VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

SYSTÉM PRO ORGANIZACI BĚŽECKÝCH ZÁVODŮ

A SYSTEM FOR ORGANIZING RUNNING RACES

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE — Bc. JIŘÍ KALUS
AUTHOR

VEDOUCÍ PRÁCE — RADEK BURGET, Ing., Ph.D.
SUPERVISOR

BRNO 2017

Abstrakt

Cílem tohoto semestrálního projektu je návrh studie technologií, analýza požadavků a návrh systému pro organizaci běžeckých závodů s podporou registrace závodníků a možností více online i offline stanovišť v průběhu závodu. Vzhledem k neexistujícímu současnému ekvivalentu systému je kladen důraz na vhodné zobrazení dat pro organizátory i závodníky. Práce obsahuje popis technologie, požadavků a návrhu systému. Mezi základní požadavky patří vytvoření a správa závodu zahrnující tvorbu kategorií, tras, položek v registračním formuláři, správa závodníků, samotná registrace závodníků a případné manuální zaznamenávání jejich časů či prostřednictvím RFID čipů v klientské aplikaci. Veškeré webové části systému budou implementovány technologií ASP.NET MVC v jazyce C#. Klientská desktopová část bude implementována technologií WPF opět v jazyce C#.

Abstract

The aim of this term project is to study technology, requirement analysis and system design for a system for organizing running races with the support of racers registration and possibility of multiple online and offline checkpoints during the race. Given the absence of any similar solution, emphasis is put on displaying data for organizers and racers. The project decribes technology, requirements and system design. The basic requirements include race inicialization and administration involving creation of categories, routes, items in the registration form, racers administration, racers registration and possible manual recording of their times or through the RFID chips in the client application. All web parts of the system will be implemented in ASP.NET MVC with language C#. Desktop client will be implemented in technology WPF also with language C#.

Klíčová slova

Organizace běžeckých závodů, offline i online stanoviště, registrace závodníků, tvorba kategorií a tras, vlastní registrační formulář, RFID čipy, ASP.NET MVC, MS.NET, WPF, C#.

Keywords

Organization running races, offline and online checkpoints, racers registration, creation of categories and routes, custom registration form, RFID chips, ASP.NET MVC, MS.NET, WPF, C#.

Citace

KALUS, Jiří. *Systém pro organizaci běžeckých závodů.* Brno, 2017. Semestrální projekt. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Burget Radek.

Systém pro organizaci běžeckých závodů

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením pana Ing.Radka Burgeta, Ph.D. Další informace mi poskytli organizátoři závodů pan Ing.Petr Volný a Ing.Lukáš Tomčík. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

. . . . . .
Jiří Kalus
10. ledna 2017

Obsah

1 Úvod **2**

2 Teorie pro vývoj informačního systému **3**
2.1 Charakteristika informačního systému . . . 3
2.2 Vybrané technologie . . . 4
2.2.1 Platforma .NET . . . 4
2.2.2 ASP.NET Framework . . . 5
2.2.3 Entity framework . . . 7
2.2.4 Windows Presentation Foundation . . . 8
2.2.5 Knihovna SQLite . . . 9
2.2.6 RFID technologie . . . 9
2.2.7 Sumarizace použitých technologií . . . 12

3 Analýza požadavků **13**
3.1 Definice požadavků . . . 13
3.2 Požadavky od webového rozhraní . . . 14
3.2.1 Kategorie závodníků . . . 14
3.2.2 Trasy závodu . . . 14
3.2.3 Registrační formulář . . . 14
3.2.4 Vyhodnocení a zanesení výsledků . . . 15
3.2.5 Platby organizátora a závodníků . . . 15
3.3 Požadavky od klientské aplikace . . . 15
3.3.1 Zanesení výsledků a jeho zobrazení . . . 15
3.3.2 Synchronizace výsledků . . . 16
3.4 Existující řešení a zařazení produktu na trhu . . . 16

4 Návrh řešení informačního systému pro organizaci závodu **17**
4.1 Schéma a popis jednotlivých elementů . . . 17
4.2 Případy užití systému . . . 18
4.3 Snímání RFID čipů . . . 22
4.4 Autorizace klientské aplikace . . . 24
4.5 Grafické rozložení webového a aplikačního rozhraní . . . 25

5 Závěr **27**

Literatura **28**

Kapitola 1

Úvod

Návrhem této práce je navrhnout a realizovat systém pro organizaci běžeckých závodů s podporou registrace závodníků a možností více online i offline stanovišť v průběhu závodu. Celý systém se skládá ze dvou webových a jednoho desktopového rozhraní. Administrační webové rozhraní slouží pro vytvoření a správu závodu. Prezenční webové rozhraní slouží pro prezentaci závodu veřejnosti, kde se mohou například registrovat závodníci. Desktopovým rozhraním je míněna aplikace pro zaznamenávání časů závodníků na jednotlivých stanovištích závodu. Zaznamenávání časů je provedeno manuálním vložením časů nebo sejmutím čipu závodníka.

Účelem systému je snadnější realizace typicky lokálních běžeckých závodů. Hlavní organizátor může využít systém pro vytvoření svého závodu, což zahrnuje například stanovení kategorií, tras, položek v registračním formuláři, pravidel či správu závodníků. Prezentační rozhraní nabízí detail závodů s důrazem na vhodné zobrazení výsledků závodníků. Primárním účelem klientské aplikace je kolekce časů závodníků bez závislosti na internetovém připojení. Zaznamenané časy jsou v ideálním případě synchronizovány se serverem a zobrazeny na webových rozhraních. Skelet systému je navrhnut pro závod Hostýnská osma, jenž je pravidelně pořádán ve Zlínském kraji.

Následující text seznámí čtenáře s vývojem výše popsaného systému. V úvodní kapitole práce lze nalézt základní informace o informačním systému a podrobnější popis použité technologii. Podrobnější analýza požadavků jednotlivých rozhraní včetně popisu existujících řešení je obsažena v kapitole Analýza požadavků. V kapitole Návrh řešení informačního systému pro organizaci závodů je čtenář seznámen s architekturou a detailnějším popisem vybraných částí systému jako je snímání čipů, autorizace, případy užití či aktualizace. Poslední dvě kapitoly Implementace navrženého systému a Testování jsou uveden jen pro přehlednost a budou doplněna v navazující diplomové práci.

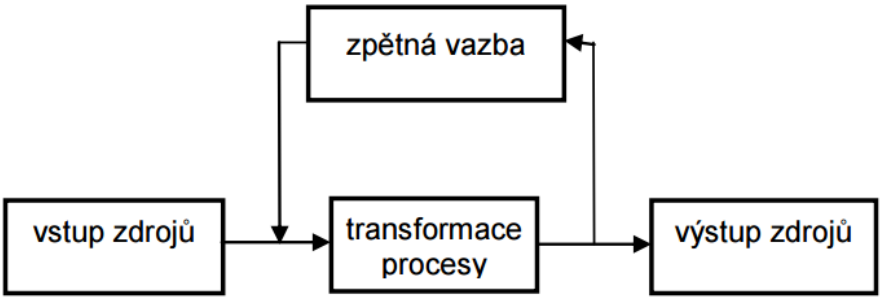
Kapitola 2

Teorie pro vývoj informačního systému

Obsahem kapitoly je analýza problému řešeného v této diplomové práci. Určíme si požadavky a hlavní cíl práce. Vytyčíme aktéry a základní funkcionalitu. Studí konkurenčního řešení se zabývá poslední sekce kapitoly.

2.1 Charakteristika informačního systému

Informační systém (IS) je soubor lidí, technických prostředků a metod (programů), zabezpečujících sběr, přenos, zpracování, uchování dat, za účelem prezentace informací pro potřeby uživatelů činných v systémech řízení. [3]. Informační systém, který je v interakci se svým okolím nazýváme otevřený IS. Takový systém je připojen k okolí oboustranným tokem zdrojů (vstup a výstup). Informační systém lze tedy chápat jako systém vzájemně propojených informací a procesů, které s těmito informacemi pracují. Procesy jsou typicky reprezentovány funkcemi, které zpracovávají informace do systému vstupující a transformují je na informace ze systému vystupující. Transformace je realizována na základě různých algoritmů, často ve formě transakcí. Aktuální uložený stav je reprezentován zpětnou vazbou. Odpovídající schéma výše popsaného systému je uvedeno níže:



Obrázek 2.1: Schéma informačního systému [14]

Informační systémy obvykle modelují skutečné zdroje jiného fyzického systému. Můžeme si IS představit jako virtuální obraz, který modeluje svůj fyzický vzor, pro který je vytvářen. Vždy se jedná o virtuální obraz na vhodné úrovni abstrakce, protože model nikdy nemůže postihnout veškeré chování a vlastnosti svého vzoru.

Model a modelovací prostředí slouží k modelování dat a procesů. Modelovací prostředí je obvykle zastoupeno některým z programovacích jazyků. Při určování modelu se musíme zabývat persistencí (v čase přetrvávající) a konzistencí, což je splňování určených pravidel o možných kombinacích hodnot údajů ve stavu. Tyto dva základní požadavky splňuje relační model dat, což nejčastější typ modelu informačního systému [14].

Neméně důležitou část IS je komunikace, jenž probíhá pomocí kontaktních bodů. Nejčastěji jsou to kontaktní obrazovky zařízení ovládané lidmi (tablet, počítač, mobil a další), čidla nebo různá měřící zařízení. Kontaktní body závisí na fyzickém systému, který IS reprezentuje. Nejvíce používaným komunikačním kanálem je Internet.

Často opomíjenou, avšak velmi podstatnou částí je koncová reprezentace dat (výstup) pro uživatele. Často je nezbytné zobrazit složitý stav systému jednoduchou a srozumitelnou formou.

2.2 Vybrané technologie

V této sekci jsou shrnuty technologie pro vývoj informačního systému a technologie pro vývoj desktopové aplikace.

2.2.1 Platforma .NET

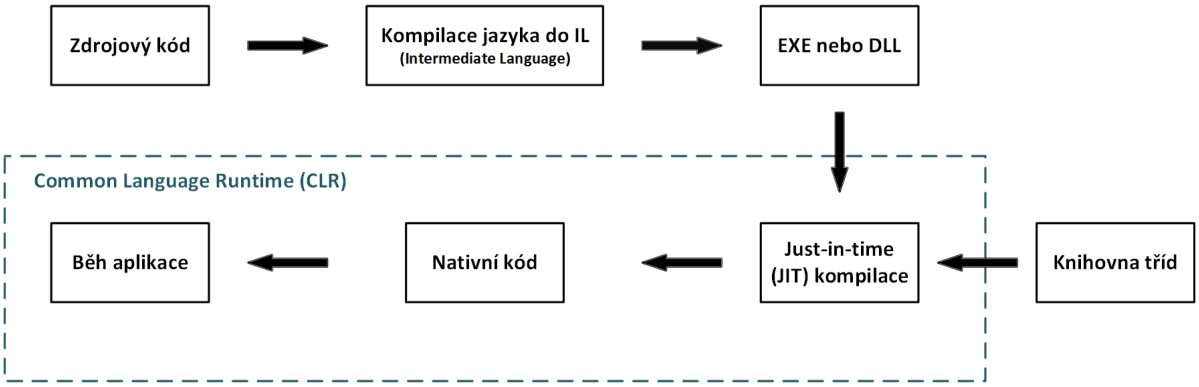
Platforma .NET [9] je složena ze dvou hlavních částí. První částí je běhové prostředí Common Language Runtime (CLR) pro vykonávání kódu. Druhou částí je knihovna tříd Framework Class Library (CLS), což je knihovna tříd a rozhraní pro přístup k funkcím systému.

Knihovna CLS je základní pilíř, na kterém jsou postaveny aplikace platformy .NET, komponenty i ovládací prvky. Jsou zde knihovny jako WPF, Windows Forms, ASP.NET a další.

Prostředí CLR je virtuální stroj, ve kterém probíhá překlad do nativního jazyka. Součástí prostředí je správa paměti, bezpečnost, správa výjimek, vláken a mnoho další. CLR je obdoba virtuálního stroje při použití programovacího jazyka Java. Zdrojové kódy tedy nejsou kompilovány přímo do nativního jazyka, ale do intermediálního jazyka (Intermediate Language - IL). Jazyk C# kódu, ve kterém je tato práce implementována, je v počáteční fázi kompilován do tzv. managed code, který je součástí assembly. Výsledek první fáze překladu je knihovna s příponout `.dll` nebo spustitelný kód s příponou `.exe`. V případě požadavku na spuštění managed code systém detekuje, že se jedná o objekt zkompilovaný do jazyka IL a spustí Just-In-Time (JIT) kompilátor. JIT kompilátor vygeneruje skutečné instrukce cílové platformy. Průběh překladu je zobrazen na obrázku níže:

Důležitým prvkem CLR je podpora společného typového prostoru (Common Type System - CLS). CLS vymezuje deklaraci, použití a správu typů. Poskytuje objektově orientovaný model, definuje pravidla, které zaručí interakci objektů napsaných v odlišných jazycích a poskytuje také knihovnu primitivních typů jako například Boolean, Char, Byte, Int32 a další. Níže je uveden seznam základních součástí platformy .NET:

- ASP.NET - serverová webová platforma pro vývoj a provozování webových aplikací.
- ADO.NET - množina tříd pro přístup k datům a tvorbu databázových aplikací.
- LINQ - technologie pro unifikované dotazování nad různými daty (databázové rozhraní, kolekce, XML, atd.).



Obrázek 2.2: CLR - kompilace a spuštění

- Windows Forms - starší knihovna tříd pro tvorbu grafického rozhraní desktopových aplikací.
- WPF - novější knihovna tříd pro tvorbu grafického rozhraní desktopových aplikací.
- WCF - framework pro komunikaci mezi aplikacemi a vytváření servisně orientovaných aplikací.

Primární prostředí pro běh platformy .NET je prostředí Windows. V současné době je však možné i v modulární verzi .NET Core, který je podporován na operačním systému Linux, Windows i Max OsX. Dalším z multiplatformním projektů je Mono, který nepoužívá běhové prostředí CLR, ale má vlastní kompilátor. Například prostředí Xamarin umožňující vývoj mobilních aplikací pro Android i iOS, podporující jazyky z rodiny .NET, je založen na prostředí Mono.

2.2.2 ASP.NET Framework

Dle oficiálního popisu [9] je jádrem platformy ASP.NET pro webové aplikace je právě platforma .NET 2.2.1. ASP. NET je serverová webová platforma [5] určená pro vývoj a provozování webových aplikací, webových služeb a webových stránek. Jedná se o nástupce technologie Active Server Pages (ASP). Základním elementem je .NET Framework, který vytváří prostředí pro běh webových aplikací a nabízí potřebné knihovny. Firma Microsoft, jenž je vyvíjí tento framework, zpřístupnila jeho zdrojový kód, čímž se stal veřejně dostupný (open source). Pro vývoj webového produktu je možné zvolit takový jazyk, který je kompatibilní s Common Language Runtime (CLR 2.2.1) prostředím, což je například: C#, VisualBasic či C++. ASP.NET disponuje následujícími aplikačními rámci:

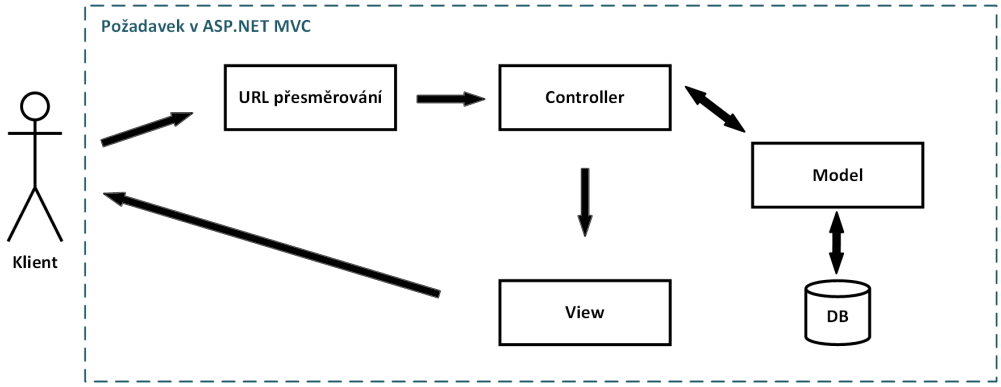
- WebForms - technologie založená na modelu řízeném událostmi. Webová aplikace je tvořena ovládacími prvky a jejich údálostmi. Logika vývoje webové aplikace je podobná technologii Windows Forms 2.2.4.
- WebPages - alternativa skriptovacích jazyků jako je například PHP. Tato technologie kombinuje HTML kód se serverovým jazyk (kompatibilním s CLR).
- MVC - technologie pro vývoj webové aplikace respektující návrhový vzor MVC. Detailnější popis je uveden níže 2.2.2 v této kapitole.

- Single Page Application - celá webová aplikace je tvořena jednou stránkou. SPA minimalizuje použití serveru a kladu důraz na klientskou část, která je tvořena jazykem Javascript. Server je implementován prostřednictvím Wep API [1]

Webová aplikace této diplomové práce je vyvíjena aplikačním rámcem ASP. NET MVC, který je podrobněji popsán níže 2.2.2.

ASP. NET MVC

Platforma ASP.NET MVC [4] je součástí kolekce ASP.NET. Tato platforma implementuje architektonický vzor MVC, jenž se skládá ze tří logických částí: Model-View-Controller. Jednotlivé části spolu se schématem jsou podrobně popsány níže:



Obrázek 2.3: Zpracování požadavku v ASP.NET MVC architektuře

- Model - objekty v této části implementují logiku pro aplikační datovou doménu. Můžeme zde zahrnout doménový model, operace nad daty, validaci, interakci s databází a mnoho dalšího. Fundamentální funkce modelu je přijetí požadavku a na základě tohoto požadavku vrátit nebo uložit data. Propojení s databází typicky zajišťuje další vrstva, v našem případě Entity Framework.

- View - česky pohled, je část, která zobrazuje výstup uživateli. Tato část neslouží jen k zobrazení uživatelského rozhraní, ale zároveň s uživatelem interaguje. Uživatelovy příkazy přeposílá na logickou část jménem Controller, který příkaz zpracuje a vrátí výsledek uživateli. Pohled je nejčastěji reprezentován HTML stránkou, do které lze vkládat příkazy v jiném jazyce (například C#). Stránka je zkompilována na serveru a vrácena ve standardním tvaru, jehož zobrazení podporuje webový prohlížeč.

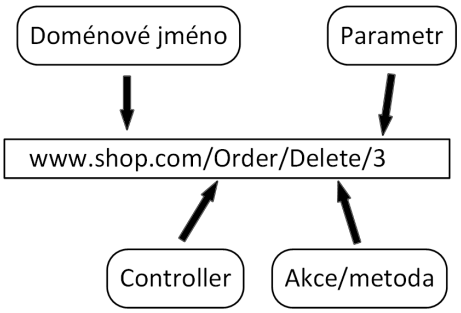
- Controller - česky ovladač, je komponenta, která přijímá uživatelské vstupy přeposlané z view a pracuje v modelem. Je to tedy mezivrstva mezi view a modelem. Detailněji informace o této logické části jsou uvedeny níže.

Controller analýza

Fungování interakce mezi View částí a Controller částí nemusí být na první pohled zřejmá. Implementované metody této třídy jsou nazvány akce (Actions). Pro vykreslení webové

[1] Web API - REST rozhraní pro komunikace a zpracování dat v serializačním jazyce XML nebo JSON.

stránky nejčastěji View předá požadavek vrstvě Controller, ve kterém je uveden název pro Controller, akce a parametry. Zaměřme se na požadavek, který je i s implementací, uveden níže:




```
public class OrderController : Controller
{
        ...
        public ActionResult Delete(int id)
        {
                ...
                return View();
        }
}
```

Obrázek 2.4: Rozložení adresy a implementace v ASP.NET MVC architektuře

První částí je doménové jméno webového serveru. Druhá část za lomítkem je Controller název, přičemž třetí část je název metody/akce, která se má zavolat. Konečně poslední částí je parametr identifikující určitý objekt. Webový server pomocí reflexe nalezne odpovídající Controller a jeho akci, ve které předá identifikující parametr.

Základní prvkem pro efektivní manipulaci zajišťuje model binding. Tato funkce umožňuje interně konvertovat parametry na objekty. V praxi tak můžeme pracovat se stejnými objekty v sekci View i Controller. Rozpoznání akce, která má být zavolána je především na základě názvu akce, počtu a typu parametrů nebo anotacích.

Platforma ASP.NET MVC nabízí rozsáhlou funkcionalitu a výše zmíněné aspekty patří mezi nejzákladnější. Tento typ architektury poskytuje rozdělení hlavní logiky aplikace a zároveň nabízí relativně nezávislé propojení mezi logickými modely. Benefitem je nesporně přehlednější zdrojový kód u rozsáhlejších aplikací. Snadnější je tak vývoj i údržba a díky odlučitelnosti logických částí i testování.

2.2.3 Entity framework

Pro práci s databází je možné využít technologii Entity framework (EF). EF [7] je open-source a zajišťuje objektově relační mapování. To umožňuje vývojářům komunikovat jen s modelem za použití známých objektově orientovaných technik. Manipulace s daty je tak jednodušší, protože EF převede dotaz z objektového rozhraní (typicky dotaz za pomocí technologie LINQ) na dotaz nad relační databází. V rámci EF můžeme u objektově orientovaných aplikací zvolit dva přístupy:

- Code first - na základě objektů EF vygeneruje databázové schéma a potřebný kontext.
- Database first - na základě databázového schématu EF vygeneruje odpovídající třídy v modelu aplikace.

V rámci této diplomové práce je použit Entity Framework s Code first přístupem.

2.2.4 Windows Presentation Foundation

Windows Presentation Foundation (WPF) [8] je platforma pro komplexní tvorbu rozsáhlých formulářových aplikací, která je součástí .NET platformy od verze 3.0. WPF je logický nástupce technologie Windows Forms. Tato technologie je však dnes stále používána vývojáři a podporována firmou Microsoft.

WPF zavádí nezávislou jednotku délky tzn. DIP (Device Independent Pixel) a všechny elementy vytváří vektorově. Díky těmto dvěma aspektům je výsledná aplikace nezávislá na rozlišení zobrazovacího zařízení. Původní grafické rozhraní Graphics Device Interface(GDI) [11] ve Windows Forms je nahrazeno a pro vykreslování grafického rozhraní je využit DirectX. V důsledku akcelerované grafiky má aplikace menší výpočetní režii na procesor a aplikace je svižnější.

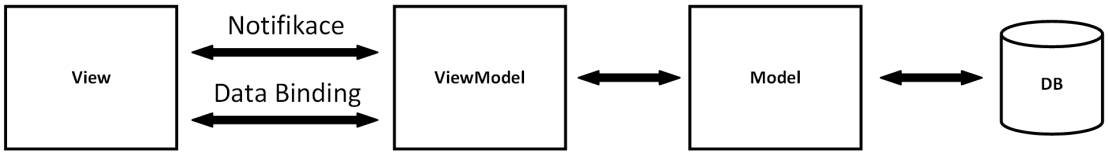
Významná technologie, jenž je součástí WPF, je Data binding [6]. Data binding umožňuje navázání a synchronizaci dat mezi zdrojem dat (modelem) a grafických uživatelským rozhraním. V rámci WPF technologie je možné rozlišit 4 módy a synchronizace dat:

- One way - uživatelské rozhraní se pouze synchronizuje se změnami hodnot ve zdroji dat. Hodnoty tedy nelze z uživatelského rozhraní modifikovat.
- Two way - uživatelské rozhraní a hodnoty ve zdroji dat jsou synchronizovány navzájem. Změní-li uživatel hodnotu v uživatelském rozhraní, pak se změní i hodnota ve zdroji dat a naopak.
- One time - Uživatelské rozhraní přečte hodnotu ve zdroji dat pouze jednou a poté jsou z pohledu uživatelského rozhraní veškeré změny ve zdroji dat ignorovány.
- One way to source - jedná se o protiklad módu One way. Pouze uživatelskému rozhraní je povoleno modifikovat hodnotu ve zdroji dat.

Pro uživatelské rozhraní je ve WPF používán jazyk XAML, který je odvozený od značkovacího jazyka XML. Všechny prvky, které definujeme v jazyce XAML, lze zapsat i v jazyce C# či Visual Basic. V kombinaci s Data binding a událostmi lze docílit oddělení implementace uživatelského rozhraní od modelu dat, což vede k možnosti poměrně nezávislého vývoje obou vrstev.

MVVM vzor

Model-View-ViewModel [2] je návrhový vzor pro WPF aplikace, jehož hlavním cílem je oddělit logiku aplikace od uživatelského rozhraní. Tento návrhový vzor vychází z principů vzoru MVC a navrhl jej softwarový architekt John Gossman. Jednotlivé logické části vzoru jsou společně se schématem popsány níže:



Obrázek 2.5: Schéma návrhového vzoru MVVM pro WPF aplikace

- View - reprezentuje uživatelské rozhraní. Definice toho, co se zobrazí uživateli. Všechny ovládací prvky jsou přes DataContext (zdroj dat) namapované na vlastnosti ve ViewModel. Tato vrstva také obsluhuje efekty a animace.
- Model - popisuje data, jejich logiku a případně validaci, se kterými aplikace pracuje. Typicky jsou to data z externího zdroje (databáze, webová služba a další). Model tedy neobsahuje popis práce s daty, ani způsob jejich zobrazování uživateli.
- ViewModel - logická vrstva mezi View a Model. Představuje prostředníka, který adaptuje Model pro potřeby View. Vrstva ViewModel je zodpovědná za stav View avšak bez přímé interakce mezi jednotlivými prvky ve View. Ovládací prvky jsou pomocí Data binding [6] propojeny a synchronizovány s ViewModel. Pro validní propojení vrstev musí být implementováno rozhraní INotifyPropertyChanged, které zajistí propagaci změn hodnot.

Klientská aplikace této diplomové práce využívá návrhový vzor MVVM.

2.2.5 Knihovna SQLite

SQLite je knihovna menšího rozsahu [12], která je napsána v jazyce C a obsahuje téměř celý relační databázový systém. Hlavním rysem je přímé připojení této knihovny k aplikaci, nikoliv oddělený běh, jak je to typické u architektury klient-server. V praxi je databáze tvořena jedním souborem , nad kterým lze provádět relační databázové dotazy. Podpora standardu SQL-92 není kompletní a některé funkce nejsou implementovány [13]. Celá knihovna je distribuována pod licencí public domain a podporována ve většině nejpoužívanějších programovacích jazycích (C#, C++, Java, C, Python, PHP a další). Výhodou je lokální nenáročná instalace a použití, čímž se knihovna stala jedním ze standardů pro jednoduchou lokální databázi. Jako zástupce použití můžeme jmenovat prohlížeče Chrome a Safari, komunikátor Skype nebo velké množství aplikací pod operačním systémem Android.

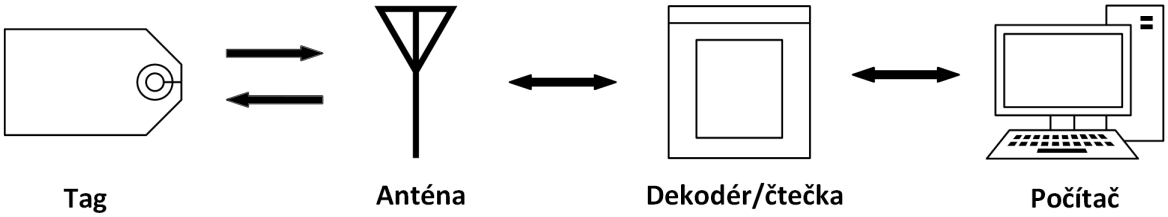
2.2.6 RFID technologie

Radio Frequency Identification (RFID) [10] je obecně používaný pojem pro identifikaci objektů pomocí elektromagnetických vln na rádiové frekvenci. Informace je uložena na mikročipu, který je připojen k anténě a zalit do substrátu. Dohromady tyto složky tvoří tzv. RFID tag, který je schopen uloženou informaci vyslat do čtečky. Vysílač čtečky vysílá elektromagnetické vlny. Anténa RFID tagu přijme signál a usměrněný elektrický proud dobije kondenzátor. Výsledkem je odeslání informace zpět do čtečky. Nejběžnější informací je unikátní standardizovaný 96-bitový kód - Electronic Produkt Code (EPC). Čtečka přemění vyslané radiové vlny přijaté od RFID tagu na formu, která může být dále zpracována.

Na obrázku výše je uveden základní princip komunikace RFID tagu s řídícím počítačem. Hardwarová realizace může být však různá a často se můžeme setkat s případy, kdy je anténa součástí čtečky.

Frekvence

Systém RFID podporuje několik frekvencí [15]. Frekvence silně ovlivňuje vlastnosti čipu (dosah, rychlost čtení a zápisu, použitelnost v různém prostředí a podobně), a proto je nutné ji stanovit již v návrhu systému. Frekvenci ovlivňuje typ a velikost antény RFID tagu. S rostoucí frekvencí jsou typicky potřeba menší antény, ovšem s vyšší frekvencí narážíme



Obrázek 2.6: Princip komunikace RFID technologie

na problém s elektromagnetickým rušením a vzniká tak problém především se snímáním z kovových materiálů a tekutin. Přehled základních frekvencí a jejich využití je uveden výčtem níže:

- **Nízkofrekvenční (LF) 125 KHz a 135 KHz**
  - Rychlost a množství dat: malá rychlost čtení a malé množství dat
  - Čtecí vzdálenost: Krátká, do 0,5 m
  - Využití: kontrola přístupu, identifikace a sledování zvířat, imobilizéry automobilů, inventura
  - Výhody: větší odolnost proti rušení, možnost upevnění v blízkosti vody, možnost upevnění na kovové podložce
  - Nevýhody: malý čtecí dosah, malá komunikační rychlost
- **Vysokofrekvenční (HF) 13.56 MHz**
  - Rychlost a množství dat: malá rychlost čtení a malé množství dat
  - Čtecí vzdálenost: Středí, do 1 m
  - Využití: chytré karty (Smart Cards), bezkontaktní placení, chytré etikety (Smart Labels), označování zavazadel při přepravě, záznam a přenos naměřených dat
  - Výhody: menší rozměry antény, větší komunikační rychlost než LF, větší čtecí dosah než LF, nízká cena RFID tagu - nejvíce rozšířené
  - Nevýhody: kovové podložky a voda již významně snižují čtecí dosah a ruší komunikaci
- **Ultrafrekvenční (UHF) 860 MHz až 960 MHz**
  - Rychlost a množství dat: vysoká rychlost čtení a malé až střední množství dat
  - Čtecí vzdálenost: do 3 m
  - Využití: sledování palet při přepravě a ve skladech, současná identifikace více zabalených produktů, elektronické mýtné, parkovací karty
  - Výhody: možnost i vzdáleného čtení = indentifikace průjezdem brány, velká přenosová rychlost = možná větší kapacita paměti RFID tagu, dipólová anténa, levná výroba
  - Nevýhody: nečitelnost přes kapaliny, obtížné čtení na kovových podložkách celosvětově nejednotná frekvence

- **Mikrovlná 2.45 GHz a 5.8 GHz**
  - Rychlost a množství dat: vysoká rychlost čtení a střední množství dat
  - Čtecí vzdálenost: do 2 m
  - Využití: elektronické mýtné, identifikace zavazadel při letecké přepravě, bezdrátový záznam a přenos dat v reálném čase
  - Výhody: vysoká přenosová rychlost až 2 Mb/s, malé rozměry dipólové antény = malé tagy
  - Nevýhody: drahá a složitá konstrukce, menší dosah než UHF, velký vliv rušení (kovu, kapalin apod.)

RFID tag

Komunikace mezi RFID tagem a čtečkou probíhá pomocí elektromagnetických vln na rádiové frekvenci [15], které jsou často využívány k napájení RFID tagu. Čtečka (vysílač) periodicky vysílá pulsy do okolí a čeká na možné odpovědi do RFID tagu (přijímač), který může být vzdálen až 3 m. V takovém případě je jedná o systém s pasivním RFID tagem. Pasivní RFID tagy jsou rozšířené pro identifikaci produktů nebo osob. Jejich výhodou je nízká pořizovací cena.

Doplněním miniaturní baterie pro napájení čipu razantně zvýšíme jeho čtecí dosah a přidáme schopnost vysílat své data a informace. V takovém případě je jedná o systém s aktivním RFID tagem. Aktivním tagům může být doplněna funkcionalita o integrovaný senzor tlaku, teploty, lokaci a podobně. Využití je například při sledování pohybu osob nebo přepravování chlazených výrobků, kdy čip vysílá periodicky data a je možné sledovat veškeré změny. Nevýhodou je vyšší cena a omezená životnost (obvykle několik let).

V současné době existují ještě semipasivní RFID tagy, které jsou doplněny jen o minibaterii, která zvýší dosah pasivního RFID tagu. RFID tagy také můžeme rozdělit podle manipulace s daty. Existují RFID tagy pouze pro čtení, pro čtení i zápis a pro jeden zápis a mnoho čtení. Poslední typ funguje tak, že uživatel nadefinuje, jaké informace mají být na čip umístěny a perzistentně je na čip zapíše.

RFID technologie a čárové kódy

Technologie RFID je další generace systému pro identifikaci objektů navazující na systém čárových kódů. Mezi přední výhody technologie patří možnost aktualizace uložených dat, není nutná přímá viditelnost, vyšší odolnost vůčí okolí, vyšší kapacita a možnost mnohačetného snímání v jednom okamžiku. V praxi je výhodné použít kombinaci RFID technologie a technologii čárových kódu. Příklad může být tzv. SmartLabel. SmartLabel je dvouvrstvá etiketa. Vrchní vrstva je čárový kód a spodní vrstva je RFID tag. Takové řešení se využívá například na oblečení v obchodech s oděvem.

Využití RFID u běžeckých závodů

Čipy (RFID tagy)jsou velmi rozšířené na sportovních akcích, především běžeckých závodech s větším počtem účastníků. U takových závodů typicky není vhodné používat manuální stopky, ale čipy, které mají závodníci připevněné na sobě. Mezi dva nejčastější případy uchycení patří integrace čipu přímo do závodníkova čísla nebo je čip zapuštěn do umělohmotného náramku.

U časových bran je umístěno čtecí zařízení, které snímá procházející závodníky. Kvalitnější a finančně náročnější je průchozí brána se zabudovanými čtecími zařízeními. Obvykle se jedná o vysokofrekvenční či ultrafrekvenční řešení. Závodník bez jakékoliv interakce projde branou a jeho čas se zaznamená. Další možností je použití levnější, nízkofrekvenční variantu. U některých, převážně dlouhých vytrvalostních přechodů, není přesnost na sekundy vyžadována. Závodník na časových branách přistoupí přímo ke čtecímu zařízení a na blízkou vzdálenost dojde k sejmutí čipu.

Tato práce je navrhnuta pro druhou, nízkofrekvenční variantu. Závodníci mají připevněný čip formou náramku na ruce a jednotlivě přistupují ke čtecím zařízením. Pro zefektivnění snímání je na jedné časové bráně umístěno několik čtecích zařízení.

2.2.7 Sumarizace použitých technologií

Webový informační systém a prezenční rozhraní využívá technologii ASP.NET MVC. Pro komunikaci s databází je použit Entity framework. Jako relační databáze byla zvolena databáze MS-SQL. Pro vzhled administračního rozhraní informačního systému i prezenční rozhraní byl využit Bootstrap framework. Klient pro stolní počítače využívá technologii WPF, kdy pro code-behind je zvolen jazyk C#a pro vzhled jazyk XAML. Lokální databáze je realizována pomocí knihovny SQLite.

Kapitola 3

Analýza požadavků

V této kapitole jsou shrnuty požadavky na systém pro organizaci závodů. Nejprve jsou uvedeny požadavky na webové rozhraní a poté na rozhraní klientské aplikace. Na konci kapitoly jsou shrnuty existující řešení.

3.1 Definice požadavků

Tento semestrální projekt se zabývá realizací webového informačního systému pro organizaci běžeckých závodů, přičemž součástí řešení je i klientská aplikace. Systém nejprve umožní vytvoření závodu hlavním organizátorem a poté je nakonfigurována klientská aplikace pro snímání výsledků. Běžecké závody mají různou, často unikátní organizaci a je nutné vytvořit systém pro univerzální správu závodů. Časy jednotlivých závodníků jsou typicky u dlouhých běžeckých závodů zaznamenávány v průběhu a je vhodné tyto výsledky zobrazovat v reálném čase ostatním uživatelům pomocí webového rozhraní.

Zaměříme-li se na hlavní požadavky při vytváření závodu, pak by každý závod měl mít určené kategorie a trasy pro závodníky. Mezi konkrétní požadavky, založené na závodě Hostýnská osma [1], této práce patří možnost vytvoření časových bran (stanovišť) na trase, kde se provede časová kontrola závodníků. V případě zájmu organizátora jsou závodníkům rozdistribuovány fyzické RFID čipy, jenž jsou na časových branách snímány. Časové brány jsou rozmístěné podél trasy, přičemž tato místa nemusí být nutně pokryta mobilním či jiným připojením k Internetu. Je tedy nezbytné, aby čipy mohly být snímány bez připojení k Internetu a po při připojení synchronizovány. Snímání čipů a synchronizaci bude mít na starosti klientská aplikace.

Po úspěšném vytvoření závodu nabídne webové rozhraní správu novinek a propozic. V momentě, kdy organizátor považuje závod za dostatečně zkompletovaný, je možné jej zveřejnit. Zveřejněním dochází ke zpřístupnění závodu běžných uživatelům, kteří se mohou v daném termínu registrovat a stát se závodníky. Závodníci si později mohou zobrazit výsledky na webovém rozhraní. Organizátor tyto závodníky spravuje a v případě nutnosti jim zasílá zprávy.

Platba za registraci je v plné moci organizátora a systém za ni není nikterak zodpovědný. Do stanového termínu po ukončení závodu je organizátor povinen zaplatit za závod pomocí platební brány, kterou bude systém disponovat. V souhrnu musí systém umožnit webovou administraci závodu, webové rozhraní pro registraci závodníků a klientskou aplikaci pro snímání čipů. Všechny zmíněné části jsou podrobněji popsány v této kapitole.

[1] http://www.hostynskaosma.cz/

3.2 Požadavky od webového rozhraní

Webové rozhraní slouží k administraci a prezentaci závodu. Organizátor nejprve založí závod, přičemž uvede základní propozice (název, místo konání, limit závodníků, startovné a další) a poté pokračuje definicí několika klíčových sekcí. Mezi klíčové sekce patří kategorie, trasy a registrační formulář pro případné závodníky. Po zveřejnění závodu je možná registrace závodníků. Každý zaregistrovaný závodník má možnost vyhledat všechny své závody.

3.2.1 Kategorie závodníků

Kategorie slouží k rozčlenění závodníků do skupin dle požadavků organizátora. Systém umožní vytvoření dvou základních typů kategorií, a to kategorie pro jednotlivce a kategorie pro týmy. V rámci každé kategorie je nezbytné uvést její název, určit akceptované pohlaví (muž/žena) a volitelně i věkové omezení. V týmu je vyjma zmíněných údajů kategorie pro jednotlivce také nutné uvést maximální počet členů, přičemž u pohlaví je možné zvolit mix (tým může být složen z obou pohlaví).

Ke kategorii je typicky přiřazena trasa 3.2.2. Asociace s trasou však není povinná, neboť existují závody, u kterých není trasa vyžadována. Například orientační běh, kde je trasa tvořena jen body, které jsou skryté a organizátor si je nepřeje zobrazovat.

3.2.2 Trasy závodu

Trasa definuje běžeckou dráhu závodníků. Organizátorovi je umožněno vytvořit libovolné množství na sobě nezávislých tras. Trasa je vytvářena na předem určeném mapovém podkladu. Kromě povinných bodů na mapě start a cíl je možné vložit i časovou bránu nebo občerstvení.

Časová brána je typ bodu na mapě, na kterém bude probíhat snímání výsledků závodníků. Dalším typem bodu je občerstvení, kde závodníci mají zázemí a mohou se občerstvit. Při vytváření dvou a více tras musí systém umožnit znovupoužití již vytvořených bodů. Častým stavem je topologie tras se společným startem, cílem a některými význačnými body (časová brána, občerstvení). Pořadí časových bran i občerstvení je dáno posloupnosti s jakou byly body vytvořeny na mapě. U každého bodu lze uvést jeho vzdálenost a výškové převýšení od startu závodu. Žádoucí je také uvedení celkové vzdálenosti a výškového převýšení pro celou trasu.

V případě menších či atypických závodů je nezbytná možnost trasy bez mapového podkladu.

3.2.3 Registrační formulář

Registrační formulář je velmi důležitá část při vytváření závodu. Organizátor si může v administraci nakonfigurovat podobu formuláře dle svých požadavků. Nakonfigurovaný formulář poté vyplňují uživatelé při registraci. Mezi hlavní položky patří jméno, příjmení, ročník, pohlaví, a kontakt. Položka ve formuláři může nabývat různého datového typu. Prvním typem je obyčejný text nebo textové pole. Dalším číslem je celé nebo desetinné číslo. Výlučný typ (ano/ne) je předposledním možným typem. Konečně posledním typem jsou možnosti, kdy organizátor nadefinuje několik možností a uživatel poté vybírá právě jednu z nich. Organizátor si u vytvořených položek stanovuje, zda je vstupní údaj povinný či volitelný. Některé položky mohou být chápany jako doplňkové služby/produkty, a proto by mělo být možné uvést u každé položky i její cenu.

3.2.4 Vyhodnocení a zanesení výsledků

Výsledky závodníků jsou implicitně seřazeny dle času, kdy nejrychlejší čas je ten nejlepší. Závodník si může pomocí grafického i tabulkového zobrazení porovnat svůj výsledek až s dvěma soupeři a vidět tak průběh závodu v závislosti na čase. Každý výsledek se započítává do osobní statistiky. Osobní statistika zobrazuje uživateli celkový počet absolvovaných výškových i pozemních metrů, a to napříč všemi závody. Systém také umožňuje shlukování výsledků. Výsledkem této funkcionality je například možnost vytvoření skupiny dvou a více stejných, avšak časově oddělených, závodů. Závodník tak vidí všechny výkonnostní změny v daných závodech o rozdílných ročnících.

Před startem má každý závodník status „neodstartoval“. Úspěšný průběh je poté status „závodí“ a nakonec „dokončil“. Má-li na konci závodu kdokoliv status „neodstartoval“ či „diskvalifikován“, pak je to indikace neúspěšného dokončení závodu a jeho výsledek není brán v potaz.

3.2.5 Platby organizátora a závodníků

Organizátor má finance svého závodu plně pod vlastní kontrolou a systém do nich nikterak nezasahuje. Systém po registraci závodníka vygeneruje a zašle email s platebními instrukcemi. Platebními instrukcemi míníme číslo účtu, dobu splatnosti, částku a podobně. Tyto údaje spravuje organizátor v administrační sekci. Po zaplacení může organizátor změnit finanční status závodníka a tím mít přehled o zaplacených a nezaplacených pohledávkách.

Naopak platba organizátora je v přímé interakci se systémem. V závislosti na počtu závodníků a použité funkcionalitě je organizátorovi vygenerována platební poptávka. Platba je realizována pomocí platební brány nebo jinou formou, která vyžaduje přímý kontakt pověřené osoby. Pokud nedojde k řádnému zaplacení, pak budou výsledky závodníků skryty.

3.3 Požadavky od klientské aplikace

Základní funkcionalita klientské aplikace je snímání čipů během závodu. Aplikace musí být zabezpečena proti nežádoucímu použití a její činnost nesmí být závislá na internetovém připojení. Přístup k aplikaci je povolen pouze organizátorem pověřeným osobám.

3.3.1 Zanesení výsledků a jeho zobrazení

Časy závodníků jsou snímány na časových branách 3.2.2. Uživatel tedy nejprve vybere konkrétní bránu a poté započne samotné zanesení časů závodníka. Časy mohou být zaneseny do aplikace dvěma způsoby. První způsob je standardní vložení závodníkova čísla do aplikace. Závodníkovo číslo je napsáno organizátorem a uloženo. Proces uložení se skládá z uložení aktuálního času a výpočtu celkového pořadí včetně ztráty na prvního závodníka. Současně se aplikace pokusí odeslat výsledek na webové rozhraní. Druhým způsobem zanesení času je snímání pomocí RFID čipů. Každý závodník obdrží při akreditaci svůj RFID čip. Časová brána je vybavena bezkontaktní čtečkou. Čip je typicky fyzickou realizací podobný hodinkám. Závodník přiloží čip ke čtečce a dojde k sejmutí čipu. Aplikace sejmutí čipu detekuje, přečte jeho přijatá data a pokračuje ve stejné posloupnosti úkonů jako ve standardním případě. Tedy výpočtem a zasláním času na webové rozhraní. Tyto dva zmíněné způsoby jsou na sobě nezávislé a lze mezi nimi přepínat.

3.3.2 Synchronizace výsledků

Pro úspěšnou činnost aplikace je nutné ji alespoň jednou připojit k Internetu. V době průběhu závodu musí být v aplikaci staženy základní propozice závodu, především identifikátor závodu a časové brány. Změna údajů závodu v administračním rozhraní by měla být v co nejkratším čase reflektována v aplikaci. Časy nově zaregistrovaných závodníků, kteří nejsou synchronizováni s aplikací, mohou být zaznamenány a korektně přeposlány na webové rozhraní.

Po úvodní inicializaci závodu může aplikace fungovat bez připojení k Internetu. Časy závodníků jsou zaznamenávány a uloženy. Při opětovném připojení k internetu proběhne synchronizace a neodeslané časy závodníků jsou odeslány.

Primární čas je čas na webovém rozhraní. Každý klient však může mít odlišný čas. Organizátor proto musí provést manuální synchronizaci časů s primárním časem. Pokud aplikace detekuje rozdílný čas, pak zobrazí výzvu k manuální synchronizaci.

Všechny zaznamenané časy jsou paralelně ukládány do souboru. Obsah souboru je záloha výsledků pro uživatele aplikace a může být kdykoliv znovu nahrán. Takto nahrané časy lze opět odeslat na webové rozhraní.

3.4 Existující řešení a zařazení produktu na trhu

V současné době existuje několik společností na měření a správu výsledků, avšak žádná nenabízí vytvoření a správu samotného závodu. Dle mého názoru patří mezi přední zástupce portály Sport-base, Sportsoft timing a Championship. Všichni tři zástupci nabízí profesionální měření časů závodníků. Výsledky jsou typicky převedeny do PDF nebo jednoduše zformátovány pro snadné vložení na cílovou stránku závodu. Součást služeb je i registrace závodníků a jejich správa. Tato řešení jsou často pro závody většího rozsahu, kde je klíčové zaznamenání přesného času závodníků.

Koncept je vždy takový, že organizátor požádá o spolupráci zmíněné společnosti, které zajistí měření časů a v případě poptávky umožní registraci a správu závodníků. Dnes považujeme za standard mít pro veřejné závody i webové stránky. Organizátor tak musí mít vlastní webové stránky, kde zpřístupní registraci, propozice, trasy, výsledky a další části náležící k závodu.

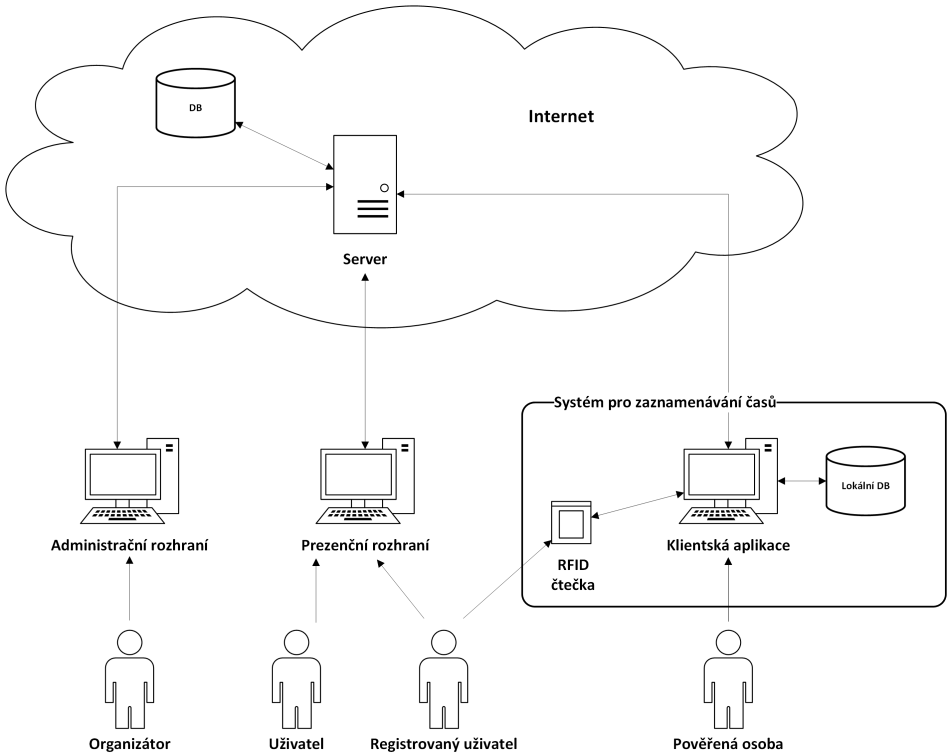
Vyvíjený produkt v této práci minimalizuje čas potřebný k vytvoření elektronické prezentace běžeckých závodů. Produkt cílí na pořadatele lokálních závodů, kteří potřebují administrativní rozhraní pro vytvoření a reprezentaci samotného závodu. Rozpočet takových závodů bývá často velmi nízký a měření probíhá individuálním způsobem. Individuálním způsobem je například ruční časomíra, kdy osoby v časové bráně zapisují časy závodníků.

Kapitola 4

Návrh řešení informačního systému pro organizaci závodu

Tato kapitola nejprve čtenáře seznámí s celkovým schématem systému a popisem jednotlivých částí. Dále se jsou podrobněji popsány aktéři a jejich činnost. Význačné prvky systému jako je autorizace nebo snímání čipů jsou zde také uvedeny. Nakonec je uvedeno základní grafické rozložení webového a aplikačního rozhraní.

4.1 Schéma a popis jednotlivých elementů



Obrázek 4.1: Schéma navrhovaného systému pro organizaci závodu

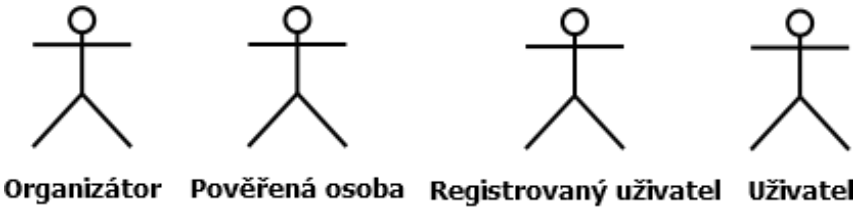
Celý systém se skládá ze sedmi částí: Serverové databáze, samotného serveru, administračního a prezenčního rozhraní, lokální databáze, klientské aplikace a RFID čtečky, přičemž administrační a prezenční rozhraní spolu s klientskou aplikací komunikují pomocí Internetu.

Pro administrační a prezenční rozhraní byla zvolena webová prezentace. Obě rozhraní jsou implementována technologií ASP.NET s návrhovým vzorem MVC 2.2.2. Jako úložiště dat byla zvolena databáze MSSQL. Komunikace mezi modelem webových rozhraní a databází zajišťuje Entity framework 2.2.3. Celý systém organizace závodů s podporou registrace závodníků je tedy rozdělen do dvou hlavních částí. Administrační rozhraní slouží k vytvoření a správě závodu. Organizátor vytvoří a zveřejní závod, který se tak zobrazí široké veřejnosti v prezenčním rozhraní. Prezenční rozhraní můžeme chápat jako alternativu osobních webových stránek konkrétního závodu.

Klientská aplikace je doplňková služba pro snímání časů závodníků. Po vytvoření závodu je služba zpřístupněna v administračním rozhraní. Vzhledem k požadavku na zálohování a snímání časů bez internetového připojení byla aplikace implementována technologií WPF 2.2.4 s datovým úložištěm využívající knihovnu SQLite 2.2.5. Je to tedy desktopová aplikace běžící na operačním systému Windows. Organizátor v administračním rozhraní stáhne a poté nainstaluje aplikaci na cílovém zařízení. Jakmile proběhne inicializační synchronizace se serverem, je aplikace připravena k použití. V případě zaznamenávání časů pomocí čipů je připojena RFID čtečka k cílovému zařízení a může dojít ke snímání čipů a přeposílání výsledků na server. Klientská aplikace s lokální databází a RFID čtečkou je součásti systému pro zaznamenávání času.

4.2 Případy užití systému

Diagram případů užití se používá k popisu chování systému z hlediska uživatele a zachycuje, které typy uživatelů se systémem pracují a jaké činnosti v rámci systému vykonávají [1]. Popisem interakce mezi uživatelem a systémem umožňuje znázornění funkčních požadavků na systém. Popis rolí uživatelů a jejich činnost je uvedena níže.



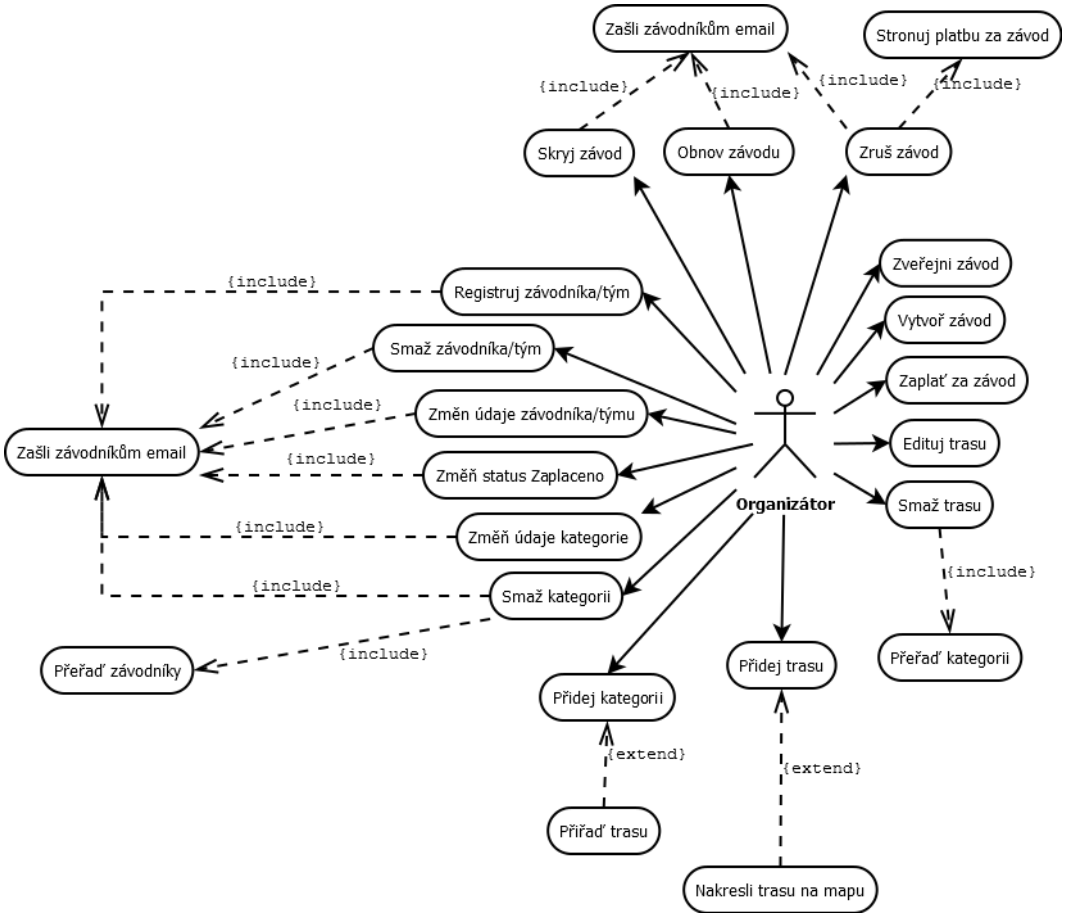
Obrázek 4.2: Výčet uživatelů systému

V celém systému pro organizaci běžeckých závodů účinkují čtyři uživatelé. Role Organizátora zastupuje uživatele, který vytváří závod a je zodpovědný za celkový průběh závodu. Správa závodu v sobě zahrnuje vytvoření novinek, stanovení propozic, určení tras a kategorií, registračního formuláře pro závodníky a správu závodníků. Ke správě závodu v administračním rozhraní má přístup pouze organizátor závodu. Pověřená osoba zastupuje uživatele, který pracuje s klientskou aplikací pro zaznamenávání výsledků. Taková osoba je určena Organizátorem a typicky se nachází v průběhu závodu na některém ze stanovišť (časových bran). V prezenčním rozhraní vystupuje role Uživatel a Registrovaný uživatel.

Uživatelé si mohou prohlížet závody a v případně se zaregistrovat do systému. Pro registraci na závod musí být uživatel zaregistrován do systému nebo zvolit jednorázovou registraci. Registrací do systému si uživatel vytvoří účet, na kterém bude mít přehled všech svých závodů a patřičných výsledků. Jednorázová registrace je pouhé přihlášení se na závod bez vytvoření osobního účtu. S podobnou politikou se můžeme setkat u celé řady komerčních webových stránek prodávající produkty. Uživatel s jednorázovou registrací či Registrovaný uživatel také interagují s klientskou aplikací.

Organizátor

Činnost organizátora je pro zpřehlednění rozdělena do dvou diagramů případu užití. V prvním je popsány základní akce pro správu závodu a správu závodníků. Druhý diagram pak doplňuje kompletní činnost organizátora.



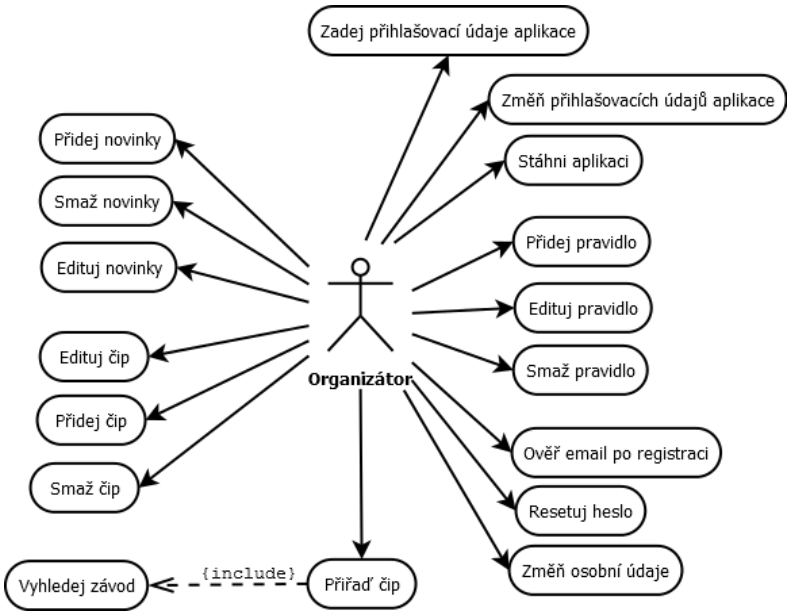
Obrázek 4.3: Diagram případu užití systému - základní činnosti Organizátora

Základní operace nad závodem jsou vytvoření, zveřejnění, skrytí, zrušení a obnovení. Závod je bezprostředně po vytvoření skryt uživatelům. Organizátor má možnost vše zkontrolovat, přidat či editovat další dílčí části (propozice, čipy, pravidla, novinky, atd.) a až poté závod zveřejnit. Jsou-li na závod již registrovaní závodníci a dojde-li ke skrytí nebo zrušení závodu, pak je nezbytné tyto závodníky kontaktovat na email. Došlo-li ke zaplacení závodu a následnému zrušení bez toho, aniž by závod proběhl, pak může organizátor podat žádost

o storno platby závodu. Při smazání či změně kategorie je nutné závodníky informovat o této skutečnosti a v případě potřeby je přeregistrovat.

Další důležitá činnost je práce s kategoriemi. Kategorii může být přiřazena trasa. Před smazáním trasy musí být změněny trasy kategorií, jenž jsou s kategorií asociovány. Tato činnost je v diagramu případu užití nazvána „Přeřazení kategorií“.

Poslední činností je správa závodníků. Jakákoliv modifikace závodníkových dat je zaslána na jeho email. Organizátor může registrovat závodníky nebo týmy do nastavené kapacity závodu. Každému závodníkovi je po registraci doručen email s výzvou k zaplacení startovného. Jakmile organizátor obdrží platbu nebo jinak klasifikuje zaplacení, pak lze změnit status závodníka na zaplaceno.



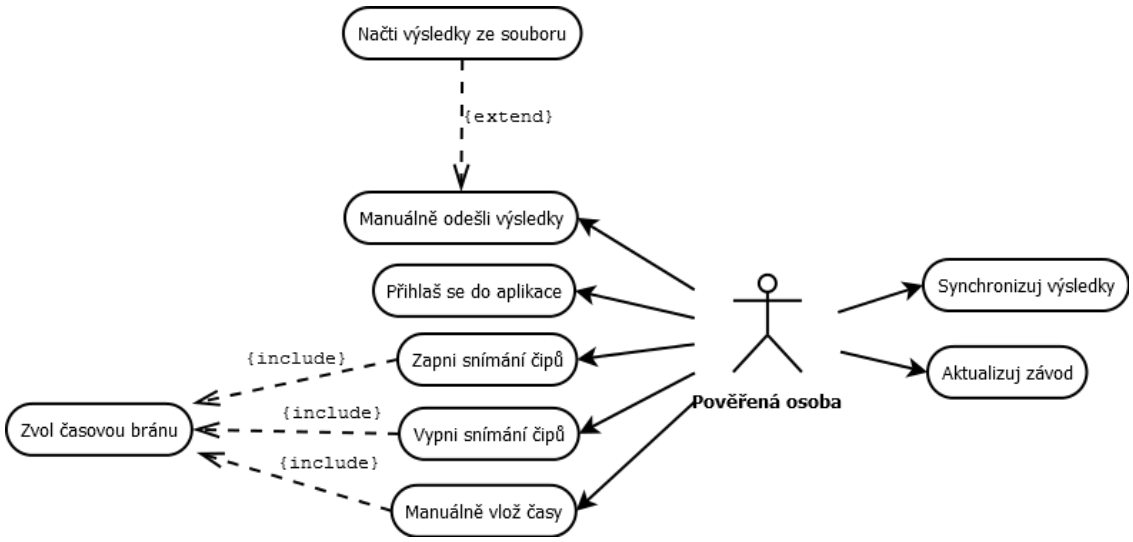
Obrázek 4.4: Diagram případu užití systému - doplňující činnosti organizátora

Doplnění kompletní činnosti organizátora je znázorněna na obrázku 4.4. Do této sekce činností spadá práce s novinkami, pravidly, čipy a manipulace s osobními údaji. Novinky a pravidla mohou být neomezeně přidávány v administračním rozhraní, přičemž výsledky se promítnou v prezenčním rozhraní. V prezenčním rozhraní si je mohou prohlédnout závodníci či běžní uživatelé. V administračním lze inicializovat a stáhnout klientskou aplikaci. Inicializací je míněno nastavení jednoznačného identifikátoru a hesla pro přístup do aplikace. Úspěšná registrace je podmíněna ověřením emailu. Je-li email úspěšně ověřen, pak je uživateli umožněno přihlásit se do systému.

Dosud popsané činnosti jsou specifické pro daný závod a s každým vytvořením závodu vznikají nové instance. To však neplatí pro čipy, které zastupují reálný fyzický produkt. Čipy jsou persistentní a mohou být použity v libovolném závodě.

Pověřená osoba

Diagram případu užití systému pro činnosti pověřené osoby

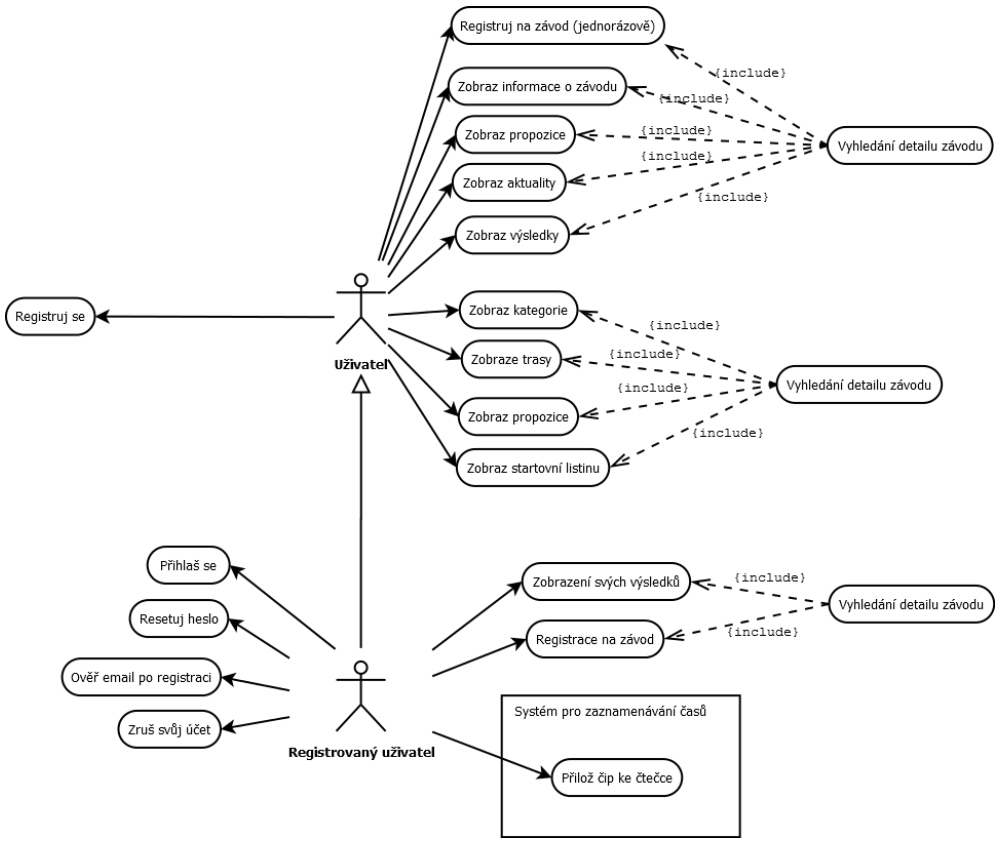


Obrázek 4.5: Diagram případu užití - pověřená osoba

Úspěšným přihlášením do aplikace je uživateli zpřístupněna kompletní funkcionalita. Aplikace provede aktualizaci dat bezprostředně po spuštění, avšak uživatel může kdykoliv provést novou aktualizaci. Zaznamenat čas lze dvěma nezávislými způsoby, jenž je snímání časů čipy nebo manuální zanesení časů. Se zaznamenaným časem je propojena brána, která musí být vybrána Pověřenou osobou. Výsledky jsou při vložení do aplikace automaticky zasílány na server. Jedním z možných scénářů je výpadek Internetu, a proto je Pověřené osobě umožněno zaslat výsledky později. V případě potřeby jsou výsledky načteny ze souboru, kde jsou v době závodu automaticky ukládány.

Uživatel a Registrovaný uživatel

Diagram případu užití systému pro činnosti uživatele a registrovaného uživatele



Obrázek 4.6: Diagram případu užití - uživatel a registrovaný uživatel

Uživateli je umožněno procházet téměř celý obsah webových stránek. Je-li vybrán závod, jsou zobrazeny pravidla, novinky, propozice kategorie, výsledky a další sekce náležící k danému závodu. Přihlášení a registrace má totožný průběh a operace, jako je tomu u Organizátora v administračním rozhraní. Registrovanému závodníkovi jsou zobrazeny aktuálně dosažené výsledky a veškerá historie závodů, kterých se zúčastnil. V průběhu závodu může Uživatel i Registrovaný uživatel přikládat své čipy k RFID čtečce, která je součástí systému pro zaznamenávání. Interakce se systémem pro zaznamenávání časů je podmíněna jednorázovou registrací nebo registrací do systému a následné registrace na závod.

4.3 Snímání RFID čipů

V rámci tohoto semestrálního projektu byla zvolena RFID čtečka 125 kHz s USB výstupem. Vzdálenost sejmutí čipů integrovanou anténou je dle specifikace 5 až 8 cm [15]. Zařízení je typu *Plug and Play*, přičemž čtečka je napájena přímo z USB kabelu, který je připojen ke zdroji. Zdrojem je v našem případě zařízení, na kterém je možné spustit klientskou aplikaci pro snímání čipů. Akceptovány jsou tagy (čipy) kompatibilní s třídou EM4100.

Po přiložení čipu je jeho unikátní hodnota přenesena jako ASCII stream 12 bytů, kde poslední dva byty jsou znaky `CR,LF` pro konec řádku na operačním systému Windows.

Výstupem je tedy desetimístné dekadické číslo, které operační systém zapíše tam, kde je logická aktivita[1]. Čtečka dočasnou změnou barvy led diodou a krátkým zvukovým pípnutím upozorní na sejmutý čip.

Po kliknutí na zahájení snímání v klientské aplikaci se přesune logická aktivita na skryté vstupní pole, kdo kterého OS zapíše dekadické číslo čipu. Aplikace číslo přečte, zpracuje a setrvá pro další vstup. Současně se sejmutím upozorní hlasitým pípnutím. Výsledky budou pro rychlou kontrolu zobrazeny v aplikaci formou tabulky. S každým zapnutím aplikace a prvním sejmutím čipu dojde v předem určené složce k otevření souboru, na jehož konec se vloží data skládající se z identifikátoru závodu, brány, čipu a časového razítka. Z výše popsaného postupu snímání čipů vyplývá, že pro bezproblémový průběh závodu je vysoce doporučené kontrolovat zaznamenaný čip pověřenou osobou, protože OS může například přesunout logickou aktivitu na nově zobrazené systémové modální okno.

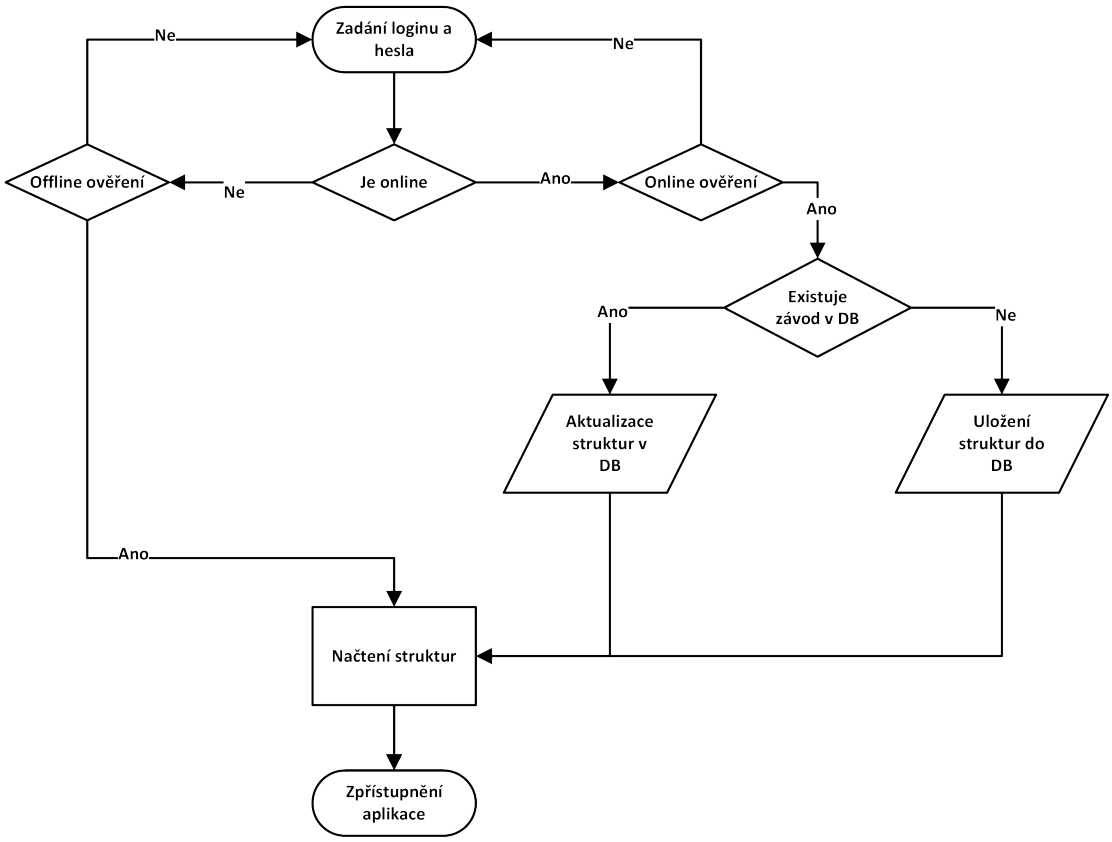
[1] OS se pokusí zapsat číslo do logicky aktivního prvku, například otevřený poznámkový blok

4.4 Autorizace klientské aplikace

Klientská aplikace pro snímání časů se autorizuje vůči autorizační části administračního rozhraní, ze kterého byla aplikace stažena. Organizátor závodu nastaví v administrační části jméno a heslo závodu, které později použije Pověřená osoba pro přihlašovací údaje.

Primární způsob autorizace je online ověření. Po úspěšném ověření v autorizační části administračního rozhraní obdrží aplikace kompletní data pro daný závod. V závislosti na pořadí přihlášení jsou data v lokální databázi vložena či aktualizována. V případě prvního přihlášení je do lokální databáze vložen nový závod. V ostatních případech je závod výhradně aktualizován.

Sekundárním způsobem přihlášení je offline ověření. Podmínkou úspěšného přihlášení je dřívější online autorizace, při které došlo ke stažení a úspěšnému uložení závodu spolu s přihlašovacími údaji do lokální databáze. Přihlášením je načten závod a zpřístupněné rozhraní pro snímání závodníků. Z výše uvedeného popisu tedy vyplývá, že aplikaci nelze použít bez minimálně jednoho online ověření. Pro přehlednost je níže uveden vývojový diagram autorizace včetně její závislosti na lokální databázi.



Obrázek 4.7: Autorizace klientské aplikace

4.5 Grafické rozložení webového a aplikačního rozhraní

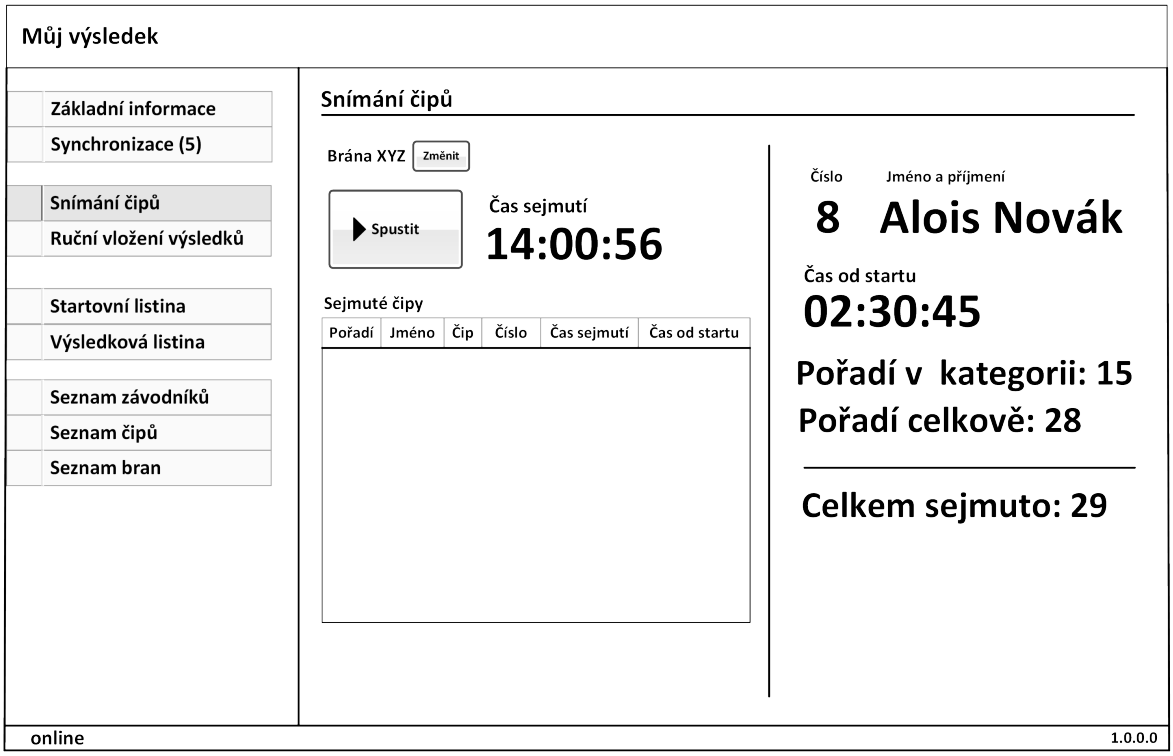
V této sekci jsou uvedeny grafická rozložení webového a aplikačního rozhraní. Jsou vybrány demonstrační obrazovky, na kterých lze popsat účel jednotlivých sekcí a v nich se vyskytujících prvků.

Aplikační rozhraní

Základním funkčním elementem je snímání čipů závodníků, na kterém je demonstrováno rozložení a funkcionalita aplikace. Celá aplikace je graficky rozložena do dvousloupcového rozložení. Levý sloupec zobrazuje menu položky, na jejichž základě se přepisuje obsah pravého sloupce. Závodní stanoviště, kde bude aplikace spuštěna, je typicky umístěno na volném prostranství s měnícím se osvětlením. Je tedy vhodné zvolit vysoký kontrast mezi ovládacími prvky.

Po zvolení brány klikne pověřená osoba na tlačítko pro spuštění snímání čipů. Vzhled tlačítka se mění tak, aby byl zřejmý zapnutý a vypnutý režim snímání. Na obrazovku mohou nahlédnout i samotní závodníci, kterým je po sejmutí čipu zobrazeno několik základních informací jako je čas, číslo a jméno závodníka nebo jeho celkové pořadí.

Dalším z důležitých požadavků je informovanost uživatele aplikace o připojení k Internetu a s tím spojená synchronizace se serverem. K tomuto účelu slouží především spodní lišta, která mění barvu v závislosti na připojení k Internetu (zelená - online, červená - offline). Při offline režimu nejsou sejmuté čipy poslány na server a každý takový výsledek navýší číslo v levém menu u položky „Synchronizace“, kde lze neodeslané výsledky manuálně odeslat. Wireframe popsaného rozložení je uveden níže.

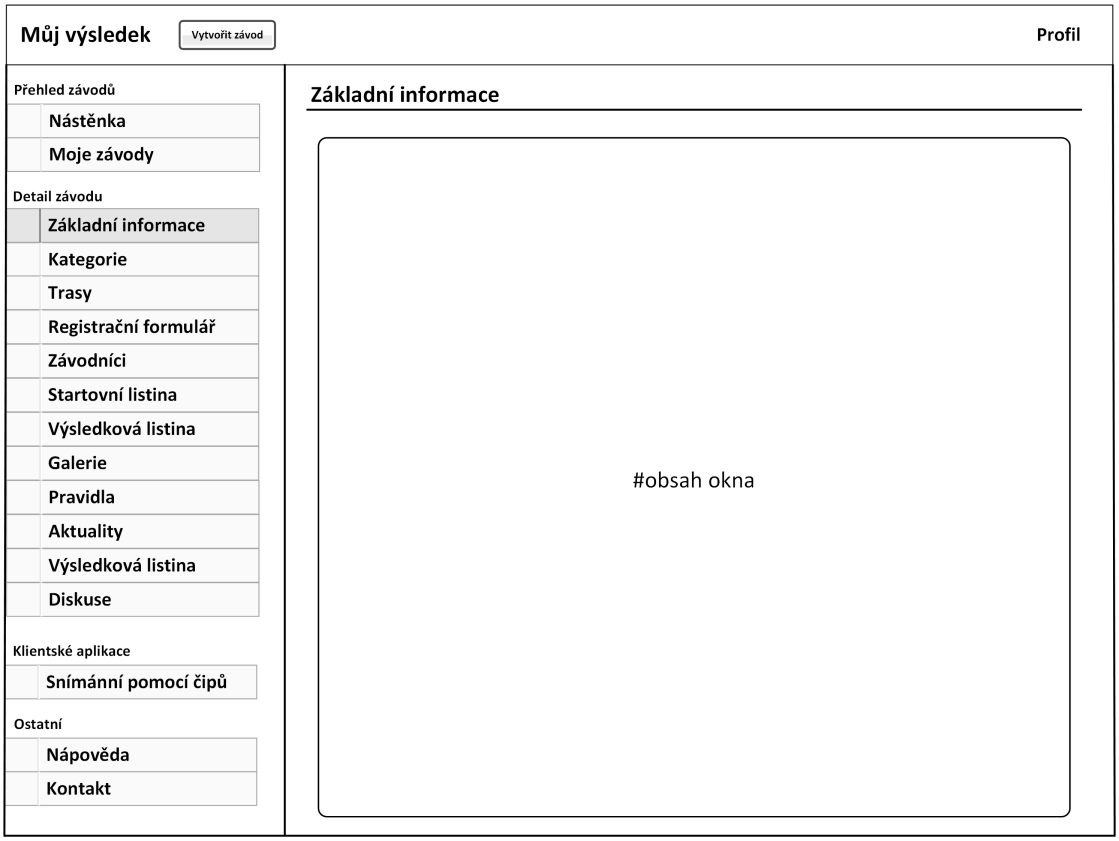


Obrázek 4.8: Uživatelské rozhraní aplikace pro snímání čipů

Webové rozhraní

Administrační rozhraní je určeno pro organizátory, kteří zde daný závod vytvoří a spravují. Jedná se o jádro celého systému. Vzhledem k jeho důležitosti je kladen vyšší důraz na funkcionalitu na úkor grafického zpracování.

Byla provedena kolekce několika šablon, určených především pro administraci informačních systémů, a následná selekce nejvhodnější. Pro svou kompatibilitu, zpracování, rozšířenost a podporu byla vybrána šablona Admin LTE [2], od autora A. Almsaeed, jenž ji zveřejnil jako open-source. Základní verze levé strany šablony obsahuje položky relevantní ke všem závodům (nástěnka, moje závody, nápověda, ...). Vytvořením, či přesměrováním na detail závodu dojde k rozšíření levého menu pro správu závodu (kategorie, trasy, závodníci, ...). Horní lišta zobrazuje přímé odkazy na často konané uživatelovy akce a přístup ke svému účtu. Wireframe popsaného rozložení je uveden níže.



Obrázek 4.9: Grafické rozložení administračního rozhraní

[2] AdminLTE - https://almsaeedstudio.com/

Kapitola 5

Závěr

V tomto semestrálním projektu byl navržen systém pro organizaci běžeckých závodů. Nejprve byl proveden průzkum trhu a nastudovány systémy s pricipiálně podobnou funkcionalitou. Následně byly prozkoumány dostupné technologie pro tvorbu informačních systémů. Na základě osobní preference a jednotnosti vzhledem ke klientské aplikaci na OS Windows byla vybrána technologie .NET firmy Microsoft. Jako další stěžejní krok byla sumarizace a ujasnění všech požadavků nejen od organizátorů, ale i analýzou vybraných závodu a vlastní zkušeností ze závodů. Nakonec byl systém rozložen na menší celky a významné z nich byly promyšleny a popsány.

Prioritou navrženého systému je správná funkcionalita. Doposud neexistuje podobný systém pro organizaci závodů. Cílem návrhu a pozdější implementace je vytvořit funkční prototyp, který projde několika cykly testování samotných organizátorů. Na prvky jako bezpečnost, výpočetní náročnost nebo kompatibilita byl brán zřetel, ale nebylo jim věnováno příliš velké úsilí. V případě úspěchu bude systém zveřejněn a postupně rozšiřován.

Mezi nejnáročnější požadavky mohu zařadit shromáždění požadavků a následného vhodného rozložení informačního systému na tři části: Administrační rozhraní, prezenční rozhraní a klientkou aplikaci. Za netriviální problém také považuji nastudování vhodných technologií z rodiny .NET.

Literatura

[1] Alena Buchalcevová, L. P., Jarmila Pavlíčková: *Základy softwarového inženýrství*. Vysoká škola ekonomická v Praze, 2007, ISBN 9878024512709.

[2] Dajbych, V.: *mvvm: model-view-viewmodel*. Duben 2009, [Online; navštíveno 06.11.2016].
URL http://www.dotnetportal.cz/clanek/4994/MVVM-Model-View-ViewModel

[3] Molnár, Z.: *Podnikové informační systémy*. České vysoké učení technické v Praze, 2009, ISBN 9788001043806, 195 s.

[4] MSDN: *ASP.NET MVC Overview* . [Online; navštíveno 17.10.2016].
URL https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx

[5] MSDN: *ASP.NET Overview* . [Online; navštíveno 17.10.2016].
URL https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx

[6] MSDN: *Data Binding Overview*. [Online; navštíveno 27.10.2016].
URL https://msdn.microsoft.com/en-us/library/ms752347(v=vs.110).aspx

[7] MSDN: *Entity Framework* . [Online; navštíveno 20.10.2016].
URL https://msdn.microsoft.com/en-us/library/gg696172(v=vs.103).aspx

[8] MSDN: *Introduction to WPF*. [Online; navštíveno 25.10.2016].
URL https://msdn.microsoft.com/en-us/library/mt149842(v=vs.110).aspx

[9] MSDN: *Overview of the .NET Framework* . [Online; navštíveno 15.10.2016].
URL https://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.110).aspx

[10] portal, R.: *Co je RFID*. [Online; navštíveno 13.09.2016].
URL http://www.rfidportal.cz/index.php?page=rfid_obecne

[11] Samaras, C.: *Differences Between WPF And Windows Forms*. Srpen 2014, [Online; navštíveno 25.10.2016].
URL http://www.myengineeringworld.net/2014/08/wpf-and-windows-forms-differences.html

[12] SQLite: *About SQLite*. [Online; navštíveno 10.11.2016].
URL https://sqlite.org/about.html

[13] SQLite: *SQL Features That SQLite Does Not Implement*. [Online; navštíveno 10.11.2016].
URL https://www.sqlite.org/omitted.htmll

[14] Tomáš Hruška, Z. K.: *Studijní opora do předmětu IIS a PIS, Pojem informačního systému, Data, Procesy, Transakce* . FIT VUT v Brně, Únor 2012, [Online; navštíveno 20.09.2016].
URL https://www.fit.vutbr.cz/study/courses/WAP/private/opory/.csl

[15] Vojáček, A.: *Používané RFID frekvence a jejich vliv na čtení a zápis tagu.* Září 2015, [Online; navštíveno 13.09.2016].
URL
http://automatizace.hw.cz/komponenty-prumyslove-sbernice-a-komunikace/vice-i-mene-bezne-rfid-frekvence-a-jejich-vliv-na-vlastnosti-tagu.html