

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

**Vývoj aplikace zaměřené na evidenci sportovní aktivity
triatlonistů**

Matěj Tvrz

© 2021 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Matěj Tvrz

Systémové inženýrství a informatika
Informatika

Název práce

Vývoj aplikace zaměřené na evidenci sportovní aktivity triatlonistů

Název anglicky

Development of a sports diary application for triathletes

Cíle práce

Cílem této bakalářské práce je vývoj desktopové aplikace, která se bude zaměřovat na sportovní činnost triatlonistů. Do aplikace bude možno zadat tréninková data triatlonistů (aktivity – plavání, cyklistika a běh) aplikace vyhodnotí tréninková data a transformuje je do přehledných grafů a statistik. Každý uživatel aplikace bude mít přístup na svůj profil, kde se mu zobrazí přehledně zobrazené počty odtrénovaných jednotek za jednotlivé sporty, statistiky a grafy. Pokud si uživatel nastaví nějaké cíle nebo roční úkoly, aplikace je bude zaznamenávat a vyhodnocovat pokrok.

Metodika

Metodika práce je založená na analyticko-syntetickém přístupu. Na základě analýzy odborných informačních zdrojů souvisejících s tématem práce a syntézy takto získaných poznatků budou popsány obecné postupy vývoje desktopových aplikací pro OS Windows. Teoretické poznatky budou následně aplikovány při vývoji prototypu aplikace. Při návrhu a vývoji hry bude využito standardních metod a postupů softwarového inženýrství.

První fáze vývoje bude zaměřena především na funkčnost celého procesu aplikace, především pak zaznamenávání nových aktivit do databáze a následné zobrazování z databáze zpět do aplikace.

Fáze dvě se zaměří na celý systém přihlašování, odhlašování a registrace, propojení aktivit s uživateli.

Finální fáze bude zaměřena na vývoj přehledného uživatelského rozhraní, zdi a prezentaci aktivit formou statistik a grafů.

Poznatky z vývoje a testování budou na závěr popsány a zhodnoceny.

Doporučený rozsah práce

40 – 50 stran

Klíčová slova

Aplikace, C#, Sport, Triatlon

Doporučené zdroje informací

ALBAHARI, Joseph and Ben ALBAHARI, 2017. C# 7.0 in a Nutshell: The Definitive Reference. O'Reilly Media. ISBN 9781491987650.

FRIEL, Joe, 2016. Triathlete's Training Bible: The World's Most Comprehensive Training Guide. 4. Colorado: VeloPress. ISBN 9781937715441.

GRIFFITHS, Ian, 2020. Programming C# 8.0: Build Cloud, Web, and Desktop Applications. Sebastopol, CA: O'Reilly Media. ISBN 9781937715441.



Předběžný termín obhajoby

2020/21 LS – PEF

Vedoucí práce

Ing. Petr Hanzlík, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 08. 03. 2021

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci " Vývoj aplikace zaměřené na evidenci sportovní aktivity triatlonistů" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.3.2021

Poděkování

Rád(a) bych touto cestou poděkoval(a) Ing. Petrovi Hanzlíkovi, Ph.D. za maximální ochotu a pomoc při vypracování této bakalářské práce. Dále za odborné poznatky Ing. Lukášovi Tvrzovi.

Vývoj aplikace zaměřené na evidenci sportovní aktivity triatlonistů

Abstrakt

Tato bakalářská práce cílí na vývoj desktopové aplikace, která bude zaměřena na zaznamenání sportovních aktivit sportovců se specializací na triatlon. Teoretická část práce se zabývá .NET Frameworkem obecně, dále se pak rozebere metodika vývoje software. Poslední část teoretické části je věnována SQL

Praktická část se zabývá vývojem desktopové aplikace. Aplikace je napsána v jazyce C# za použití standardních metod a postupů systémového inženýrství. Dále jsou zde popsány poznatky z vývoje, testování a samotné zhodnocení.

Klíčová slova: Aplikace, C#, Sport, Triatlon, SQL, Microsoft Azure, MSSQL

Development of a sports diary application for triathletes

Abstract

This bachelor's thesis is thematically focused on the development of a desktop application, which will focus on recording sports activities of athletes with a focus on triathlon. The theoretical part of the thesis deals with the .NET Framework in general, then the methodology of software development. The last part of the theoretical part is devoted to SQL. The practical part deals with the development of a desktop application. The application is written in C # using standard methods and systems engineering procedures. Parts from development, testing and evaluation itself are also described here.

Keywords: Application, C#, Sport, Triathlon, SQL, Microsoft Azure, MSSQL

Obsah

1. Úvod	10
2. Cíl práce a metodika	11
2.1. Cíle práce	11
2.2. Metodika	11
3. Teoretická východiska	12
3.1. .NET Framework.....	12
3.1.1. Windows Forms	14
3.1.2. Windows Presentation Foundation	15
3.2. Relační datový model.....	17
3.2.1. Jazyk SQL.....	17
3.2.2. DBMS	18
3.2.3. Příkazy jazyka SQL	19
3.2.3.1. DML – Příkazy pro práci s daty	19
3.2.3.2. DDL – příkazy pro definici dat	19
3.2.3.3. DCL – příkazy pro řízení dat.....	19
3.3. Metodika vývoje software.....	21
3.3.1. Prototypování.....	21
3.3.2. Spirálový model.....	21
3.3.3. Vodopádový model.....	22
4. Vlastní práce	23
4.1. Analýza	23
4.2. Tvorba databáze	25
4.2.1. Datový slovník.....	25
4.2.2. Azure SQL Server.....	27
4.3. Přihlašovací obrazovka	28
4.3.1. Registrace.....	28
4.3.1.1. Zkrácená registrace.....	29
4.3.1.2. Kompletní registrace	30
4.3.2. Přihlášení	31
4.4. Hlavní obrazovka	33
4.4.1. Sportovec	33
4.4.1.1. Dashboard.....	33
4.4.1.2. Statistika	34
4.4.1.3. Zed'	36
4.4.1.4. Nová aktivita	38

4.4.1.5. Nastavení	39
4.4.2. Administrátor	41
4.4.2.1. Uživatelé.....	41
4.4.2.2. Aktivity.....	43
4.5. Testování aplikace.....	44
5. Závěr.....	47
6. Seznam použitých zdrojů	48

Seznam obrázků

Obrázek 1 - .NET (25)	12
Obrázek 2 - .NET Framework (26).....	13
Obrázek 3 - Windows Forms (27)	15
Obrázek 4 - Windows Presentation Foundation (28).....	16
Obrázek 5 - Relační datový model (32).....	17
Obrázek 6 - Odlišnosti v syntaxi PostgreSQL a TSQL (33).....	18
Obrázek 7 - Prototypování (29)	21
Obrázek 8 - Model spirála (30).....	22
Obrázek 9 - Vodopádový model (31)	22
Obrázek 10 - ER diagram [archiv autora].....	24
Obrázek 11 - Wireframe - dashboard [archiv autora].....	24
Obrázek 12 - Dashboard azure [archiv autora].....	27
Obrázek 13 - Úvodní obrazovka [archiv autora]	28
Obrázek 14 - Zkrácená registrace [archiv autora]	29
Obrázek 15 - Kompletní registrace [archiv autora]	30
Obrázek 16 - Přihlášení [archiv autora].....	31
Obrázek 17 - Přihlášení 2 [archiv autora].....	32
Obrázek 18 - Dashboard [archiv autora].....	33
Obrázek 19 - Statistiky [archiv autora].....	34
Obrázek 20 - Zeď [archiv autora]	36
Obrázek 21 - Aktivita [archiv autora].....	37
Obrázek 22 - Nová aktivita [archiv autora]	38
Obrázek 23 - Nastavení 1 [archiv autora].....	39
Obrázek 24 - Nastavení 2 [archiv autora].....	40
Obrázek 25 - Uživatelé [archiv autora].....	41
Obrázek 26 - Uživatel [archiv autora]	42
Obrázek 27 - Aktivity [archiv autora].....	43

1. Úvod

Nacházíme se v době, kdy ve volném čase mezi lidmi stoupá na popularitě sportovní aktivita. Lidé taktéž tyto aktivity chtějí zaznamenávat a sdílet mezi přáteli. Nespočet existujících aplikací je webových, nebo čistě mobilních. Musíme však věnovat pozornost i uživatelům, kteří dávají přednost čistě desktopovým aplikacím.

První část bakalářské práce se zabývá problematikou .NET Frameworku. Rozeberou se zde skupiny instrukcí mezijazyka CIL. Dále pak práce představí jako součásti platformy .NET Framework Windows Forms Application a Windows Presentation Foundation. Další část této práce shrnuje metodiku vývoje software. Rozeberou se zde základní rozdíly mezi metodickými frameworky softwarového vývoje. Popíšou se zde základní principy prototypování, spirály, která byla následně užita v praktické části, a vodopádového modelu. Následně práce rozebere a vysvětlí relační datový model společně s představením jazyka SQL a následným popisem základních příkazů.

Druhá kapitola této práce, na základě poznatků z teoretické části, je věnována vývoji desktopové aplikace v jazyce C#. Jedná se o aplikaci umožňující zaznamenat data sportovní aktivity s následným uložením do databáze. Práce nám představí jednotlivé obrazovky této aplikace společně s nejvíce problematickými metodami a jejich následným popisem. Rovněž se zde odkryjí nejkompexnější dotazy pro jazyk SQL, které aplikace vyžaduje pro správný chod. Jsou zde vypíchnuty poznatky z vývoje a následné testování aplikace.

Cílem této bakalářské práce je vyvinout desktopovou aplikaci za použití .NET Framework, která umožní zadávat sportovní data, odeslat je do databáze a následně s nimi dále pracovat.

2. Cíl práce a metodika

2.1. Cíle práce

Cílem této bakalářské práce je vývoj desktopové aplikace, která se bude zaměřovat na sportovní činnost triatlonistů. Do aplikace bude možno zadat tréninková data triatlonistů (aktivity – plavání, cyklistika a běh) aplikace vyhodnotí tréninková data a transformuje je do přehledných grafů a statistik. Každý uživatel aplikace bude mít přístup na svůj profil, kde se mu zobrazí přehledně zobrazené počty odtrénovaných jednotek za jednotlivé sporty, statistiky a grafy. Pokud si uživatel nastaví nějaké cíle nebo roční úkoly, aplikace je bude zaznamenávat a vyhodnocovat pokrok.

2.2. Metodika

Metodika práce je založená na analyticko-syntetickém přístupu. Na základě analýzy odborných informačních zdrojů souvisejících s tématem práce a syntézy takto získaných poznatků budou popsány obecné postupy vývoje desktopových aplikací pro OS Windows. Teoretické poznatky budou následně aplikovány při vývoji prototypu aplikace. Při návrhu a vývoji hry bude využito standardních metod a postupů softwarového inženýrství.

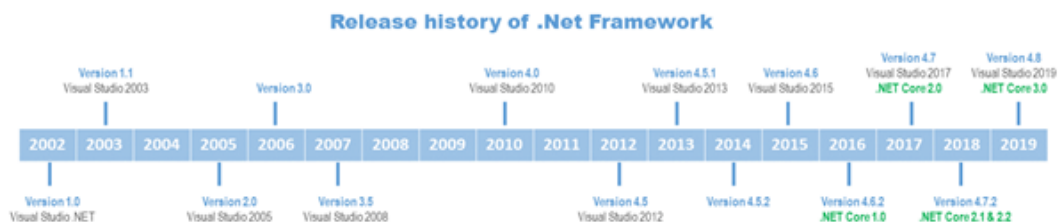
V praktické části této bakalářské práce bude využito databázové řešení firmy Microsoft, jedná se o produkt SQL Server v prostředí Microsoft Azure. Pro vývoj samotné aplikace bude využit objektově orientovaný programovací jazyk .NET Framework C#. Pro vývoj bylo zvoleno vývojové prostředí Visual Studio 2019, ladění SQL skriptů probíhalo v Microsoft SQL Management Studio. Pro vytvoření E-R diagramu databáze byla zvolena aplikace Visual Paradigm Enterprise.

3. Teoretická východiska

3.1. .NET Framework

.Net Framework je balík aplikačních programovacích rozhraní neboli API a sdílených knihoven které lze při vývoji aplikace využít. Framework je vyvíjen firmou Microsoft, původně v 90 letech 20. století byl vyvíjen pod názvem NGWS - Next generation Windows Services. První beta verze .NET 1.0 byla vydána začátkem roku 2001, o rok později byla vydána první verze .NET Frameworku určeného pro operační systémy Windows NT 4, 98, 2000, ME, XP.

Od vydání první verze frameworku Microsoft uvedl na trh již devět nových verzí, většinou společně s vývojovým prostředím Microsoft Visual Studio.(20,23)



Obrázek 1 - .NET (25)

Při vývoji za použití frameworku .NET nezáleží, jaký programovací jazyk si programátor zvolí, původní kód je vždy přeložen do mezijazyka (bytekód) CIL - Common Intermediate Language. CIL je soubor instrukcí, který je procesorově a platformě nezávislý, může tedy být realizován v jakémkoliv prostředí umožňujících práci a podporující Common Language Infrastructure. V prvním kroku je zdrojový kód převeden do CIL, z mezijazyka je vytvořen bytecode společně s .NET assembly. Po provedení .NET assembly pomocí JIT kompilátoru je generován nativní kód, ten je v poslední části zpracován pomocí procesoru. (1,2,20,24)

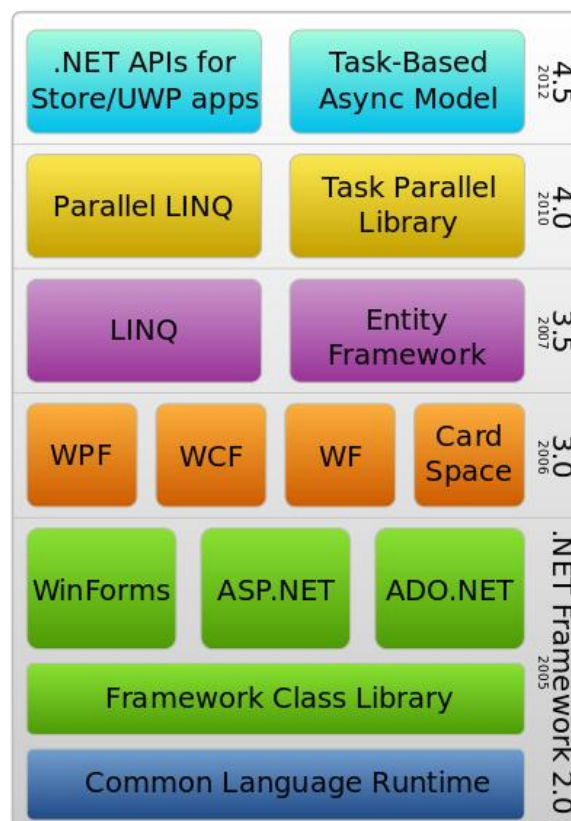
Bytecode mezijazyka CIL obsahuje následující skupiny instrukcí:(23,24)

- načítání a ukládání
- Aritmetické
- Konverze
- Vytváření objektů
- Volání metod a vyvolání vyjímek
- Synchronizace atd.

Nejpoužívanějšími programovacími jazyky, které se využívají pro vývoj aplikací za pomoci .NET frameworku jsou C#, Visual Basic .NET, F#, J#, C++/CLI.

Hlavní součásti .NET frameworku jsou: (1,20,21)

- **ASP.NET** - technologie využívaná k vývoji webových aplikací.
- **Windows Communication Foundation** – komunikační infrastruktura webových aplikací.
- **Windows Workflow Foundation** – využití pro definování sekvenčních procesů.
- **Windows Presentation Foundation** – technologie k vytváření grafických uživatelských rozhraní GUI.
- **Windows CardSpace** – implementování Information Cards
- **LINQ** – využití k objektovému přístupu k datům v databázi, XML, objektech



Obrázek 2 - .NET Framework (26)

Platformy v rámci .NET Frameworku (21)

- **.NET Core** – open source standart spustitelná kdekoliv (Windows, Linux, macOS)
- **.NET Framework** – určeno pro webové stránky, služby a aplikace spustitelné pod operačním systémem Windows
- **Xamarin/Mono** - .NET varianta pro mobilní zařízení (iOS, Android, AppleTV atd.)

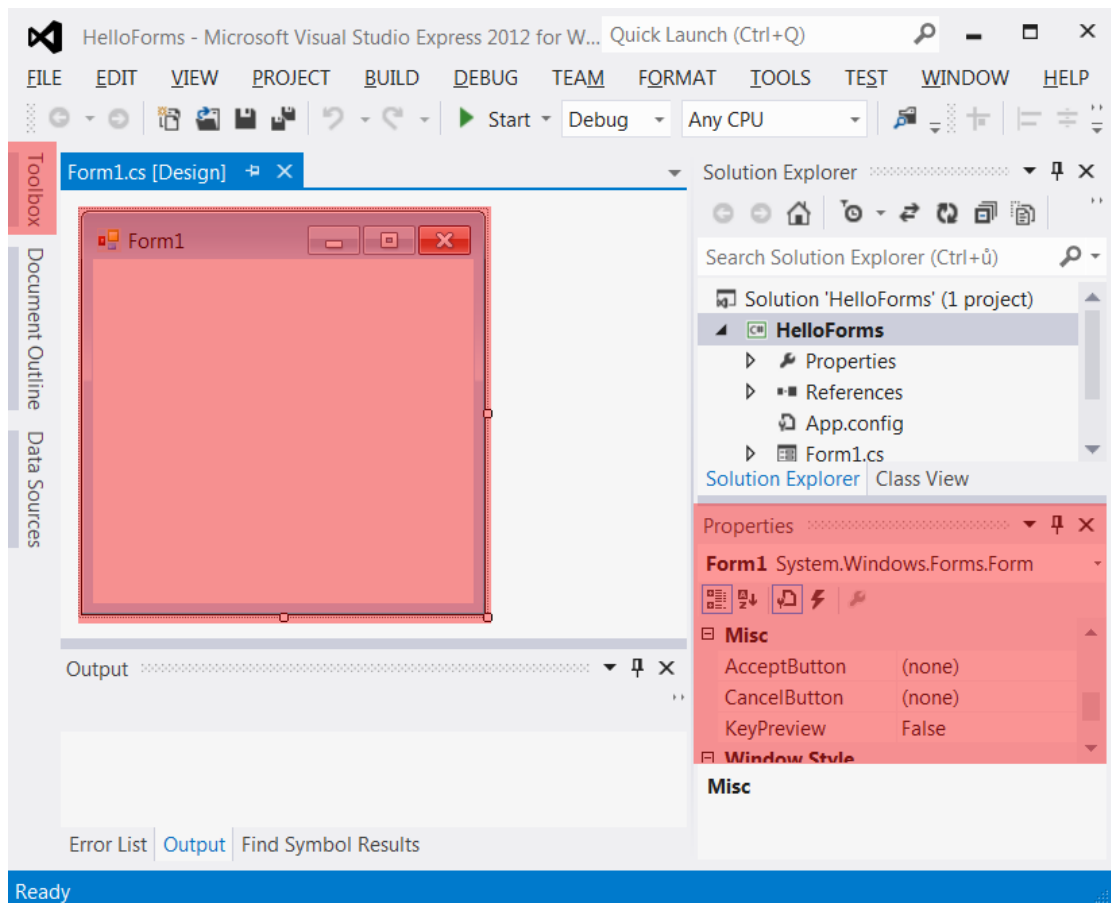
Využitelnost .NET frameworku je velice rozmanitá, díky přenositelnosti mezi jednotlivými druhy zařízení se může jednat o aplikace: (1,4,21,24)

- Webové
- Desktopové
- Hry
- Cloudová řešení
- Mobilní aplikace
- Micro služby pracující s Docker kontejnerem
- IoT

3.1.1. Windows Forms

Windows Forms jsou součástí platformy .NET Framework umožňující pomocí grafického designéru vytvoření jednoduchých aplikací využívajících formulářových prvků. Nachází se zde plná sada ovládacích komponent využitelných pro většinu situací. (20,24)

Windows Forms .NET slouží pouze jako platforma pro uživatelská rozhraní při řešení vícevrstvé architektury. Když uživatel v rámci formulářového pole případně jiného ovládacího prvku provede nějakou činnost, tato akce vyvolá událost. Aplikaci vždy na tyto události reaguje a s kódem dané události ji zpracovává. V rámci modelu Windows Forms lze využít různých ovládacích prvků jako jsou například: textová pole, tlačítka, rozevírací seznamy, přepínače atd. Pomocí třídy UserControl lze vytvořit i vlastní ovládací prvky. Pro snadné vytváření Windows Forms aplikací lze využít Návrhář formulářů, který je součástí vývojového prostředí Visual Studio. (1,24)



Obrázek 3 - Windows Forms (27)

3.1.2. Windows Presentation Foundation

Nástupcem platformy Windows Forms .NET je Windows Presentation Foundation, která oproti WF poskytuje nový vzhled, nové principy přizpůsobení ovládacích prvků a v neposlední řadě nové grafické funkce. (22,24)

Technologie Windows Presentation Foundation (WPF) je rozhraní pro návrh a zobrazování uživatelského prostředí. Důvodem pro vznik WPF byly problémy platformy WF, která začala narážet na různá omezení, ty vycházely z podstaty tohoto řešení. WF například primárně pracuje s elementy uživatelského rozhraní operačního systému Windows, která není snadné upravovat po vzhledové stránce ani z funkčního hlediska.

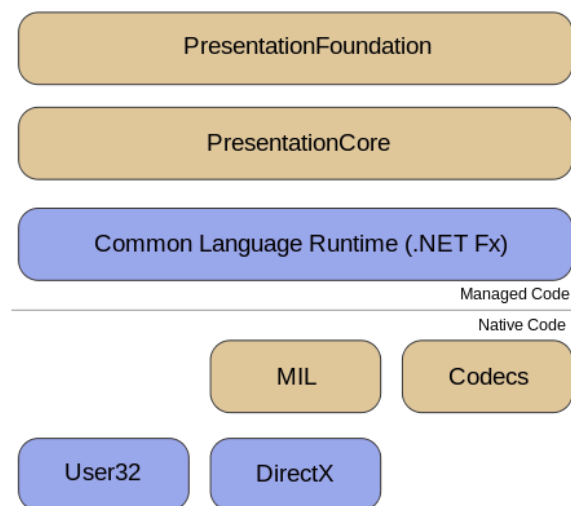
Hlavní vlastnosti WPF: (22,23)

- **Komplexní objektový model** – oproti WF zde existuje mnoho nových událostí, parametrů, vlastností, které lze přizpůsobit.
- **Binding** – slouží pro vázání vlastností datových objektů na uživatelské rozhraní
- **Styly** – možnost měnit vzhled i funkčnost již existujících prvků
- **Dispatchers** – zajištění synchronizace úkolů do hlavního aplikačního vlákna

- **XAML** – návrh uživatelského prostředí pomocí jazyka XAML, který je založený na jazyce XML
- **Layout** – správné rozložení ovládacích prvků při změně velikosti formuláře či jednotlivých elementů
- **Vektorová transformace** – jelikož WPF je celé vektorové, elementy je tedy možno libovolně zmenšovat či zvětšovat. Finální aplikace je nezávislá na rozlišení zobrazovací stanice.

Architektura Windows presentation foundation se skládá ze dvou typů komponent: (22)

- Windows komponenty (CLR, User32, DirectX)
- WPF komponenty (PresentationFoundation, PresentationCore, MIL, Codecs)



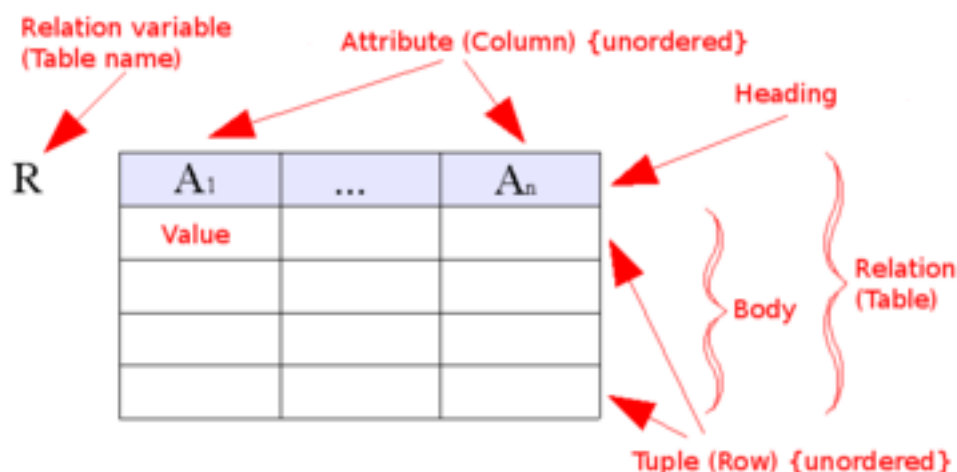
Obrázek 4 - Windows Presentation Foundation (28)

3.2. Relační datový model

Relační datový model je nejrozšířenější způsob, jakým se ukládají data v databázi. S představou o využití matematického aparátu relačních množin a predikátové logiky přišel informatik E.F. Codd v roce 1969. Databázová relace oproti matematické využívá pomocného aparátu, který se nazývá schéma relace. Schéma nám udává, jaký je název relace, kolik se v ní nachází sloupců včetně jejich jmen a přípustných hodnot pro daný sloupec. (10,18)

Data jsou sdružena do relací (tabulek), které obsahují jednotlivé řádky. Jednotlivé tabulky jsou tvořeny strukturou záznamů s pevně danými položkami (sloupci) neboli atributy. Tabulky jsou naplněny vlastním obsahem – daty. V rámci relačního modelu je kladen důraz na minimalizaci redundance dat, zachování integrity dat, využívání primárních klíčů, tvoření vazeb mezi jednotlivými tabulkami za pomoci cizích klíčů.

Pro práci s relačním datovým modelem se využívá jazyk SQL (Structured Query Language), kterým je strukturovaný dotazovací jazyk. (9,11,13)



Obrázek 5 - Relační datový model (32)

3.2.1. Jazyk SQL

Strukturovaný dotazovací jazyk je standardizovaný jazyk, který se využívá primárně pro práci s daty uloženými v relačních databázích. Jazyk SQL byl vyvinut firmou IBM v 70. letech 20. století v rámci výzkumu relačních databází. Původně pod názvem SEQUEL (Structured English Query Language), po rozšíření firem, které se na vývoji podíleli se jazyk přejmenoval na SQL. (12,13)

Na potřeby nejmodernějších relačních databází reaguje zatím nejnovější standart, který je SQL3 (SQL-99). Prakticky každá relační databáze podporuje standardy SQL3, obvykle ale nejsou

veškeré normy, které SQL3 obsahuje implementovány. Omezuje to tak bezproblémovou přenositelnost složitějších SQL dotazů mezi jednotlivými databázemi. (23,24)

- PL/SQL – procedurální jazyk používaný v relačních databázích Oracle
- PL/pgSQL – procedurální jazyk používaný v relačních databázích PostgreSQL
- T-SQL – procedurální jazyk využívaný Microsoft SQL Server (23)

Každý z těchto rozšíření standardizovaného jazyka SQL obsahuje rozšiřující funkcionalitu použitelnou pouze na konkrétní relační databázi. (11,13,17)

Jedná se především o:

- Procedury
- Funkce
- Syntaktické odlišnosti
- Způsob implementace transakcí
- Podporu příkazů
- Datové typy atd. (17)

Syntax	PostgreSQL	SQL Server
SELECT	col1, col2	Select [col1], [col2]
Aliases for columns and tables	SELECT AVG(col1) AS avg1	SELECT AVG(col1)=avg1
Working with dates	CURRENT_DATE() CURRENT_TIME() EXTRACT()	GETDATE() DATEPART()

Obrázek 6 - Odlišnosti v syntaxi PostgreSQL a TSQL (33)

Na obrázku výše je znázorněna odlišnost v syntaxi základních příkazů mezi PostgreSQL a SQL Server. (17)

3.2.2. DBMS

System řízení báze dat je softwarový aparát využívaný a zajišťující práci s relační databází. Tvoří rozhraní mezi fyzicky uloženými daty a aplikačním software, který s daty pracuje. Hlavní přednosti SRBD jsou: (19)

- definice datových modelů (fyzické, logické, objektové)
- správa klíčů (primární, cizí, fulltext)
- autentizace a autorizace uživatelů
- správa transakcí
- práce se spouštěmi (trigger), procedurami
- zajištění integrity dat (datová integrita)
- využití pokročilých funkcí (CTE)
- atomicita příkazů

3.2.3. Příkazy jazyka SQL

Jednotlivé příkazy, které umožňují plnou kontrolu systému řízení báze dat se dělí do několika hlavních skupin. (11,12,13)

3.2.3.1. DML – Příkazy pro práci s daty

Data Manipulation Language – obsahují veškeré příkazy pro práci s daty, získávání dat z databáze a jejich následnou úpravu. (11,12)

- SELECT
- UPDATE
- INSERT
- MERGE
- DELETE

3.2.3.2. DDL – příkazy pro definici dat

Data Definition Language – příkazy využívané k vytváření struktury databáze, strukturu lze následně upravovat, doplňovat či mazat. (11,12)

- CREATE
- ALTER
- DROP

3.2.3.3. DCL – příkazy pro řízení dat

Data Control Language – příkazy pro nastavování práv a využívané ke správě transakcí spouštěných pod SQL databází. (11,12)

- GRANT
- REVOKE
- COMMIT
- ROLLBACK
- START TRANSACTION/BEGIN TRANSACTION

Pro práci s daty a vytváření složitějších příkazů se využívají kromě výše zmíněných DML příkazů následující klauzule: (11,12,13)

- WHERE – filtrování dat
- GROUP BY – seskupení dat podle určité podmínky
- HAVING – podmínka při využití agregačních funkcí (např. COUNT, SUM atd.)
- ORDER BY – řazení dat (vzestupně – ASC/sestupně DESC)

Níže je ukázka komplexního SQL dotazu využívajícího filtrování dat, seskupení, podmínek agregačních funkcí atd. (11,12,13)

```

SELECT [sloupec_1],[sloupec_2],SUM([sloupec_3])
FROM [název_tabulky]
WHERE [sloupec_1] BETWEEN [hodnotaA] and [hodnotaB]
GROUP BY [sloupec_1],[sloupec_2]
HAVING SUM([sloupec_3]) > [hodnota]
ORDER BY [sloupec_1] ASC

```

Logické zpracování

SQL dotazu databázovým strojem probíhá v následujícím pořadí: (11,12,13)

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT
6. ORDER BY

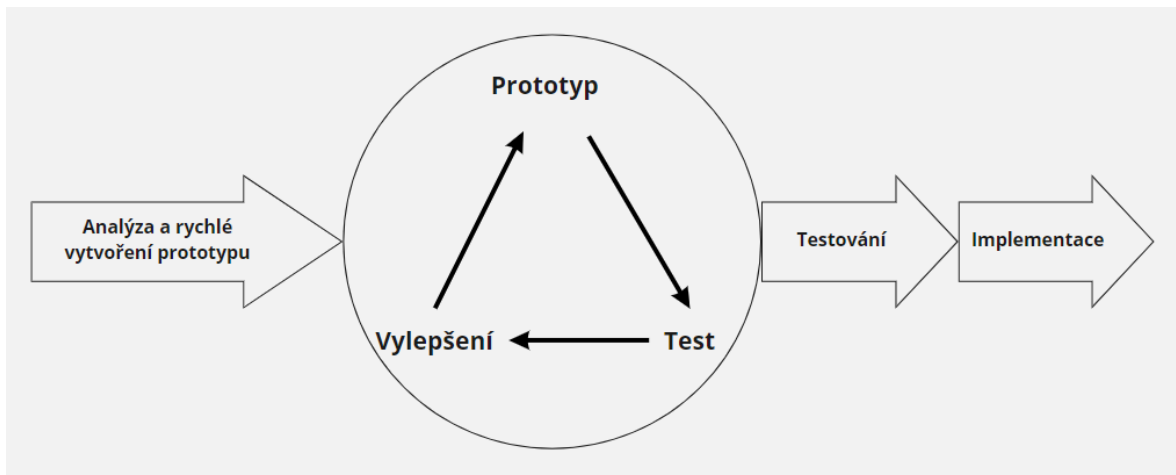
3.3. Metodika vývoje software

Vývoj softwaru se skládá z různých specifických postupů, pravidel a nástrojů, pomocí kterých se řídí celý cyklus vývoje aplikace od prvotního návrhu přes plánování, samotný vývoj software a testování. Metodické frameworky softwarového vývoje lze rozdělit na: (6)

- prototypování
- spirála
- vodopádový model

3.3.1. Prototypování

V případě metodiky prototypování se vytváří „prototyp“, čím se rozumí částečná implementace software, který by měl reprezentovat všechna dostupná vnější rozhraní. Uživatelé jsou zapojeni do procesu testování na základě, kterého mohou vznést připomínky na úpravy SW. (4,6)



Obrázek 7 - Prototypování (29)

3.3.2. Spirálový model

Spirálový model je odvozený od vodopádového modelu, oproti tomu je však doplněn o některé odlišné vlastnosti: (5,6)

- iterativní přístup
- podrobná analýza rizik

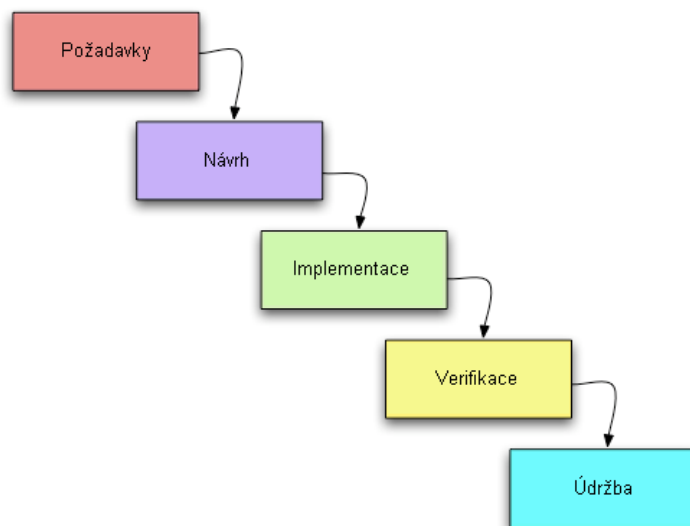
V případě spirálové metodiky vývoje software se opakují po sobě čtyři hlavní činnosti. V první řadě se provede podrobná analýza problému, ta se následně vyhodnotí a pomocí algoritmických technik probíhá její vývoj. V posledním bodě probíhá plánování dalšího cyklu spirály. (5,6)



Obrázek 8 - Model spirála (30)

3.3.3. Vodopádový model

Tento model znázorňuje posloupnost vzájemně na sebe navazujících etap, posun je pouze vpřed. Neexistují zde žádné cyklické návraty zpět. Výhodou vodopádového modelu je jeho jednoduchost z hlediska řízení projektu, v případě stálých a jasných požadavků je nejlepší struktura výsledného produktu. Nevýhodou tohoto přístupu je, že zákazník není schopen přesně stanovit všechny požadavky dopředu. Při změnách v zadání se protahuje doba realizace. Funkční spustitelná verze je až v závěrečných fázích projektu. (6,7,8)



Obrázek 9 - Vodopádový model (31)

4. Vlastní práce

Tato část práce se zabývá problémem, kdy je většina vytvořených aplikací pro zaznamenávání sportovní aktivity webových. Často jsou opomíjené klasické desktopové aplikace. Praktická část práce se zabývá vývojem desktopové aplikace pro zaznamenání sportovních aktivit se zaměřením na triatlon. Aplikace je vytvořena jako Windows Forms Application na softwarové struktuře .NET Framework.

4.1. Analýza

Analýza a samotný návrh aplikace popsany v následující kapitole se týká desktopové aplikace pro evidenci tréninkových dat sportovců. Sportovec po registraci získá přístup do uživatelského rozhraní systému. Bude zde mít k dispozici možnost přidat nový trénink, mazání již zadaných, získá informace o detailních statistikách v jednotlivých disciplínách. Druhým typem uživatele je Administrátor systému, ten bude moci mazat jednotlivé uživatele a spravovat veškerý obsah aplikace.

Požadavky na aplikaci jsou následující:

Funkční požadavky:

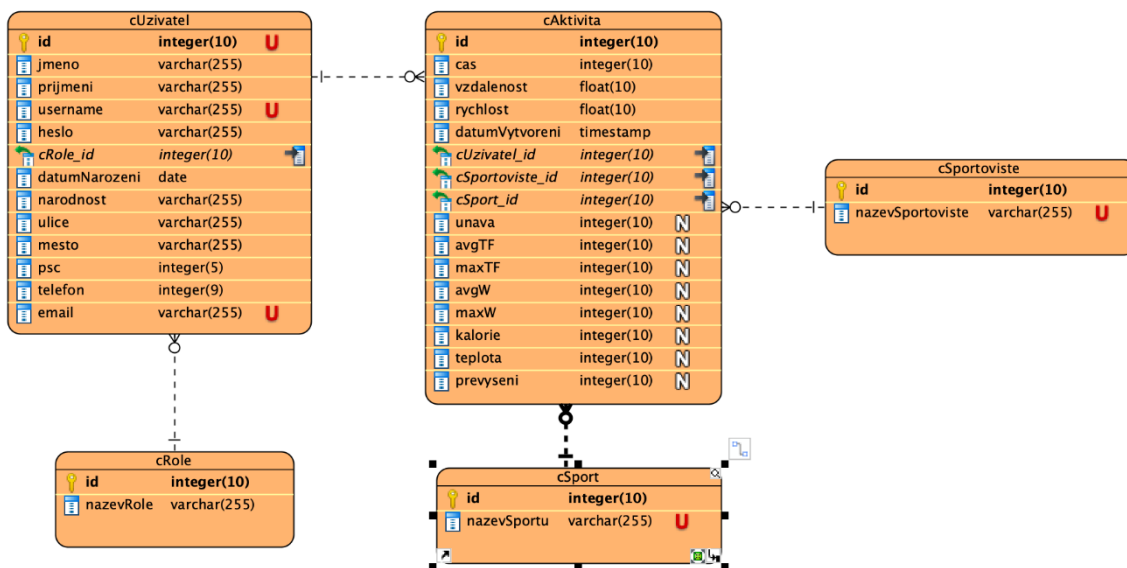
- Registrace sportovce
- Možnost úpravy osobních údajů
- Vytvoření nové aktivity
- Smazání již vytvořených aktivit
- Vedení kompletních statistik o aktivitách
- Správa uživatelů

Systém se bude skládat ze dvou typů uživatelských účtů (Sportovec, Administrátor), každý bude mít své rozhraní.

Nefunkční požadavky:

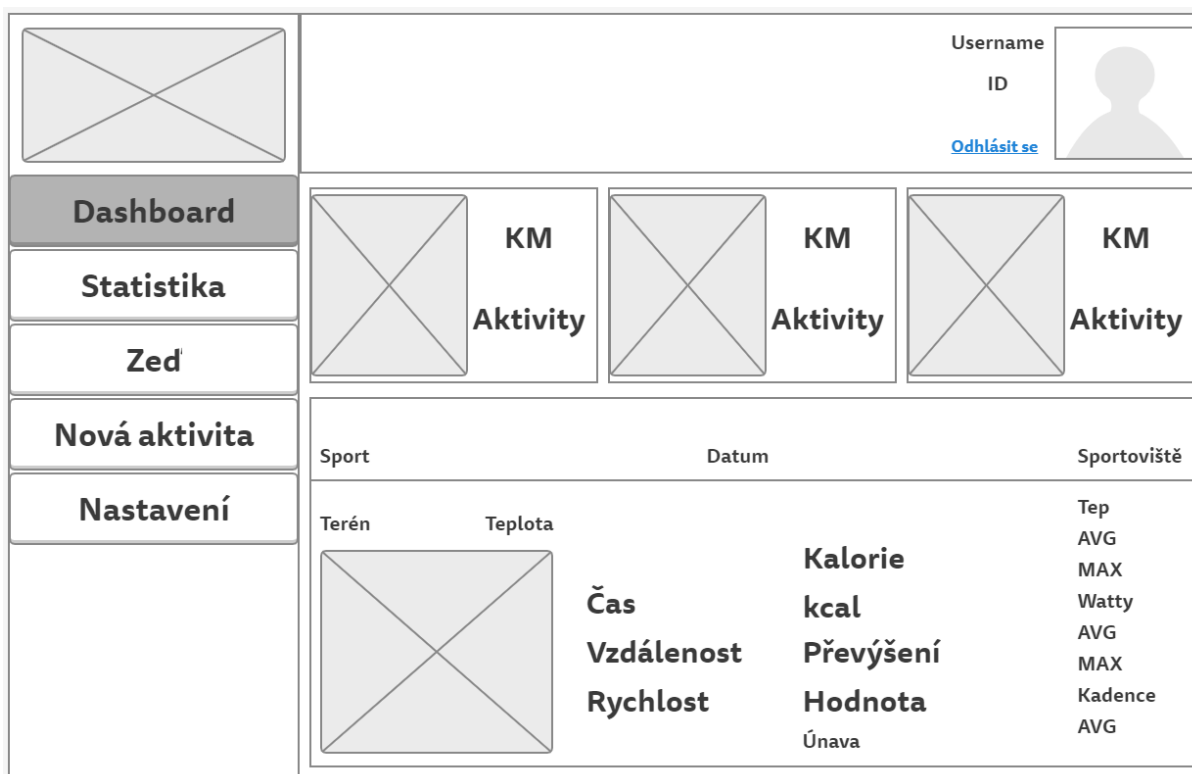
- Využití frameworku .NET Windows Forms
- Databáze Azure SQL Server
- Aplikace pro operační systém Windows 10

Struktura databáze je znázorněna v entitně relačním diagramu níže, dále pak je detailněji popsána v další kapitole návrhu, kterým je datový slovník.



Obrázek 10 - ER diagram [archiv autora]

Aplikace bude vyvíjena za pomoci spirálové metody vývoje software. Jednotlivé funkcionality po vyvinutí projdou fází testování a následného doladění. Postupovat se tak bude od prvního formuláře přihlášení a registrace. Jelikož aplikace požaduje pro správný chod na hlavní obrazovce ID uživatele, musí být nejprve vyvinuto a doladěno přihlašování uživatele. Následně bude vyvinuto i samotné prostředí s postupně přibývajícimi funkcionalitami.



Obrázek 11 - Wireframe - dashboard [archiv autora]

4.2. Tvorba databáze

4.2.1. Datový slovník

V následující kapitole je podrobně popsána struktura jednotlivých tabulek, které se v databázi nacházejí.

cUzivatel

Hlavní tabulkou v aplikaci je tabulka uživatelů, obsahující veškeré informace o dané osobě, pomocí vazby do tabulky cRole je určeno, jakou roli má uživatel přiřazenou.

Název	Datový typ	Nulový	Výchozí	Poznámka
id	int(10)	Ne	Autoincrement	Primární klíč
jmeno	varchar(255)	Ne		
prijmeni	varchar(255)	Ne		
username	varchar(255)	Ne		Unique
heslo	varchar(255)	Ne		
datumNarozeni	date	Ano		
narodnost	varchar(255)	Ano		
ulice	varchar(255)	Ano		
město	varchar(255)	Ano		
psc	int(5)	Ano		
telefon	int(9)	Ano		
email	varchar(255)	Ano		Unique
cRole_id	int(10)	Ne		Cizí klíč

cAktivita

Tabulka cAktivita v sobě udržuje veškeré informace o všech aktivitách sportovců, nachází se zde také propojení s tabulkami sportů a sportovišť.

Název	Datový typ	Nulový	Výchozí	Poznámka
id	int(10)	Ne	Autoincrement	Primární klíč
cas	integer(10)	Ne		
vzdalenost	float(10)	Ne		
rychlost	float(10)	Ne		
datumVytvoreni	timestamp	Ne		

unava	int(10)	Ano	NULL	
avgTF	int(10)	Ano	NULL	
maxTF	int(10)	Ano	NULL	
avgW	int(10)	Ano	NULL	
maxW	int(10)	Ano	NULL	
kalorie	int(10)	Ano	NULL	
teplota	int(10)	Ano	NULL	
prevyseni	int(10)	Ano	NULL	
cUzivatel_id	int(10)	Ne		Cizí klíč
cSport_id	int(10)	Ne		Cizí klíč
cSportoviste_id	int(10)	Ne		Cizí klíč

Tabulky sportů a sportovišť v sobě udržují informaci o jednotlivých dostupných sportech a sportovištích v aplikaci.

cSportoviste

Název	Datový typ	Nulový	Výchozí	Poznámka
id	int(10)	Ne	Autoincrement	Primární klíč
nazevSportoviste	varchar(255)	Ne		

cSport

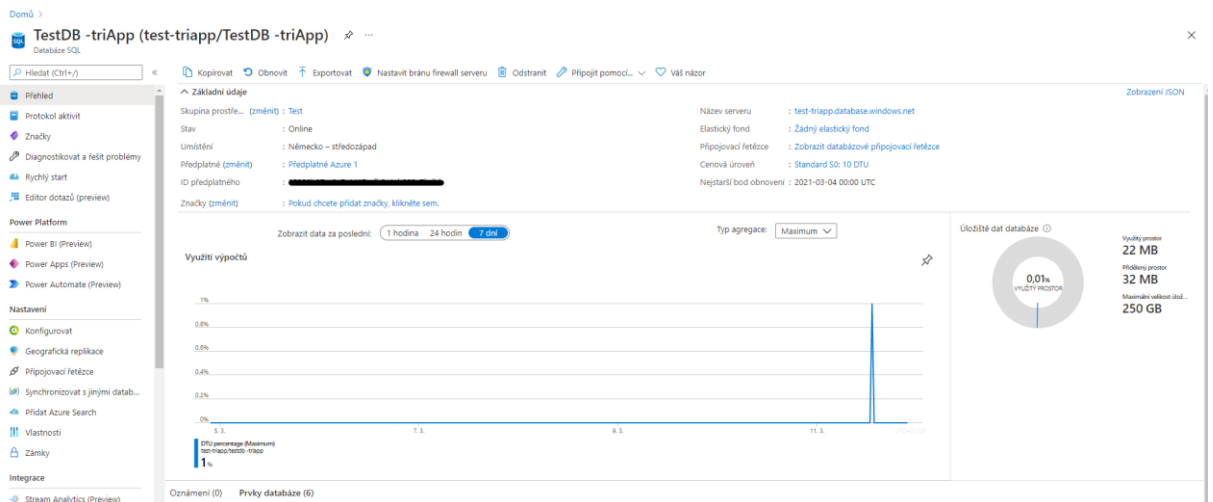
Název	Datový typ	Nulový	Výchozí	Poznámka
id	int(10)	Ne	Autoincrement	Primární klíč
nazevSportu	varchar(255)	Ne		Unique

cRole

Poslední tabulkou, která se v této databázi nachází je tabulka s informací o jednotlivých rolích použitých v aplikaci.

Název	Datový typ	Nulový	Výchozí	Poznámka
id	int(10)	Ne	Autoincrement	Primární klíč
nazevRole	varchar(255)	Ne		Unique

4.2.2. Azure SQL Server

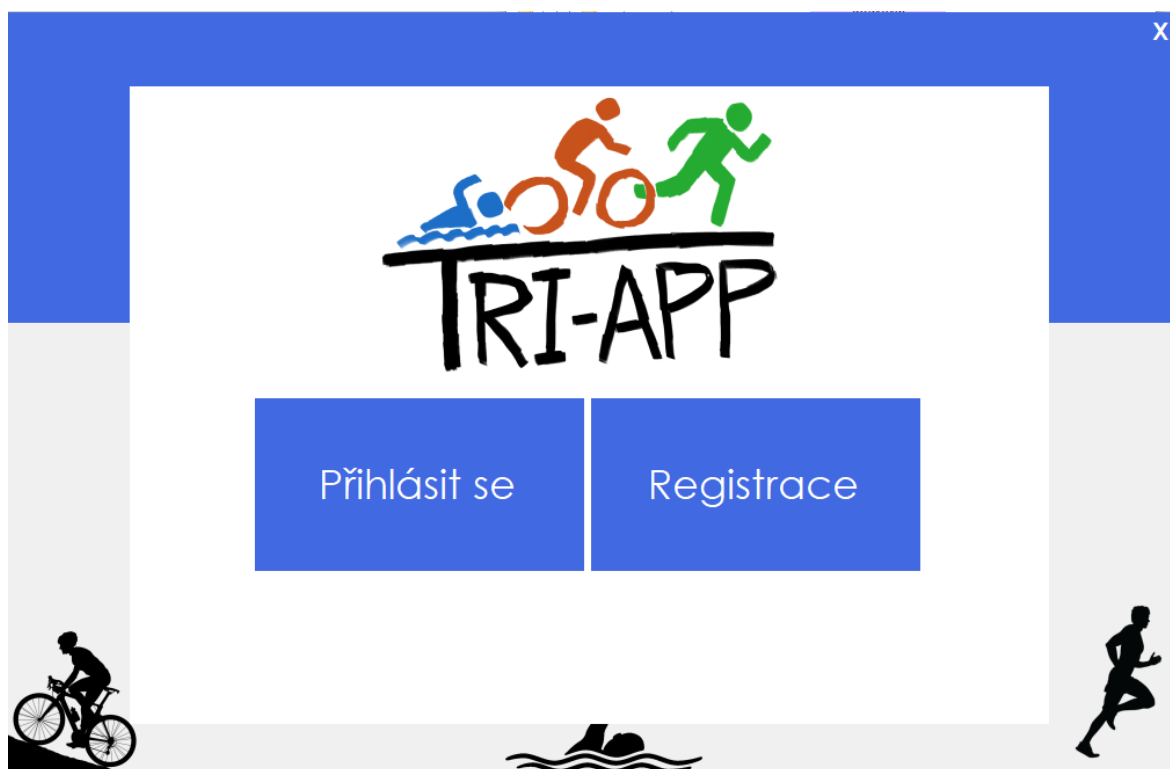


Obrázek 12 - Dashboard azure [archiv autora]

Databázová část aplikace je vytvořena pomocí cloudového řešení Microsoft Azure, ve kterém je užit produkt Microsoft SQL Server. Databáze je tvořena tabulkami cUzivatel, cAktivita, cRole, cSport a cSportoviste. V tabulce cSport a cSportoviste se nacházejí konkrétní údaje o sportech a sportovištích. Tabulka cUzivatel definuje uživatele aplikace a tabulka cAktivita definuje jednotlivé vytvořené aktivity v aplikaci. Tabulka cRole definuje roli samotného uživatele a následně se podle ní načítá hlavní obrazovka. V aplikaci existuje role obyčejného koncového uživatele a administrátora. Uživateli je umožněno přidávat nové aktivity, měnit své údaje a sledovat všechny vytvořené aktivity. Administrátor má právo sledovat veškeré vytvořené aktivity, vytvořené uživatelské účty a libovolně je mazat. U vytvořených účtů má rovněž právo změnit roli uživatele.

4.3. Přihlašovací obrazovka

Po spuštění aplikace nabízí možnost přihlášení, nebo registrace jako nový uživatel. Přihlašovací obrazovka po kliknutí na konkrétní tlačítko přenesení počáteční panel do pozadí a zobrazí zvolený proces, který reprezentuje uživatelský ovládací prvek. Po kliknutí na jedno z tlačítek je možné se pomocí tlačítka zpět vrátit na úvodní obrazovku



Obrázek 13 - Úvodní obrazovka [archiv autora]

4.3.1. Registrace

Existují dvě možnosti registrace. Při zkrácené registraci se vyplňuje pouze jméno, příjmení, uživatelské jméno a heslo. V kompletní registraci se pak dále vyplňuje email, telefonní číslo, ulice, město, PSČ, národnost a datum narození. Aplikace vyžaduje vyplnění všech údajů pro danou registraci. V rámci databáze se nachází atribut role. Při registraci je automaticky nastaven na hodnotu 1 (sportovec)

4.3.1.1. Zkrácená registrace

Registrace

Jméno * Příjmení *

Username * Heslo *

Souhlasím s vyplněním nepovinné části registrace Uživatel

Vytvořit účet

Obrázek 14 - Zkrácená registrace [archiv autora]

Při zkrácené registraci nám aplikace zobrazí čtyři základní údaje pro vyplnění. Po stisku tlačítka vytvořit účet se jednotlivá pole zkontrolují, zda se v nich nenachází nevyplněné údaje.

```
if (validace == true && namevalidace == true && passvalid == true)
{
    jmeno = textBoxjmeno.Text;
    prijmeni =textBoxprijmeni.Text;
    username=textBoxusername.Text;
    pass=textBoxpass.Text;

    sestaveniDotazu = "INSERT INTO cUzivatel (id,jmeno,prijmeni,username,heslo,cRole)
VALUES ((SELECT MAX(id) FROM cUzivatel)+1,'" + jmeno + "','" + prijmeni + "','" +
username + "','" + pass + "','1)";

    cnn.Open();
    command = new SqlCommand(sestaveniDotazu, cnn);
    adapter.InsertCommand = new SqlCommand(sestaveniDotazu, cnn);
    adapter.InsertCommand.ExecuteNonQuery();
    command.Dispose();
    cnn.Close();

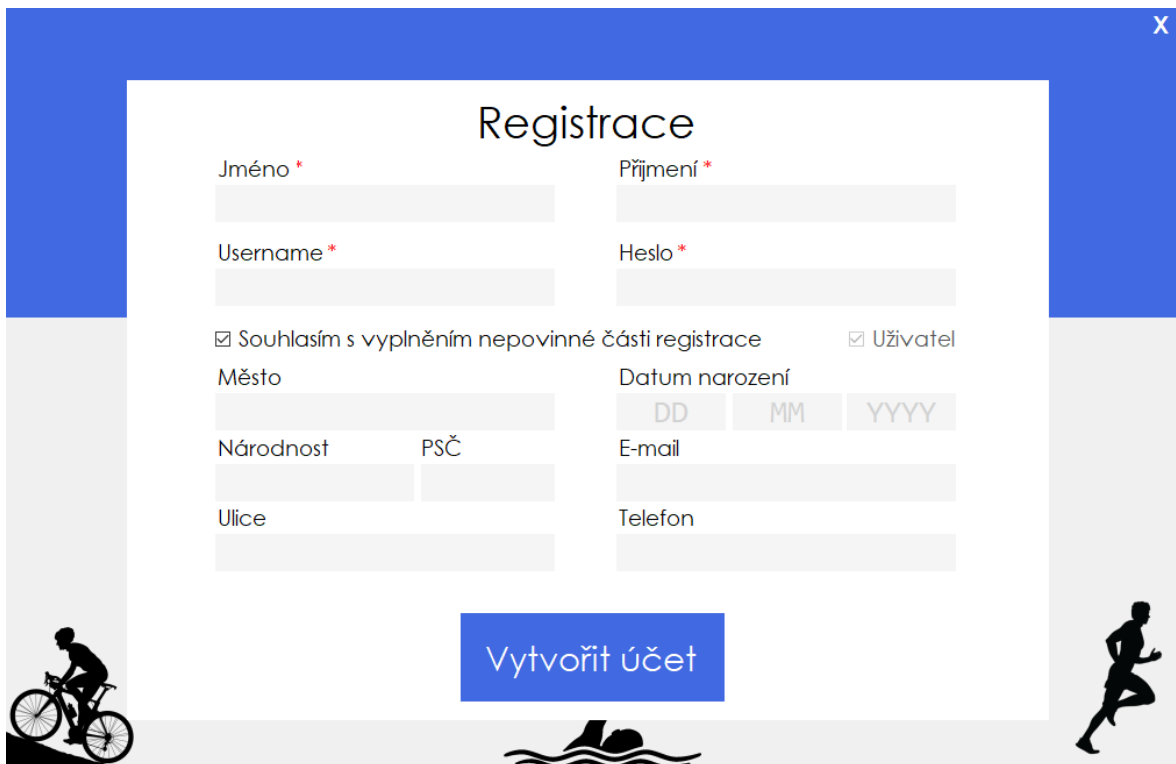
    this.Visible = false;
}
```

Po kontrole všech prvků registrace se provolá podmínka, zda jsou všechny pole vyplněné, a zároveň zda není uživatelské jméno již využito a heslo je dostatečně silné. Pokud jsou všechny tyto podmínky splněny, proměnné se naplní hodnotami z text-boxů a následně je s jejich pomocí sestaven konkrétní dotaz pro databázi.

```
INSERT INTO cUzivatel (id,jmeno,prijmeni,username,heslo,cRole) VALUES ((SELECT MAX(id)
FROM cUzivatel)+1,'jmeno','prijmeni','username','pass',1)
```

Po sestavení dotazu aplikace spustí komunikaci s AZURE databází a následné vložení nových údajů za pomoci sestaveného dotazu.

4.3.1.2. Kompletní registrace

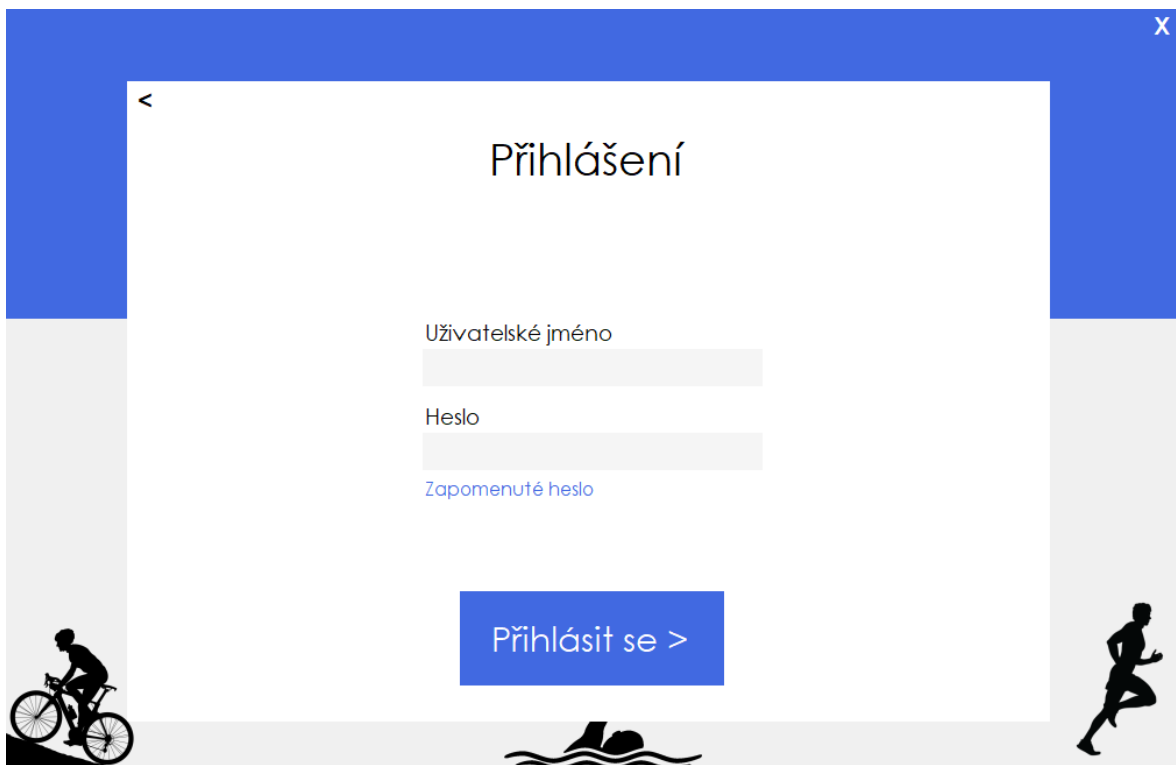


Obrázek 15 - Kompletní registrace [archiv autora]

Kompletní registrace se od té zkrácené liší pouze v sestavení dotazu, kdy nejsou vybrány pouze konkrétní atributy, ale jsou vyplněny všechny. Sestavený dotaz má následnou podobu.

```
INSERT INTO [dbo].[cUzivatel] VALUES ((SELECT MAX(id) FROM
cUzivatel)+1,'jmeno','prijmeni','username','pass',1,'YYYY-MM-
DD','narodnost','ulice','mesto','PSC','telefon','email')
```

4.3.2. Přihlášení



Obrázek 16 - Přihlášení [archiv autora]

Po stisku na tlačítko Přihlášení nám aplikace zobrazí přihlašovací formulář. Po stisku tlačítka se načte hodnota z text-boxu „Uživatelské jméno“ a následně bude porovnána s databází. Aplikace spustí dotaz, který načte počet uživatelů v tabulce cUzivateL.

```
SELECT Count(ID) FROM cUzivateL
```

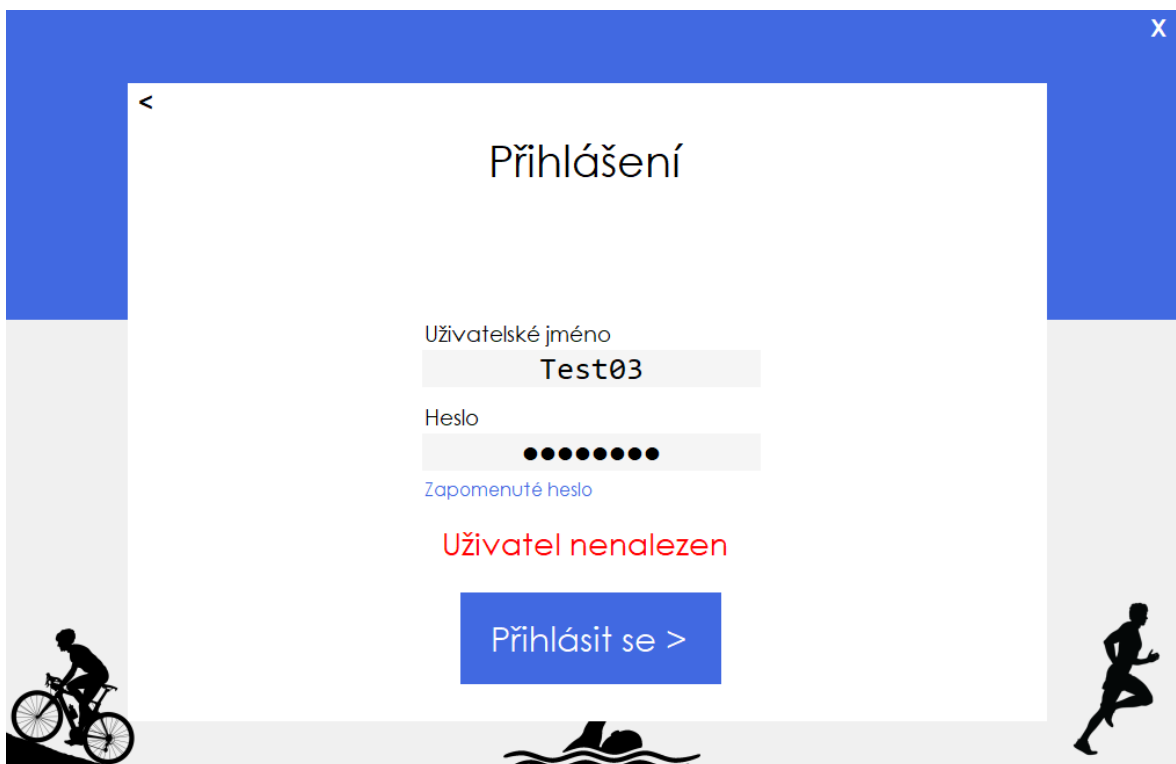
Následně je tento počet použit v cyklu for, kdy aplikace projde všechny všechny uživatele a porovnává, zda se shoduje uživatelské jméno a heslo přidružené k němu. Pokud aplikace uživatele najde, otevře nový form „Dashboard“ a zároveň do jeho veřejných proměnných pošle důležitá vstupní data. Mezi tyto data patří ID uživatele pro zobrazení konkrétních aktivit uživatele a taktéž role uživatele.

```

for (int i = 1; i <= IDcount; i++)
{
    string commandos = "SELECT username,heslo FROM cUzivatel WHERE id='" + i.ToString() +
    """;
    comLog.CommandText = commandos;
    SqlDataReader dr = comLog.ExecuteReader();
    if (dr.Read())
    {
        if (textBoxUser.Text.Equals(dr["username"].ToString()) &&
        textBoxPassword.Text.Equals(dr["heslo"].ToString()))
        {
            d.Show();
            d.ID = GetID;
            d.Role = getRole;
            d.refresh();
            d.roleCheck();
            d.dash = d;
            d.dash2 = d;
            this.Visible = false;
        }
        else
        {
            labelWrong.Visible = true;
        }
    }
}
dr.Close();
}

```

Pokud uživatel nalezen není, aplikace nám zobrazí chybovou hlášku.



Obrázek 17 - Přihlášení 2 [archiv autora]

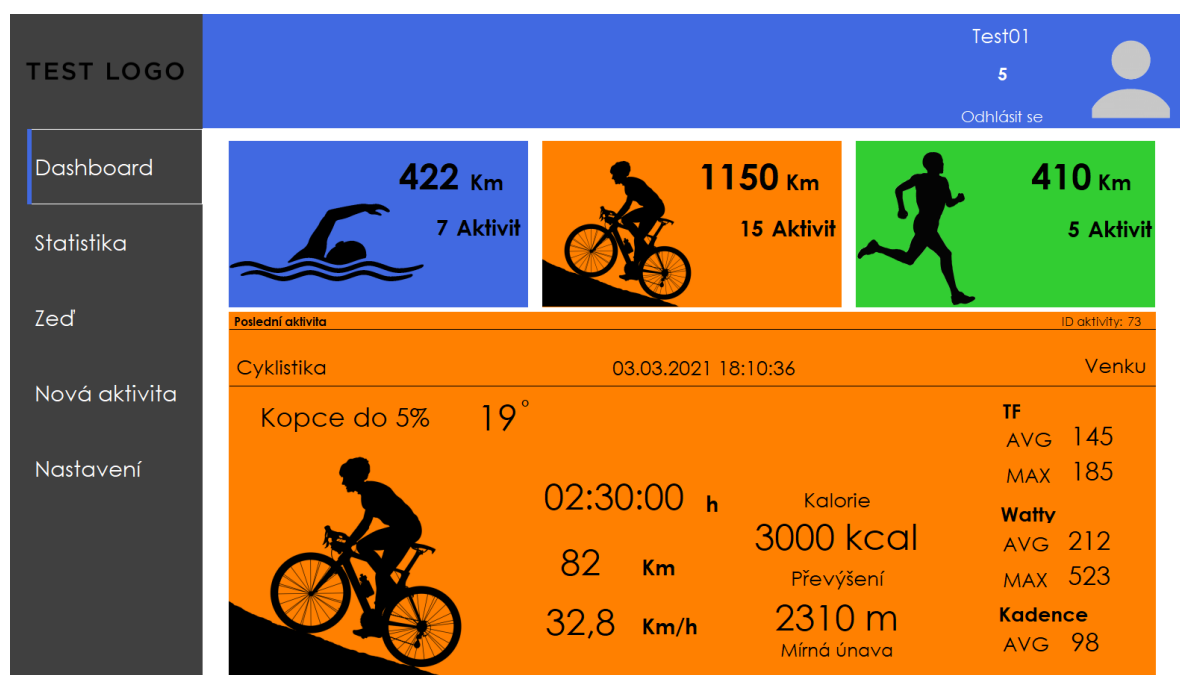
4.4. Hlavní obrazovka

Po úspěšném přihlášení aplikace načte hlavní obrazovku v závislosti na konkrétní roli uživatele. Uživatel s rolí admin tedy uvidí hlavní obrazovku jinak, než obyčejný koncový uživatel.

4.4.1. Sportovec

4.4.1.1. Dashboard

Aplikace po načtení hlavní obrazovky pro uživatele zobrazí Dashboard, kde se nachází krátké přehledy za jednotlivé disciplíny a taktéž nám zobrazí detail poslední uložené aktivity.



Obrázek 18 - Dashboard [archiv autora]

Na hlavní obrazovce se nachází veřejná proměnná ID, kterou aplikace naplní při přihlašování uživatele a následně je tato hodnota používána při sestavování většiny dotazů v aplikaci.

Za pomoci propojení tabulek v databázi vypíšeme poslední aktivitu daného uživatele i s konkrétními údaji o sportovišti a sportu. Sestavený dotaz má podobu následující.

```

SELECT
a.id,u.jmeno,u.username,s.nazevSportu,sp.nazevSportoviste,a.cas,a.vzdalenost,a.rychlost,
a.datumvytvoreni,a.kalorie, a.unava,a.AVGTF,a.MAXTF, a.AVGWatt,a.MAXWatt,
a.AVGcad,a.teplota,a.prevyseni,a.teren,a.plavaniTypBazenu from caktivita a

join [dbo].[cSport] s on s.id = a.cSport
join cSportoviste sp on sp.id = a.cSportoviste
join cUzivatel u on u.id = a.cUzivatel

WHERE a.cUzivatel='ID'
AND a.id = (SELECT MAX(id) FROM cAktivita WHERE cUzivatel = 'ID')

```

Dále pak nám aplikace zobrazí údaje za jednotlivé sporty (plavání, cyklistika a běh). V přehledu se zobrazí počet aktivit a celkový počet kilometrů jednotlivých sportů. Pro každý sport je sestrojen následný dotaz.

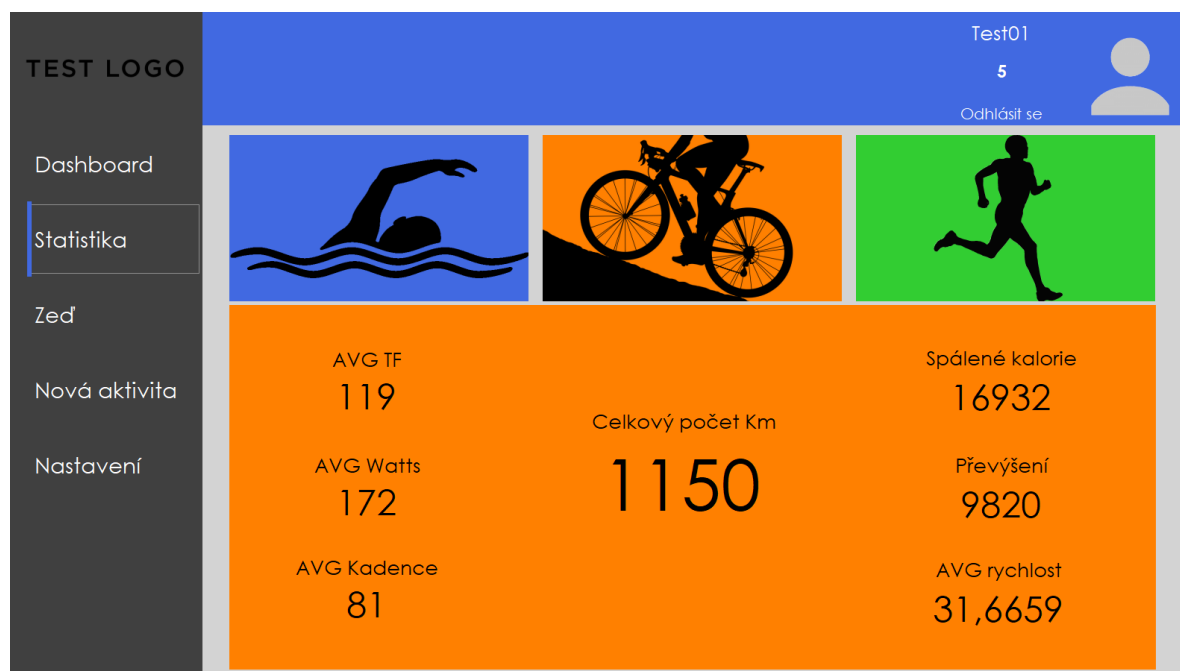
```

SELECT SUM(vzdalenost) FROM cAktivita WHERE cSport = '1' AND cUzivatel = 'ID'

```

4.4.1.2. Statistika

Obrazovka statistiky nám zobrazí všechny čísla a přehledy za jednotlivé sporty.



Obrázek 19 - Statistiky [archiv autora]

Aplikace pomocí dotazu vytáhne z databáze potřebná data pro konkrétní ID a přiřadí je konkrétním sportům v záložkách.

```
SELECT  
  
SUM(kalorie),AVG(NULLIF(AVGTF,0)),SUM(vzdalenost),AVG(NULLIF(AVGWatt,0)),  
AVG(NULLIF(AVGcad,0)),SUM(prevyseni),AVG(NULLIF(rychlost,0)) FROM cAktivita  
  
WHERE cSport = 2 AND cUzivatel = ID
```

Takto sestavený dotaz aplikace pošle do databáze a následným výstupem naplní jednotlivé proměnné.

```
con.Open();  
commbeh.Connection = con;  
commbeh.CommandText= "SELECTSUM(kalorie),  
AVG(NULLIF(AVGTF,0)),SUM(vzdalenost),AVG(NULLIF(AVGWatt,0)),AVG(NULLIF(AVGcad,0)),SUM(  
prevyseni),AVG(NULLIF(rychlost,0)) FROM cAktivita WHERE cSport = 2 AND cUzivatel = " +  
ID + """;  
  
    SqlDataReader behread = commbeh.ExecuteReader();  
    while (behread.Read())  
    {  
        behkalorie = behread.GetValue(0).ToString();  
        behAVGTF = behread.GetValue(1).ToString();  
        behVzdalenost = behread.GetValue(2).ToString();  
        behAVGW = behread.GetValue(3).ToString();  
        behAVGCAD = behread.GetValue(4).ToString();  
        behprevyseni = behread.GetValue(5).ToString();  
        behrychlost = behread.GetValue(6).ToString();  
  
    }  
    behread.Close();
```

4.4.1.3. Zed'

Na obrazovce „Zed“ se zobrazí poslední vytvořené aktivity všech uživatelů. Každá aktivita představuje jeden uživatelský ovládací prvek, který se dynamicky vytvoří podle počtu všech aktivit v databázi. Každá stránka na zdi zobrazí devět aktivit a aplikace nám umožní načítat další aktivity do doby, než dojde k načtení nejstarší aktivity v databázi



	Datum	Uživatel	Sport	Čas	Km	ID
Dashboard	03.03.2021 18:10:36	Test01	Cyklistika	02:30:00 h	82 K m	73
Statistika	10.02.2021 20:26:43	M_tvrz	Cyklistika	03:23:12 h	107 K m	72
Zed'	10.02.2021 20:24:46	Test02	Plavání	02:30:00 h	82 K m	71
	10.02.2021 19:03:13	Test01	Plavání	02:30:00 h	6 K m	70
Nová aktivita	10.02.2021 19:00:01	Test01	Běh	02:30:00 h	82 K m	69
	10.02.2021 17:03:19	Test01	Cyklistika	02:30:00 h	82 K m	68
Nastavení	10.02.2021 16:26:05	Test01	Běh	02:30:00 h	82 K m	67
	10.02.2021 16:12:40	Test01	Plavání	02:30:00 h	6 K m	66
	10.02.2021 16:12:11	Test01	Plavání	02:30:00 h	82 K m	65

Obrázek 20 - Zed' [archiv autora]

Po kliknutí na tlačítko „Zed“ nám aplikace přenesse do popředí uživatelský ovládací prvek zed', kterému předáme hodnotu, kterou bude vypisovat a následně zed' spustí metodu dynamického vykreslování uživatelských ovládacích prvků aktivita

	00:00:00 00-00-00	Username	Sport	00:00:00 h	22.3 K m		ID
---	-------------------	----------	-------	------------	-------------	---	----

Obrázek 21 - Aktivita [archiv autora]

V prvku aktivita se zobrazí miniatura sportu, obecné informace o aktivitě jako jsou čas, vzdálenost, sport a uživatel, který aktivitu vytvořil. Dále pak podle role je zobrazen koš pro možnost smazání aktivity.

```

con.Open();
idcheck.Connection = con;
idcheck.CommandText = "SELECT (id) FROM caktivita ORDER BY id DESC";
SqlDataReader dat = idcheck.ExecuteReader();
    while (dat.Read() && count < 9 * nas)
    {
        idecko = dat.GetValue(0).ToString();
        if (podminka == true)
        {
            aktivit A = new aktivit(idecko, role); ;
            this.Controls.Add(A);
            A.Location = new Point(0, i);
            i += 70;
            A.dash = Dash;

            if (idecko == idmin)
            {
                Dash.enaButtdale();
            }
        }
        count++;
    }
con.Close();

```

Následující část metody pomocí dotazu vypíše všechny aktivity v databázi, ze kterých poté vytáhne devět aktivit podle aktuální stránky, kterou reprezentuje v kódu násobek. Taktéž aplikace u každého ID aktivity ověří, zda se nejedná o nejnižší hodnotu v databázi. Pokud aplikace načte nejnižší hodnotu, tlačítko další stránky se zpřístupní.

4.4.1.4. Nová aktivita

Na záložce nové aktivita nám aplikace dovoluje zadat nová data, která budou spojeny s ID dotyčného uživatele. Po otevření záložky se nám objeví pouze část polí pro zadání údajů. Rozhodující jsou dva combo-boxy „Sport“ a „Sportoviště“. Po zvolení sportu nám aplikace odkryje druhý combo-box sportoviště naplněný odpověďmi pro daný sport. V závislosti na tyto dvě odpovědi se nám zobrazí konkrétní datová pole pro konkrétní sporty. Například kombinací cyklistika venku se zobrazí všechna možná datová pole

The screenshot shows a web application interface for creating a new activity. The main content area is titled "Nová aktivita" and contains a form with the following fields:

- Uživatel:** Text input with "Test01" and a timestamp "03.03.2021 19:10:36".
- Sport:** Dropdown menu with "Cyklistika" selected.
- Sportoviště:** Dropdown menu with "Venku" selected.
- Čas:** Text input field.
- Vzdálenost:** Text input field.
- Rychlost:** Text input field.
- Kalorie:** Text input field.
- Únava:** Dropdown menu.
- AVG TF, MAX TF, AVG W, MAX W, AVG CAD:** Five text input fields.
- Teplota, Převýšení, Terén:** Three dropdown menus.

A blue button labeled "Vytvořit aktivitu" is located at the bottom right of the form. The interface also features a sidebar on the left with navigation options: "Dashboard", "Statistika", "Zed", "Nová aktivita" (highlighted), and "Nastavení". The top right corner shows the user "Test01" with a profile icon and the number "5", along with a link to "Odhlásit se".

Obrázek 22 - Nová aktivita [archiv autora]

Pole se dělí na povinná a nepovinná. Základní údaje o aktivitě jako je čas a vzdálenost musí být vyplněné vždy. Ostatní atributy jsou v základu nastavené jako nulové a pouze, pokud je uživatel vyplní, proměnné se přepíšu při sestavování finálního dotazu, který má podobu následující.

```
INSERT INTO cAktivita VALUES ((select MAX(id) from
cAktivita)+1,ID,sport",sportoviste','HH:MM:SS',vzdalenost,rychlost,getdate(),kalorie,'u
nava',avgTF,maxTF,avgW,maxW,avgCAD,teplota,prevyseni,'teren',typBazenu)
```

4.4.1.5. Nastavení

Aplikace nabízí taktéž možnost změnit údaje uživatele. Uživateli se tak nabízí tři možné změny. První změna je změna obecných údajů. Zde se mohou nastavit atributy ulice, město, PSČ, národnost, telefonní číslo, email, jméno a příjmení. Druhá možnost je změna uživatelského jména, kdy se po zadání hodnoty do text-boxu spustí validace, zda se v databázi nenachází uživatel se stejným jménem. Pokud se tak stane, aplikace nám nedovolí změnu uložit. Třetí možnost je změna hesla, kdy se po zadání hesla spustí validace na sílu hesla a po následném znovu vyplnění zkontroluje, zda se hesla shodují. Pokud splní tyto dvě podmínky, aplikace změnu uloží, v opačném případě nikoli.

TEST LOGO

Test01
5
Odhlásit se

Dashboard
Statistika
Zed'
Nová aktivita
Nastavení

Úprava profilu

Jméno * Test01 Příjmení * -
Username * Test01 Heslo * ●●●●●●●●
 Uživatel

Město Plzen Datum narození 04.01.1999 0:00:00
Národnost CZE PSČ 19000 E-mail test01@triapp
Ulice - Telefon -

Změna uživatelského jména

Username *

Změna hesla

Heslo *
Potvrzení hesla *

Uložit

Obrázek 23 - Nastavení 1 [archiv autora]

```

{
    namevalid = true;
    labelWrong.Visible = false ;
    SqlCommand cmd = new SqlCommand();
    con.Open();
    cmd.Connection = con;
    cmd.CommandText = "SELECT username FROM cUzivatel";
    SqlDataReader drt = cmd.ExecuteReader();
    while (drt.Read())
    {
        if (textBox3.Text == drt.GetValue(0).ToString())
        {
            namevalid = false;
            labelWrong.Visible = true;
        }
    }
    con.Close();
}

```

Aplikace sestaví dotaz pro zobrazení všech uživatelských jmen existujících v databázi. Pomocí cyklu while projde všechny hodnoty, kdy aplikace zkontroluje podmínku, zda se uživatelské jméno rovná jménu v databázi. Pokud se tak stane, proměnná typu bool změní hodnotu na false a není dovoleno změnu uložit

Obrázek 24 - Nastavení 2 [archiv autora]

4.4.2. Administrátor

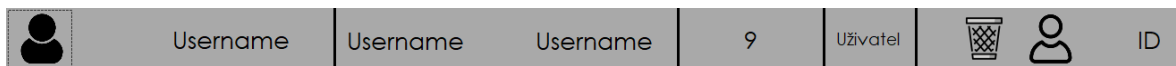
Administrátor je uživatel s právy na mazání uživatelů i jednotlivých aktivit. Místo klasické hlavní obrazovky se po přihlášení objeví takzvaná servisní obrazovka. Z této obrazovky může správce otevřít přehled všech registrovaných uživatelů, kde má právo uživatele smazat, uživatele má možnost povýšit na roli správce nebo uživateli tuto roli sebrat a degradovat na klasického uživatele. Dále pak má správce možnost otevřít přehled všech aktivit a libovolné aktivity má možnost smazat.

4.4.2.1. Uživatelé

Username	Jméno a příjmení	Aktivity	Role	ID
admin	Administrátor -	0	Admin	1
M_tvrz	Matej Tvrz	1	Uživatel	2
fullreg	FullREG -	0	Uživatel	3
L_tvrz	Lukáš Tvrz	0	Uživatel	4
Test01	Test01 -	27	Uživatel	5
Test02	Test02 -	1	Uživatel	6
Test03	Test03 -	0	Uživatel	7

Obrázek 25 - Uživatelé [archiv autora]

Uživatelé se obdobně jako aktivity dynamicky vykreslí na uživatelský ovládací prvek zeď. Uživatelský ovládací prvek tvoří miniatura, uživatelské jméno, jméno a příjmení, celkový počet aktivit, role, ID a tlačítko pro smazání a změnu role.



Obrázek 26 - Uživatel [archiv autora]

```

{
    DialogResult result = MessageBox.Show("Opravdu chcete smazat uživatele " +
    UserNAME.Text + " ?", "?", MessageBoxButtons.YesNo);
    if (result == DialogResult.Yes)
    {
        con.Open();
        SqlDataAdapter adapter = new SqlDataAdapter();
        SqlCommand delAkt = new SqlCommand();
        string dotaz = "DELETE FROM cAktivita WHERE cUzivatel = '"+ID+"'";
        delAkt = new SqlCommand(dotaz, con);
        adapter.InsertCommand = new SqlCommand(dotaz, con);
        adapter.InsertCommand.ExecuteNonQuery();
        adapter.Dispose();
        con.Close();

        con.Open();
        SqlDataAdapter adapter1 = new SqlDataAdapter();
        SqlCommand deluser = new SqlCommand();
        string dotaz1 = "DELETE FROM cUzivatel WHERE id = '" + ID + "'";
        deluser = new SqlCommand(dotaz1, con);
        adapter1.InsertCommand = new SqlCommand(dotaz1, con);
        adapter1.InsertCommand.ExecuteNonQuery();
        adapter1.Dispose();
        con.Close();

        dash.UserRefresh();
    }
    else if (result == DialogResult.No)
    {
    }
}

```

Pokud se správce rozhodne smazat uživatele, provolá se metoda uvedená výše. V první fázi aplikace získá ID vybraného uživatele a následně sestrojí dotaz pro vymazání všech vytvořených aktivit pod tímto uživatelem. V druhé fázi aplikace vezme ID uživatele a sestaví dotaz pro smazání uživatele. Po těchto krocích se provede obnovení stránky a znovu vykreslení dynamických uživatelských ovládacích prvků.

4.4.2.2. Aktivity

Aplikace nám na obrazovce „Aktivity“ zobrazí obdobné prostředí, jako pro obvyčejného uživatele zed'. Změnu však tvoří právo na viditelnost tlačítka pro smazání aktivity.

	Datum	Uživatel	Sport	Čas	Km	ID		
Uživatelé		03.03.2021 18:10:36	Test01	Cyklistika	02:30:00 h	82 Km		73
Aktivita		10.02.2021 20:26:43	M_tvrz	Cyklistika	03:23:12 h	107 Km		72
		10.02.2021 20:24:46	Test02	Plavání	02:30:00 h	82 Km		71
		10.02.2021 19:03:13	Test01	Plavání	02:30:00 h	6 Km		70
		10.02.2021 19:00:01	Test01	Běh	02:30:00 h	82 Km		69
		10.02.2021 17:03:19	Test01	Cyklistika	02:30:00 h	82 Km		68
		10.02.2021 16:26:05	Test01	Běh	02:30:00 h	82 Km		67
		10.02.2021 16:12:40	Test01	Plavání	02:30:00 h	6 Km		66
		10.02.2021 16:12:11	Test01	Plavání	02:30:00 h	82 Km		65

Obrázek 27 - Aktivity [archiv autora]

```
{
    DialogResult result = MessageBox.Show("Opravdu chcete danou aktivitu smazat?",
        "?", MessageBoxButtons.YesNo);

    if (result == DialogResult.Yes)
    {
        con.Open();
        SqlDataAdapter ad = new SqlDataAdapter();
        SqlCommand deleteaktivity = new SqlCommand();
        string dotaz = "DELETE FROM cAktivita WHERE id = '" + AktivitaID.Text +
            "'";
        deleteaktivity = new SqlCommand(dotaz, con);
        ad.InsertCommand = new SqlCommand(dotaz, con);
        ad.InsertCommand.ExecuteNonQuery();
        ad.Dispose();
        con.Close();
        dash.ZedRefresh();
    }
    else if (result == DialogResult.No)
    {
    }
}
```

4.5. Testování aplikace

Testování proběhlo pomocí testovacích scénářů s následným otestováním pěti koncovými uživateli.

Testovací scénář 1 – Registrace uživatele

Uživatel otevře aplikaci, pokusí se zaregistrovat a následně se přihlásí.

Kroky:

- Zvolit možnost registrace (zkrácená/ kompletní)
- Vyplnit údaje a zaregistrovat se
- Zvolit možnost přihlášení
- Přihlásit se pomocí nově vytvořeného profilu

Očekávaný stav:

Po odkliknutí tlačítka registrace nám aplikace nabídne možnost zkrácené i kompletní registrace. V případě nevyplnění údajů nás varuje validační hláška. Po úspěšné registraci nás aplikace přesměruje zpět na úvodní stranu. Po kliknutí tlačítka přihlášení nám aplikace zobrazí formulář pro přihlášení. Při úspěšném přihlášení aplikace zobrazí obrazovku dashboard. V opačném případě se zobrazí chybová hláška.

Poznatky z testování:

- Aplikace při registraci nevaliduje strukturu e-mailu (exampe@seznam.cz) [3/5]
- Při kompletní registraci není viditelně označeno, že jsou všechna pole povinná [4/5]

Testovací scénář 2 – Vytvoření aktivity

Uživatel otevře aplikaci, přihlásí se, a poté vytvoří libovolnou novou aktivitu.

Kroky:

- Přihlásit se
- Přejít na záložku „Nová aktivita“
- Vytvořit libovolnou aktivitu

Očekávaný stav:

Po úspěšném přihlášení nám aplikace zobrazí dashboard. Uživatel má možnost přejít na záložku nové aktivity. Podle vybraného sportu nám aplikace zobrazí přidružené sportoviště. Následně kombinací sportu a sportoviště zobrazí dodatečné údaje pro vyplnění. Po úspěšném vytvoření nové aktivity nás aplikace přesměruje zpět na dashboard, kde bude zobrazena poslední vytvořená aktivita.

Poznátky z testování:

- Není jasně označeno, které pole jsou povinné a které nepovinné [5/5]
- Pokud jsou vyplněny všechny povinné údaje, ale combobox sport je na hodnotě „(none)“, aplikace nám nedovolí vytvořit aktivitu, ale nezobrazí se validační hláška [1/5]
- Aplikace na záložce „Dashboard“ má staticky nastaven label „aktivit“. V případě 1-4 aktivit tudíž nezobrazuje srozumitelně (př. 1 aktivit) [3/5]

Testovací scénář 3 – Změna údajů

Uživatel otevře aplikaci, přihlásí se, a poté se pokusí změnit své údaje v nastavení.

Kroky:

- Přihlásit se
- Přejít na záložku „Nastavení“
- Změnit libovolný údaj
- Uložit změny

Očekávaný stav:

Po úspěšném přihlášení nám aplikace zobrazí dashboard. Uživatel má možnost přejít na záložku nastavení. Zde nám aplikace nabídne změnit údaje, heslo nebo uživatelské jméno. Na údaje se nevztahuje žádná validace. Pokud uživatel zadá již existující uživatelské jméno, aplikace zobrazí chybovou hlášku. Rovněž aplikace zobrazí validační chybu, pokud uživatel nezadá dostatečně silné heslo, nebo po znovuzadání hesla neshodují. Pokud není zobrazena chybová hláška, aplikace nám dovolí změnu údajů uložit.

Poznátky z testování:

- V úpravě profilu (1. úprava) se nachází pole s uživatelským jménem a heslem, ale není zde možnost je změnit [3/5]
- V úpravě profilu (1. úprava) je možné hodnotu odebrat a uložit jako nulovou hodnotu [1/5]

5. Závěr

V první části se tato bakalářská práce zabývala teoretickým pohledem na téma .NET Framework. Dále byla popsána metodika vývoje software. Vysvětlení spirálového a vodopádového modelu a jejich hlavních rozdílů. V závěrečné pasáži teoretické části této práce byl představen dotazovací jazyk SQL a následně vysvětleny jeho hlavní příkazy.

V praktické části této práce byla za využití spirálové metodiky vývoje SW vyvinuta desktopová aplikace pro vedení tréninkového deníku sportovce v jazyce C#. Aplikace pro zaznamenání a následné uložení sportovní aktivity slouží pro triatlonisty a jsou zde zahrnuty tři hlavní disciplíny. Aplikace komunikuje s SQL databází vytvořenou pomocí služby Microsoft Azure. Aplikace byla následně otestována pomocí testovacích scénářů, které poslouží pro zlepšení aplikace. V rámci testování bylo odhaleno několik drobných nedostatků aplikace. Hlavní funkčnost aplikace po testování nevykazuje žádné závažné chyby.

Cílem této práce bylo vyvinutí desktopové aplikace, postavené na .NET Frameworku, pro vedení sportovního deníku triatletů. Bylo využito spojení technologie C# s databází Microsoft SQL Server.

6. Seznam použitých zdrojů

- [1] SHARP J., Microsoft Visual C# Step by Step (Developer Reference) 9th Edition
- [2] EEGH, J., GIANNINI, M. OOP Objektově orientované programování bez předchozích znalostí - Průvodce pro samouky, Computer Press, 2006, 224 s. ISBN: 80- 251-0973-9
- [3] RUPARELIA, Nayan B. Software development lifecycle models. ACM SIGSOFT Software Engineering Notes. 2010, 35(3). ISSN 01635948. Dostupné také z: <http://portal.acm.org/citation.cfm?doid=1764810.1764814>
- [4] BIRREL, N. D. and Ould, M.A. (1988): A practical handbook to software development; Cambridge University Press. ISBN 978-0521347921
- [5] BOEHM, BARRY W. (1986): A spiral model of software development and enhancement. In ACM SigSoft Software Engineering Notes, Vol. II, No. 4, 1986
- [6] MARTINŮ, Jiří a Petr ČERMÁK. METODIKY VÝVOJE SOFTWARE. Olomouc, 2018. Studijní opora pro kombinované studium. Moravská vysoká škola Olomouc.
- [7] HLAVA, Tomáš. Vodopádový model. Testování softwaru [online]. 2016 [cit. 2021-02-21]. Dostupné z: <http://testovanisoftwaru.cz/manualnitestovani/modely-zivotniho-cyklu-softwaru/vodopadovy-model/>
- [8] Vodopádový model (Waterfall model).Management Mania [online]. 2015 [cit. 2021-02-05]. Dostupné z: <https://managementmania.com/cs/vodopadovymodel-waterfall-model>
- [9] Introduction to Databases E-R Data Modeling [online]. [cit. 2021-08-01] Dostupné z: <http://labe.felk.cvut.cz/~stepan/AE3B33OSD/Lesson08-IntroDatabases.pdf>
- [10] BEAULIEU Alan, Learning SQL: Generate, Manipulate and Retrive data, 2020, ISBN: 978-1492057611
- [11] SHIELD Walter, SQL QuickStart Guide, 2019 ISBN: 978-1945051753
- [12] W3schools.com, SQL Tutorial, [cit. 2021-03-01] Dostupné z: (www.w3schools.com)
- [13] ORACLE, What is database? [cit. 2021-03-01] Dostupné z: (www.oracle.com/cz/database/what-is-database/)
- [14] PŠENIČKOVÁ Jana, Algritmizace, 2009 Computer media ISBN:978-80-7402-034-6
- [15] ČADA Ondřej, Objektové programování, Grada 2009, ISBN: 978-80-247-2745-5
- [16] VIRIUS Miroslav, Programování v C#, Grada 2020, ISBN: 978-80-271-1216-6

- [17] Microsoft SQL Server (MSSQL) vs. PostgreSQL Comparison in Details - [online], [2020] <https://www.enterprisedb.com/blog/microsoft-sql-server-mssql-vs-postgresql-comparison-details-what-differences>
- [18] Relační datový model [online] [cit. 2021-03-07] <http://wiki.cs.vsb.cz/images/2/2a/Inz2.pdf>
- [19] Databázové systémy [online] [cit. 2021-03-07] https://www.ped.muni.cz/wtech/03_studium/cvt4/databaze.pdf
- [20] .NET Framework [online] [cit. 2021-03-11] Dostupné z: <docs.microsoft.com/dotnet>
- [21] .NET Framework verze 5.0 - představení [online] [cit. 2021-03-11] Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/desktop/winforms/overview/?view=netdesktop-5.0>
- [22] Úvod do Windows Presentation Foundation [online] [cit. 2021-03-11] Dostupné z: <https://www.dotnetportal.cz/clanek/196/Uvod-do-Windows-Presentation-Foundation-WPF->
- [23] Historie .NET Framework [online] [cit. 2021-03-11] Dostupné z: <https://www.itnetwork.sk/csharp/historie-dotnet/?all-comments>
- [24] .NET Framework [online] [cit. 2021-03-11] Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/framework/get-started/overview>
- [25] .NET [online] Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/architecture/modernize-desktop/media/whats-new-dotnet-core/release-history.png>
- [26] .NET Framework [online] Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/thumb/d/d3/DotNet.svg/440px-DotNet.svg.png>
- [27] Windows Forms [online] Dostupné z: https://www.itnetwork.cz/images/5/csp/visual_studio_windows_forms.png
- [28] Windows Presentation Foundation [online] Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/thumb/d/d5/WPF.svg/440px-WPF.svg.png>
- [29] Prototypování [online] Dostupné z: https://www.itnetwork.cz/images/77422/rad_diagram.png

- [30] Model spirála [online] Dostupné z: https://upload.wikimedia.org/wikipedia/commons/thumb/a/a6/Software_Development_Spiral_cz.svg/220px_Software_Development_Spiral_cz.svg.png
- [31] Vodopádový model [online] Dostupné z: https://upload.wikimedia.org/wikipedia/commons/2/24/Vodopadovy_model.png
- [32] Relační datový model [online] Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/thumb/8/8d/Relational-model-concepts.png/360pxRelational-model-concepts.png>
- [33] Odlišnosti v syntaxi PostgreSQL a TSQL [online] Dostupné z: <https://www.enterprisedb.com/blog/microsoft-sql-server-mssql-vs-postgresql-comparison-details-whatdifferences>