových nástrojů pro ni. Jedná se o prostředí potřebné pro běh aplikací a nabízející jak spouštěcí rozhraní, tak potřebné knihovny. Platforma .NET měla být odpovědí na řadu problémů, které se nahromadily v programování pro Windows. Například:

- koexistence různých verzí knihoven,
- spolupráce různých programovacích jazyků,
- podpora webových aplikací,
- zabezpečení softwaru a další.

Prostředí .NET představovalo zpočátku nadstavbu nad operačním systémem. Od Windows Vista je již jeho nedílnou součástí.



Obrázek 1.1 - Základní struktura prostředí .NET (Zdroj: [2])

Z Obrázku 1.1 vidíme, že na hardware počítače je položen operační systém a nad ním stojí .NET. Tato platforma se skládá z několika vrstev. Jejím základem je společný běhový systém CLR². Ten definuje společný typový systém, nástroje pro správu paměti a další součásti.

Nad CLR stojí základní knihovna tříd BCL³. Tato knihovna obsahuje řadu nástrojů, počínaje atomickými datovými typy, jako je System.Int32 (kterému odpovídá typ int v C#), přes typ System.Object, který je společným předkem všech datových typů v .NET, až po třídy představující datové kontejnery, nástroje pro vstupní a výstupní operace, práci s regulárními výrazy, zpracování souborů v jazyce XML, práci se sítí, zpracování lokálního nastavení atd.

Nad touto knihovnou stojí knihovny grafického uživatelského rozhraní, webových služeb, přístupu k databázím apod. Nejvyšší vrstvu tvoří překladače programovacích jazyků, které překládají zdrojový kód do mezijazyka IL⁴. Jedná se v podstatě o strojový

² Z angl. Common Language Runtime

³ Z angl. Basic Class Library

⁴ Z angl. Intermediate Language

(binární) kód, který má ale o poznání jednodušší instrukční sadu a přímo podporuje objektové programování. [3] [4]

1.2.2 Jazyk C#

Jazyk C# spadá do kategorie jazyků s virtuálním strojem. Jedná se o nejmodernější kategorii jazyka, která je v současné době také nejrozšířenější a nejlepší volbou pro vývoj většiny aplikací.

Standard ECMA⁵ definuje současný design C# jako jednoduchý, moderní, mnohoúčelový a objektově orientovaný programovací jazyk. Jazyk a jeho implementace poskytuje podporu pro principy softwarového inženýrství, jakými jsou kupř. hlídání hranic polí, detekce použití neinicizovaných proměnných a automatický garbage collector⁶. Jazyk C# výrazně navyšuje produktivitu programátorů. Jazyk je vhodný pro vývoj softwarových komponent distribuovaných v různých prostředích. [5]

Zdrojový kód je nejprve přeložen do mezikódu. Tento mezikód je potom díky jednoduchosti relativně rychle interpretovatelný tzv. virtuálním strojem (tedy interpretem, v případě .NET je to CLR). Výsledkem je strojový kód pro náš procesor. Tímto přístupem získáváme následující výhody. [6] [4]

- Odhalení chyb ve zdrojovém kódu – Díky kompilaci do IL jednoduše odhalíme chyby ve zdrojovém kódu.
- Stabilita – Díky tomu, že interpret kódu rozumí, zastaví nás před vykonáním nebezpečné operace a na chybu upozorní.
- Jednoduchý vývoj – Máme k dispozici pokročilé datové struktury a knihovny, správu paměti za nás provádí garbage collector.
- Málo zranitelný kód – Aplikace se šíří jako zdrojový kód v IL, není tedy úplně jednoduše lidsky čitelná.

⁵ Z angl. European Computer Manufacturers Association – Evropská asociace výrobců počítačů

⁶ Způsob automatické správy paměti – speciální algoritmus vyhledává a uvolňuje úseky paměti, které již program nebo proces nepoužívá.

1.3 Visual Studio

Visual Studio je vývojové prostředí (IDE⁷) od Microsoftu. Obsahuje editor kódu podporující IntelliSense⁸ a refaktorování⁹. Integrovaný debugger¹⁰ pracuje jak na úrovni kódu, tak na úrovni stroje. Další vestavěné nástroje zahrnují návrhář formulářů pro tvorbu aplikací s GUI¹¹, návrhář webu, tříd a databázových schémat.

Nová verze vychází přibližně jednou za rok až za dva. Každá verze Visual Studia se distribuuje s různými verzemi a přívlasky. Většina verzí je draze placených a jsou určený firmám. Pro potřeby diplomové práce byla využita verze Community z roku 2017, která je zdarma.

Visual Studio má ve znaku symbol nekonečna a jeho možnosti se tomu podobají. Ve výchozím nastavení Visual Studio podporuje několik programovacích jazyků a platform. Navíc díky různým doplňkům je možné si do Visual Studia doinstalovat ještě několik dalších.

Kvalita podpory různých technologií se dá rozdělit do tří skupin. Výtečná, dobrá a špatná. Technologie, které patří Microsoftu mají podporu zpravidla vynikající, Důvodem je relativně jednodušší vývoj nástrojů pro vlastní produkt. Příkladem takového produktu je programovací jazyk C#. O trochu těžší to mají nástroje kde Microsoft kontrolu nemá, ale i přesto je v jeho zájmu nabídnout co nejlepší podporu. Jsou to například nástroje pro vývoj webů (HTML, CSS, JavaScript). Úplně nejhůře dopadají nástroje, které Microsoft vůbec nezajímají. Navíc jsou pro něj dokonce konkurencí. Příkladem takové technologie je programovací jazyk Java, který ani přes spoustu komunitních pokusů stále ve Visual Studiu provozovat nelze. [7]

1.4 Databáze

Databáze je systém souborů s pevnou strukturou záznamů. Tyto soubory jsou na sebe navzájem navázány pomocí klíčů. V širším smyslu považujeme za součást databáze

⁷ Z angl. Integrated Development Environment

⁸ Funkce při editaci kódu mající informativní charakter

⁹ Proces modifikace kódu usnadňující údržbu, pochopení a rozšiřitelnost

¹⁰ Softwarový nástroj, který se používá pro hledání chyb při vývoji software ve fázi ladění

¹¹ Z angl. Graphical User Interface – Grafické uživatelské rozhraní

i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Takový software označujeme pojmem systém řízení báze dat (SŘBD). Slovem databáze mohou být myšlena jak samotná uložená data, tak i software SŘBD. Hlavní funkce SŘBD jsou tyto:

- definice dat – tvorba, úprava a odstranění definic, které určují organizaci dat,
- aktualizace dat – vkládání, úprava a odstraňování dat,
- získání dat – poskytování informací pro přímé použití nebo pro další zpracování jinými aplikacemi,
- administrace – registrace a monitorování uživatelů, vynucování zabezpečení dat, sledování výkonu, zachování integrity dat atd.

Po fyzické stránce jsou databáze provozovány databázovými servery. Jedná se o jednoúčelové počítače, které uchovávají aktuální databázi, provozují SŘBD a související software. Databázové servery jsou většinou multiprocessorové počítače s velkou pamětí a diskovými poli typu RAID¹². RAID se používá pro obnovu dat v případě selhání některého z disků. [8]

1.4.1 Modely

Předchůdcem databází byly papírové kartotéky. Umožňovaly uspořádávání dat podle různých kritérií a přidávání nových položek. Veškeré operace s nimi prováděl přímo člověk. Správa takových kartoték byla v mnohém podobná správě dnešních databází.

S nástupem informačních technologií se systémy pro správu databáze klasifikují podle databázových modelů, které podporují. Relační databáze byly dominantní v 80. letech. Data tohoto modelu jsou uspořádána do řádků a sloupců v sérii tabulek. Většina z nich používá dotazovací jazyk SQL¹³ pro vkládání a výběr dat. Na relačním databázovém modelu je založen program vyvíjený v rámci diplomové práce.

V devadesátých letech začalo nabývat popularitu objektově orientované programování. Tento způsob programování zanedlouho začal prostupovat i do prostředí databází. Programátoři a návrháři začali považovat data ve svých databázích za objekty. Objektově orientované databáze měly podle předpokladů vytlačit relační systémy. Tento

¹² Z angl. Redundant Array of Independent Disks – vícenásobné diskové pole nezávislých disků

¹³ Z angl. Structured Query Language

přístup však zaznamenal obtíže týkající se překládání mezi programovými objekty a databázovými tabulkami. Původní předpoklady se tak nenaplnily a vznikla kompromisní objektově-relační technologie. [9]

V roce 2000 se staly populární nerelační databáze, označované jako NoSQL. Databáze NoSQL poskytuje mechanismus pro ukládání a vyhledávání dat, který je modelován jinými prostředky, než je tabulkové schéma u relačních databází. Rozdílem je například stromová nebo grafová struktura pro ukládání dat.

Dalším moderním systémem je systém pro správu relačních databází NewSQL. Takové systémy se zaměřují na dosažení takové úrovně škálovatelnosti jako nabízí systémy NoSQL a současně dodržování vlastností ACID¹⁴, které zajišťují tradiční relační databáze. [10]

1.4.2 Výběr databáze

Jedním z hlavních témat při úvahách nad budoucí aplikací bylo téma výběru správné databáze. Výběr vhodné databáze pro aplikaci je závažným a dlouhodobým rozhodnutím, neboť jakékoli pozdější změny mohou být obtížné, a dokonce i poměrně drahé.

Existuje rozdíl mezi výběrem libovolné databáze pro aplikaci a výběrem správné databáze pro tuto aplikaci. Podívejme se na některé z důležitých faktorů, které je třeba vzít v úvahu při výběru databáze pro aplikaci: [11]

- Struktura dat – rozhodujeme se o tom, jak se data budou ukládat a načítat,
- Velikost dat – bereme v úvahu množství dat, které potřebujeme k ukládání a načítání kritických aplikačních dat,
- Rychlost a škálovatelnost – určíme rychlost, kterou potřebujeme pro čtení dat z databáze a zápis dat do databáze,
- Dostupnost dat – známe počet osob nebo uživatelů, kteří budou současně přistupovat k databázi,
- Bezpečnost a zabezpečení dat – zkontrolujeme úroveň zabezpečení, kterou databáze poskytuje uloženým datům.

¹⁴ Skupina vlastností, které musí dodržet každá databázová transakce (atomicita, konzistence, izolovanost a trvalost)

1.4.3 Microsoft SQL Server

Po zvážení všech faktorů padnula volba na MS SQL. V C# se pracuje téměř výhradně s databází MS SQL. Teoreticky lze v C# pracovat s libovolnou databází (např. SQLite, MySQL a podobně). Prakticky jsou však všechny nástroje a technologie optimalizovány pro MS SQL. Například platforma Entity Framework. [12]

MS SQL Server je databázový a analytický systém, speciálně vyvinutý pro internetové obchody nebo datové sklady. SQL Server dokáže díky správě podnikových dat zvýšit efektivitu firmy. Jeho vývoj započal roku 1985 firmami Microsoft a IBM. Databáze je založená na architektuře klient/server a je v ní integrována podpora XML. Celý systém je navržen tak, aby zvládl uskutečňovat velké množství transakcí.

MS SQL Server je primárně postavený na struktuře navzájem propojených tabulek. Pomocí cizích klíčů jsou propojeny související datové prvky v různých tabulkách. Tím je vyloučena potřeba redundantního ukládání dat na více místech v databázi. Relační model také poskytuje referenční integritu a další omezení integrity pro zachování přesnosti dat. Tyto vlastnosti jsou součástí zásad ACID, které zaručují, že databázové transakce jsou spolehlivě zpracovány. [13]

1.5 Microsoft SQL Server Management Studio

SQL Server Management Studio je nástroj pro konfiguraci a správu všech komponent v rámci databáze Microsoft SQL Server. Nástroj obsahuje jak editory skriptů, tak grafické nástroje, které pracují s objekty a funkcemi databáze.

Centrálním prvkem produktu SQL Server Management Studio je objektový průzkumník, který umožňuje uživateli procházet, vybírat a ovlivňovat kterýkoli z objektů v rámci databáze. Nástroj může být použit k vizuálnímu sledování a analýze dotazů nebo optimalizaci výkonu databáze. SQL Server Management Studio lze také použít k vytvoření nové databáze, ke změně existující schématu databáze přidáním nebo úpravou tabulek a indexů. [14]

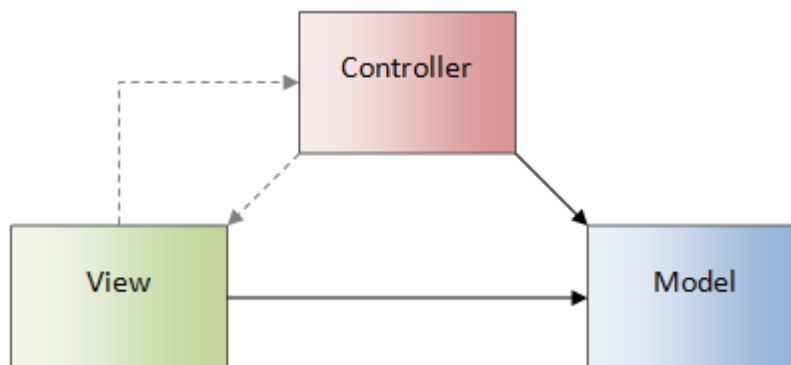
1.6 MVC architektura

MVC je oblíbený architektonický vzor, který se uchytil zejména na webu, ačkoli původně vznikl pro vývoj aplikací pro stolní počítače. Architektura MVC dělí aplikaci na 3 logické části tak, aby je šlo upravovat samostatně a dopad změn byl na ostatní části co nejmenší. Tyto tři části jsou Model, View a Controller. [15]

Model reprezentuje data a business logiku aplikace. Mohou to být výpočty, databázové dotazy, validace a podobně. Model neví, odkud data v parametrech přišla a ani jak budou výstupní data zformátována a vypsána.

View zobrazuje uživatelské rozhraní. Data získaná pomocí modelu zobrazuje uživateli a také mu umožňuje data upravovat. Tato vrstva je aktivována požadavkem na zobrazení údajů, který přichází od Controlleru.

Controller má na starosti tok událostí v aplikaci a obecně aplikační logiku. Zprostředkovává komunikaci mezi vrstvami Model a View.



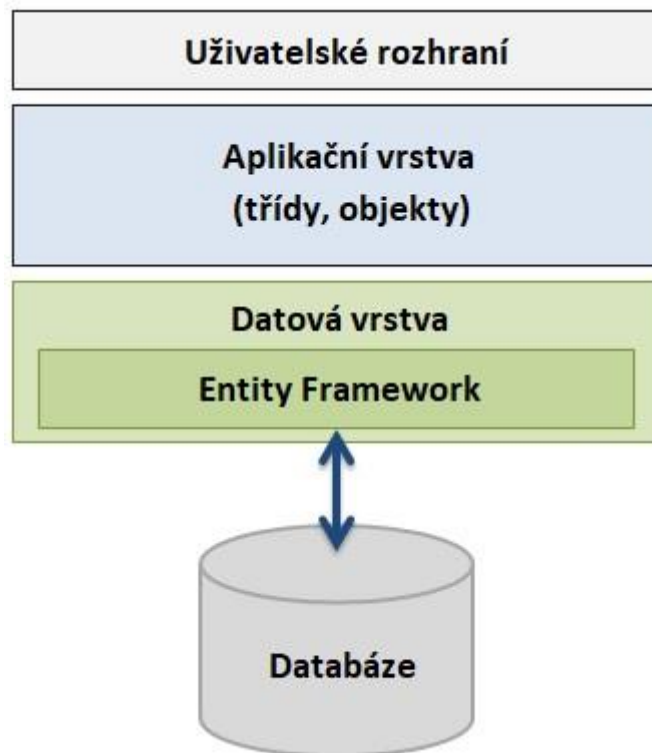
Obrázek 1.2 - Návaznost jednotlivých komponent (Zdroj: [15])

Jak naznačují šipky na *Obrázku 1.2*, v architektuře MVC v principu existují pouze dvě přímé vazby. Controller a View přistupují do Modelu, aby upravili, respektive zobrazili jeho data. v praxi se občas setkáme ještě s poměrně častou vazbou mezi Controllerem a View. Nikdy nesmí existovat přímá vazba Modelu na ostatní dvě komponenty. Když se data Modelu změní, příslušná View jsou upozorněna nějakým notifikačním mechanismem. [16]

1.7 Entity Framework

Entity Framework je otevřený software¹⁵ představující ORM¹⁶ rámec pro .NET aplikace podporované společností Microsoft. ORM zajišťuje automatickou konverzi dat mezi relační databází a objektově orientovaným programovacím jazykem. Entity Framework díky tomu umožňuje vývojářům pracovat s daty pomocí objektů tříd, aniž by se soustředili na podkladové tabulky databáze a sloupce, ve kterých jsou tato data uložena. Pomocí Entity Framework mohou vývojáři při práci s daty uvažovat na vyšší úrovni abstrakce. K vytváření a údržbě aplikace orientované na data stačí méně kódu ve srovnání s tradičními aplikacemi. [17]

Obrázek 1.3 znázorňuje, jak Entity Framework zapadá do naší aplikace. Entity Framework automaticky ukládá data z objektů aplikace do databáze a naopak.



Obrázek 1.3 - Pozice Entity Frameworku v aplikaci (Zdroj: [17])

Nyní uvedme některé z hlavních vlastností, které charakterizují Entity Framework:

¹⁵ Anglicky open-source software – programy, jejichž zdrojový kód byl poskytnut dalším vývojářům ke studiu, úpravě nebo vylepšení

¹⁶ Z angl. Object-relational mapping – objektově relační mapování

- Napříč platformám – EF funguje na systémech Windows, Linux a Mac.
- Dotazování – EF nám umožňuje používat dotazy jazyka LINQ (viz dále) k načtení dat z podkladové databáze. EF nicméně také umožňuje provádět surové SQL (viz dále) dotazy přímo do databáze.
- Sledování změn – EF sleduje změny provedené v instancích objektů, které je třeba uložit do databáze.
- Uložení – Voláním metody `SaveChanges()` provádí EF příkazy INSERT, UPDATE a DELETE do databáze .
- Ukládání do mezipaměti – Opakované dotazování vrátí data z vyrovnávací paměti namísto opětovného dotazu do databáze.

1.7.1 Přístupy k vývoji databázové aplikace

Existují tři různé přístupy, které můžeme použít při vývoji aplikace pomocí Entity Frameworku – Database First, Model First a Code First. Každý z přístupů vytváří databázi jinou posloupností úkonů.

Database First přístup spočívá v existenci databáze, ze které nám EF vygeneruje třídy a kontext. V Model First přístupu vytvoříme entity, vztahy a hierarchie dědičnosti přímo na vizuálním návrháři integrovaném v aplikaci Visual Studio a pak generujeme entity, kontextovou třídu a databázový skript z tohoto vizuálního modelu. Zbývajícím třetímu přístupu se říká Code First, pomocí kterého můžeme vytvořit C# třídu a EF nám podle ní automaticky vygeneruje databázovou tabulku a potřebný kontext. Tímto přístupem byla vytvořena aplikace v diplomové práci. [18]

1.7.2 Konfigurace databázového modelu

Konfigurace umožňuje přepsat výchozí chování EF při vytváření databáze. Konfigurace lze použít dvěma způsoby, a to pomocí rozhraní Fluent API a prostřednictvím atributů `DataAnnotation`.

Atributy jsou druh značky, která se umísťuje do hranatých závorek nad třídy nebo vlastnosti. Specifikujeme tak metadata dané třídy nebo vlastnosti. [19] Například:

- `[Key]` – určí sloupec jako primární klíč tabulky,

- *[Required]* - anotace informuje EF, že daná databázová položka musí být vyplněná,
- *[MaxLength(10),MinLength(5)]* – určí maximální a minimální délku položky

Fluent API je pokročilý způsob, jak specifikovat konfiguraci modelu. Pokrývá všechno, co jsme schopni dosáhnout s atributy, a navíc umožňuje i pokročilejší konfigurace, které s atributy nedokážeme. Anotace dat a rozhraní Fluent API mohou být použity společně, ale Code First přístup dává přednost Fluent API. V diplomové práci jsou využity oba způsoby. [20]

1.8 SQL

SQL¹⁷ je databázový programovací jazyk, který umožňuje definovat data, dotazovat se na ně a manipulovat s nimi. Díky syntaktickým a sémantickým pravidlům, která jsou odvozená z jednoduchých anglických vět, je srozumitelným a snadno pochopitelným. Jeho výhodou je také možnost volat jej z jiných programovacích jazyků jako C, JAVA, PHP apod. [21]

Jazyk SQL byl vyvinut firmou IBM na počátku 70. let jako dotazovací jazyk pro práci s velkými databázemi na počítačích střediskového typu. Cílem tvůrců SQL bylo vyvinout takový nástroj pro koncové uživatele, který by jim umožnil vybírat data z databáze přesně podle jejich individuálních požadavků a byl přitom co nejjednodušší. [22]

1.9 LINQ

LINQ¹⁸ je jednotný dotazovací jazyk integrovaný v C# pro načítání dat z různých zdrojů a formátů. Ten samý dotaz spustíme jak nad obyčejným polem, tak nad XML souborem nebo databází. [23]

LINQ je velmi podobný jazyku SQL a je to tedy jazyk deklarativní. Programu sdělíme, co hledáme a už nás většinou nezajímá, jakým způsobem pro nás data opravdu vyhledá. Z dotazu zapsaném v jazyce LINQ je vygenerován SQL dotaz, který je následně

¹⁷ Z angl. Structured Query Language

¹⁸ Z angl. Language Integrated Query

spuštěn nad databází. Pokročilejší programátor může vidět problém v generování neoptimálních dotazů určených pro velké objemy dat. Naopak výhodou je syntaktická kontrola dotazů při překladu programu. [24]

Dotazy LINQ vrací výsledky ve formě objektů. To nám umožňuje použít objektově orientovaný přístup na sadu výsledků a nemusíme se obtěžovat transformací rozdílných formátů výsledků na objekty.

1.10 CSV

Formát CSV¹⁹ je univerzální a velmi rozšířený textový formát pro výměnu dat mezi dvěma systémy. Je podporovaný téměř všemi nástroji, které pracují s daty. Je standardizovaný v RFC-4180.

Soubor CSV se skládá z libovolného počtu řádků, které obsahují libovolný počet sloupců. Standard RFC uvádí, že oddělovačem řádků je posloupnost znaků CRLF²⁰ a oddělovačem sloupců je čárka (.). Většina nástrojů nám však umožňuje libovolně zvolit tyto znaky podle potřeby. Často se setkáme s případem, kdy se jako oddělovače sloupců používají svislítka či středníky.

Na prvním řádku může být umístěna hlavička, která popisuje názvy jednotlivých sloupců. Formát hlavičky je stejný jako formát ostatních sloupců. V případě, že je v souboru definována hlavička, měl by být počet sloupců na každém řádku stejný jako v hlavičce.

Formát CSV nespecifikuje žádnou znakovou sadu a soubor sám o sobě neobsahuje žádné údaje o tom, jak jej strojově načíst. Proto je nutné před výměnou CSV souboru přesně dohodnout formát sloupců a řádků včetně kódování souboru. Tento problém je znatelný i při práci na diplomové práci. Textový soubor ve formátu CSV s výsledky ze závodů je na straně pořadatelů generován programem pro vedení závodů. Naštěstí je v dnešní době po republice rozšířen pouze jeden druh programu. Takže i když nikde není explicitně popsán význam a formát jednotlivých polí souboru, dá se ze vzorových dat vypořádat. [25] [10]

¹⁹ Z angl. Comma-separated values

²⁰ Sekvence znaků, která v počítačovém souboru označuje konec řádku

1.11 XML

XML²¹ pochází z oblasti zaměřené na uchování a zpracování textových dokumentů. Jeho „otec“ je SGML²² a jeho „bratr“ je všeobecně známý a používaný jazyk HTML²³.

XML je standardem W3C²⁴ a v současné době se jedná o jeden z nejdůležitějších formátů výměny dat strukturovaným způsobem. XML během posledních deseti let doznal značného rozšíření také díky svým velmi dobrým vlastnostem. Jedná se zejména o následující vlastnosti:

- XML popisuje data nezávisle na platformě
- XML je otevřený formát (značky lze doplňovat libovolně podle potřeby – rozdíl oproti HTML)
- Značky mají pevnou gramatiku – je možné programově kontrolovat správnost struktury
- Značky jasně popisují význam uložených dat.

1.11.1 XML versus CSV

XML je modernější způsob výměny dat než popsané CSV. Na rozdíl od CSV je poměrně dobře čitelný. Za tuto čitelnost platíme diskovým prostorem. Stejný objem informace uložíme ve formátu XML na třikrát větším prostoru než v případě CSV. Je tomu tak proto, že každý datový prvek má přidruženou značku pro začátek a konec. Dalšími výhodami XML oproti CSV jsou například možnost ověřit správnost pomocí XSD schémat nebo skutečnost, že XML je nativně podporován .NET frameworkem. [26]

1.11.2 XML Schema

Díky pevné struktuře XML lze snadno ověřovat správnost dat. Toto ověření lze provést nezávisle na jakémkoliv programu, který bude v budoucnu XML dokument zpracovávat. Kontrola dat se tak přesouvá na jejich pořizovatele, tj. subjekt, který o pořizovaných

²¹ Z angl. eXtensible Markup Language – rozšiřitelný značkovací jazyk

²² Z angl. Standard Generalized Markup Language – univerzální značkovací metajazyk, který umožňuje definovat značkovací jazyky jako své vlastní podmnožiny

²³ Z angl. Hypertext Markup Language – značkovací jazyk pro tvorbu webových stránek

²⁴ Z angl. World Wide Web Consortium – mezinárodní konsorcium, jehož členové společně s veřejností vyvíjejí webové standardy

datech ví nejvíce informací. Z odběratele se tím snímá povinnost v případě chyby vstupních dat tyto problémy řešit.

XML schéma je obecné označení pro jazyk, který popisuje strukturu dokumentu XML. Strukturou dokumentu XML se myslí popis elementů a jejich atributů, které se mohou v XML dokumentu objevit, spolu s dalšími omezeními. Pokud soubor XML splňuje konkrétní XML schéma, říkáme, že XML soubor je dle schématu validní.

Dnes hojně používané XSD²⁵ je alternativou k jiným XML schématům jako je např. DTD²⁶. Definiuje soustavu specifikací a pravidel, jak má vypadat XML dokument, aby byl validní. XSD je robustnější jazyk než DTD. Je rozšiřitelný vůči změnám v budoucnu, podporuje datové typy a jmenné prostory. XSD se zapisuje pomocí jazyka XML. [27]

1.12 Funkční modelování

Funkční modelování se zabývá ději v informačním systému, zkoumá a analyzuje všechny činnosti, které musí informační systém v jednotlivých úlohách provádět. Po analýze činností následuje textové či grafické zobrazení algoritmizovatelných činností a procesů.

Při popisu činností v informačním systému můžeme provádět hierarchický rozklad funkcí od nejobecnějších až po elementární funkce, které jsou pro uživatele k dispozici. Například od modulu konfigurace obecně až k přepočtu bodových hodnot časů v celé databázi. Funkce vyšší úrovně vznikne pojmenováním skupiny funkcí na bezprostředně podřízené úrovni. Počet úrovní dekompozice modelu není omezen a liší se podle konkrétního řešeného případu. Funkci na nejnižší úrovni takového členění říkáme elementární funkce. [28]

1.12.1 Slovní popis funkčního modelu

Metodu slovního popisu je vhodné použít při řešení úloh menšího rozsahu a také pro komunikaci uvnitř analytického týmu. V dokumentaci informačního systému se její

²⁵ Z angl. XML Schema Definition

²⁶ Z angl. Document Type Definition

použití nedoporučuje z důvodu nižší přehlednosti. Pro popis funkcí aplikace vytvářené v rámci diplomové práce bude postačující. [29]

1.12.2 Diagram případů užití

Graficky můžeme funkcionalitu a chování systému popsat třeba diagramem případů užití (anglicky Use Case Diagram). Diagram vypovídá o tom, co má systém umět, ale neříká, jak to bude dělat. Use Case diagram se skládá z případů užití, aktérů a vztahů mezi nimi.

Případ užití (zkráceně UC) je sada několika akcí, které vedou k dosažení určitého cíle. Definuje jednu funkcionalitu, kterou by měl navrhovaný systém umět. Tu lze rozložit na další akce, které již diagram nezachycuje. Případ užití je nejčastěji zakreslován jako elipsa s jeho názvem uvnitř. Mezi případy užití se můžeme setkat s vazbami typu:

- *Include* – případ užití může obsahovat jiný
- *Extend* – případ užití může rozšiřovat jiný
- *Generalization* – případ užití může být speciálním případem jiného

Aktér je role, která komunikuje s jednotlivými případy užití. Roli může zastávat uživatel nebo externí systém. Aktér může být aktivní nebo pasivní. Aktivní aktér inicializuje nějaký případ užití. Pokud je aktér případem užití iniciován, hovoříme o pasivním aktérovi. V tomto případě ho zakresluje v diagramu napravo. Aktéry znázorňujeme jako postavu z čar s názvem napsaným pod ní. Podobně jako u případů užití, může mezi aktéry také existovat vazba typu generalizace. [30]

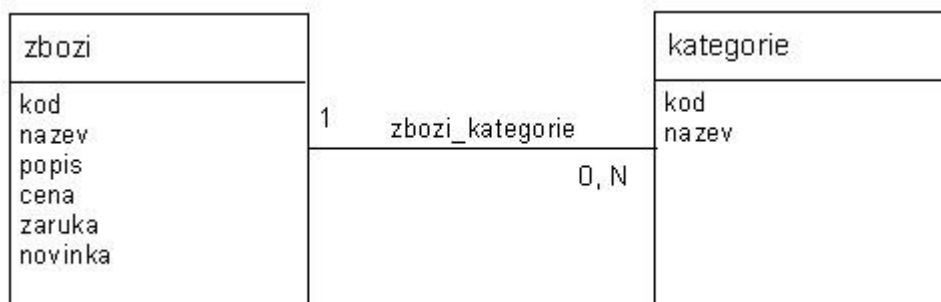
1.13 Datové modelování

Cílem datového modelování je navrhnout kvalitní datovou strukturu pro konkrétní aplikaci a databázový systém, který bude tato aplikace využívat k uložení dat. Je to proces, při němž se definují a analyzují požadavky na strukturu dat. Cílem je zachytit a popsat tu část reality, o které chceme uchovávat informace. Snažíme se tedy převést reálné objekty na objekty datové. Podobně jako funkční modelování tak i to datové usnadňuje komunikaci mezi jednotlivými osobami podílejícími se na návrhu informačního systému.

1.13.1 Konceptuální model

K tvorbě datového modelu na konceptuální úrovni se nejčastěji používá ER²⁷ diagram. Ten je tvořen sadou entit, které jsou mezi sebou provázány pomocí vazeb s příslušnou kardinalitou (násobností). Entita je označována podstatným jménem a reprezentuje určitou skupinu objektů reálného světa, které se označují jako instance entity. Každá entita je popsána svým názvem a sadou atributů. Každý z těchto atributů má přiřazen datový typ. Pro datový model se používá grafické zobrazení. Entitu zobrazujeme pomocí obdélníku, ve kterém jsou vepsány název a pod čarou atributy. [28]

Vztah mezi entitami zachycuje, jakým způsobem jsou dvě nebo více entit vztažené mezi sebou. Vztahy se označují slovesy. Kardinalita vztahu neboli mocnost vztahu vyjadřuje, kolik výskytů jedné entity může vstoupit do vztahu s kolika výskyty druhé entity. Definuje se vždy maximální a minimální kardinalita. Minimální kardinalita může nabývat hodnot 0 nebo 1 a maximální kardinalita 1 nebo N.



Obrázek 1.4 - Ukázka vazby mezi dvěma entitami

Na ukázce vidíme pojmenovanou vazbu, která vyjadřuje vztah mezi entitami zboží a kategorie. Říká, že jeden druh zboží může odpovídat žádné až teoreticky nekonečnu kategorií.

1.13.2 Fyzický model

Fyzický návrh vzniká převodem logického návrhu²⁸ a je použitelný v rámci zvoleného databázového systému. Dalo by se říci, že logické modelování se zaměřuje na otázku co.

²⁷ Z angl. Entity-relationship diagram – entitně vztahový diagram

²⁸ Komplexnější než konceptuální návrh a může být také reprezentován ER diagramem

Kdežto fyzický návrh spíše odpovídá na otázku jak. Pro fyzický návrh je nutná dobrá znalost funkcí a možností cílového databázového systému. [31] [32]

Na počátku fyzického modelování je nutné shromáždit veškeré informace o tabulkách vytvořených během logického návrhu. Každá tabulka obsažená v logickém modelu by měla minimálně obsahovat následující informace:

- jméno tabulky,
- seznam jednoduchých sloupců,
- primární, popřípadě alternativní klíče,
- cizí klíče,
- referenční omezení pro všechny identifikované klíče.

Zároveň by pro každý sloupec tabulky měla být k dispozici následující specifikace:

- definice domény (typ dat, délka, omezení domény),
- implicitní hodnota sloupce, pokud je definována,
- zda se ve sloupci smí vyskytnout hodnota NULL,
- zda se jedná o tzv. odvozený sloupec, a pokud ano, tak jakým způsobem bude získána jeho hodnota.

1.14 UML

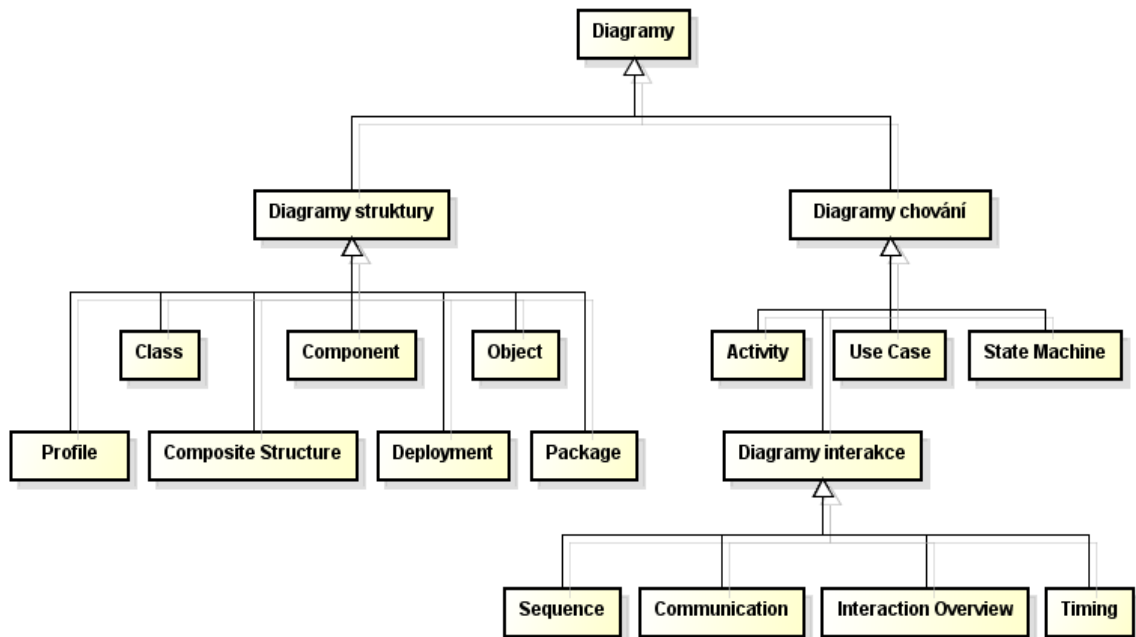
UML²⁹ je v softwarovém inženýrství grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů. V oblasti analýzy a návrhu se stal standardem. UML nám pomůže ve fázi analýzy, kdy komunikujeme s klientem a zjišťujeme, co budeme vlastně programovat. A pomůže nám i ve fázi designu, kdy řešíme otázku, jak to naprogramujeme. UML lze obecně použít třemi způsoby:

- UML jako náčrt – ručně kreslené diagramy používané při jednání s klientem,
- UML jako plán – detailnější diagramy jsou vytvářeny v CAD³⁰ nástrojích a slouží jako plán implementace pro programátory,

²⁹ Z angl. Unified Modeling Language

³⁰ Z angl. Computer-aided drafting – počítačem podporované kreslení

- UML jako programovací jazyk – z detailního UML diagramu lze vygenerovat šablonu kódu, která slouží jako základ pro implementaci.



Obrázek 1.5 - UML diagramy (Zdroj: [33])

UML se skládá ze čtrnácti diagramů, které dělíme na dvě základní skupiny. Diagramy struktury popisují, z čeho je systém složený. Diagramy chování popisují fungování systému. V rámci diagramů chování ještě najdeme skupinu diagramů interakce, které znázorňují interakce mezi jednotlivými částmi systému. [313]

1.15 Enterprise architect

Enterprise Architect (EA) od společnosti Sparx Systems je kompletní nástroj pro systémovou analýzu a návrh, který pokrývá celý životní cyklus vývoje systému, tzn. od zadání požadavků přes analýzu stavů, návrh modelů, testování a údržbu. Vše s využitím diagramů v UML. EA poskytuje podporu pro týmový vývoj a pro jednotlivé role (analytik, tester, projektový manažer, kontrola kvality, vývojový tým). [34]

Uživatelská základna se pohybuje od programátorů a obchodních analytiků až po podnikové architektky. Od malých developerských společností, nadnárodních korporací a vládních organizací až po orgány mezinárodního průmyslu. Společnost Sparx Systems poprvé uvolnila Enterprise Architect v roce 2000. [35]

1.16 Material Design

Material design je návrhový jazyk, který společnost Google vyvinula v roce 2014. Materiálový design je inspirován fyzickým světem a tím, jak jeho textury odrážejí světlo a vytvářejí stíny. Podle společnosti Google je materiálový design určen k lepšímu využití volného místa a přináší konzistentní uživatelský dojem, ať už je zobrazován na smartphonu, tabletu nebo stolním počítači. Autoři za hlavní cíle považují tyto: [36]

- Vytvořit vizuální jazyk, který spojuje klasické zásady dobrého designu s inovacemi na poli technologií a vědy,
- Vytvořit jeden základní systém, který sjednocuje zkušenosti uživatelů napříč platformami a zařízeními,
- Poskytnout inovativní a flexibilní možnosti pro prezentaci.

1.17 SWOT analýza

SWOT analýza je technika strategického plánování používaná k tomu, aby pomohla osobě nebo organizaci identifikovat silné a slabé stránky, příležitosti a hrozby související s konkurenceschopností podniku nebo plánováním projektů. Cílem je upřesnit cíle podnikatelského záměru nebo projektu a identifikovat vnitřní a vnější faktory, které jsou pro dosažení těchto cílů příznivé a nepříznivé. Uživatelé analýzy SWOT se často ptají a odpovídají na otázky, aby získali smysluplné informace pro každou kategorii a identifikovali tak jejich konkurenční výhodu, popřípadě nevýhodu. SWOT analýza je považována za osvědčený nástroj strategické analýzy.

Aby byla SWOT analýza efektivní, zakladatelé a vedoucí společnosti se jí musí aktivně účastnit. Toto není úkol, který lze delegovat na jiné. Na druhou stranu by vedení společnosti ale nemělo dělat práci samo o sobě. K dosažení nejlepších výsledků, je potřeba shromáždit skupinu lidí, kteří zastávají různé pozice a názory ve firmě. Jsou to osoby, které reprezentují různé aspekty podniku, od prodeje a zákaznického servisu až po marketing a vývoj produktů. Každý by měl mít místo u stolu.

Analýza SWOT byla dříve používána pouze k analýze podniků. Nyní je také často využívána vládami, neziskovými organizacemi a jednotlivci, včetně investorů a podnikatelů. Například v případech:

- Prozkoumání možností nové iniciativy nebo řešení problémů.
- Rozhodování se o nejlepší cestě pro iniciativu.
- Identifikace příležitostí k úspěchu
- Určování možných změn.
- Upravování a upřesňování plánů za běhu. Nová příležitost by mohla otevřít širší cesty, zatímco nová hrozba by mohla uzavřít cestu, která kdysi existovala.

SWOT také nabízí jednoduchý způsob komunikace o iniciativě nebo programu a způsob, jak organizovat informace, které byly získány ze studií nebo průzkumů.

SWOT je zkratka složená z počátečních písmen anglických slov Strength, Weakness, Opportunity a Threat. Silné a slabé stránky jsou často cíleny na vnitřní prostředí, zatímco příležitosti a hrozby se běžně zaměřují na vnější prostředí. [37]



Obrázek 1.6 - Grafická podoba SWOT analýzy (Zdroj: [38])

1.17.1 Silné stránky

Silné stránky popisují, v čem předmět zájmu analýzy vyniká a co ho odděluje od konkurence. Například silná značka, loajální zákaznická základna, jedinečná technologie a tak dále. Je nutné učinit rozhodnutí, jak tyto výsledky využít k získání nových investorů. Cílem je maximalizace silných stránek. Vytěžit co nejvíce z toho co už umíme.

1.17.2 Slabé stránky

Slabé stránky brání organizaci, aby vykonávala svou optimální úroveň. Jedná se o oblasti, ve kterých se podnik musí zlepšit, aby zůstal konkurenceschopný. Například slabá značka, obrat vyšší než průměr, vysoká úroveň zadlužení, nedostatečný dodavatelský řetězec nebo nedostatek kapitálu. Obvykle jsou měřeny interním hodnotícím systémem, benchmarkingem nebo srovnáním s konkurencí. Cílem je jejich minimalizace.

1.17.3 Příležitosti

Příležitosti se týkají příznivých vnějších faktorů, které mohou organizaci poskytnout konkurenční výhodu. Například, pokud země sníží sazby, může výrobce automobilů exportovat své automobily na nový trh, čímž se zvýší prodej a podíl na trhu. Podnik se proto snaží o jejich správnou identifikaci a maximální využití.

1.17.4 Hrozby

Hrozby se týkají faktorů, které mohou poškodit organizaci. Běžnými hrozbami jsou například rostoucí náklady na materiály, narůstající konkurence, omezená nabídka pracovních sil a tak dále. Hrozby musí společnost systematicky dlouhodobě řídit a předcházet jim. Při řízení hrozeb lze uplatnit tzv. Paretovo pravidlo 80/20, kdy 20 % potenciálních hrozeb způsobí 80 % finančních ztrát.

1.18 Řetězec data – informace – znalosti

Jelikož se diplomová práce věnuje problematice informačních systémů, je vhodné definovat pojmy data, informace a znalost a určit jejich vzájemný vztah. Informační systémy ukládají data. Pokud je uživatel určitým způsobem interpretuje a přidá jim význam, tak se z dat stávají informace. Aplikací informací se nakonec vytvoří znalosti.

1.18.1 Data

Data jsou objektivní fakta o událostech. Mohou to být čísla, písmena, symboly apod. Datům se snažíme porozumět, interpretovat je a přiřadit jim význam. Představují něco, co se dá v praxi získat experimentem, měřením, pozorováním nebo šetřením. Data objektivně zobrazují vlastnosti, stavy objektů a probíhající procesy v reálném

prostředí kolem nás. Data lze chápat jako jednoduché reprezentační nástroje faktů s jednosměrným a jedinečným významem. Data lze z pohledu práce s daty dělit na strukturovaná a nestrukturovaná. V práci se zabýváme daty strukturovanými.

"Data jsou vlastně surovinou, ze které mohou vyvstávat informace. Například data 0212345678 nebo paegas jistě reprezentují něco reálného z okolního světa, ale bez dalšího popisu nebo kontextu nedávají smysl." [39]

1.18.2 Informace

Pojem informace je používán v mnoha disciplínách a oborech, a proto existuje i mnoho definic pojmu.

Informace jsou jistým způsobem formátovaná, filtrovaná a sumarizovaná data. Jsou to data v kontextu. Proto je číslo 0212345678 užitečné tomu, kdo hledá telefonní číslo na firmu XY a současně zná strukturu, kde 02 je meziměstská předvolba a 12345678 je číslo v rámci Prahy.

Protože informace nejsou často k dispozici, jsou vyhledávány v externích zdrojích. K tomuto účelu slouží právě informační systémy, které vybírají informace z uložených dat. Tuto transformaci od dat k informacím lze chápat jako proces přidávání hodnoty.

Hodnota informace je součástí procesu transformace dat na informace, proto má subjektivní charakter. Hodnota informace nemá přímou souvislost s případnou cenou dat. Data jsou jen nositeli potenciální hodnoty.

1.18.3 Znalosti

Pokud dokážeme informace prakticky využít, jedná se již o znalosti. Znalost tedy poznání či poznatek, vědění či dovednost vyplývá z porozumění zákonitostem. Je v lidské mysli uspořádána tak, aby bylo možné ji používat. Na jejich základě je možné se rozhodovat. Znalosti jsou založené na interpretaci, zkušenostech, poznávání a porozumění. Dále jsou závislé na inteligenčních schopnostech a na schopnostech dávat si věci do souvislosti. Znalost je širší, hlubší, bohatší než data nebo informace. Znalosti se velmi těžko předávají. [39]

1.19 GDPR

Obecné nařízení na ochranu osobních údajů neboli GDPR³¹ je dosud nejvíce uceleným souborem pravidel na ochranu dat na světě. GDPR se dotýká každého, kdo shromažďuje nebo zpracovává osobní údaje Evropanů, včetně společností a institucí mimo území EU, které působí na evropském trhu. Nařízení míří na firmy, instituce i jednotlivce, kteří zacházejí s osobními údaji – zaměstnanců, zákazníků, klientů či dodavatelů, a to napříč segmenty a odvětvími. Cílem GDPR je chránit digitální práva občanů EU.

Ať už jde o bankovní instituce, zdravotnictví, veřejnou správu, nebo e-shopy, všichni se budou v dohledné době potýkat s nutností upravit způsob zpracovávání osobních údajů. V případě závažného porušení pak budou firmám hrozit vysoké pokuty.

GDPR je v celé EU jednotně účinné od 25. května 2018. V Česku tak nahradilo právní úpravu ochrany osobních údajů v podobě směrnice 95/46/ES a související zákon č. 101/2000 Sb., o ochraně osobních údajů. To, že nová pravidla byla přijata formou evropského nařízení, znamená především jejich jednotnou platnost ve všech státech EU, aby je národní vlády a zákonodárci nemohli jakkoli ohýbat a přizpůsobovat místním zájmům nebo lobbistům. [40]

³¹ Z angl. General Data Protection Regulation

2 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE

V kapitole provedeme analýzu problému, který je tématem práce. Nejprve představíme prostředí, ve kterém vznikl a kde probíhá jeho vývoj. Abychom se nebavili pouze v obecné rovině, tak si problém jasně definujeme a následně rozložíme. Podrobněji analyzujeme stav, v jakém se nachází v současné době. Uvedeme si způsoby, jakými se s problémem vypořádává v současnosti nejčastěji. Další druhy analýz zamíří do oblasti konkurence a osobních údajů. Z provedených analýz vyvodíme závěry, které vložíme do kontextu požadavků zadavatele.

2.1 Představení plaveckého klubu

Problém, o kterém je v úvodu kapitoly řeč, vznikl v prostředí blanenského plaveckého klubu. Blansko je město ležící přibližně 19 kilometrů severně od Brna a má asi 21 tisíc obyvatel. Téměř 80 let je město domovem pro plavecký oddíl Asociace sportovních klubů Blansko (dále ASK Blansko), který co do velikosti spadá do kategorie menších oddílů. Navzdory tomu se však v posledních letech začíná prosazovat i na republikové úrovni.

2.1.1 Vznik a vývoj

Počátek oddílu se datuje do roku 1940. V té době však ještě nebyl postaven krytý bazén a plavci tak trénovali na venkovním koupališti. Kryté lázně, jak je známe dnes byly zbudovány až v roce 1976. Jedná se o šesti dráhový dvacetipětimetrový bazén s posunovacím dnem a pětmetrovým skokanským můstkem. Plavci mohli od té doby v klidu trénovat i v zimě.

Blanenský oddíl se stal odrazovým můstkem pro několik plavců a plavkyň, kteří následně dosáhli republikových, evropských i světových úspěchů.

- Yveta Hlaváčová – několikanásobná mistryně České republiky v bazénovém i dálkovém plavání, dvojnásobná medailistka z mistrovství Evropy (dále jen ME) v plavání, držitelka světového rekordu v přeplavání kanálu La Manche a další
- Dana Chalupová – držitelka československého rekordu z roku 1990 na 400 m polohový závod, stříbrná medailistka z ME juniorů (Francie) v roce 1990

- Michal Špaček – vítěz Českého poháru (dále jen ČP) v dálkovém plavání v r. 1995 a 1996, stříbrný medailista z ME v dálkovém plavání (Chorvatsko – ostrov Hvar) na 21 km v r. 1989
- Silvie Rybářová – několikanásobná mistryně České republiky v bazénovém i dálkovém plavání, vítězka ČP v dálkovém plavání v roce 2011 a 2012, medailistka z nejtěžších světových závodů v dálkovém plavání v r. 2014 (2 x Argentina, Makedonie, Kanada)

2.2 Popis současného stavu

V současné době čítá oddíl okolo stovky plavců a deseti trenérů. Ročně se průměrně vymění kolem dvaceti pěti plavců. To znamená, kolik plavců se na začátku roku přihlásí a kolik se jich na konci roku odhlásí. Ukazuje se, že členská základna v posledních letech stále pozvolna narůstá. To lze přisuzovat mimo jiné propagaci v novinách, sociálních sítích nebo prostřednictvím městské televize.

Jedná se o zájmovou organizaci, kde trenéři vykonávají svou práci ve svém volném čase za symbolickou odměnu. Jinými slovy, není to jejich primární zaměstnání. Nicméně všichni trenéři jsou proškolení a disponují dlouholetými zkušenostmi s trénováním nebo se závodním plaváním.

2.3 Organizační struktura oddílu

Vzhledem k velikosti oddílu není struktura příliš rozvětvená. Na špici pyramidy se nachází vedoucí oddílu, která schvaluje a vydává zásadní rozhodnutí týkající se oddílu a jeho činnosti. Spravuje finanční toky. Prosazuje zájmy plaveckého klubu v městské radě. Jmenuje do funkcí trenéry. Zaujímá také roli mluvčího.

Činnost oddílu je rozdělena do několika sekcí. Plavecká příprava, závodní plavání a kurz pro dospělé. Podrobněji jsou jednotlivé sekce rozebrány dále v kapitole 2.42.4. Každá sekce má svoje trenéry. Někteří trenéři působí ve více sekcích současně.

Je určen jeden hlavní trenér, který vykonává největší objem práce v sekci závodní plavání. V této sekci je totiž nejsložitější agenda v porovnání s ostatními sekcemi. Práce

se týká přihlašování plavců na závody, vedení databáze výkonů, správa internetové i jiné prezentace a organizace závodů pořádaných oddílem.

Jednou do měsíce jsou pořádány trenérské rady, kde se mohou trenéři společně s vedoucí oddílu vyjádřit k chodu oddílu.

2.4 Činnost oddílu

Výuku plavání v oddílu mohou děti započít od šesti let. Jsou zařazeny do takzvané plavecké přípravy, kde se dvakrát týdně pod dohledem zkušených trenérů učí základům plaveckých stylů, které postupně zdokonalují. Plaveckou přípravku navštěvuje v průměru kolem 60 dětí. Starší děti z přípravy se již zúčastňují některých závodů spolu s plavci závodního plavání, do kterého postupně podle dovedností přechází.

Závodní plavání je sekce plaveckého oddílu vytvořená pro děti, které prošly přípravkou a chtějí se dále zlepšovat v plavání a soutěžit na závodech. Má asi čtyřicet plavců, kteří trénují čtyřikrát až pětkrát týdně. Ročně se zúčastňují kolem čtyřiceti závodů po celé ČR i na Slovensku.

Jak přípravka, tak závodní plavání rozděluje plavce do tréninkových skupin podle dovedností a výkonosti. Každou skupinu má na starosti jeden či více trenérů. Přestup dětí z jedné skupiny do druhé probíhá na základě změny výkonosti na začátku nové sezóny.

Oddíl taktéž organizuje kurzy plavání pro dospělé. Kurzy jsou určeny pro ty, kteří se chtějí naučit plavat, zdokonalit svůj styl nebo se připravit k absolvování zkoušky na sportovní škole. Kurz probíhá každým rokem v únoru až dubnu a sestává z dvanácti lekcí.

Oddíl každoročně pořádá několik závodů v blanenském bazénu:

- Blanenský závod (březen)
- Cena města Blanska (duben – květen)
- Krajský přebor žactva (květen)
- Krajský přebor žactva (listopad)
- Memoriál Jaroslava Starého (listopad)

Jednou až dvakrát do roka se závodní plavci účastní soustředění na suchu, popřípadě ve vodě.

2.5 Členství v ČSPS

Český svaz plaveckých sportů (dále jen ČSPS) zastupuje Českou republiku od roku 1993 v Mezinárodní plavecké federaci (dále FINA). FINA sdružuje národní svazy vodních sportů. Zastřešuje celosvětové aktivity bazénového plavání, dálkového plavání, skoků do vody, synchronizovaného plavání a vodního póla. Federace má sídlo ve švýcarském Lausanne. ČSPS sdružuje vodní sporty v České republice.

Každý sportovní oddíl, klub nebo tělovýchovná jednota, které se chtějí podílet na činnosti ČSPS (zúčastnit se soutěží v plaveckých sportech, školení a dalších akcí), musí být registrovány v ČSPS. Proto na začátku každého kalendářního roku provádí vedoucí oddílu ve spolupráci s hlavním trenérem registraci nových a aktualizaci stávajících členů plaveckého oddílu. [41] [42]

2.5.1 Krajský svaz ČSPS

Krajské svazy ČSPS jsou založeny jako pobočné spolky ČSPS. Sídla krajských svazů ČSPS jsou krajská města. Členem příslušného krajského svazu se stává sportovní oddíl souběžně s přijetím oddílu do ČSPS. Plavecký oddíl ASK Blansko jedná s Krajským svazem Jihomoravského kraje při příležitosti pořádání letních a zimních krajských přeborů žactva v plavání v Blansku.

2.6 Definice procesu

Nyní si specifikujeme proces, který je předmětem zájmu této práce a potažmo aplikace vyvíjené v rámci práce. Zjednodušeně řečeno, procesem zde rozumíme evidenci výsledků a jejich následné využití pro analýzy, statistiky a přihlášky.

2.6.1 Uložení výsledků

Po skončení plaveckých závodů jsou pořádajícím klubem vydány výsledky, které jsou v různých formátech publikovány na serveru. Odtud si je zúčastněné kluby mohou stáhnout pro vlastní potřebu. S využitím prostředků pro správu databáze si kluby uloží obdržené výsledky.

2.6.2 Získání požadované informace

Důvodem ukládání výsledků do databáze je jejich pozdější využití. Proces získání požadované informace zahrnuje procházení databáze, výběr a třídění požadovaných dat podle zvolených kritérií. Formátováním, filtrováním a sumarizováním získají tato data určitou interpretaci a význam. Tím se z dat stanou informace, které lze jejich aplikací převést na znalost. Ta se může v řízení klubu projevit například tak, že hlavní trenér přihlásí plavci na závody takové disciplíny, v kterých dosahuje nejvyšších bodových hodnot a má v nich tak nejvyšší naději na úspěch.

2.7 Prostředky k výkonu procesu

Výzkumem bylo zjištěno, že k výkonu výše definovaného procesu kluby nejčastěji používají následující trojici prostředků. V mnoha případech šlo o jejich kombinované využití.

2.7.1 Papírová kartotéka

V dnešní době už výrazně zastaralý způsob vedení databáze. Jedná se o uspořádaný soubor papírových lístků s údaji řazenými podle potřeb. Je velice nepraktické uchovávat oddílovou databázi tímto způsobem, když jsou výsledky dostupné elektronické podobě. Nemluvě o nutnosti zajištění fyzických prostorů přizpůsobených k uchovávání a třídění množství lístků. Vyhledání a natož analýzy z takto uskladněných dat jsou nadmíru obtížné až nemožné.

Proto s příchodem a s rozšířením informačních technologií do prostředí klubů se začalo od evidence výsledků formou papírové kartotéky upouštět. Na scénu začali nastupovat uživatelsky relativně dostupné programy od firmy Microsoft jako jsou Excel a Access. Jak se dále dozvíme, i tyto způsoby do jisté míry ustupují a kluby dávají přednost centralizované online evidenci.

Po přechodu z papírové evidence na modernější způsoby vyvstává otázka. Co s existující papírovou kartotékou? Té je plná skříň v kanceláři oddílu a sedá na ni prach. Přitom je v této skříní schovaná mnohdy několik desítek let stará minulost klubu se všemi hodnotnými výkony úspěšných a často už vysloužilých plavců. Pokud bychom chtěli uvažovat o převedení dat do elektronické podoby, musíme počítat s tím, že to bude

mechanická, časově náročná a na chyby náchylná práce založená na opisování časů do počítače. Z těchto důvodů tuto činnost řada starších klubů disponujících objemnou kartotékou výkonů odkládá nebo vůbec neplánuje.

Plavecký oddíl ASK Blansko má svoji papírovou evidenci výkonů vedenou do roku 2006, kdy převzaly svou roli počítače. Převodem z papíru do počítače byl pověřen dobrovolník. Přepisuje časy do programu MS Excel. Přepis probíhá ve volném čase dobrovolníka, proto tomuto úkolu nebyl přidělen koncový termín, aby se minimalizovala chybovost přepisu. Předpokládá se, že by přepsané výsledky mohli být časem nahrány do novodobé oddílové elektronické databáze vedené od zlomového roku 2006.

2.7.2 MS Excel a MS Access

Multifunkční tabulkový procesor Excel a nástroj pro správu relačních databází Access převzali po vyřazení papírové kartotéky ve většině klubů vládu nad správou oddílových databází.

Obecně platí, že Access je nejlepší volbou, pokud chceme pravidelně sledovat a zaznamenávat data a potom zobrazovat, exportovat nebo tisknout výběry z těchto dat. Formuláře v Accessu poskytují pohodlnější rozhraní než pracovní list aplikace Excel pro práci s daty. Aplikaci Access můžeme použít k automatizaci často prováděných akcí a reporty umožňují shrnutí dat v tištěné nebo elektronické podobě. [43]

Aplikace Excel může ukládat velké množství dat do sešitů, které obsahují jeden nebo více listů. Místo toho, aby sloužil jako systém správy databáze, jako Access, je aplikace Excel optimalizována pro analýzu a výpočet dat. Tento flexibilní program lze použít k sestavování modelů pro analýzu dat, psaní jednoduchých a složitých vzorců k výpočtu těchto dat a k prezentaci dat v řadě profesionálně vyhlížejících grafů. [43]

Každý oddíl si podle požadavků na zpracování dat ve většině případech zvolil jeden z uvedených programů. Důležitým faktorem volby byla zkušenost a úroveň znalostí daného programu osobou pověřenou správou oddílové databáze.

Blanenský oddíl se vrhnul cestou vytvoření jednoduché relační databáze v MS Access. Propojením tří tabulek dimenzí s jednou faktovou tabulkou vznikl základ pro budoucí oddílovou databázi, která má v současné době přes třicet tisíc záznamů.

2.7.3 Centrální evidence

Stejnou možnost evidovat výsledky z textových souborů jako mají plavecké kluby po republice má i ČSPS. A taky ji využívá. V roce 2004 zprovoznil ČSPS na svých internetových stránkách aplikaci pro centrální evidenci závodních plavců a závodů konaných na území České republiky. Nicméně, smysluplná databáze je pro veřejnost přístupná až od roku 2009 prostřednictvím internetové aplikace. Před tímto rokem lze v databázi vyhledat pouze ojedinělé časy převážně plavců z reprezentačního výběru. Pravděpodobně má ČSPS starší výsledky v jiné formě.

V dnešní době je online evidence výsledků od ČSPS již na takové úrovni, že dokáže zastoupit individuální evidenci výsledků na klubové úrovni. Navíc v roce 2018 proběhla kompletní aktualizace internetové aplikace pro vedení výsledků. Takže ty kluby, které nemají prostředky nebo zájem vést vlastní oddílovou databázi, mohou využívat tu centrální.

V nabídce jsou ale jenom omezené možnosti výběrů a třídění dat. Jelikož aplikace slouží všem klubům v České republice, nenajdeme zde pokročilejší sestavy zaměřené výhradně na jeden oddíl. Například, nedokážeme vytvořit tabulku oddílových rekordů nebo vybírat a třídit plavce na základě bodové hodnoty výkonu. Nepříjemností může být také občasná chybovost zmíněné aktualizované verze internetové aplikace. V některých internetových prohlížečích nefunguje posuvník v seznamu, který je delší než okno náhledu. Také byla zjištěna nefunkčnost modulu pro export vybrané větší množiny záznamů (kolem jednoho tisíce). Pravděpodobně překročení časového limitu úkolu.

2.8 Analýza konkurenčních aplikací

V české republice se v současné době nenachází odpovídající počítačový program vyhovující požadavkům oddílu. Naopak v zahraničí existuje relativně velké množství aplikací zabývajících se řízením plaveckých klubů a správou jejich databází. Tyto aplikace poskytují rozsáhlou nabídku možností, jsou otestovány a aktivně využívány množstvím zahraničních plaveckých klubů. Problém těchto aplikací je jejich lokalizace. Lokalizace nikoli ve smyslu jazykové (i když i to je problém), ale hlavně ve smyslu prostředí, pro které byly vyvinuty. Jiné potřeby, pravidla a konvence má závodní plavání v USA (odkud pochází většina konkurenčních aplikací), v Evropě a v České republice.

Z analyzovaných konkurenčních aplikací vyplynulo, že každá obsahuje část funkcionality, která by splňovala požadavky oddílu. Nicméně se nepodařilo najít aplikaci, která by je splňovala všechny. Už jen fakt, že ČSPS má svůj textový formát k výměně plaveckých dat, je důvod k úvahám nad vytvořením aplikace podporující tento formát a sledující konvence plavání v České republice. Cena těchto aplikací se pohybuje v řádech stovek dolarů, přičemž průměr je okolo 650 dolarů což je asi 15 000 českých korun. [44] [45] [46]

Zajímavou konkurenční aplikací je aplikace *iSwimManager*, která vznikla na Slovensku. Funguje na operačním systému iOS, tedy na zařízeních od společnosti Apple. Z nalezených konkurenčních aplikací je nejbližší požadavkům oddílu ASK Blansko. Pozitivem je podpora formátu plaveckých dat Lenex, který se poslední dobou po Evropě stále více využívá. Nicméně podpora v Česku nejpoužívanějšího textového rozhraní formátu CSV chybí. Navíc je celá aplikace velmi komplexní a poskytuje mnoho pokročilých funkcí, které by menší blanenský oddíl nevyužil. Také nutnost provozu aplikace na platformě iOS by nevyhnutelně musela vyústit k zakoupení nového zařízení od společnosti Apple. [47]

Skutečnosti jako nedostupnost plně vyhovujícího řešení a výše finančních prostředků nutných k pořízení licence vedli vedení plaveckého oddílu ASK Blansko k rozhodnutí zahájit spolupráci se mnou s cílem získat řešení formou studentského projektu. Lákadla pro oddíl byla v první řadě nízká cena řešení, možnost přizpůsobení aplikace potřebám oddílu nebo moje dostupnost a flexibilita při schůzích s vedením oddílu ve fázích návrhu, vývoje nebo zkušebního i ostrého provozu aplikace z důvodu mého členství a aktivní činnosti v klubu.

2.9 Analýza současného stavu procesu

Proces evidence výsledků v plaveckém oddílu ASK Blansko se po upuštění od papírové kartotéky přenesl na bedra MS Access. Činnost má na starosti výhradně jedna osoba, a to hlavní trenér v sekci závodní plavání. Pokud by se na procesu podílelo více osob mohlo by docházet k chybám vedoucím k nejednoznačnosti dat v databázi.

V případě větších klubů však všechnu práci spojenou s procesem evidence výsledků jedna osoba nezvládne. V takovém případě se určí více osob a jasně se jim

specifikuje náplň práce. Například jeden člověk se bude starat pouze o plnění databáze výsledky, druhý z nich bude generovat statistiky a třetí bude vytvářet přihlášky na závody. Aplikace vyvíjená v rámci diplomové práce si dává za cíl minimalizovat počet osob nutných účasti na popsaném procesu a v případě menších oddílů zjednodušit a sjednotit činnosti tak, aby byly zvládnutelné jednou osobou.

2.9.1 Přihlášky na závody

Každý oddíl na začátku sezóny vydá kalendář závodů, kterých se plánuje zúčastnit. Závody se dělí na dva druhy z hlediska způsobu přihlašování na vedené a nevedené v termínové listině ČSPS. Většina závodů už je v dnešní době vedena v termínové listině ČSPS a neustále se poměr vedených/nevedených zvětšuje. Přihlašování na tento druh závodů je jak pro oddíly, tak pro organizátory pohodlnějším a dnes preferovaným způsobem. Neevidovány jsou pouze některé okresní závody často pro menší děti.

V dobách, kdy nebyly informační technologie dostupné pro širokou veřejnost se přihlášky posílaly klasickou poštou. Výsledky se měřily, zapisovaly a vyhodnocovaly pouze ručně.

Později oddíly musely elektronickou poštou posílat organizátorům textovou soupisku přihlašovaných plavců. Přihlašované časy museli trenéři často odhadovat nebo dohledávat ve svých oddílových databázích. Výsledkem bylo množství odlišně naformátovaných textových dokumentů, které často obsahovaly překlepy. Organizátor si musel sjednotit formátování souborů podle předem daného rozhraní a importovat je do počítačového programu na pořádání závodů. Vzhledem k tomu, že formát rozhraní je veřejně známý, mohou oddíly přihlášky posílat už formátované. Jenomže se jedná o CSV soubor (čárkami oddělené hodnoty). Ruční tvorba takového souboru je nepřehledná a náročná. Proto vznikl nápad generovat přihlášky automaticky z oddílové databáze rovnou v tomto rozhraní, což bude jednou z funkcí připravovaného programu.

Situace je jiná a jednodušší případě evidovaných závodů v termínové listině ČSPS. Pořadatel v centrálním informačním systému ČSPS nadefinuje všechny potřebné náležitosti závodu – program, kategorie, kluby s povolenou účastí a mnoho dalšího. Potom spustí přihlášky, které se uzavírají většinou týden před zahájením závodů. Oddíly v průběhu této doby s využitím uživatelské rozhraní informačního systému přihlašují

požadované plavce k disciplínám v programu závodů. Přihlašované časy jsou brány rovnou z centrální databáze ČSPS. Místa nenalezené časy musí uživatel doplnit ručně. Po uzavření přihlášek si pořadatel stáhne přihlášku všech klubů v jednom souboru v požadovaném rozhraní.

Někdo by mohl namítat, proč se zabývat způsobem přihlášek, který se používá čím dál méně. Důvody jsou následující. Stále existují závody, které tento způsob používají, a my nevíme, jak dlouho tomu tak ještě bude. Plavecký oddíl ASK Blansko, kterému bude aplikace v první řadě sloužit, si tuto funkcionalitu výslovně vyžádal.

Na příkladu níže je uvedena vzorová přihláška splňující formát pro bezproblémový import do programu pro řízení závodů.

```
171214
Mikulášský víceboj Kuřim
Kuřim
14.12.2017
ASKBI
```

```
"ANTOŇŮ";"Nela";"2007";"Z";"8";"50Z";"00:53,13";"Brno Cup v Brně Lesné";"16.12.2017"
"ANTOŇŮ";"Nela";"2007";"Z";"12";"50P";"00:56,74";"Brno Cup v Brně Lesné";"16.12.2017"
"BURGETOVÁ";"Sára";"2007";"Z";"12";"50P";"00:51,37";"Brno Cup v Brně Lesné";"16.12.2017"
"BURGETOVÁ";"Sára";"2007";"Z";"16";"50K";"00:38,00";"Velká cena Kuřimi";"20.05.2017"
"BURGETOVÁ";"Sára";"2007";"Z";"4";"50M";"00:42,06";"Brno Cup v Brně Lesné";"16.12.2017"
"HANZLÍČKOVÁ";"Stella";"2009";"Z";"4";"50M";"01:01,47";"Moravskoslezská liga nejml. žactva v Brně";"04.11.2017"
"HANZLÍČKOVÁ";"Stella";"2009";"Z";"16";"50K";"00:46,63";"Městské přebory žactva v Brně";"21.10.2017"
```

Obrázek 2.1 - Vzorová podoba přihlášky

Jednotlivé řádky přihlášky zastupují disciplíny, které si plavci přihlašují. Kolik disciplín si plavec přihlašuje, tolik řádků mu náleží. Pole na řádku jsou oddělena středníky a obsahují – příjmení, jméno, ročník, pohlaví, číslo disciplíny dle programu závodů, zkratka disciplíny, přihlašovaný čas, místo zaplávání přihlašovaného času, datum zaplávání přihlašovaného času

2.9.2 Výsledky ze závodů

Po ukončení závodů pořadatel vydá výsledky ve formátech PDF a CSV. U nevidovaných závodů se toto posílá elektronickou poštou zúčastněným oddílům. U většiny se však tyto soubory nahrají do informačního systému ČSPS. Odtud jsou k dispozici pro ČSPS a oddíly.

V zahraničí je rozšířen formát zvaný Lenex. Lenex je formát výměny plaveckých dat založený na XML. Verze 1.0 se používá od roku 1999 v různých zemích Evropy. S využitím získaných zkušeností s formátem Lenex byla koncem roku 2008 vydána nová

závodů, zkratka disciplíny, umístění, příjmení, jméno, ročník, pohlaví, plavecký klub, čas, body, číslo rozplavby, číslo dráhy, datum.

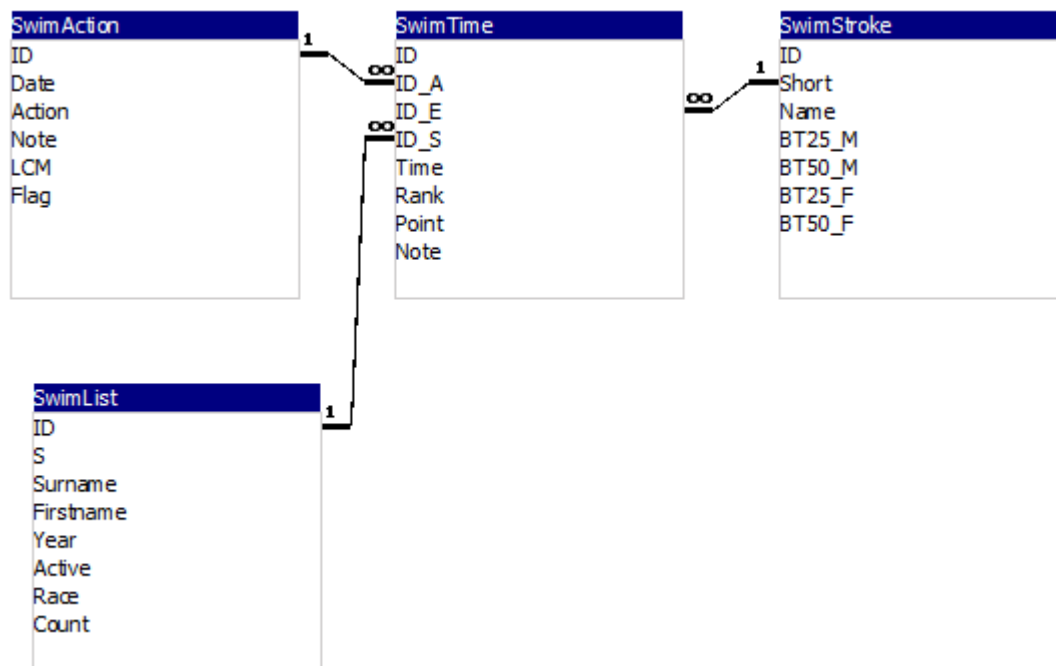
Zbylá pole se využívají u štafet, kde jsou uvedeny jména plavců a jejich ročník. Z důvodu zkrácení řádku byla tato pole zkrácena na pouhá označení *PLAVEC1-4* z původního formátu *PŘÍJMENÍ Jméno ročník*. Další rozdíl štafetových záznamů od individuálních je v polích číslo 4, 5, 6, kde je uveden pouze název štafety. První úsek štafety se bere jako klasický individuální start a je uvozen číslem 999. Ukázka záznamu jedné štafety a odpovídajícího prvního úseku:

```
11;"4x1000";1;"PKS1áviaVŠPlzeň";"";"Z";"S1P1";04:23,53;683;2;4;06.07.2017;"00:32,83";"01:07,31";"";"PLAVEC1";"PLAVEC2";"PLAVEC3";"PLAVEC4"
999;100 Z; ;"PŘÍJMENÍ1";"Jméno1";"2001";"Z";"S1P1";"01:07,31"; ; ; ;06.07.2017;"00:32,83";"";"";"";"";"";"";"";
```

Obrázek 2.3 - Vzorová podoba výsledků štafetového startu

2.9.4 Evidence výsledků

Jakmile jsou dostupné textové výsledky ze závodů, kterých se blanenský oddíl účastnil, je na hlavním trenérovy, aby je nahrál do databáze. CSV soubor si po odstranění hlavičky otevře v programu Excel. Odstraní záznamy týkající se jiných klubů a ponechá pouze časy plavců ASK Blansko. Následně musí data ručně upravit, aby vyhovovala struktuře tabulek v databázi vedenou v programu MS Access. Konečně může provést import upraveného excelovského souboru do databáze.



Obrázek 2.4 - ER diagram databáze v programu Access

SwimTime na *Obrázku 2.4* je faktová tabulka časů. Tabulky dimenzí SwimAction, SwimStroke a SwimList jsou v pořadí tabulky závodů, stylů a plavců. Tabulky dimenzí jsou s faktovou spojeny vazbou typu 1: N. To znamená, že jeden záznam v dimenzi odpovídá 0 až N záznamům v tabulce faktů.

2.9.5 Výběr dat

Výběr dat provádí hlavní trenér pomocí sestav v programu MS Access. Jedná se o v celku propracovaný aparát sloužící k zobrazování, formátování a vytváření souhrnů informací.

Při vytváření nové sestavy hlavní trenér nejprve zvolí zdroj záznamů sestavy. Tím může být tabulka nebo dotaz. Dále pomocí nástrojů sestav vytvoří návrh, konkrétní nastavení a popisky sestavy. Po vytvoření sestavy lze přidat seskupení, řazení nebo souhrny. Přehlednost a čitelnost se dá vylepšit nástroji pro zvýrazňování dat v sestavách, přidáním pravidel podmíněného formátování a vlastním přizpůsobením barev a písma. Hlavní trenér také využívá možnosti přidat logo, která mu umožňuje přidat oddílové logo do záhlaví dokumentu. Před samotným tiskem je dobré zobrazit náhled výsledné sestavy.

Pochvalovanou funkcí aparátu sestav programu MS Access je možnost uložit pracně vytvořený návrh sestavy. Jakmile hlavní trenér návrh sestavy uloží, může sestavu

spouštět tak často, jak potřebuje. Návrh sestavy se nezmění, ale při každém tisku nebo zobrazení sestavy získá aktuální data.

2.9.6 Prezentace informací

Prezentace informací získaných analýzou dat z oddílové databáze probíhá několika způsoby. Tradičním portálem pro publikaci vytvořených sestav jsou internetové stránky oddílu. Zde se počátkem každého kalendářního roku publikují aktualizované sestavy a statistiky. Tištěná podoba sestav visí na nástěnkách ve vestibulu blanenských lázní.

V rámci méj bakalářské práce vznikla internetová aplikace pro veřejnost a příznivce blanenského plavání. Aplikace poskytuje zájemcům náhled do databáze zaplavaných výkonů, aktivních i neaktivní plavců a závodů, kterých se oddíl zúčastnil. Tato data dokáže aplikace přehledně zobrazit a vytvořit z nich zajímavé statistiky. Nevýhodou je, že se údaje aktualizují duplicitně na serverové i lokální databázi. Jinými slovy, trenér importuje výsledky ze závodů nově ještě do internetové aplikace.

2.9.7 SWOT analýza před začátkem projektu

Nyní provedeme SWOT analýzu z pohledu stávajícího řešení ve vztahu k novému řešení.

Kdybychom vztahovali SWOT analýzu současného řešení k původní papírové kartotéce, tak by v oblasti silných stránek bylo podstatně více bodů. Jenomže my současné řešení porovnáváme s připravovaným novým řešením. Proto je zde uveden pouze fakt, že hlavní trenér má stávající řešení téměř zautomatizováno.

Naopak příležitostí pro zlepšení je identifikováno více, což je určitě pozitivum. Stávající proces se z podstatné části skládá z manuálních činností, a to dává hlavnímu trenérovi prostor udělat chybu. Automatizace těchto činností by také trenérovi ušetřila čas. Původní návrh databáze byl hodně zjednodušen a tabulky neobsahovali některé podstatné údaje. S rozmachem blanenského plaveckého oddílu databáze postupem času začala postrádat prostor pro uložení časů štafet a jejich prvních úseků jako samostatných startů. Mezičasy nejsou rovněž evidovány.

Při stávajícím řešení hrozí publikování nepravdivých a zavádějících informací v případě výskytu chyby v průběhu procesu. To může vést k nedorozumění nejen mezi trenéry a plavci v průběhu tréninkového procesu nebo závodů. Nepřehledné přihlášky

s chybami mohou vytvářet negativní pověst oddílu ASK Blansko u oddílů zpracovávajících přihlášku. Vše dohromady vede k poklesu úrovně sekce závodní plavání.

Podle definice SWOT analýzy by měli části s příležitostmi a hrozbami obsahovat položky týkající se vnějšího prostředí oddílu. Z důvodu hlubší analýzy se však v těchto částech SWOT analýz prováděných v rámci diplomové práce mohou vyskytnout i položky z prostředí vnitřního.

Tabulka 2.1 - SWOT analýza před projektem

Silné stránky	Slabé stránky
- řešení, které hlavní trenér zná	- neefektivní a zdlouhavý proces - větší pravděpodobnost chyby - chybí štafety, mezičasy a 1. úseky štafet
Příležitosti	Hrozby
- automatizace procesu, úspora času - více zájemců o členství v oddíle - větší povědomí u ostatních klubů - větší kvalita dat	- technické problémy, ztráta databáze - špatná rozhodnutí kvůli špatným informacím - zhoršení vztahů s ostatními oddíly - nezájem nových plavců, trenérů o členství

2.9.8 Zhodnocení procesu

Analýzou současného stavu procesu jsme identifikovali nedostatky a slabá místa v současném řešení evidence výsledků v plaveckém oddílu ASK Blansko. Jako velice neefektivní byla shledána činnost vytváření přihlášek na závody nevidované ČSPS. Způsob nahrávání výsledků do databáze přes program MS Excel je neefektivní a bohatý na chyby. Dále navzdory existenci mocného nástroje MS Access pro správu a výběr dat lze některé charakteristické činnosti nad plaveckou databází výrazně zautomatizovat a tím do určité míry odstranit časové a znalostní nároky kladené na hlavního trenéra.

2.10 Analýza zpracovávaných osobních údajů

Záměrem této analýzy je věnovat pozornost nařízení o ochraně osobních údajů GDPR. Kromě výsledků bude aplikace uchovávat i databázi plavců daného oddílu s jejich osobními údaji. Tato funkcionality je nezbytná z důvodu navázání konkrétního výkonu na plavce a následné vytváření smysluplných statistických analýz.

Co už není pro funkčnost aplikace zásadní je uchovávání detailnějších informací o plavci jako datum narození, adresa bydliště, telefon a jiné. Navzdory tomu je ale oddíl ASK Blansko potřebuje uchovávat, protože pracuje s nezletilými dětmi. Kontakt na rodiče, popř. jejich adresa jsou proto z bezpečnostních důvodů vyžadovány uvést na přihlášce dítěte do oddílu. S přihláškou do oddílu zájemci také odevzdávají souhlas pro zpracování osobních údajů jak pro výše uvedené potřeby, tak pro potřeby propagace oddílu.

2.11 Závěr provedených analýz

Z výsledků provedených analýz vyplývá, že absence komplexní aplikace pro tvorbu, správu a analýzu oddílové databáze způsobuje neefektivitu procesu evidence výsledků spojeným s jejich využíváním v dalších aktivitách oddílu. Vzniklá neefektivita způsobuje, že trenéři, a především hlavní trenér vynakládají čas a úsilí na činnosti, které je možno výrazně zjednodušit a urychlit. Navíc, v případě plaveckého oddílu ASK Blansko, kde trenérské seskupení tvoří dobrovolníci, je efektivita prováděných činností nanejvýš žádoucí. S rostoucím počtem členů oddílu a také zvyšujícím se počtem oddílem navštívených závodů je příhodné zavedení takové aplikace, která přispěje ke zvýšení efektivity zmíněného procesu.

2.12 Požadavky oddílu na funkcionalitu aplikace

Na závěr druhé kapitoly si shrňme požadavky plaveckého oddílu ASK Blansko na funkcionalitu aplikace, které byly stanoveny na trenérské radě v přítomnosti vedoucí oddílu. Zúčastněným byly předneseny výsledky provedených analýz. Spojením jejich představ s mými doporučeními vzniknul následující závazný seznam požadovaných funkcí:

- Vytvořit a spravovat databázi plavců, stylů, závodů, časů
- Vybírat a třídit data podle různých kritérií
- Export informací do formátů TXT a PDF
- Poskytovat sestavy a žebříčky generované nad aktuální podobou databáze
- Import výsledků
- Vytvářet přihlášky na závody, které nejsou evidovány přes ČSPS

- Import FINA základních časů
- Import dat ze stávající podoby databáze
- Záloha, obnova databáze

3 VLASTNÍ NÁVRH ŘEŠENÍ

Následuje pasáž diplomové práce, ve které se zaměříme na volbu, návrh a realizaci vhodného řešení problému identifikovaného v kapitole 2. Ze všeho nejdříve si uvedeme a zdůvodníme volbu konkrétního řešení. Navážeme vytvořením diagramů, které využijeme v procesu návrhu a tvorby databáze pro aplikaci. Dále v kapitole Funkční návrh představíme veškerou funkcionalitu aplikačních modulů. Pokračovat budeme grafickým návrhem aplikace, se kterým souvisí i logo aplikace. Bude také popsán postup při zavádění aplikace do reálného provozu, které zahrnuje jak testování a zkušební provoz, tak i samotné nasazení a údržbu aplikace. Závěrečnou kapitolou této sekce diplomové práce bude kapitola vyhrazená zhodnocení celého projektu. Zde si projekt zhodnotíme po ekonomické stránce, uvedeme jeho přínosy a podíváme se na možnosti budoucího rozvoje.

3.1 Volba řešení

Po seznámení vedení plaveckého oddílu s náklady na pořízení požadovaného softwaru od cizí společnosti bylo řešení typu studentského projektu pro oddíl jasnou volbou. Tomuto rozhodnutí rovněž přispěl fakt, že já jakožto autor aplikace jsem dlouholetým členem oddílu a dobře tak znám pozadí většiny probíhajících oddílových procesů a dokáži se vcítit do požadavků oddílu stanovených v předchozí kapitole.

Řešení bude realizováno formou desktopové³² aplikace. Uživatel si bude muset nejprve nainstalovat databázi MS SQL Server na svůj počítač. Ta bude distribuována společně s aplikačním softwarem. Pro spuštění aplikace bude dále potřeba provést instalaci vedenou instalačním průvodcem. Ta se postará o nakopírování nezbytných složek a souborů do uživatelova aplikačního adresáře.

3.1.1 Způsob užití aplikace

Jeden oddíl, jedna kopie aplikace. Taková je představa použití aplikace ke správě oddílové databáze plaveckých dat. Není žádoucí mít více instancí aplikace na různých počítačích a vytvářet tak více databází k jednomu oddílu. V takových případech by mohlo

³² Programy, které běží díky operačnímu systému na našem stolním počítači nebo notebooku.

docházet k nekonzistenci dat v jednotlivých databázích. Z praktického hlediska takový postup nedává smysl. Pokud je záměrem databázi zálohovat, použije uživatel funkci v rámci aplikace, která se stará o zálohu i obnovu databáze.

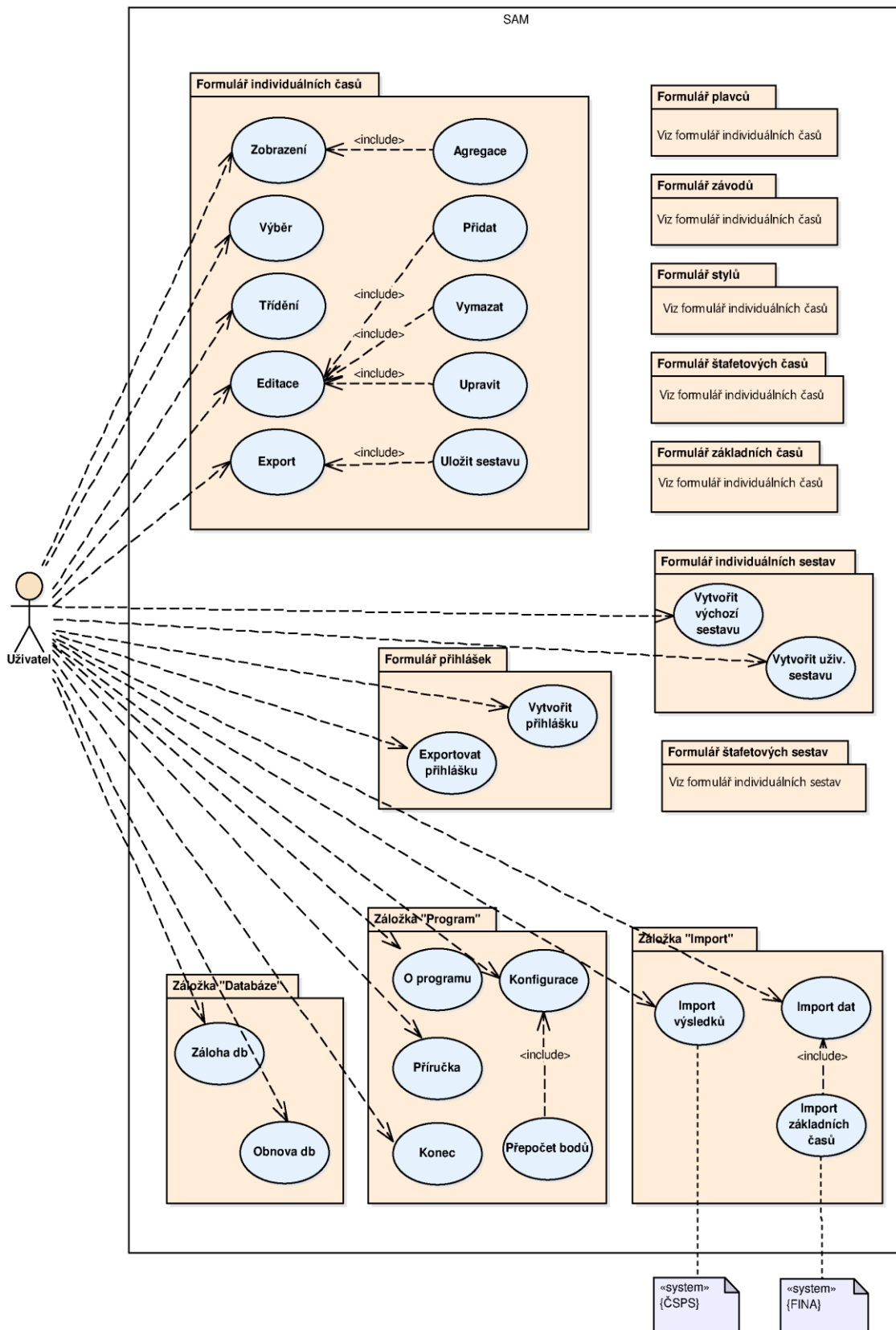
Oddíl si nainstaluje aplikaci na jeden zvolený počítač a zvolí zodpovědnou osobu nebo osoby. V případě oddílu ASK Blansko se určí jedna zodpovědná osoba, která bude mít na starosti obsluhu programu. V případě větších oddílů může aplikaci obsluhovat více osob. Toto však není primární způsob užití aplikace. Že tento přístup není doporučován podtrhuje fakt, že aplikace neumožňuje vytvářet uživatelské účty. Pokud tedy zodpovědné osoby nebudou mít jasně určené role nebo je nebudou dodržovat může dojít ke vzniku chyb v databázi a následných komplikací v rámci provozu oddílu. Cílem je vytvořit aplikaci tak, aby její provoz zvládal jeden člověk.

3.2 Návrh databáze

Než začneme navrhovat a programovat funkce aplikace je potřeba vytvořit kompletní návrh databáze. Návrh bude prezentován formou tří diagramů. První bude diagram případů užití a zbylé dva budou ER diagramy pro různé úrovně návrhu. Všechny diagramy byly vytvořeny v prostředí programu Enterprise Architect.

3.2.1 Diagram případů užití

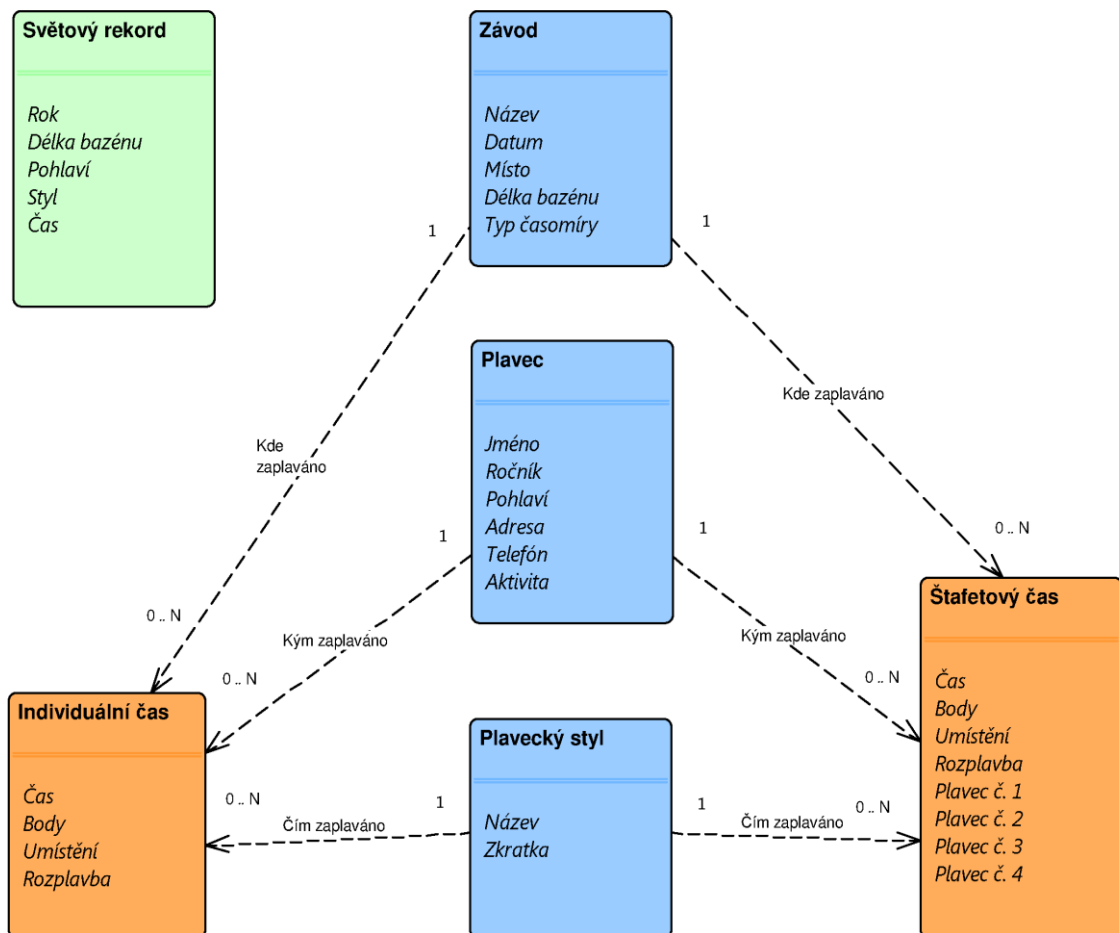
Tento typ diagramu nám pomůže znázornit dekompozici aplikace na jednotlivé funkce a zjednoduší komunikaci se zákazníkem. Systém má pouze jeden typ uživatele, který obsluhuje veškerou funkcionalitu. Některé formuláře jsou minimalizovány z prostorových důvodů. To lze provést, protože minimalizované formuláře mají velice podobnou funkcionalitu jako odkazované formuláře. Jediným rozdílem ve formulářích plavců, závodů a stylů je nedostupnost agregace a možnosti uložení sestav. Formulář základních časů obsahuje pouze možnost zobrazení dat. Položky *<system>* představují vstup dat původem z externího systému. Systém ČSPS poskytuje výsledky ze závodů a systém FINA základní časy.



Obrázek 3.1 – Diagram případů užití

3.2.2 Konceptuální návrh

ER diagram na *Obrázku 3.2* na konceptuální úrovni představuje první pohled na databázovou strukturu. Vyčteme z něj, že databáze bude čítat šest tabulek z toho dvě faktové (oranžová), tři tabulky dimenzí (modrá) a jednu oddělenou (zelená). Bez vazby je tabulka obsahující základní časy (světové rekordy) z externího systému.

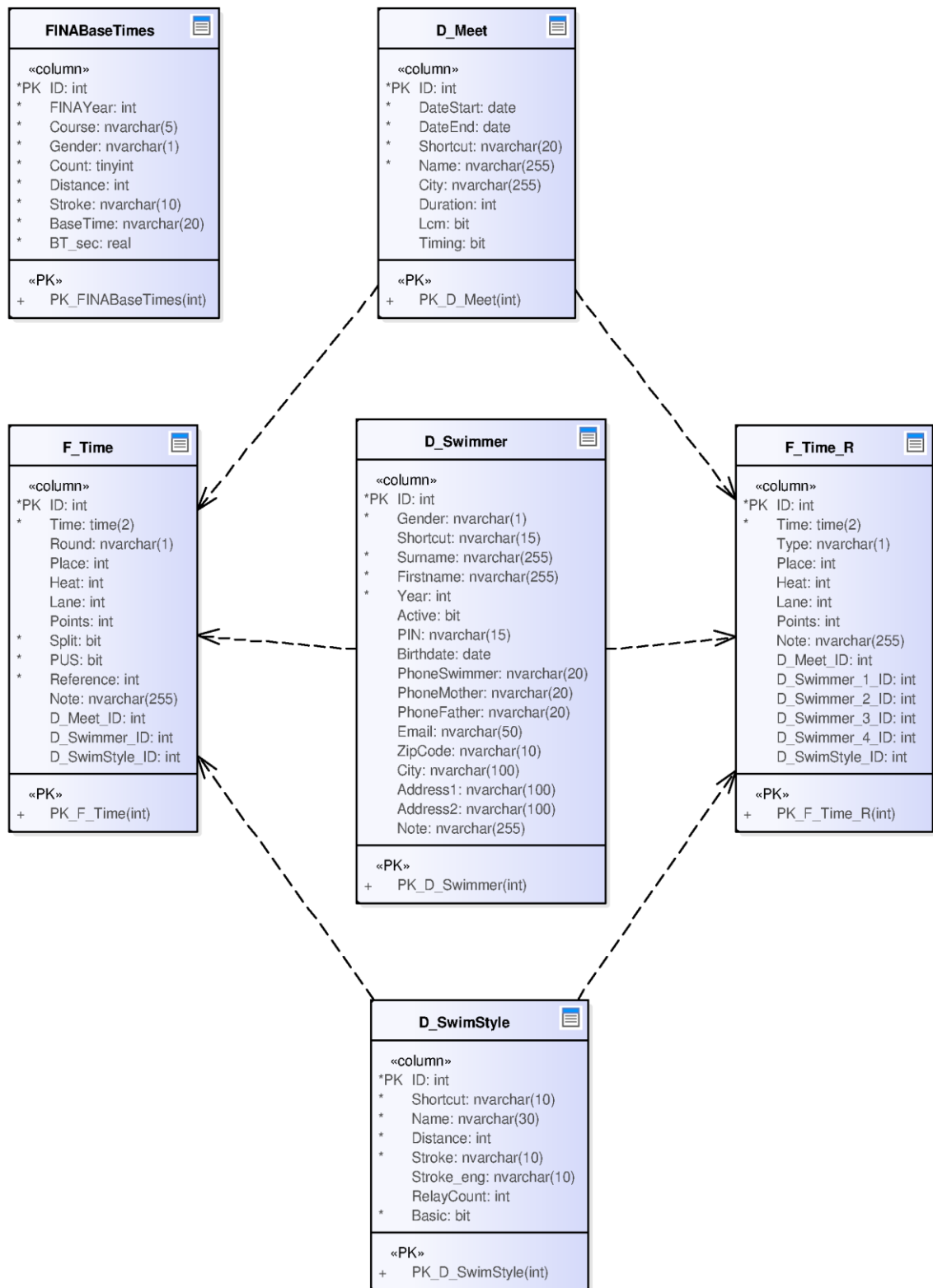


Obrázek 3.2 – ER diagram konceptuálního návrhu

3.2.3 Fyzický návrh

V návrhu databáze postupujeme dále převodem konceptuálního návrhu na fyzický. Všechny názvy tabulek i atributů odpovídají těm v databázi. Atributy se skládají z názvu a datového typu. U textových typů je navíc v závorce uveden maximální počet znaků. Ve srovnání s konceptuálním návrhem byly přidány některé doplňující atributy. Atributy uvozené hvězdičkou musí obsahovat data čili nesmí mít hodnotu *NULL*. Primární klíče

jsou uvozeny navíc označením *PK*. Cizí klíče jsou u faktových tabulek umístěny na konci výčtu atributů



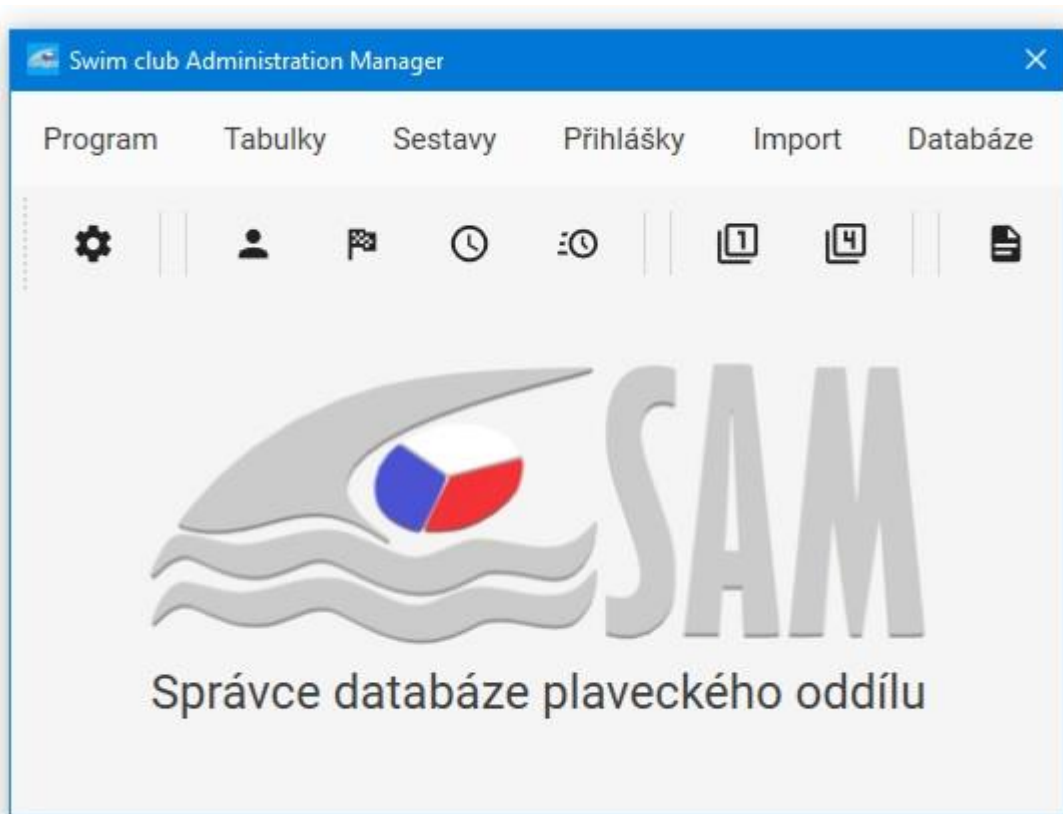
Obrázek 3.3 - ER diagram fyzického návrhu

3.3 Funkční návrh

Kapitola popisuje, jakým způsobem jsou navrženy základní funkční moduly aplikace. Funkční návrh reflektuje požadavky plaveckého oddílu na funkcionalitu aplikace. V následujících podkapitolách budou představeny jednotlivé moduly, jejich funkce a obsluha. Vše doplníme snímky z aplikace.

3.3.1 Hlavní okno

Hlavní okno je okno, které vidí uživatel jako první hned po spuštění aplikace. Cílem formuláře³³ je zapůsobit na uživatele v rámci prvního dojmu a poskytnou přehledný rozcestník do všech koutů aplikace.



Obrázek 3.4 – Formulář Hlavní okno

Horní řádek okna patří rozbalovací nabídce. Při kliknutí myši na příslušnou sekci se ukáže její obsah s odkazy na další formuláře. Druhý řádek tlačítek pro rychlý přístup

³³ Formulářem se také označuje okno v kontextu programování formulářových (okenních) aplikací na OS Windows

nabízí nejčastěji používané formuláře. Tlačítka všech formulářů aplikace včetně těch v hlavním okně jsou opatřena tzv. tooltipy. To jsou dodatečné informace o funkci tlačítka mající podobu menšího okna s textem zobrazovaného vedle ukazovátka myši. Největší prostor zabírá logo programu společně s jeho názvem. Okno má pevné rozměry a nejde tedy zmenšovat nebo zvětšovat.

Aplikace se jmenuje SAM. Název vzniknul z prvních písmen anglického pojmenování *Swim club Administration Manager*. Český překlad zní *Správce databáze plaveckého oddílu*.

3.3.2 Tabulky

Nabídka tabulky umožňuje přístup ke všem tabulkám v databázi. Jsou to dimenzní tabulky plavců, závodů, stylů a faktové tabulky individuálních časů a štafetových časů. Pro každou zmíněnou tabulku je vytvořen samostatný formulář, díky kterému má uživatel možnost prohlížet, vybírat, třídit, editovat a exportovat obsah tabulky. Nabídka tabulky obsahuje ještě dva formuláře, které se vymykají popsané charakteristice.

První je formulář *Přehled*, který zobrazuje v jednom okně tabulky plavců, závodů, stylů, individuálních časů. Uživateli je umožněno klikat na data v tabulkách. Na základě uživatelské volby se přepíše obsah ostatních tabulek. Tak lze získat například informaci: Zvoleného závodu se zúčastnili tyto plavci, kde dále zvolený plavec plaval tyto disciplíny. Formulář je pouze přehledový a neumožňuje proto žádné editace tabulek ani export dat.

Závody

Začátek	Konec	Název	Město	50m bazén	El. časomíra
13.01.2019	13.01.2019	MČR družstev 1. kolo	Brno	<input type="checkbox"/>	<input type="checkbox"/>
28.12.2018	29.12.2018	Silvestrovská cena Vsetína	Vsetín	<input type="checkbox"/>	<input type="checkbox"/>
13.12.2018	16.12.2018	Zimní MČR dorostu a dospělých v Plzni	Plzeň	<input type="checkbox"/>	<input checked="" type="checkbox"/>
24.11.2018	25.11.2018	Vánoční cena Zlína	Zlín	<input type="checkbox"/>	<input checked="" type="checkbox"/>
03.11.2018	04.11.2018	AXIS Cup Jihlava	Jihlava	<input type="checkbox"/>	<input checked="" type="checkbox"/>
20.10.2018	21.10.2018	Velká cena Brna	Brno	<input type="checkbox"/>	<input checked="" type="checkbox"/>
15.10.2018	15.10.2018	Memoriál Jaroslava Starého	Blansko	<input type="checkbox"/>	<input type="checkbox"/>
27.09.2018	27.09.2018	Roskovická čtyřstovka	Roskovice	<input type="checkbox"/>	<input type="checkbox"/>

Počet: 251

Plavci

Příjmení	Jméno	Ročník
Demová	Kateřina	2002
Kučera	Milan	2003
Švarc	Radim	2002
Vencel	Jan	1992

Počet: 4

Styl

Zkratka
50P
100P
50M
100M
100PZ

Počet: 5

Časy

Čas	Kolo	Umístění	Rozplavba	Dráha	Body	Mezičas	1. úsek štafety	Diskvalifikace
00:29,64	Rozplavba	15	4	6	618	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
00:29,64	Semifinále	18	1	0	618	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
00:30,01	Rozplavba	5	6	6	596	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
00:29,46	Semifinále	7	1	6	630	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
00:29,65	Finále	8		1	618	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
01:03,04	Finále	8		1	686	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
01:03,37	Rozplavba	5	6	6	676	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
01:02,89	Semifinále	7	1	6	691	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
00:25,45	Rozplavba	14	4	8	629	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
00:25,49	Semifinále	14	1	1	626	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Počet: 15

Obrázek 3.5 - Formulář Přehled

Druhým odlišným formulářem je ten, který zobrazuje data tabulky základních časů. Jeho cílem je jenom ukázat uživateli obsah tabulky z databáze, aby věděl, jak aktuální či neaktuální její obsah je. Editace těchto dat není povolena, protože se jedná o data zveřejňované centrálně organizací FINA, která si musí uživatel do databáze nahrát přes aplikační rozhraní v sekci *Import dat*. Tuto operaci je doporučováno udělat jako první věc po instalaci aplikace. Bez existence základních časů totiž nebude aplikace schopná počítat body u časů (individuálních i štafetových) což se děje například při importu výsledků do databáze. Organizace FINA vydává každý rok soubor s novými základními časy. Základní čas je nejlepší světový čas v každé plavecké disciplíně převedený na vteřiny. Tyto časy se nejenom v programu používají k výpočtu bodové hodnoty plaveckého výkonu.

ID	FINAYear	Course	Gender	Count	Distance	Stroke	BaseTime	BTsec
1	2008	SCM	M	1	50	FREE	21.10	21.1
2	2008	SCM	M	1	100	FREE	46.45	46.45
3	2008	SCM	M	1	200	FREE	1:42.47	102.47
4	2008	SCM	M	1	400	FREE	3:37.85	217.85
5	2008	SCM	M	1	800	FREE	7:36.85	456.85
6	2008	SCM	M	1	1500	FREE	14:27.07	867.07
7	2008	SCM	M	1	50	BACK	23.46	23.46
8	2008	SCM	M	1	100	BACK	50.47	50.47
9	2008	SCM	M	1	200	BACK	1:50.18	110.18
10	2008	SCM	M	1	50	BREAST	26.56	26.56
11	2008	SCM	M	1	100	BREAST	57.82	57.82
12	2008	SCM	M	1	200	BREAST	2:05.13	125.13
13	2008	SCM	M	1	50	FLY	22.79	22.79
14	2008	SCM	M	1	100	FLY	50.20	50.2
15	2008	SCM	M	1	200	FLY	1:51.38	111.38
16	2008	SCM	M	1	100	MEDLEY	52.23	52.23
17	2008	SCM	M	1	200	MEDLEY	1:53.86	113.86
18	2008	SCM	M	1	400	MEDLEY	4:02.85	242.85
19	2008	SCM	M	4	50	FREE	1:26.74	86.74
20	2008	SCM	M	4	100	FREE	3:10.40	190.4
21	2008	SCM	M	4	200	FREE	7:02.46	422.46
22	2008	SCM	M	4	50	MEDLEY	1:34.75	94.75
23	2008	SCM	M	4	100	MEDLEY	3:28.00	208
24	2008	SCM	F	1	50	FREE	23.81	23.81

Obrázek 3.6 - Formulář Základní časy

Nyní se vraťme k formulářům z úvodu této podkapitoly. Všech pět (*Plavci, Závody, Styly, Individuální časy a Štafetové časy*) má velice podobnou strukturu i funkcionalitu. Proto bude stačit, když si popíšeme pouze jednoho zástupce. Vybereme *Individuální časy*, protože se jedná o nejrozsáhlejší formulář a co se týče funkcionality, ostatní jsou jeho podmnožinou.

Příjmení	Jméno	Ročník	Styl	Čas	Body	Mezičas	Datum	Závody
Vencel	Jan	1992	50M	00:31,35	336	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Vencel	Jan	1992	100M	01:08,15	351	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Vencel	Jan	1992	100P	01:09,65	509	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Vencel	Jan	1992	50P	00:32,40	473	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Vencel	Jan	1992	100M	01:01,99	467	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Vencel	Jan	1992	50M	00:29,34	410	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Vencel	Jan	1992	100VZ	00:56,76	496	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Vencel	Jan	1992	50VZ	00:27,29	409	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Švarc	Radim	2002	200P	02:36,38	457	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Švarc	Radim	2002	50P	00:34,97	376	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Švarc	Radim	2002	100P	01:15,17	405	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Švarc	Radim	2002	400PZ	05:07,89	447	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Švarc	Radim	2002	50M	00:31,34	337	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Švarc	Radim	2002	100M	01:07,30	365	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Švarc	Radim	2002	100P	01:10,01	501	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Švarc	Radim	2002	50P	00:32,81	456	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Švarc	Radim	2002	50VZ	00:25,76	486	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Kučera	Milan	2003	100VZ	00:54,84	550	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Kučera	Milan	2003	50VZ	00:26,28	458	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Kučera	Milan	2003	100Z	01:03,35	460	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Kučera	Milan	2003	50Z	00:30,73	378	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Kučera	Milan	2003	200VZ	02:02,19	538	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Kučera	Milan	2003	50VZ	00:27,84	385	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Kučera	Milan	2003	100VZ	00:58,29	458	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Kučera	Milan	2003	100M	00:59,86	518	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Kučera	Milan	2003	50M	00:28,21	461	<input checked="" type="checkbox"/>	13.01.2019	MČR družstev 1. kolo
Vencel	Michal	1996	100VZ	00:56,15	513	<input type="checkbox"/>	13.01.2019	MČR družstev 1. kolo

Obrázek 3.7 - Formulář Individuální časy

Rozdělme si formulář z orientačních důvodů na část ovládací a datovou. V levé ovládací části jsou umístěny záložky obsahující ovládací prvky. V pravé datové části jsou zobrazena samotná data pocházející nejen z faktové tabulky, ale i z jejich dimenzí. Šířka formuláře se přizpůsobuje množství zobrazovaných sloupců, které si uživatel může nastavit. Zápatí formuláře je vyhrazeno pro informační panel, který počítá celkové množství záznamů tabulky individuálních časů v databázi a také aktuální počet zobrazených řádků v datové části okna.

První záložka, která je vybrána na *Obrázku 3.7* je záložka *Zobrazení*. Nabízí uživateli nastavit, které sloupce ho zajímají a které se tak mají zobrazovat v datové části. Toto nastavení lze uložit a mít ho tak v nezměněné podobě při dalším spuštění aplikace. Hned vedle je panel *Seskupit* s prostorem pro volbu sloupců, které bude obsahovat klauzule Group By. Je možné seskupovat jenom podle těch sloupců, které jsou zobrazeny. Klauzule je součástí agregační funkce, kterou lze spustit v panelu *Agregace*. Vybereme, jestli chceme agregovat body nebo čas spolu s agregační funkcí *Min* nebo *Max*. Pomocí tohoto aparátu a s přispěním ostatních záložek jsme schopni vytvářet pokročilé dotazy do

databáze. Jedním z nich může být třeba výběr nejlepších mužských časů v jednotlivých disciplínách na 25metrovém bazénu v konkrétním období. Možnost agregací mají pouze formuláře *Individuální časy* a *Štafetové časy*, protože agregace mají smysl, pokud jsou prováděny na větším množství dat.

Pole pod panelem *Agregace* s názvem *Max. počet záznamů* umožňuje uživateli zadat až čtyřciferné číslo představující nejvyšší počet záznamů, které vrátí dotaz do databáze. Protože není možné zobrazovat v datové části desetitisíce záznamů.

Posuneme se doprava na záložku *Výběr*. Záložka má za úkol umožnit uživateli vybírat data podle rozličných kritérií. Každý řádek v ovládací části představuje jedno výběrové kritérium. Ty lze přidávat a ubírat pomocí tlačítek plus a minus. Mezi kritérii je možno nastavit spojení představující logické spojky AND a OR. Každé kritérium se skládá ze tří položek – *Sloupec*, *Operátor* a *Hodnota*. Ukázkovému výběru na *Obrázku 3.8* odpovídá 21 záznamů v databázi.

The screenshot shows a web application window titled "Časy - individuální starty". The interface is divided into a sidebar for filter configuration and a main table of results.

Filter Configuration (Sidebar):

- Buttons: **Zobrazení**, **Výběr** (active), **Třídění**, **Editace**, **Export**
- Rule 1: Sloupec: **Příjmení**, Operátor: **začíná na**, Hodnota: **Venc**. Logic: **a zároveň** (selected).
- Rule 2: Sloupec: **Styl**, Operátor: **rovná se**, Hodnota: **50VZ**. Logic: **a zároveň** (selected).
- Rule 3: Sloupec: **50m bazén**, Operátor: **rovná se**, Hodnota: **Ano** (checked). Logic: **a zároveň** (selected).
- Rule 4: Sloupec: **Body**, Operátor: **je větší**, Hodnota: **500**. Logic: **a zároveň** (selected).
- Buttons: **+**, **-**, **Zrušit výběr**, **Vybrat**

Table of Results:

Příjmení	Jméno	Styl	Čas	Body	Datum	Závody
Vencel	Michal	50VZ	00:25,51	551	25.05.2018	Velká cena města Pardubic
Vencel	Jan	50VZ	00:26,08	515	05.05.2018	Krajský přebor dorostu a dospělých v Brně
Vencel	Michal	50VZ	00:26,19	509	02.06.2017	Praha 2017 - 4. kolo ČP
Vencel	Michal	50VZ	00:26,10	514	06.05.2017	Krajský přebor dorostu a dospělých
Vencel	Michal	50VZ	00:26,26	505	29.04.2017	Velká cena Ostravy
Vencel	Michal	50VZ	00:25,20	571	01.07.2016	Letní MČR open v Pardubicích
Vencel	Michal	50VZ	00:25,68	540	27.06.2016	České akademické hry v Pardubicích
Vencel	Michal	50VZ	00:25,67	540	27.06.2016	České akademické hry v Pardubicích
Vencel	Michal	50VZ	00:25,90	526	07.05.2016	Krajský přebor dorostu a dospělých v Brně
Vencel	Michal	50VZ	00:25,33	563	30.04.2016	Velká cena Ostravy
Vencel	Michal	50VZ	00:25,62	544	30.04.2016	Velká cena Ostravy
Vencel	Michal	50VZ	00:25,52	550	15.04.2016	Velká cena města Pardubic
Vencel	Michal	50VZ	00:25,83	531	12.03.2016	Velká cena Hradce Králové
Vencel	Michal	50VZ	00:25,11	577	17.07.2015	Letní MČR open v Praze - 5. kolo ČP
Vencel	Michal	50VZ	00:25,28	566	29.05.2015	Praha 2015 - 4. kolo ČP
Vencel	Michal	50VZ	00:25,32	563	18.04.2015	Velká cena Ostravy
Vencel	Michal	50VZ	00:25,73	537	28.03.2015	Velká cena Olomouce
Vencel	Michal	50VZ	00:25,61	544	20.06.2014	Letní mistrovství ČR dorostu v Ústí nad Labem
Vencel	Michal	50VZ	00:25,68	540	17.05.2014	Velká cena města Pardubic
Vencel	Michal	50VZ	00:25,97	522	26.04.2014	Velká cena Ostravy
Vencel	Michal	50VZ	00:25,98	521	29.03.2014	Velká cena Olomouce

Summary: Počet všech záznamů v databázi: 24265; Počet zobrazených záznamů: 21

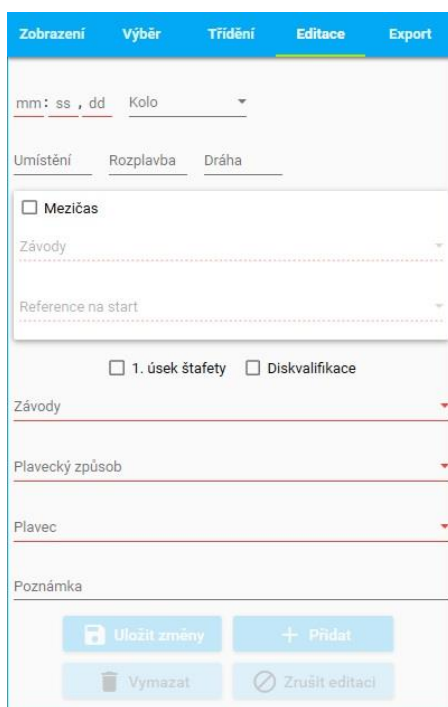
Obrázek 3.8 - Zadávání výběrových kritérií

Záložka *Třídění* pracuje velmi podobně. Třídící kritéria se skládají jenom ze dvou položek – *Sloupec* a *Třídění*. Zde můžeme specifikovat podle kterého sloupce chceme

data setřídít. Máme na výběr různé způsoby třídění. Třeba A-Z, vzestupně nebo sestupně. Lze si povšimnout, že pokud aktivujeme ovládací prvky na kterékoliv záložce, zbarví se nadpis záložky zeleně. Je to proto, aby uživatel věděl, kde všude provedl nastavení ovlivňující datovou část.

Předposlední záložka *Editace* slouží ke vkládání, upravování a mazání záznamů. Výchozí podoba ovládací části zahrnuje prázdné pole odpovídající sloupcům databázové tabulky. Po jejich vyplnění může uživatel vložit nový záznam do databáze. V případě zvolení řádku myší v datové části se automaticky vyplní prázdná nebo přepíše vyplněná pole v části ovládací. Tato pole lze editovat a změněný záznam uložit nebo úplně vymazat z databáze.

Všechna pole v aplikaci, kde je uživatel žádán o vstup z klávesnice, jsou validována. Pokud tedy uživatel vloží jiný typ znaku, než je očekáván (např. místo čísla písmeno), aplikace zobrazí okno upozornění a je nutné znak změnit. Pole podtržená červeně jsou povinná.



Obrázek 3.9 – Možnosti editace tabulky

Může se stát, že uživatel chce vymazat záznam z dimenzní tabulky, který má navázaný záznam v tabulce faktové skrz cizí klíč. V takovém případě je uživatel na tuto

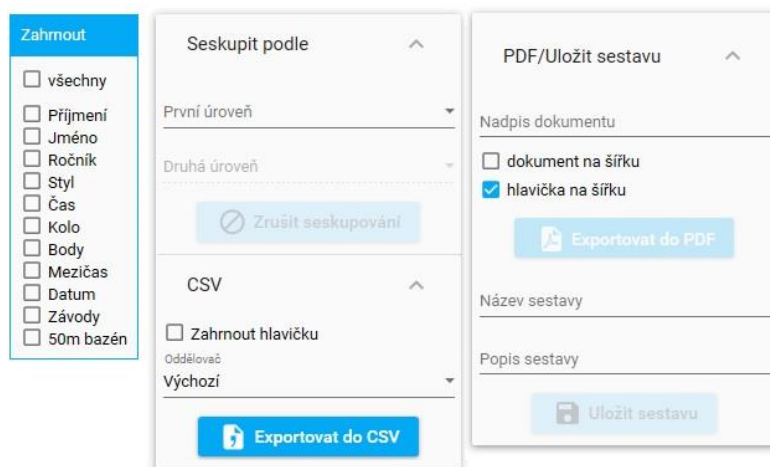
skutečnost upozorněn a je mu dovoleno záznam vymazat pouze tak, že se smažou i všechny navázané záznamy. Tím zůstane zachována integrita databáze.

Export je poslední záložka. Zde můžeme exportovat data, která jsme získali jako výsledek operací agregace, výběr a třídění. Před exportem jakéhokoliv typů je potřeba nejprve vybrat sloupce nabízené v kartě *Zahrnout*. Tím určíme, které sloupce chceme vidět ve výsledných dokumentech.

Pro další zpracování dat, například programem MS Excel, použijeme export do formátu CSV. Lze nastavit, jestli má výsledný dokument obsahovat hlavičku (názvy sloupců na prvním řádku) a také je možné specifikovat konkrétní oddělovač polí.

Na kartě *PDF/Uložit sestavu* použijeme export do formátu PDF, když chceme data zveřejnit a prezentovat. Vymyslíme nadpis dokumentu, který bude uveden v jeho horní části tučně. Podle množství sloupců vybereme, zda má být dokument nebo hlavička sloupců na šířku. Tímto způsobem jde na papír dostat i poměrně široká data. Na této kartě také můžeme uložit sestavu. Do XML dokumentu se uloží nastavení agregace, výběru a třídění, kterému odpovídá určitá podmnožina dat z databázové tabulky. Tomuto souboru nastavení říkáme sestava. Uloženou sestavu si uživatel může později spustit znovu nad aktualizovanou podobou databáze. Podrobněji je problematika sestav rozebrána v následující kapitole 3.3.3.

Ještě pár slov k volbě *Seskupit podle*. Tu mohou využít ti, kteří chtějí obsah svých PDF dokumentů strukturovat podle vybraných sloupců. Je k dispozici maximálně dvou úrovněová struktura.



Obrázek 3.10 - Možnosti exportu dat

3.3.3 Sestavy

Nabídka *Sestavy* obsahuje dva formuláře pro sestavy vzniklé z individuálních času a štafetových časů. Díky těmto formulářům může uživatel přistupovat k uloženým sestavám. Oba rozlišují dva typy uložených sestav – *Výchozí* a *Uživatelské*. Výchozí jsou předem definovány autorem aplikace a uloženy v aplikačních datech. Jedná se o nejčastěji používané sestavy generované z plavecké databáze. Vlastní sestavy si uživatel vytvoří volbou *Uložit sestavu* v záložce *Export* u formulářů *Individuální časy* nebo *Štafetové časy*. Tyto sestavy potom najde zde pod záložkou *Uživatelské*.

Po vybrání kterékoliv sestavy je k dispozici rozbalovací panel obsahující její specifikaci. Uživatel odtud může vyčíst podrobnější popis a jiné vlastnosti sestavy. Popis je volitelný a vytváří ho autor sestavy. Zbylé vlastnosti se generují automaticky, což mírně snižuje jejich čitelnost. Před vygenerováním některých výchozích sestav je požadováno zvolit plaveckou disciplínu sestavy. Toto je vlastnost, které uživatel při vytváření vlastních sestav nedosáhne. Uživatelské sestavy mohou být smazány, výchozí nikoliv.

Uložené individuální sestavy

Výchozí Uživatelské

Rekordy - krátký bazén - muži

Rekordy - krátký bazén - ženy

Pořadí v dané disciplíně podle časů - krátký bazén - muži

Pořadí v dané disciplíně podle časů - krátký bazén - ženy

Pořadí v dané disciplíně podle časů - dlouhý bazén - muži

Pořadí v dané disciplíně podle časů - dlouhý bazén - ženy

Specifikace sestavy

Popis sestavy: Pořadí plavců ve zvolené disciplíně podle časů bez ohledu na věk. Časy z mezičasů a 1. úseků štafet jsou uvažovány. Časy jsou brány od počátku databáze.

Formát dokumentu: A4 svisle

Výběrová kritéria: Gender == "Muž" and Lcm == "False"

Třídící kritéria: Time ASC

Seskupování výběru podle: Surname, Firstname, SwimStyleShortcut, Lcm

Agregační funkce: Čas - min

Seskupování pdf exportu podle: nazadáno

Maximální počet záznamů: 1000

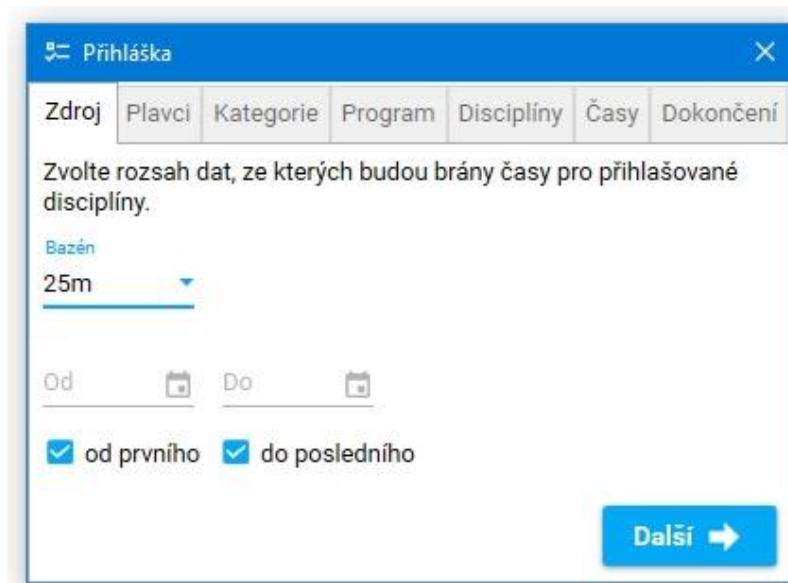
Disciplína Vytvořit

Obrázek 3.11 - Formulář *Sestavy*

3.3.4 Přihlášky

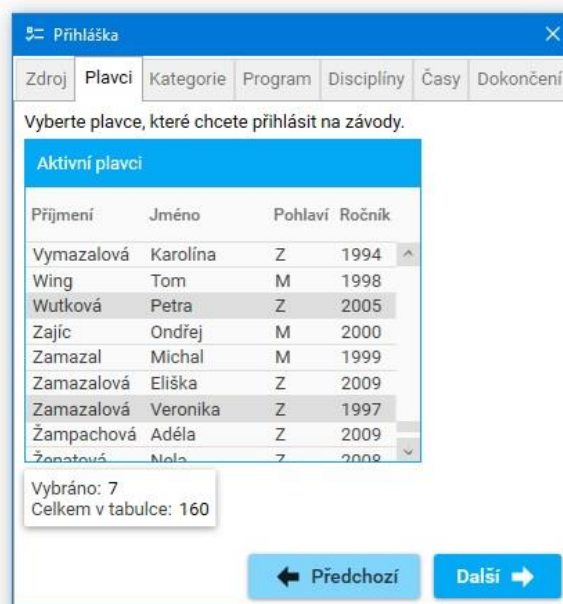
Formulář *Přihlášky* se skládá ze série záložek, jejichž postupné procházení a vyplňování tvoří proces tvorby přihlášky na závody. K posuvu mezi záložkami slouží tlačítka *Další* a *Předchozí* v pravém dolním rohu okna.

Začneme volbou parametrů určujících rozsah dat, ze kterých budou brány časy pro přihlašované disciplíny. Vybereme typ bazénu a časové období. Pokud nám nevyhovují hranice období určené konkrétními daty, je možné uvažovat celý obsah databáze od jejího počátku až do po nejnovější záznam.



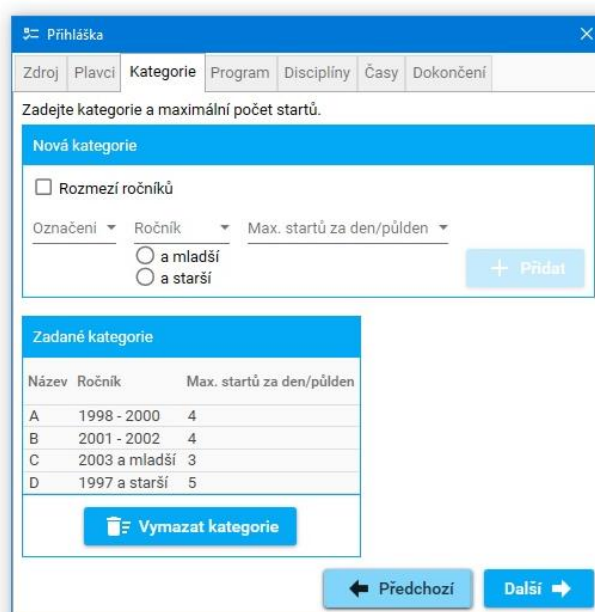
Obrázek 3.12 - Obsah záložky *Zdroj*

Následuje výběr plavců, které chceme na závody přihlásit. Více jich vybereme s přidržením klávesy Ctrl.



Obrázek 3.13 - Obsah záložky *Plavci*

Nyní je potřeba vytvořit kategorie odpovídající závodům, na které tvoříme přihlášku. Panel *Nová kategorie* slouží k vytváření jednotlivých kategorií. Zvolíme její jednopísmenné označení, určíme její rozsah a maximální počet startů za den nebo půlden (záleží na typu závodů). Způsob zadávání rozsahu kategorie můžeme měnit přepínačem *Rozmezí ročníků*. Vytvořené kategorie se shromažďují v panelu *Zadané kategorie*. Obsah tohoto panelu lze celý smazat tlačítkem *Vymazat kategorie*.



Obrázek 3.14 - Obsah záložky *Kategorie*

Další záložka *Program* nás provede tvorbou programu disciplín, který najdeme na rozpise závodů. Začneme tím, že zadáme název, město a datum začátku závodů. Vytváření programu závod probíhá podobně jak vytváření kategorií. V panelu *Nová disciplína* zadáme číslo, disciplínu, pohlaví a jednu nebo více kategorií, které mohou danou disciplínu plavat. Jednotlivé disciplíny postupně přidáváme do panelu *Program závodů*.

č.	Disciplína	Pohlaví	Kategorie
1	50P	M	A, B, D, C
2	50P	Z	A, B, C, D
3	100M	M	A, B, D
4	100M	Z	A, B, D
5	400PZ	M	A, D
6	400PZ	Z	A, D

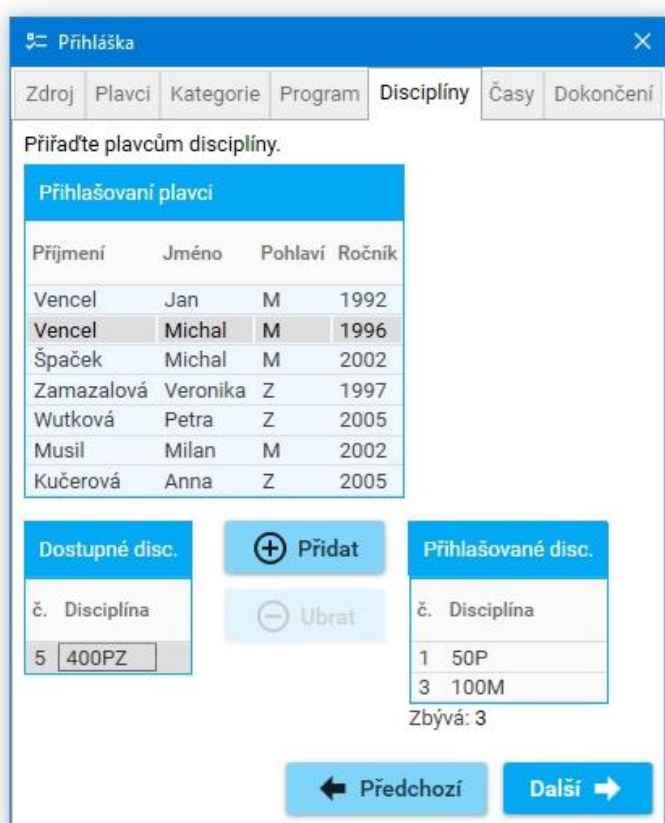
Obrázek 3.15 - Obsah záložky *Program*

Program lze vymazat jako celek nebo po jednom řádku, který musíme nejdříve označit kliknutím myší. Hotový program jde uložit do souboru na disk počítače. Načtení programu usnadní pozdější tvorbu přihlášky na stejné závody, u kterých zůstává program rovněž stejný.

Pokračováním na další záložku bude obsah této a předchozích záložek uzamknut. Jejich editace v pozdější fázi tvorby přihlášky by způsobila datové konflikty. Nicméně, ve výjimečných případech je umožněna editace po kliknutí na tlačítko *Editovat*, které se nově objeví. Cenou za editaci uzamčených záložek je výmaz obsahu všech záložek, které následují za aktivní záložkou.

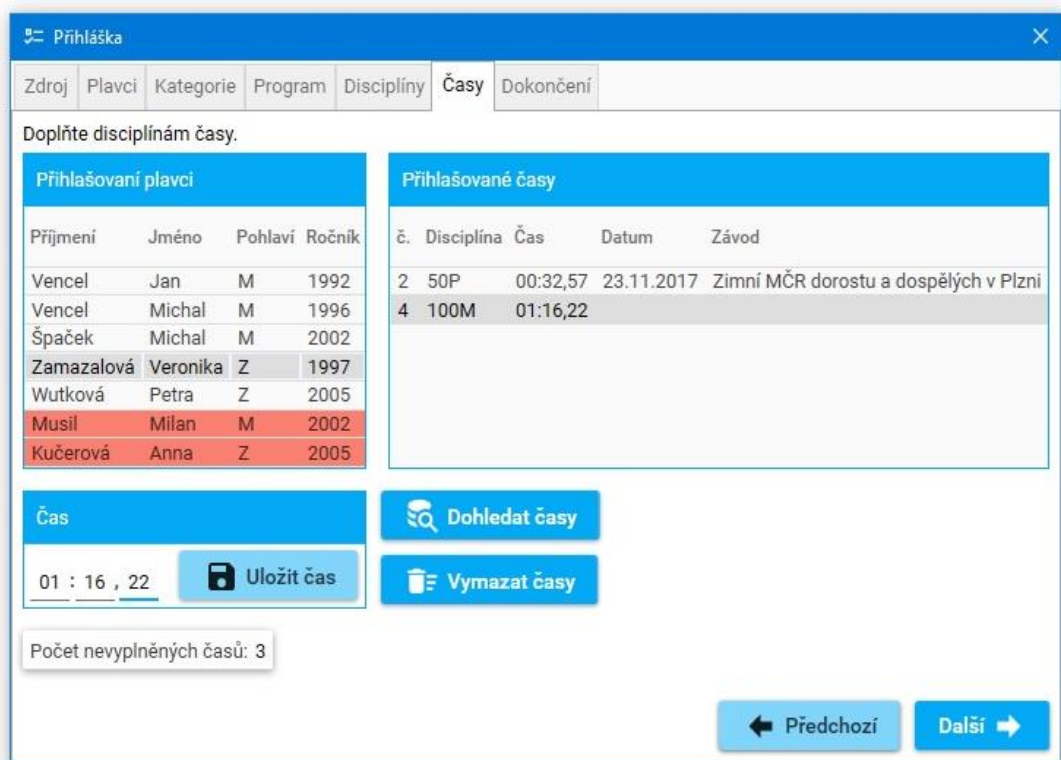
Na první pohled se může zdát, že vytváření kategorií a následně programu závodů není nutné k sestavení přihlášky. Což je pravda. Proč to i tak děláme si zdůvodníme na

další záložce *Disciplíny*. Máme zde seznam plavců, které jsme na začátku vybrali. Když na některého z nich klikneme myší, naplní se panel *Dostupné disciplíny*. To jsou disciplíny, které plavci mohou být přihlášeny. Právě toto aplikace vyhodnotí na základě dříve vytvořených kategorií a programu závodů. Samotné přihlášení disciplíny proběhne po jejím označení (více přes tlačítko Ctrl) a stisknutí tlačítka *Přidat*. Maximální počet přihlášených disciplín reflektuje ukazatel pod panelem *Přihlašované disciplíny*. Tuto podmínku aplikace opět hlídá díky existenci kategorií. Přihlašovanou disciplínu lze i odebrat.



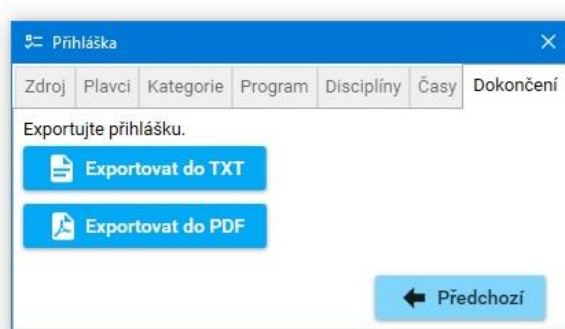
Obrázek 3.16 - Obsah záložky *Disciplíny*

Na předposlední záložce *Časy* přidáme časy všem přihlášeným disciplínám u každého plavce. Zadávají se takové časy, které se očekávají od daného plavce. Časy používá pořadatel závodů k rozlosování plavců do rozplaveb.



Obrázek 3.17 - Obsah záložky *Časy*

Aplikace nabízí možnost přiřadit časy automaticky stisknutím tlačítka *Dohledat časy*. V tom případě jsou v databázi vyhledány osobní rekordy plavců a ty jsou přiřazeny přihlašovaným disciplínám. K nalezenému času je ještě doplněno datum a místo zaplavání. Pokud plavec onu disciplínu ještě neplaval (není záznam v databázi), je potřeba čas přiřadit ručně. Plavci s jedním a více nenalezených časů jsou označeni červeně. Po zvolení plavce a následně disciplíny, lze čas editovat a posléze uložit. Tlačítkem *Vymazat časy* se smažou všechny přiřazené časy u všech plavců. Jakmile panel s počtem nevyplněných časů ukazuje nulu, lze přejít na poslední záložku.



Obrázek 3.18 - Obsah záložky *Dokončení*

Na závěr na záložce *Dokončení* provedeme export přihlášky do textového a PDF souboru. Textový soubor pošleme pořadatelům a PDF soubor vystavíme na internetových stránkách oddílu.

190316

Velká cena města Blanska

Blansko

16.03.2019

ASKB1

```
"Vencel";"Jan";"1992";"M";"1";"50P";"00:29,37";"Velká cena Znojma";"08.09.2018"
"Vencel";"Jan";"1992";"M";"3";"100M";"00:55,79";"Zimní MČR dorostu a dospělých v Plzni";"13.12.2018"
"Vencel";"Jan";"1992";"M";"5";"400PZ";"04:38,30";"Silvestrovská cena Vsetína";"29.12.2015"
"Vencel";"Michal";"1996";"M";"1";"50P";"00:31,54";"Velká cena Znojma";"19.09.2015"
"Vencel";"Michal";"1996";"M";"3";"100M";"00:59,77";"Velká cena Znojma";"10.09.2016"
"Špaček";"Michal";"2002";"M";"1";"50P";"00:39,35";"Silvestrovská cena Vsetína";"28.12.2017"
"Špaček";"Michal";"2002";"M";"3";"100M";"01:11,73";"Silvestrovská cena Vsetína";"28.12.2017"
"Zamazalová";"Veronika";"1997";"Z";"2";"50P";"00:32,57";"Zimní MČR dorostu a dospělých v Plzni";"23.11.2017"
"Zamazalová";"Veronika";"1997";"Z";"4";"100M";"01:16,22";"";""
"Wutková";"Petra";"2005";"Z";"2";"50P";"00:42,06";"Brno Cup v Brně Lesné";"16.12.2017"
"Musil";"Milan";"2002";"M";"1";"50P";"00:34,36";"";""
"Musil";"Milan";"2002";"M";"3";"100M";"01:08,82";"Silvestrovská cena Vsetína";"28.12.2017"
"Kučerová";"Anna";"2005";"Z";"2";"50P";"00:38,67";"";""
```

Obrázek 3.19 - Ukázka obsahu exportovaného textového souboru

Přihláška ASKB1 - Velká cena města Blanska

Ročník	Jméno	Číslo	Disciplína	Čas
1992	Vencel Jan	1	50P	00:29,37
		3	100M	00:55,79
		5	400PZ	04:38,30
1996	Vencel Michal	1	50P	00:31,54
		3	100M	00:59,77
2002	Špaček Michal	1	50P	00:39,35
		3	100M	01:11,73
1997	Zamazalová Veronika	2	50P	00:32,57
		4	100M	01:16,22
2005	Wutková Petra	2	50P	00:42,06
2002	Musil Milan	1	50P	00:34,36
		3	100M	01:08,82
2005	Kučerová Anna	2	50P	00:38,67

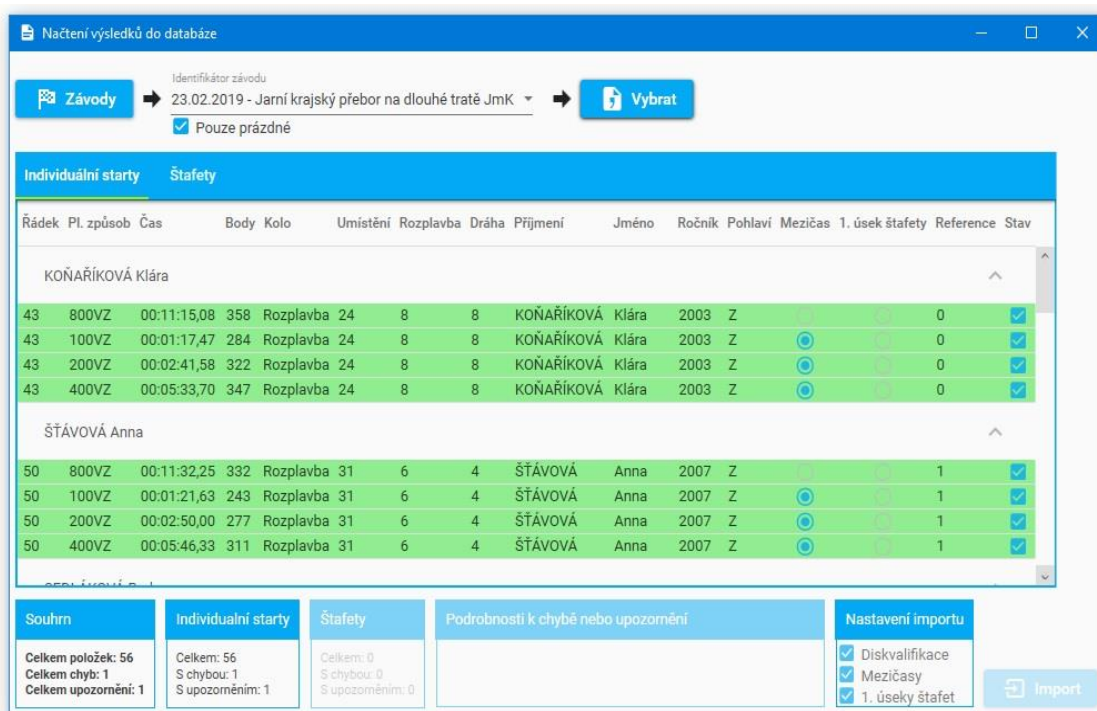
Obrázek 3.20 - Ukázka obsahu exportovaného PDF dokumentu

3.3.5 Import

Nabídka Import v sobě zahrnuje dva formuláře. Vedle ručního vkládáním řádků do jednotlivých tabulek poskytuje tato nabídka další možnosti k tvorbě plavecké databáze. Na rozdíl od ručního vkládání samostatných řádků, které je zdlouhavé a náchylné na chyby, je import automatizovaný a kontrolovaný proces zvládající nahrávat do databáze velké množství řádků v řádu sekund.

První formulář *Import výsledků* bude pravděpodobně nejpoužívanější, proto je u něj obzvláště důležité zajistit jeho srozumitelnost a bezporuchovost. Uživatel v první řadě potřebuje získat výsledky v textové podobě ze stránek ČSPS. Nyní přes odkaz Závody spustí formulář závodů a vloží nový závod do tabulky. Zavře formulář Závody a pokračuje výběrem identifikátoru právě vloženého závodu. Přepínač Pouze prázdné je vhodné mít stále aktivován, protože tím je zajištěno, že uživateli budou nabízeny pouze závody bez nahraných výsledků. Tlačítko Vybrat nás vyzve k zvolení získaného textového souboru s výsledky a zahájí jeho čtení. Za pár vteřin (v závislosti na velikosti souboru) se v tabulce objeví načtená data.

Tabulka s načtenými záznamy je rozdělena do záložek s individuálními a štafetovými časy. U těch individuálních jsou časy seskupeny do oddílů podle jednotlivých plavců. Oddíly lze pro lepší přehlednost sbalit i rozbalit. Nejlepší přehlednosti však dosáhneme zvětšením okna na celou obrazovku monitoru. Štafetových časů bývá několikanásobně méně, proto nejsou strukturovány do oddílů. Také zde neexistují mezičasy, a tudíž i reference. Navíc je zde naopak sloupec, určující typ štafety.



Obrázek 3.21 - Formulář *Import výsledků*

Popišme si strukturu řádku individuálního času. Na první pozici je číslo řádku vstupního souboru, ze kterého byl záznam získán. Řádky dále obsahují jak přečtená pole

ze vstupního souboru (disciplína, čas, umístění, rozplavba, dráha, příjmení, jméno, ročník, pohlaví), tak odvozené (disciplína u mezičasů, kolo) nebo vypočítané (body) údaje. Body sice vstupní soubor obsahuje, ale mohou být zastaralé, proto si je aplikace počítá sama. Čas může pocházet i z mezičasu nebo prvního úseku štafety, což signalizují puntíky ve sloupcích *Mezičas* a *1. úsek štafety*. Na *Obrázku 3.21* závodnice plavaly na závodech pouze jednu disciplínu (800 volný způsob), ale do databáze budou nahrány čtyři záznamy místo jednoho. Zbylé tři jsou totiž mezičasy na 100, 200 a 400 metrů. Ke které disciplíně mezičasy patří poznáme jak z čísla řádku, tak z čísla ve sloupci *Reference*. Související záznamy mají stejná čísla reference. Poslední sloupec *Stav* shrnuje, zda řádek splňuje požadavky pro import do databáze.

Obarvení řádků je trojí. Zelené jsou ty, ve kterých aplikace nenašla chybu a mohou být importovány do databáze. Žluté řádky obsahují varování, které ovšem nebrání importu. Nejčastěji se jedná o řádek diskvalifikace nebo podezření z chybného času. Toto podezření se zobrazí u mezičasů³⁴, který jsou lepší než osobní rekord plavce v dané disciplíně. A konečně červeně obarvené řádky jsou řádky obsahující chybu. Chybné řádky blokují import výsledů do databáze (pole ve sloupci *Stav* není vyplněno). Chyby mohou být způsobeny neexistencí načteného plavce nebo stylu v příslušných databázových tabulkách. V takovém případě je nejčastějším řešením vložení plavce nebo stylu do tabulek. Také může být chyba ve formátu času ve vstupním souboru. Chyby tohoto typu je nutné opravit editací vstupního souboru.

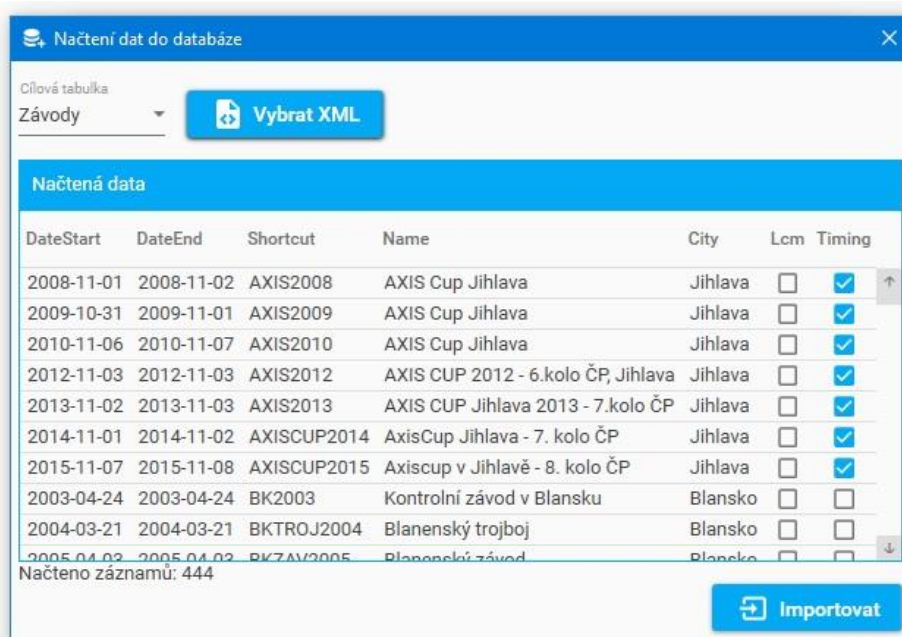
K získání seznamu varování nebo chyb lze rozkliknout řádek záznamu. Dalším kliknutím na vybranou chybu nebo varování se zobrazí podrobnější informace v panelu *Podrobnosti k chybě nebo varování*. Tím jsme se dostali k popisu panelů ve spodní části formuláře. První tři panely zleva obsahují přehled o počtu načtených záznamů, varování a chyb v rámci individuálních startů, štafet i celkově. Panel *Nastavení importu* není editovatelný a zobrazuje pouze nastavení provedené ve formuláři *Konfigurace*. Nastavení určuje, jaké typy záznamů se mají načítat. Výchozí nastavení povoluje všechny typy.

Jakmile načtené záznamy neobsahují chyby, je možné provést import do databáze tlačítkem *Import*. Záznamy se nahrají do faktových databázových tabulek pro

³⁴ Časy z mezičasů lze brát jako plnohodnotné časy odpovídajících disciplín

individuální a štafetové časy. Pokud uživatel provede import časů k závodům, které už nějaké výsledky obsahují, tyto výsledky se přepíší.

Druhý formulář *Import dat* v nabídce *Import* slouží k nahrávání dat do tabulek plavců, závodů a základních časů. Vybereme cílovou tabulku a přes tlačítko *Vybrat XML* vybereme zdrojový soubor formátu XML. Struktura a datové typy v souboru jsou validovány podle XML schématu, které je součástí aplikace. Pokud jsou vstupní data v pořádku, zobrazí se v tabulce. Od tohoto okamžiku je povolen import do databáze. Aplikace importuje jenom položky, které v databázi neexistují.



Obrázek 3.22 - Formulář *Import dat*

3.3.6 Export

Export dat nemá vlastní samostatnou nabídku ani formulář, nicméně možnosti exportu dat jsou prostoupeny většinou tabulkových formulářů (*Plavci, Závodů, Styly, Individuální časy, Štafetové časy*). Dále mají zastoupení při tvorbě sestav nebo přihlášek. Postup exportu dat do formátů CSV a PDF u tabulkových formulářů je popsán v kapitole 3.3.2. *Tabulky*. Postup při exportu sestav je jednoduchý, stačí pouze zvolit požadovanou sestavu z tabulky a kliknout na tlačítko *Vytvořit*. Export přihlášky jsme si také popsali dříve v kapitole 3.3.4. *Přihlášky* a ukázali i podobu výstupu. Zde si ukažme podobu jedné stránky formátu A4, která vznikla exportem sestavy. Černé linky značí okraje stránky.

Rekordy - krátký bazén - ženy

Příjmení	Jméno	Ročník	Styl	Čas	Body	Datum	Závod
Sedláková	Barbora	2002	50Z	00:32,62	487	24.11.2018	Vánoční cena Zlína
Pokorná	Petra	1998	100Z	01:10,50	476	23.04.2016	Cena města Blanska
Sedláková	Barbora	2002	200Z	02:35,51	451	20.10.2017	Plzeňské sprinty - 6. kolo ČP
Zamazalová	Veronika	1997	50P	00:32,57	680	23.11.2017	Zimní MČR dorostu a dospělých v Plzni
Zamazalová	Veronika	1997	100P	01:12,66	632	18.12.2014	Zimní mistrovství ČR open v Plzni
Zamazalová	Veronika	1997	200P	02:41,88	574	15.11.2014	Velká cena Brna
Pokorná	Petra	1998	50M	00:30,43	514	19.11.2016	Velká cena Brna
Kopřivová	Hana	1998	100M	01:08,68	503	15.12.2016	Zimní MČR dorostu a dospělých v Plzni
Kopřivová	Hana	1998	200M	02:31,83	489	10.12.2015	Zimní MČR dorostu a dospělých v Plzni - 11. kolo ČP
Zamazalová	Veronika	1997	50VZ	00:27,44	608	18.12.2014	Zimní mistrovství ČR open v Plzni
Zamazalová	Veronika	1997	100VZ	01:00,14	607	14.10.2017	Memoriál Jaroslava Starého
Zamazalová	Veronika	1997	200VZ	02:10,88	606	14.10.2017	Memoriál Jaroslava Starého
Demová	Kateřina	2002	400VZ	04:37,17	606	20.10.2017	Plzeňské sprinty - 6. kolo ČP
Demová	Kateřina	2002	800VZ	09:44,49	552	13.12.2018	Zimní MČR dorostu a dospělých v Plzni
Demová	Kateřina	2002	1500VZ	18:22,03	581	13.12.2018	Zimní MČR dorostu a dospělých v Plzni
Zamazalová	Veronika	1997	100PZ	01:06,74	612	18.12.2014	Zimní mistrovství ČR open v Plzni
Zamazalová	Veronika	1997	200PZ	02:28,13	557	15.12.2016	Zimní MČR dorostu a dospělých v Plzni
Zamazalová	Veronika	1997	400PZ	05:25,10	508	28.12.2016	Silvestrovská cena Vsetína

Obrázek 3.23 - Příklad exportovaného PDF dokumentu

Každý PDF dokument, vytvořený v aplikaci SAM má nadpis, obsah a zápatí. Nadpis je generován automaticky (výchozí sestavy) nebo volen uživatelem (export vlastních dat z formulářů tabulek, uživatelské sestavy). Obsah dokumentu může být různě

velký. Výšku a potažmo počet stran určuje množství položek a šířku ovlivňuje počet sloupců. Pokud obsah přetéká bočními okraji, je nutno při exportu nastavit formát listu na šířku. V uvedeném příkladě od popsané situace nejsme daleko. Spodní část dokumentu uzavírá patička, která obsahuje datum, logo aplikace s verzí a aktuální stránku společně s celkovým počtem stran.

3.3.7 Zálaha a obnova databáze

Pro zálohu a obnovu databáze není vytvořen samostatný formulář. Jedná se pouze o dvě volby v hlavní nabídce programu. Po zvolení zálohy je uživatel vyzván k pojmenování souboru obsahujícího zálohu a k zadání místa jeho uložení. K obnově databáze použijeme dříve vytvořený zálohovací soubor. Ten má binární podobu a je nepřenosný. To znamená, že obnovit můžeme pouze databázi ze které byl vytvořen zálohovací soubor, nikoli jinou databázi na jiném počítači. Obnova databáze spočívá v tom, že po provedení se databáze bude nacházet v takovém stavu jako při vytvoření zálohy.

3.3.8 Konfigurace

K nastavení chování aplikace slouží samostatný formulář *Konfigurace*. Měl by to být jeden z prvních formulářů, které uživatel spustí po instalaci aplikace. Je zde potřeba nastavit parametry, které se využívají na jiných místech aplikace. Jedním takovým je třeba pole *Zkratka oddílu*. Obsah tohoto pole je potřeba při načítání výsledků ze závodů z textového souboru. Aplikace hledá řádky se zkratkou oddílu, která se shoduje s obsahem zmíněného pole. Obsah pole *Celý název oddílu* má v současné verzi aplikace pouze informační význam. Rozbalovací nabídka *Ročník FINA bodů* obsahuje seznam jednotlivých ročníků základních časů, které obsahuje tabulka základních časů. Podle vybraného ročníku aplikace počítá bodové hodnoty výkonů. Při změně toho pole proběhne přepočítání všech bodových hodnot v databázi na zvolený ročník. Tato akce může trvat několik vteřin v závislosti na velikosti databáze časů.

Obrázek 3.24 - Formulář *Konfigurace*

Ročník vybraný v nabídce *Spodní hranice ročníku* se využívá v aplikaci v tabulkových formulářích při výběru ročníku. Aplikace zde nabízí k výběru rozsah ročníků od aktuálního roku po hodnotu zvolenou v konfiguraci. Je zbytečné a nepřehledné nabízet k výběru ročníky v rámci celého století, když oddíl má nejstaršího plavce například ročníku 1990. V takovém případě uživatel nastaví spodní hranici ročníku na hodnotu 1990.

Nabídka *Zobrazení hodin* obsahuje tři možnosti – *Automaticky*, *Zobrazovat* a *Nezobrazovat hodiny*. U klasických bazénových disciplín (nejdelší má 1500 metrů) se neočekává, že by plavec přesáhl svým časem jednu hodinu. Může se tak stát při dálkových tratích plavaných v bazénu (např. 5000 metrů). Právě pro případ, kdy uživatel nahraje do databáze časy přesahující hodinu, je zde tato nabídka. Výchozí nastavení je *Automaticky*, při němž aplikace kontroluje obsah databáze a když objeví čas přesahující hodinu, tak zobrazuje všechny časy ve formátu s hodinami.

Dále na formuláři najdeme tři zaškrtačací pole, která ovlivňují, zda bude aplikace při importu výsledků ze závodů nahrávat příslušné typy záznamů – diskvalifikace, mezičasy a 1. úseky štafet. Poslední nabídka *Upozornění na body* umožňuje zvolit bodovou hranici. Když tuto hranici aplikace při výpočtu bodů přesáhne, upozorní uživatele, aby překontroloval reálnost zadávaného času. Ve většině českých oddílů není časté, aby se bodová hodnota zaplavaného času vyšplhala nad nastavenou výchozí

hodnotu 800 bodů. Proto pokud se tak stane, je pravděpodobné, že v čase je chyba. Potom je na uživateli, aby to ověřil.

3.3.9 Uživatelská podpora

Podporu pro uživatele představují volby *O programu* a především *Příručka*, které obsahuje nabídka *Program* na hlavním okně aplikace. Formulář *O programu* obsahuje název programu, jeho verzi a kontakt na autora. Volba *Příručka* spustí PDF dokument popisující funkce aplikace.

3.4 Grafický návrh

Vizuální podoba aplikace je navržena tak, aby co nejvíce zjednodušila a zpřehlednila práci uživatelů. Souběžně s tím je cílem grafického návrhu zaujmout oko uživatele a učinit tak práci s aplikací příjemnou a zábavnou.

Základní paleta barev aplikace obsahuje šedou, černou a odstíny modré. Ovládací prvky jsou laděny do modra, protože modrá evokuje vodu nebo bazén. Pozadí oken je světle šedé, aby nepůsobilo rušivý dojem. Text je zbarven do černa, čímž na pozadí dobře vyniká.

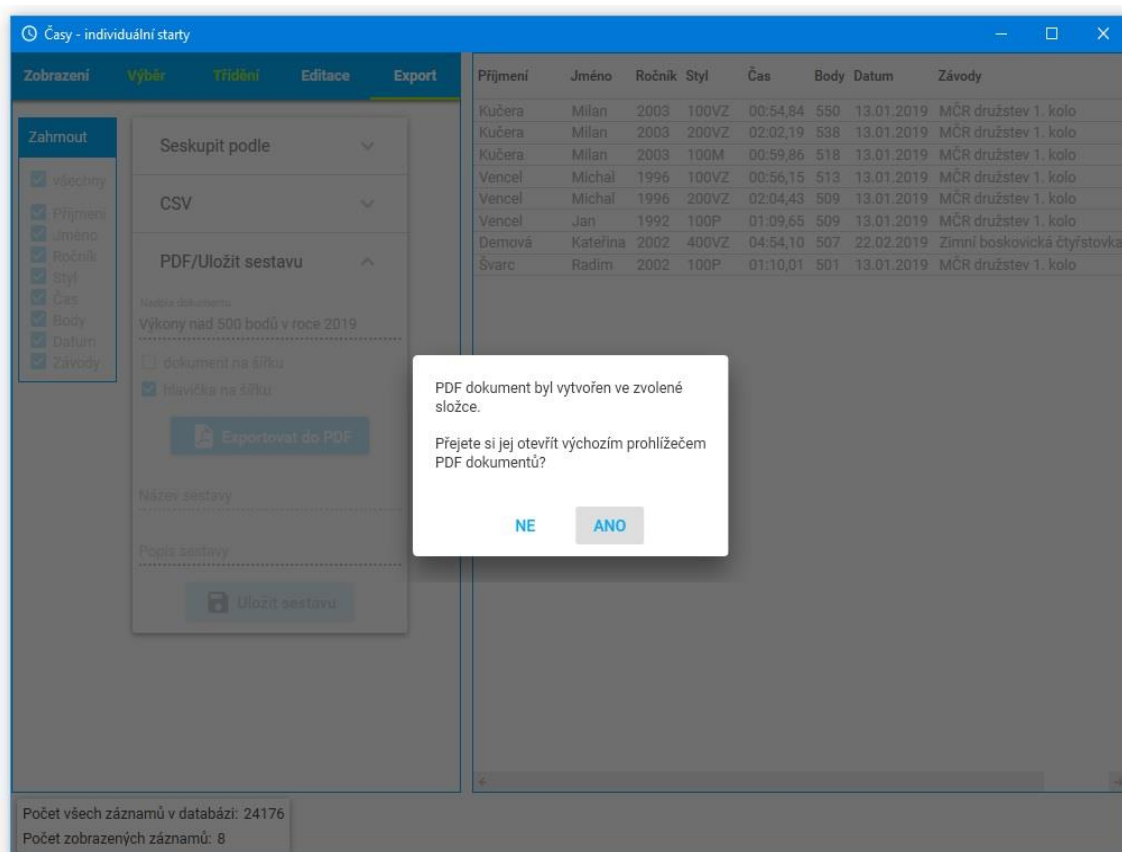
Konkrétně se vizuální podoba řídí zásadami návrhového jazyka Material design. Minimalistické ikony, tenké linie, animované ovládací prvky, lehké stínování tlačítek. To jsou některé charakteristické znaky grafického návrhu aplikace podle Material designu.

Formuláře aplikace mají pevně dané rozměry nebo je uživatel může měnit v závislosti na velikosti monitoru. Záleží to na druhu formuláře. U některých není potřeba měnit rozměry, protože tím uživatel nic nezíská a zároveň by se porušilo optimální rozložení ovládacích prvků. Toto se týká formulářů *Hlavní okno*, *Přehled*, *Sestavy*, *Přihlášky a Import*. U jiných se naopak vyplatí především možnost zvětšit okno na celou obrazovku monitoru. Jsou to všechny tabulkové formuláře, do kterých se tak vleze více dat a uživatel nemusí tolik používat posuvníky.

3.4.1 Dialogová okna

Dialogová okna se používají k interakci aplikace s uživatelem. Jakmile dojde během užívání aplikace k situaci, kdy je vhodné nebo nutné dát uživateli zprávu, zobrazí aplikace

dialogové okno s textem, popřípadě tlačítka. Zprávy mohou být různého typu. Informativní („*Databáze byla zálohována.*“) slouží k uvědomění uživatele o úspěšně vykonané akci. Potvrzující („*Čas bude vymazán z databáze.*“) vyžadují potvrzení před provedením důležité akce. Varovné („*Nejprve zvolte styl.*“) se snaží předejít možným problémům. Chybné („*Takový plavec již existuje.*“) reprezentují vzniklou chybu v aplikaci, která nedovoluje pokračovat v prováděné činnosti.



Obrázek 3.25 - Ukázka dialogového okna

Dialogové okno se objeví v rámci formuláře, který ho vyvolal. Nelze s ním posouvat a zavřeme ho pouze kliknutím na příslušné tlačítko. Formulář v pozadí je deaktivován a čeká na vyřízení dialogového okna.

3.4.2 Logo aplikace

K lepší identifikaci a rozšíření povědomí o aplikaci bylo vytvořeno logo. První polovinu zaujímá obrázek připomínající plavce, který má hlavu v barvách české vlajky. Zbytek je samotný název aplikace. Logo se drží principů minimalistického návrhu. Lehké stíny

dávají logu plastický nádech. Z převážné části je v šedé barvě což zapadá do barevné palety aplikace.



Obrázek 3.26 - Logo aplikace

3.5 Zavedení do provozu

Aplikace byla zavedena do ostrého provozu plaveckým oddílem ASK Blansko v dubnu roku 2019. Tomu předcházelo testování aplikace autorem aplikace a pověřenou osobou, které trvalo dva týdny. Následně byl vyhrazen týden pro opravu nalezených chyb. Po dokončení oprav nalezených závad se v procesu zavádění do provozu pokračovalo nasazením aplikace do zkušebního provozu. Toto období započalo v únoru a trvalo dva měsíce.

3.5.1 Testování

Testování prováděl autor aplikace souběžně s externím specialistou. Na testování byly vyhrazeny dva týdny. Testování autorem mělo výhodu v tom, že autor nejlépe ví, které funkce aplikace jsou stěžejní a nejpoužívanější, a proto na ně zaměří zvýšenou pozornost. Obdobně znalost vnitřního fungování aplikace přispěje k sestavení nejvhodnějších testovacích procedur, které mají největší pravděpodobnost nalezení chyby programu. Naopak ale v takovém případě může dojít k zanedbání otestování vedlejších funkcí aplikace, kde autor neočekává velkou chybovost. Naproti tomu externí odborník nemá subjektivní vztah k aplikaci a testování provede stejně na všech modulech aplikace.

Výstupem fáze testování je dokument od každé osoby provádějící testování. Obsahuje popis chyby, postup, jak chyba vznikla a způsob jakým ovlivnila běh programu. Následoval týden, kdy byly provedeny opravy chyb popsanych v dokumentech z testovací fáze.

3.5.2 Zkušební provoz

Počátkem února roku 2019 byla aplikace zavedena do zkušebního provozu. Ten si vyžadoval instalaci aplikace na počítač používaný v oddílu k uchování plavecké databáze. Trenérovi, který se o její správu stará byla přidělena úloha vést souběžně se starou Access databází i tu novou spravovanou aplikací SAM.

Důležitým krokem bylo zkopírovat aktuální podobu staré databáze do nové. Jedině tak získá obsluhující osoba plný rozsah funkcí aplikace a bude schopna absolvovat fázi zkušebního provozu takovou, jaká byla naplánována. Data ze staré databáze vedené aplikací MS Access byla vyexportována do XML souborů, které byly posléze importovány přes aplikační rozhraní *Import dat* do nové databáze. Data ve faktové tabulce časů ve staré databázi nejsou ve srovnání s modelem tabulky nové databáze tak podrobná. Chybí sloupce identifikující mezičas, první úsek štafety nebo kolo závodu, ve kterém byl čas zaplavan. Řešením bylo vytvořit část databáze znovu. Výsledky z posledních čtyř let byly v nové databázi smazány a vzápětí znovu postupně nahrány prostřednictvím formuláře *Import výsledků*.

Původně byly domluveny pravidelné schůzky trenéra obsluhujícího aplikaci s autorem aplikace. Z důvodu nedostatku času na obou stranách se z pravidelných schůzek staly občasné v závislosti na problémech s aplikací. Na schůzkách je projednávána míra uspokojení požadavků na funkcionalitu aplikace. V rámci fáze zkušebního provozu byly vysloveny drobné korekce těchto požadavků, které autor zohlednil v příští aktualizaci aplikace. Spolu s úpravami požadavků zákazník narazil na jisté prvky aplikace, které vyhodnotil jako nevyhovující, a naopak některé mu v řešení chyběly. Jakmile byla většina podobných nesrovnalostí vyřešena, nastal čas přejít do fáze ostrého provozu, která započala v dubnu roku 2019.

3.5.3 Ostrý provoz a údržba

Jednou z hlavních odlišností fází ostrého a zkušebního provozu je ukončení souběžného vedení staré a nové databáze. Vedení staré databáze se ukončuje a její obsah se pro všechny případy zálohuje. Zákazník je na přelomu těchto fází seznámen s veškerou funkcionalitou a nemá žádné výhrady k aktuální verzi aplikace.

Údržba aplikace bude prováděna na základě podnětů od zákazníka nebo autorovi iniciativy. Podněty od zákazníka se mohou týkat třeba náhle objevené chyby v aplikaci, kterou bude potřeba urychleně řešit. Autorova iniciativa může vzejít z výsledků monitorování aktuálních trendů a potřeb komunity závodního plavání v Česku a může vést k vylepšení, opravě nebo zavedení nových funkcí v aplikaci. Aktualizace aplikace iniciovaná některým z popsaných způsobů bude na straně zákazníka prováděna instalací aktualizacího balíku publikovaného autorem aplikace.

3.6 Zhodnocení projektu

Pro zhodnocení projektu po ekonomické stránce vyčíslíme jednotlivé náklady plaveckého oddílu, které byly vynaloženy k uskutečnění projektu. Dále rozebereme přínosy pro oddíl, které byly identifikovány po zavedení aplikace do ostrého provozu. Na konec kapitoly uvedeme možnosti a záměry týkající se budoucího rozvoje aplikace.

3.6.1 Ekonomické zhodnocení

Začneme tím, že si rozdělíme celý projekt do úkonů. Každý z úkonů představuje pro vývojáře rozdílnou časovou náročnost, která je dána složitostí úkonu. Náročnost je udávána v jednotkách MD³⁵. V našem případě budeme počítat s jednotkou MD, které odpovídají 4 pracovní hodiny. Což odpovídá pracovnímu zatížení jako při práci na poloviční úvazek. A to z toho důvodu, že autor projektu byl v období vývoje studentem se studijními povinnostmi. Hodinová sazba byla stanovena na 40 Kč. Opodstatněním takto nízkého hodinového ohodnocení jsou fakta jako neschopnost oddílu vyplatit vyšší částku, autorovo dlouholeté aktivní členství v oddílu, zpracovávání projektu v rámci školní práce, autorův osobní zájem v problematice a zájem se naučit novým technologiím.

V *Tabulce 3.1* vidíme soupis jednotlivých úkonů s různou náročností. Cena každého úkonu byla získána vynásobením jeho náročnosti s počtem hodin jednotky MD a hodinovou sazbou. Celkovým součtem vypočtených hodnot dostáváme celkovou náročnost projektu v podobě 83 jednotek MD a celkovou cenu za projekt ve výši 13 280 Kč.

³⁵ Člověkodenní (anglicky Man-day) je pracovní čas jedné osoby odpovídající jednomu pracovnímu dni, tedy typicky 8 hodin

Tabulka 3.1 - Výčet celkových nákladů

Úkon	Náročnost [MD]	Cena [Kč]
Analýza současného stavu	2	320
Návrh aplikace	9	1440
Implementace	46	7360
Příprava a import dat do DB	6	960
Testování	8	1280
Tvorba dokumentace a programu školení	10	1600
Školení	2	320
CELKEM	83	13280

Cenu za úkon *Analýza současného stavu* bylo možné výrazně snížit díky mojí předchozí aktivitě v oddíle. Výsledná částka 13 280 Kč je pro klub v rámci možností přijatelná. Stále je menší jak průměrná cena konkurenčních aplikací identifikovaných v kapitole 2.8. Analýza konkurenčních aplikací. Navíc je toto řešení dělané na míru nejen klubu, ale i prostředí a konvencím tvořených ČSPS. Případná další zaškolování a úpravy aplikace budou ohodnoceny podle výše uvedených pravidel.

3.6.2 Přínos projektu

Jelikož ostrý provoz aplikace započal o více jak měsíc dříve, než je termín odevzdání diplomové práce, jsme nyní schopni identifikovat a zhodnotit přínosy aplikace pro plavecký oddíl ASK Blansko. Většina přínosů byla rozpoznatelná už ve fázi zkušebního provozu. Ze schůzky s vedením klubu a trenéry konané měsíc po uvedení aplikace do ostrého provozu vyplynuly následující přínosy:

- **Vytváření přihlášek** – Proces vytváření přihlášek na závody je nyní jednodušší, rychlejší a pohodlnější na zpracování na straně pořadatele závodů. Za cenu vytvoření kategorií a programu závodů, uživatel nemusí kontrolovat, které a kolik disciplín může danému plavci přihlásit. Výstupem procesu tvorby přihlášky je textový dokument odpovídající stanovenému rozhraní, který je pořadateli obratem zpracován. Navíc tento modul poskytuje export přihlášky do formátu PDF, který je vystavován na webových stránkách oddílu.
- **Nahrávání výsledků** – Jedná se o nejocetňovanější modul celé aplikace. Výsledky ze závodů mohou být nyní nahrány do databáze během několika vteřin. Ve srovnání se starým přístupem (výrazná editace v MS Excel a následný import do

MS Access) je tento plně automatizovaný. Pokud se ve vstupním textovém souboru nenachází neočekávaná chyba, tak není vůbec potřeba ho editovat. Modul vybere pouze záznamy týkající se blanenského oddílu. Navíc kontroluje jejich správnost, kdy v případě nesrovnalostí vypíše varování ještě před samotným importem do databáze.

- **Orientace v datech, jednoduché výběry, třídění, editace** – Trenér obsluhující aplikaci zdůraznil lepší orientaci v plaveckých datech při provádění výběrů a zároveň více možností při zadávání výběrových a třídících kritérií. Vyzdvihována byla existence validací editovatelných polí nejen v rámci editace dat. Po úspěšném zaškolení v oblasti agregování dat byla oceněna možnost datových agregací, které jsou optimalizovány výhradně na plavecká data.
- **Možnosti exportu konkrétní podmnožiny dat** – Velký přínos přinesla aplikace SAM v oblasti exportu dat do PDF formátu. Komplikovaný proces tvorby PDF výstupů v MS Access s omezenými možnostmi byl nahrazen uživatelsky přístupným postupem. Ten zahrnuje možnosti vložit libovolný nadpis, vytvořit až dvouúrovňovou strukturu exportovaných dat nebo optimalizovat rozložení obsahu tím, že nastavíme dokument nebo hlavičky sloupců na šířku.
- **Existence pokročilých výchozích sestav, možnost vytvářet vlastní** – Velmi využívanými se staly pokročilé sestavy, které je obtížné vytvořit prostřednictvím MS Access. Díky těmto novým sestavám získali trenéři podklady k vytváření inovativních tréninkových plánů a plavci podrobnou zpětnou vazbu svého snažení na závodech. Možnost tvorby vlastních sestav přispívá k uspokojení případné kreativity trenérů a zároveň umožňuje lépe přizpůsobit data konkrétního plaveckého oddílu v tu chvíli, kdy předpřipravené výchozí sestavy selhávají (přetečení textu za okraje stránky).
- **Počítání bodů, možnost jejich přepočtu** – Na rozdíl od dřívějšího přístupu, kdy byly body přebírány z výsledkového textového souboru, jsou v aplikaci SAM body počítány. To umožňuje mít bodové hodnoty zaplavaných výkonů stále v aktuální podobě v závislosti na základních časech publikovaných organizací FINA. Jakmile je aktualizace vydána, je proveden přepočet bodů v celé databázi do aktuálního stavu. Výpočet bodů při vkládání časů také zpřístupňuje funkcionalitu, která zajišťuje hlídání reálnosti zadaného času.

- **Snížení nároků na trenéra** – Výše popsané automatizace a zjednodušení procesů spojených se správou oddílové databáze přináší snížení pracovních nároků na trenéra spravujícího oddílovou databázi, úsporu jeho času i ostatních trenérů a v neposlední řadě snížení pravděpodobnosti chyb.
- **Zefektivnění administrace v rámci závodní činnosti klubu** – Dalším přínosem zavedení aplikace do provozu je celkové zefektivnění administrativních činností klubu, které mohou brzdit rozvoj oddílu a zvyhodňovat konkurenci.
- **Podrobnější datová struktura** – S novým modelem databáze v aplikaci SAM přichází i podpora evidence tří nových druhů záznamů. Mezičasy a první úseky štafet jsou brány jako plnohodnotné časy vkládané do databázové tabulky individuálních časů. Časy štafet jsou též nově evidovány a mají i samostatnou databázovou tabulku. Mezi nové položky záznamu v tabulce individuálních časů patří informace o kolu (rozplavba, semifinále, finále), dráze a rozplavbě.
- **Snížení množství chybných dat v databázi** – Výsledkem používání aplikace SAM ke správě oddílové databáze bylo sníženo množství chyb vznikajících překlepy při ručním vkládání záznamů do tabulek nebo absencí kontrolního mechanismu při importu výsledků.

3.6.3 SWOT analýza po ukončení projektu

Podruhé v diplomové práci použijeme SWOT analýzu. Nyní ovšem ve vztahu nového řešení k původnímu. Silné stránky do určité míry korespondují s kapitolou 3.6.2. Přínos projektu, kterou bychom také mohli chápat jako výčet silných stránek aplikace. Z úsporných důvodů je v tabulce uvedeno pouze několik hlavních. Pokud bychom se striktně nedrželi konvencí SWOT analýzy pro pole příležitostí, které mají být z vnějšího prostředí oddílu, dali by se zde uvést i body z kapitoly 3.6.4. Budoucí rozvoj. Většina uvedených hrozeb se vyskytovala už v případě původního řešení, protože se jedná o hrozby, které se nedají aplikací ovlivnit.

Tabulka 3.2 - SWOT analýza po projektu

Silné stránky	Slabé stránky
- efektivní a přehledná práce s daty - nízká pravděpodobnost chyby - podpora záznamů typu štafeta, mezičas a 1. úsek štafety	- více práce pro trenéra při zkušeb. provozu - uživatel se musí učit novým postupům
Příležitosti	Hrozby
- expanze aplikace do dalších oddílů - získání náskoku před ostatními oddíly	- technické problémy - ztráta nezálohované databáze - nedostupnost autora pro školení, řešení potíží

3.6.4 Budoucí rozvoj

Z důvodu omezeného časového rámce odpovídajícímu diplomové práci, byly implementovány pouze funkce stanovené na začátku ve specifikaci požadavků od zákazníka. Další funkce a vylepšení vhodné do aplikace, které byly objeveny v průběhu vývoje, musely být odloženy. Jejich případné zavedení do aplikace je otázkou míry spokojnosti se stávající verzí aplikace, vnějšího prostředí aplikace tvořeného ČSPS a českou plaveckou komunitou, zájmu uživatelů o nové funkce a v neposlední řadě zájmu a možností autora. Projdeme si několik hlavních oblastí aplikace, které mají potenciál pro budoucí rozvoj.

- **Lenex** – Doplnit modul pro export přihlášek a import výsledků ve formátu Lenex.
- **Konektivita s ČSPS** – Navázáním na centrální evidenci by bylo možno zjednodušit řadu stávajících postupů při obsluze aplikace i úkony prováděné v informačním systému ČSPS.
- **Konektivita s iSAM** – Realizovat propojení aplikace SAM s internetovou aplikací iSAM, která prezentuje výsledky oddílu široké veřejnosti. Synchronizovat strukturu a obsah jejich databází, aby nemusel trenér nahrávat výsledky dvakrát. Jinými slovy vše, co by se změnilo na lokální databázi v aplikaci SAM by se synchronizovalo na server aplikace iSAM.

- **Evidence trenérů** – Vytvořit databázovou tabulku obsahující seznam trenérů, kteří prošli branami klubu. Tabulka by měla podobnou strukturu jako tabulka plavců.
- **Evidence akcí** – Zřídit databázovou tabulku obsahující seznam všech akcí kromě závodů, kterých se klub zúčastnil nebo je pořádal. Jsou myšleny soustředění, výlety, školení atd. Na tabulku akcí by byly navázány tabulky plavců a trenérů.
- **Evidence tréninkových skupin** – Zřídit databázovou tabulku obsahující seznam tréninkových skupin. Na tabulku by byly navázány tabulky trenérů a plavců. Vztahy by určovali trenéra (respektive trenéry), přiřazeného dané skupině plavce patřící do skupiny. Záznam v tabulce tréninkových skupin by obsahoval mimo jiné sloupce s datem počátku a konce existence skupiny.
- **Evidence tréninků** – Založit databázovou tabulku obsahující údaj o celkové vzdálenosti uplavané za trénink s volitelným vložením textového souboru nebo fotky s podrobným rozpisem tréninku. Jelikož přístup do aplikace SAM by měl mít pouze hlavní trenér, který nemá přímý přístup k tomuto typu údajů, museli by editaci tabulky provádět trenéři jednotlivých skupin prostřednictvím aplikace iSAM. Tabulka by byla navázána na tabulku tréninkových skupin.
- **Evidence docházky** – Vytvořit databázovou tabulku obsahující docházku plavců na jednotlivé tréninky v průběhu sezóny. Editaci by obdobně jako v případě evidence tréninků prováděli trenéři jednotlivých skupin v prostředí aplikace iSAM. Tabulka by byla navázána na tabulku plavců.
- **Výpočty odtrénovaných hodin trenérů i plavců** – Z vylepšeného databázového modelu popsaného výše by následně šel vyčíslit počet odtrénovaných hodin každého trenéra a plavce. Tyto výpočty by se potom dali využít při ohodnocování trenérů i plavců.
- **Grafy výkonnosti a fungování oddílu**– Doplnit možnost generování grafů, využívajících nová data z vylepšeného modelu databáze.

3.6.5 Licence

Jestliže bude realizován záměr rozšířit aplikaci i do jiných klubů, tak protože se jedná o komerční produkt, bude nutné do aplikace zabudovat systém licencování. Tím se zajistí, aby si uživatelé aplikaci samovolně mezi sebou nešířili. Jelikož v první fázi projektu je

zákazníkem pouze jeden konkrétní oddíl, je možné systém licencování obejít. Aplikace byla nainstalována osobně autorem na cílový počítač pomocí instalačního souboru, jehož kopie v současné době vlastní pouze autor.

ZÁVĚR

V závěru dostáváme prostor k ohlédnutí se zpět v čase na průběh činností vykonaných v rámci diplomové práce, jejich zhodnocení a okomentování výsledků praktické části. Pořadí, průběh i struktura úkonů se odvíjeli od obsahu práce. Činnosti probíhaly podle předem stanoveného pracovního plánu i časového harmonogramu. V průběhu vývoje úspěšně proběhlo několik schůzek se zákazníkem i s vedoucím diplomové práce k projednání aktuálních témat ohledně aplikace a potažmo diplomové práce.

Především je ale důležité závěrem konstatovat, že všechny cíle z úvodu práce se podařilo naplnit. Splnění cíle předcházela analýza současné situace zahrnující seznámení se s oddílem, jeho procesy a potřebami spolu s analýzou konkurenčních aplikací. Jakmile byly stanoveny požadavky na aplikaci bylo nutné věnovat čas výběru a studiu technologií. Poté už byla cesta volná samotnému řešení, v jehož rámci byly provedeny databázové, funkční i grafické návrhy. V hodnocení výsledného řešení byly vyčísleny náklady oddílu, identifikovány přínosy aplikace a představeny možnosti jejího budoucího rozvoje.

Aplikace se nyní nachází ve fázi ostrého provozu v plaveckém oddíle ASK Blansko. Na základě zpětné vazby od uživatele aplikace plní požadované úkony a její ovládání je přehledné a srozumitelné. To vyústilo ve snížení nároků na trenéry a zefektivnění administrativních činností v oddíle.

LITERATURA

- [1] *Co je to DLL?* [online]. 2018 [cit. 2019-03-17], Dostupné na URL: <<https://support.microsoft.com/cs-cz/help/815065/what-is-a-dll>>
- [2] MAIER, Z., *Architektura .NET Framework rozebrána dopodrobna* [online]. 2010 [cit. 2019-02-15], Dostupné na URL: <<https://www.swmag.cz/670/architektura-net-framework-rozebrana-do-podrobna/>>
- [3] VIRIUS, Miroslav. *C# 2010 Hotová řešení*. Vyd. 1. Brno: Computer Press, 2012. ISBN 978-80-251-3730-7
- [4] KAČMÁŘ, D., *Programujeme .NET aplikace ve Visual Studiu .NET*. Vyd. 1. Praha: 2001. ISBN 80-7226-569-5
- [5] *C# Language Specification* [online]. 2017 [cit. 2019-02-25], Dostupné na URL: <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-334.pdf>>
- [6] ČÁPKA, D., *Úvod do C# a .NET frameworku* [online]. 2012 [cit. 2019-02-27], Dostupné na URL: <<https://www.itnetwork.cz/csharp/zaklady/c-sharp-tutorial-uvod-do-jazyka-a-dot-net-framework>>
- [7] ŽŮREK, M., *Visual Studio – Úvod do vývojového prostředí* [online]. 2015 [cit. 2019-02-28], Dostupné na URL: <<https://www.itnetwork.cz/csharp/visual-studio/tutorial-visual-studio-uvod>>
- [8] ČÁPKA, D., *Úvod do databází v C#* [online]. 2013 [cit. 2019-03-02], Dostupné na URL: <<https://www.itnetwork.cz/csharp/databaze/c-sharp-tutorial-uvod-do-databazi>>
- [9] CONOLLY, T., BEGG, C., HOLOWCZAK, R., *Mistrovství databáze*. Vyd. 1. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7
- [10] HOLUBOVÁ, I., KOSEK, J., MINAŘÍK, K., NOVÁK, D., *Big Data a NoSQL databáze*. Vyd. 1. Praha: Grada Publishing, 2015. ISBN 978-80-247-5466-6
- [11] RATAN, V., *Choose the Right Database for Your Application* [online]. 2017 [cit. 2019-03-01], Dostupné na URL: <<https://opensourceforu.com/2017/05/choose-right-database-application/>>

- [12] ROUSE, M., HUGHES, A., STEDMAN, C., *Microsoft SQL Server* [online]. 2017 [cit. 2019-03-03], Dostupné na URL: <<https://searchsqlserver.techtarget.com/definition/SQL-Server>>
- [13] LACKO, L., *Mistrovství v Microsoft SQL Server 2012*. Vyd. 1. Brno: Computer Press, 2013. ISBN 978-80-251-3773-4
- [14] ROUSE, M., HUGHES, A., SIRKIN, J., *Microsoft SQL Server Management Studio* [online]. 2017 [cit. 2019-03-15], Dostupné na URL: <<https://searchsqlserver.techtarget.com/definition/Microsoft-SQL-Server-Management-Studio-SSMS>>
- [15] BERNARD, B., *Úvod do architektury MVC* [online]. 2009 [cit. 2019-03-08], Dostupné na URL: <<https://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>>
- [16] *MVC Architecture* [online]. 2019 [cit. 2019-03-09], Dostupné na URL: <<https://www.tutorialsteacher.com/mvc/mvc-architecture>>
- [17] *What is Entity Framework?* [online]. 2019 [cit. 2019-03-06], Dostupné na URL: <<https://www.entityframeworktutorial.net/what-is-entityframework.aspx>>
- [18] *Development Approaches with Entity Framework* [online]. 2019 [cit. 2019-03-06], Dostupné na URL: <<https://www.entityframeworktutorial.net/choosing-development-approach-with-entity-framework.aspx>>
- [19] VEGA, D., *Code First Data Annotations* [online]. 2016 [cit. 2019-03-07], Dostupné na URL: <<https://docs.microsoft.com/cs-cz/ef/ef6/modeling/code-first/data-annotations>>
- [20] *Entity Framework – Fluent API* [online]. 2019 [cit. 2019-03-07], Dostupné na URL: <https://www.tutorialspoint.com/entity_framework/entity_framework_fluent_api.htm>
- [21] LONEY, Kevin. *Oracle Database: kompletní průvodce*. Brno: Computer Press, 2010. Administrace (Computer Press). ISBN 978-80-251-2489-5.
- [22] KROENKE, David a David J. AUER. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

- [23] *What is LINQ?* [online]. 2019 [cit. 2019-03-09], Dostupné na URL: <<https://www.tutorialsteacher.com/linq/what-is-linq>>
- [24] ČÁPKA, D., *LINQ v C# – revoluce v dotazování* [online]. 2013 [cit. 2019-03-10], Dostupné na URL: <<https://www.itnetwork.cz/csharp/kolekce-a-linq/c-sharp-tutorial-linq-dotazy>>
- [25] SHAFRANOVICH, Y., *RFC4180 – Common Format and MIME Type for Comma-Separated Values (CSV) Files* [online]. 2005 [cit. 2019-03-11], Dostupné na URL: <<https://tools.ietf.org/html/rfc4180>>
- [26] HEROUT, P., *Java a XML*. Dotisk 1. vydání. České Budějovice: KOPP, 2012. ISBN 978-80-7232-307-4
- [27] COVER, R., *XML Schemas* [online]. 2016 [cit. 2019-03-10], Dostupné na URL: <<http://xml.coverpages.org/schemas.html>>
- [28] KOCH, M., NEUWIRTH, B., *Datové a funkční modelování*. Vyd. 4. rozšířené. Brno: Akademické nakladatelství Cerm, 2010. ISBN 978-80-214-4125-5
- [29] KOCH, M., *Informační systémy a technologie*. Vyd. 1. Brno: Akademické nakladatelství Cerm, 2005. ISBN 80-214-3003-6
- [30] ČÁPKA, D., *Use Case Diagram* [online]. 2013 [cit. 2019-03-13], Dostupné na URL: <<https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>>
- [31] ŽÁK, D., ZECHMEISTER, J., VÁŇA, T., *Fyzický návrh databáze* [online]. 2012 [cit. 2019-03-15], Dostupné na URL: <<https://docplayer.cz/1527446-12-blok-fyzicky-navrh-databaze.html>>
- [32] *Conceptual, Logical and Physical Data Model* [online]. 2019 [cit. 2019-03-15], Dostupné na URL: <https://www.visual-paradigm.com/support/documents/vpuserguide/3563/3564/85378_conceptual,1.html>
- [33] ČÁPKA, D., *Úvod do UML* [online]. 2013 [cit. 2019-03-12], Dostupné na URL: <<https://www.itnetwork.cz/navrh/uml/uml-uvod-historie-vyznam-a-diagramy>>
- [34] *Enterprise Architect* [online]. 2019 [cit. 2019-03-11], Dostupné na URL: <<https://sparxsystems.com/products/ea/index.html>>

- [35] *Case Study* [online]. 2009 [cit. 2019-03-11], Dostupné na URL: <<https://sparxsystems.com/downloads/pdf/GeoSciMLEACaseStudy.pdf>>
- [36] *Material Design – Introduction* [online]. 2019 [cit. 2019-03-12], Dostupné na URL: <<https://material.io/design/introduction/#>>
- [37] GRASSEOVÁ, Monika; DUBEC, Radek; ŘEHÁK, David. *Analýza podniku v rukou manažera*. 2. vyd. Brno: BizBooks, 2012. ISBN 978-80-265-0032-2.
- [38] *SWOT* [online]. 2019 [cit. 2019-02-13], Dostupné na URL: <<https://cs.wikipedia.org/wiki/SWOT>>
- [39] SKLENÁK, V. a kol. *Data, informace, znalosti a Internet*. Vyd. 1. Praha: C.H. Beck, 2001. xvii, 507 s. C.H. Beck pro praxi. ISBN 80-7179-409-0.
- [40] ŠKORNIČKOVÁ, E., *Co je to GDPR?* [online]. 2019 [cit. 2019-03-23], Dostupné na URL: <<https://www.gdpr.cz/gdpr/>>
- [41] *Stanovy* [online]. 2016 [cit. 2019-01-20], Dostupné na URL: <<http://www.czechswimming.cz/index.php/dokumenty/dokumenty-a-rady-csps?download=86:stanovy-csps-2016>>
- [42] *Registrační řád* [online]. 2017 [cit. 2019-01-22], Dostupné na URL: <<http://www.czechswimming.cz/index.php/dokumenty/dokumenty-a-rady-csps?download=14:smernice-4-2017-a-registracni-rad>>
- [43] *Using Access or Excel to manage your data* [online]. 2019 [cit. 2019-02-17], Dostupné na URL: <<https://support.office.com/en-us/article/using-access-or-excel-to-manage-your-data-09576147-47d1-4c6f-9312-e825227fcaea?ui=en-US&rs=en-US&ad=US>>
- [44] *Team Management Software* [online]. 2019 [cit. 2019-03-20], Dostupné na URL: <<https://hytek.active.com/swimming-management-software.html>>
- [45] *Swimclub Manager* [online]. 2019 [cit. 2019-03-26], Dostupné na URL: <<https://www.swimclubmanager.co.uk/>>
- [46] *Swim Office* [online]. 2019 [cit. 2019-03-27], Dostupné na URL: <<https://www.teamunify.com/swim-team-management-software/>>

[47] RAJCOK, V., FRONEK, T., *iSwimManager* [online]. 2017 [cit. 2019-03-28],
Dostupné na URL: <<https://www.iswimmanager.com/>>

SEZNAM OBRÁZKŮ

Obrázek 1.1 - Základní struktura prostředí .NET (Zdroj: [2]).....	16
Obrázek 1.2 - Návaznost jednotlivých komponent (Zdroj: [15])	22
Obrázek 1.3 - Pozice Entity Frameworku v aplikaci (Zdroj: [17]).....	23
Obrázek 1.4 - Ukázka vazby mezi dvěma entitami	30
Obrázek 1.5 - UML diagramy (Zdroj: [33])	32
Obrázek 1.6 - Grafické podoba SWOT analýzy (Zdroj: [38]).....	34
Obrázek 2.1 - Vzorová podoba přihlášky	47
Obrázek 2.2 - Vzorová podoba výsledků individuálních startů.....	48
Obrázek 2.3 - Vzorová podoba výsledků štafetového startu	49
Obrázek 2.4 - ER diagram databáze v programu Access	50
Obrázek 3.1 – Diagram případů užití.....	57
Obrázek 3.2 – ER diagram konceptuálního návrhu	58
Obrázek 3.3 - ER diagram fyzického návrhu	59
Obrázek 3.4 – Formulář Hlavní okno	60
Obrázek 3.5 - Formulář Přehled	62
Obrázek 3.6 - Formulář Základní časy	63
Obrázek 3.7 - Formulář Individuální časy	64
Obrázek 3.8 - Zadávání výběrových kritérií	65
Obrázek 3.9 – Možnosti editace tabulky	66
Obrázek 3.10 - Možnosti exportu dat	67
Obrázek 3.11 - Formulář Sestavy	68
Obrázek 3.12 - Obsah záložky Zdroj	69
Obrázek 3.13 - Obsah záložky Plavci	70
Obrázek 3.14 - Obsah záložky Kategorie	70
Obrázek 3.15 - Obsah záložky Program	71
Obrázek 3.16 - Obsah záložky Disciplíny	72
Obrázek 3.17 - Obsah záložky Časy	73
Obrázek 3.18 - Obsah záložky Dokončení	73
Obrázek 3.19 - Ukázka obsahu exportovaného textového souboru.....	74
Obrázek 3.20 - Ukázka obsahu exportovaného PDF dokumentu	74
Obrázek 3.21 - Formulář Import výsledků	75

Obrázek 3.22 - Formulář Import dat	77
Obrázek 3.23 - Příklad exportovaného PDF dokumentu	78
Obrázek 3.24 - Formulář Konfigurace	80
Obrázek 3.25 - Ukázka dialogového okna	82
Obrázek 3.26 - Logo aplikace	83

SEZNAM TABULEK

Tabulka 2.1 - SWOT analýza před projektem	52
Tabulka 3.1 - Výčet celkových nákladů	86
Tabulka 3.2 - SWOT analýza po projektu	89