

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Inovace webové aplikace T. K. Průzkumník – analýza

Jakub Koláčný

Bakalářská práce

2017

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2015/2016

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub Koláčný**
Osobní číslo: **I13151**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Inovace webové aplikace T. K. Průzkumník - analýza**
Zadávající katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Primárními cíli této bakalářské práce je provést počáteční analýzu současného stavu webového portálu neziskové organizace T. K. Průzkumník spuštěného na adrese www.pruzkumnik.cz a navrhnout inovaci této rozsáhlé aplikace.

V teoretické části práce se pak autor zaměří na popis aktuálních trendů v oblasti vývoje webových aplikací a měl by též zahrnout popis v současné době používaných standardů.

V praktické části pak bude provedena samotná analýza, která poslouží jako podklad pro tvorbu inovovaného portálu. Analýza by měla zahrnovat jednak současný stav programového kódu (z hlediska optimalizace výkonu a zabezpečení), návrhu databáze a případů užití. Návrh inovace by měl obsahovat UML modely případů užití, diagramů aktivit a analytických tříd.

Pro dosažení cíle by měla analýza vycházet z principů metodiky RUP.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

[1] ARLOW, Jim a Ila NEUSTADT. UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007, 567 s. ISBN 978-80-251-1503-9.

[2] BOOCH, Grady a Grady BOOCH. Object-oriented analysis and design with applications: objektově orientovaná analýza a návrh prakticky. 3rd ed. Upper Saddle River, NJ: Addison-Wesley, c2007, xxiii, 691 p. ISBN 978-020-1895-513.

[3] WEST, David, Gary POLLICE a Brett MCLAUGHLIN. Head First Object-Oriented Analysis and Design. 1st Edition. Sebastopol, CA 95472: O'Reilly Media Inc., 2007. ISBN 0596008678.

Vedoucí bakalářské práce:

Ing. Jan Hříděl

Katedra softwarových technologií

Datum zadání bakalářské práce:

31. října 2015

Termín odevzdání bakalářské práce:

13. května 2016



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Mgr. Josef Hořálek, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2016

PROHLÁŠENÍ AUTORA

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 9. května 2017

Jakub Koláčný

PODĚKOVÁNÍ

Rád bych poděkoval Ing. Janu Hřídelovi za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce. Mé poděkování patří též Mgr. Ondřeji Chobotovi za poskytnutí historických údajů a materiálů pro praktickou část práce. A dále děkuji všem ostatním členům vedení T. K. Průzkumník za jejich cenné rady a návrhy při vytváření nového konceptu.

ANOTACE

Na začátku práce je představena většina používaných standardů, důležitých při vývoji webových aplikací. Práce dále pojednává o rozdílech vývoje a nových trendech v oblasti webových aplikací. Hlavním cílem praktické části bakalářské práce je provést počáteční analýzu současného stavu webového portálu neziskové organizace T. K. Průzkumník, spuštěného na adrese www.pruzkumnik.cz, a navrhnout inovaci nevyhovujících částí této rozsáhlé aplikace.

KLÍČOVÁ SLOVA

T. K. Průzkumník, standardy webových aplikací, UML, případy užití, diagramy aktivit, databáze

TITLE

The Innovation of web application of T. K. Průzkumník – analysis

ANNOTATION

Thesis introduces the most important standards applied in the development of web applications. It discusses the differences of development and new trends in web applications. The main goal of the practical part of the thesis is to perform an initial analysis of the current condition of the web portal of the non-profit organization T. K. Průzkumník running at www.pruzkumnik.cz and suggest upgrading non-conforming parts of this extensive application.

KEYWORDS

T. K. Průzkumník, web applications standards, UML, use cases, activity diagrams, database

OBSAH

0	Úvod.....	11
1	Používané standardy webových aplikací	13
1.1	HTML5	13
1.2	CSS3.....	14
1.3	Skriptovací jazyky.....	16
1.3.1	PHP7	16
1.3.2	Perl.....	17
1.3.3	ASP.NET	17
1.3.4	JavaScript.....	18
1.3.5	TypeScript.....	18
1.3.6	CoffeeScript.....	19
1.4	XML, JSON	19
1.4.1	XML.....	19
1.4.2	JSON.....	20
1.5	Relační databáze.....	20
1.5.1	Normalizace databáze	21
1.6	NoSQL databáze	22
2	Trendy vývoje webových aplikací.....	23
2.1	Framework	23
2.1.1	Model view controller (MVC).....	23
2.2	Responzivní web design (RWD).....	23
2.3	Search Engine Optimization (SEO)	24
2.4	Progressive Web Apps (PWA).....	25
3	Unifikovaný proces vývoje aplikací	27
3.1	Unified Modeling Language (UML).....	27
3.2	Unified Process (UP).....	28

3.3	Funkční a nefunkční požadavky.....	28
3.4	Případy užití	29
3.5	Diagramy aktivit.....	30
4	T. K. Průzkumník.....	31
4.1	Historie webu	31
4.2	Odchyly oproti dnešním trendům.....	32
4.3	Stávající funkcionalita.....	33
4.4	Diagramy případů užití a aktivit	34
4.5	Část upraveného relačního modelu databáze	41
5	Závěr	43
6	Použitá literatura	44
7	Seznam příloh	47

SEZNAM ILUSTRACÍ

Obrázek 1 – Diagram případu užití (Karta člena).....	34
Obrázek 2 – Diagram případu užití (Karta družiny).....	35
Obrázek 3 – Diagram případu užití (Správa výprav).....	36
Obrázek 4 – Diagram aktivit (Systém výpravy)	37
Obrázek 5 – Diagram případu užití (Karta výpravy)	38
Obrázek 6 – Diagram případu užití (Bodování)	38
Obrázek 7 – Diagram případu užití (Upozornění).....	39
Obrázek 8 – Diagram případu užití (Fotogalerie).....	40
Obrázek 9 – EER diagram	41

SEZNAM ZKRATEK A ZNAČEK

API	Application Programming Interface
CERN	European Organization for Nuclear Research
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
EER	Enhanced Entity Relationship
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IETF	Internet Engineering Task Force
IIS	Internet Information Services
JS	JavaScript
JSON	JavaScript Object Notation
MVC	Model view controller
OMG	Object Management Group
OOP	Objektově orientované programování
PHP	PHP: Hypertext Preprocessor
PWA	Progressive Web Apps
RDBMS	Relational database management system
RFC	Request for Comments
RMD	Relační model dat
RWD	Responsive Web Design
SEO	Search Engine Optimization
SGML	Standard Generalized Markup Language
SQL	Simple Query Language
UML	Unified Modeling Language
UP	Unified Process
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

0 ÚVOD

Technologie pro vývoj webových stránek, případně aplikací, se každým rokem vyvíjí kupředu a již velmi dlouho nám k vývoji nestačí pouze dva nebo tři jazyky či technologie, ale potřebujeme jich celou škálu. Cílem teoretické části práce je přiblížit některé vybrané standardy pro vývoj webových aplikací dnešní doby. Nejdříve je potřeba představit dva nejpoužívanější jazyky pro práci se statickými webovými stránkami. Prvním z nich je HTML neboli hypertextový značkovací jazyk pro tvorbu obsahu webových stránek. Druhým je CSS, česky řečeno kaskádové styly, které určují, jaký vzhled budou data a celkově webová stránka mít. Některé větší projekty vyžadují, aby byla data zobrazována dynamicky, tudíž takto vytvořená statická prezentace nestačí. Pro tyto účely dobře poslouží dále popsané skriptovací jazyky PHP nebo JavaScript. Mezi další důležité a zajímavé skriptovací jazyky patří např. Perl nebo jazyky vycházející právě z JavaScriptu – TypeScript a CoffeeScript. Nedílnou součástí vývoje webových aplikací je i komunikace a předávání dat mezi jednotlivými aplikacemi. Pro předávání informací je zapotřebí mít jednoduchý a srozumitelný formát dat, který poskytuje například jazyk XML nebo JSON. Data je zapotřebí nejenom předávat, ale také někde uchovávat a následně znovu načíst a použít. Nejčastěji se data ukládají do relačních databází, jako jsou např. MySQL, Oracle, PostgreSQL nebo MS SQL.

Velkým trendem při vývoji webových aplikací je co největší ulehčení procesu. K takovému usnadnění dojde např. při použití tzv. frameworků, které slouží jako základní stavební rámec a vývojáři tak nemusí řešit běžné úkony a problémy, ale mohou se naplno věnovat danému projektu. Jelikož je dnešní doba přehlcena mobilními telefony, tablety a jinými chytrými zařízeními, je zapotřebí se tomuto trendu přizpůsobit i při vývoji nových webových aplikací. Přizpůsobením je myšlena možnost zobrazit si čitelně a přehledně obsah webových stránek právě na jakémkoliv zařízení. Záleží už na vývojářích, jestli aplikaci vytvoří s pomocí responzivního designu anebo z ní udělají přímo progresivní webovou aplikaci (PWA) neboli aplikaci tvářící se jako běžná mobilní aplikace.

Při vývoji větších projektů je důležité mít systém nebo osnovu pro plánování. Jednou z metod, které lze použít, je metodika UP neboli unifikovaný proces vývoje aplikací, založený na iterativním a přírůstkovém procesu. To znamená, že se projekt rozdělí na menší sekce (iterace) a poté se vykonává více menších procesů, kde je každý rozdělen na zpracování požadavků, analýzu, návrh, implementaci a testování.

Praktická část práce se zabývá analýzou současného stavu rozsáhlého webového portálu volnočasové dětské organizace T. K. Průzkumník a nového návrhu nevyhovujících funkcionalit právě za využití některých vhodných metod používaných v unifikovaném procesu vývoje aplikací, jakou jsou funkční a nefunkční požadavky, diagramy užití nebo diagramy aktivit. Nejdříve je nahlédnuto do historického vývoje portálu a následné porovnání stávající funkcionality vůči dnešním trendům. Dále jsou popsány již samotné diagramy užití a aktivit nových funkcionalit a inovací sestavených v programu Enterprise Architect od společnosti Sparx System pro systémovou analýzu a návrh. V poslední řadě je představena upravená část databázového modelu, vytvořeného v programu MySQL Workbench od společnosti Oracle Corporation, který momentálně čítá desítky tabulek, a díky vytvořeným inovacím bylo zapotřebí některé upravit anebo připojit nové.

1 POUŽÍVANÉ STANDARDY WEBOVÝCH APLIKACÍ

1.1 HTML5

HTML (HyperText Markup Language) neboli hypertextový značkovací jazyk je jedním z hlavních jazyků WWW (World Wide Web). S každou novou verzí tohoto jazyka se internet posune o velký krok vpřed.

První krůčky jazyka HTML lze datovat od roku 1990, kdy byl vytvořen spolu s protokolem HTTP. První oficiální verze z roku 1991 nesla označení HTML 0.9. Specifikace zatím neobsahovala grafiku a bylo možné v ní vytvářet pouze primitivní dokumenty. Šlo o součást projektu propojeného informačního systému pro výzkumné centrum fyziky CERN. Verze 2.0 byla vydána v listopadu 1995 komunitou IETF (Komise pro technickou stránku internetu) a již podporovala jednoduchou grafiku a formulářové prvky. Verze 3.2 byla vydána 14. ledna 1997 komunitou W3C. Vycházela z verze 3.0, která byla natolik složitá, že nebyla nikdy přijata jako standard. Tato verze začala podporovat formátování textu, tabulky a stylové prvky pro úpravu vzhledu. V roce 1997 vznikla verze 4.0, do které přibyly nové formulářové prvky, byly rozšířeny prvky pro tvorbu tabulek a specifikace začala podporovat rámy (frames). Tato verze se snažila dosáhnout původního záměru – prvky by měly vyznačovat význam jednotlivých částí dokumentu. Vzhled byl ovlivňován připojovanými styly (CSS). Verze 5 se začala tvořit okolo roku 2007, kdy W3C založilo novou pracovní skupinu HTML Working Group jejímž cílem bylo v roce 2010 uvolnit specifikaci nové verze HTML. Tato verze ukončuje závislost HTML na SGML (Standard Generalized Markup Language), opravuje mnoho chyb předešlé verze, vyřazuje mnoho zastaralých a již nepoužívaných prvků a přidává nové sémantické prvky.

[1][2]

HTML5 zavedlo zcela nové tagy, které slouží pro zpřehlednění a optimalizaci celého projektu. Místo rozdělování částí stránky do tzv. divů (<div>) umožňuje HTML používat tagy přesně podle toho, jaká část stránky je jimi strukturována. Mezi takové tagy patří například: <section>, <article>, <main>, <header>, <footer>, <nav>. Dále přibyly úplně nové tagy pro práci s multimédií: <video>, <audio>, <track> a pro práci s grafikou: <canvas>. Další novinkou je <time> pro formát a práci s časem. S novými tagy byly zavedeny nové globální atributy nebo mnoho nových atributů pro práci s formuláři.

Specifikace HTML5 též podporuje řadu nových API (Application Programming Interface):

- FILE API – umožňuje nahrávat neomezené množství souborů najednou nebo rozdělení velkého souboru na straně klienta a postupné nahrání těchto dílů na server.

- FULLSCREEN API – odstraňuje nutnosť používať Flash pri prepnutí dokumentu přes celou obrazovku.
- MEDIA API – umožňuje vytvoriť vlastné prostredie prehrávače a v kombinaci s Fullscreen API i nahradit dodnes využívaný Flash.
- TEXT TRACK API – umožňuje pripojovať k prehrávaným multimédiám popisky, titulky nebo metadata.

Jak už bylo řečeno, specifikace HTML5 vyřazuje některé zastaralé prvky předchozích verzí, jako jsou `<frame>` a `<frameset>` nebo prvky nahraditelné CSS: ``, `<center>`, `<big>` a mnoho dalších.

Další zásadní změnou je i změna hlavičky. Zde se upravila a zjednodušila deklarace *DOCTYPE* a meta tagu *charset*.

Zápis ve verzi 4.01:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

A ve verzi HTML5:

```
<!DOCTYPE html>
<meta charset="utf-8">
```

[3][4]

1.2 CSS3

Kaskádové styly, běžně označované jako CSS, jsou jednoduchým jazykem určeným ke zjednodušení tvorby vzhledu webových stránek. CSS popisují způsob zobrazení elementů na webových stránkách. Nejčastěji se používají v kombinaci se značkovacím jazykem jako je HTML nebo XHTML. Pomocí CSS můžete stanovit barvu textu, styl písma, mezery mezi odstavci, šířku sloupců, obrázek na pozadí, rozestavení prvků, změny v zobrazení pro různá zařízení s rozdílným rozlišením obrazovky a mnoho dalšího.

Nápad o zavedení kaskádových stylů měl 10. října 1994 Håkon Wium Lie a dnes je spravována prostřednictvím skupiny W3C pod názvem CSS Working Group. Tato skupina vytváří dokumenty tzv. specifikace. Když tato specifikace projde diskuzí, kontrolou a oficiálním schválením členy W3C, stává se tzv. doporučením. Doposud byly vydány tři verze:

- Cascading Style Sheets, level 1 (CSS1). Tato verze vyšla jako doporučení v prosinci 1996. Popisuje jazyk CSS jako jednoduchý vizuální formátovací model pro každý z HTML tagů.
- CSS2 se v květnu 1998 stala W3C doporučením a je založena na předchozí verzi CSS1. Tato verze přidává podporu pro specifické mediální šablony (tiskárny a zvuková zařízení), styly písma, uspořádání prvků a tabulky.
- CSS3 se stala W3C doporučením v červnu roku 1999. První stabilní verze byla vydána až po více než 12 letech. Tato verze je rozdělena na části, které se nazývají *moduly*. Každý z modulů obsahuje rozšiřující prvky definované v CSS2. Postupem času jsou přidávány do specifikace další rozšiřující moduly. Momentálně tato specifikace obsahuje okolo 50 modulů.

Mezi hlavní výhody použití CSS patří úspora času – stačí jednou napsat CSS a stejný vzor používat pro více stránek HTML. Další výhodou je rychlejší načítání stránek – stačí napsat jedno pravidlo CSS pro daný tag a tento styl se použije pro všechny výskyty téhož tagu, čímž se ušetří mnoho kódu. S tím je spojena další výhoda, a to globální změny – jednoduše změňte styl a všechny prvky na celých webových stránkách budou automaticky aktualizovány. Jsou lepší než styly v HTML – CSS má mnohem širší pole atributů pro úpravu vzhledu a lepší kompatibilitu s novými verzemi webových prohlížečů. Mimoto si prohlížeč může do své mezipaměti dané styly uložit a následně je rychleji načíst bez nutnosti znovunačtení.

[5]

Mezi hlavní novinky ve verzi CSS3 patří:

- Kulaté rohy (*border-radius*),
- Stín (*box-shadow*),
- Stín u textu (*text-shadow*),
- Pohyb, rotace (*transform*) – bez využití JavaScriptu,
- Animace (*animation*),
- Průhlednost (*opacity*),
- Modely barev:
 - RGBA: klasické RGB, ke kterému přibývá čtvrtá hodnota – průhlednost,
 - HSL (Hue, Saturation, Lightness): neboli barva, sytost a světlost,
 - HSLA: HSL obohacené o průhlednost,
- Sloupce (*Multiple Columns*),

- Počítání hodnot – *calc()*,
- Webové fonty – nyní možnost použití libovolného fontu.

[6]

1.3 Skriptovací jazyky

Jsou to vyšší programovací jazyky, které jsou interpretovány jiným programem v době běhu, nikoli kompilovány jako jiné programovací jazyky (například C nebo C++). Tento druh jazyků ovlivňuje data, která jsou zobrazena v prohlížeči na straně klienta (client-side). Existují i serverové skriptovací jazyky, které obvykle zpracovávají data například v databázích.

1.3.1 PHP7

PHP je serverový skriptovací jazyk navržený pro potřeby tvorby dynamických webových stránek. Počátky vývoje PHP sahají do roku 1994, kdy Rasmus Lerdorf vytvořil jednoduchý systém pro evidenci přístupů na svůj web. Ten se rozšířil mezi další uživatele, kteří přicházeli s požadavky na jeho vylepšení. Tím vznikl systém Personal Home Page Tools, častěji označován jako PHP Tools. V polovině roku 1995 se systém PHP Tools spojil s programem stejného autora pro zpřístupnění databázi na serveru – Form *Interpreter*, neboli zkráceně FI. To dalo za vznik PHP/FI 2.0. Dále následovalo PHP 3.0 z roku 1998. V této verzi dostává původní zkratka nový rekurzivní význam – PHP: Hypertext Preprocessor. PHP 4.0 bylo oficiálně vydáno v květnu 2000 a bylo vylepšenou verzí 3.0 postavené na novém enginu firmy Zend Technologies – Zend Engine 1.0. V roce 2004 následovalo vydání verze 5.0 postavené na Zend Engine 2.0 s novým objektovým modelem, který se přibližuje ostatním jazykům podporující OOP.

[7]

Předešlá stabilní verze před PHP 7 nese označení PHP 5.6. Vývojový tým rozhodl, že v příštím vydání název PHP 6 vynechá, protože v minulosti už toto označení existovalo jako experimentální projekt, který ale nikdy nedospěl do produkční fáze. Aby se předešlo tomu, že by uživatelé míchali dohromady předešlý pokus s nejnovějším vývojem, běží nové hlavní vydání pod názvem PHP 7.

Mezi hlavní změny v nové verzi patří především přechod na zbrusu novou verzi enginu přicházející pod kódovým názvem PHPNG (Next Generation). Vývojový tým PHPNG předělal původní Zend Engine, optimalizoval využívání paměti a přidal kompilaci *just-in-time*, která umožňuje kompilaci kódu při běhu, nikoliv před zahájením vykonávání, jako tomu bylo doposud. Díky tomu se kód vykoná dvojnásobně rychleji a zároveň je potřeba méně serverů k obsluze stejného množství uživatelů oproti starším verzím. Dále přibýly nové objekty pro snazší

odchytávání a zpracování chyb a PHP 7 zavádí konsistentní 64 bitovou podporu, čímž je myšlena jak nativní podpora 64 bitových celých čísel, tak i velkých souborů. Aktuální stabilní verze je verze PHP 7.1.3.

[8]

1.3.2 Perl

Perl je praktický a jednoduchý interpretační programovací jazyk. Je využíván převážně na WWW serverech a při zpracování textu. V oblasti zpracovávání souborů patří k těm nejlepším nástrojům. S vývojem internetu se Perl stal velmi oblíbeným nástrojem pro vytváření CGI skriptů. CGI již dnes nemá takovou popularitu, protože ho z jeho postavení vytlačují jazyky typu PHP. Perl byl původně napsán pro unixovou platformu, ale nyní již běží na většině ostatních. Mezi platformami se dá poměrně dobře přesunovat. Syntaxe Perlu je velmi svobodná – běžná podmínka se dá napsat mnoha způsoby, tudíž je vhodné držet se předem určených pravidel. Perl člověku dovolí mnohdy až příliš.

Název Perl vznikl ze slov: Practical Extraction and Report Language. Autorem je Kanadčan Larry Wall, který v roce 1987 publikoval Perl 1.0 pro veřejnost. Při vytváření jazyka se řídil heslem: „*dá se to udělat více způsoby*“ („*there's more than one way to do it*“). Číslování verzí bylo následně dosti přeskočeno a další verze byla rovnou 4. Pak následovala verze 5, která přinesla četná vylepšení, obzvláště výkonné datové struktury a schopnost objektového programování. Od té doby se přidávaly nové a nové podoby 5. verze a poslední stabilní verze Perlu je verze 5.24 ze dne 5. 9. 2016.

[9]

1.3.3 ASP.NET

Je součástí .NET Frameworku pro tvorbu webových aplikací a služeb. Jednoduše řečeno se jedná o soubor knihoven, které umožňují tvorbu webových aplikací v jazyce C#. Knihovny obsahují připravená řešení mnoha základních problémů, které ve webových technologiích nastávají. Tím je myšlena například bezpečnost, správa formulářů, práce s databází, autentifikace uživatele apod. První verze byla vydána v roce 2002 a dnešní nejnovější verze je 4.6. Technologie je založena na architektuře klient-server. Aplikace v ASP.NET je tedy program, na jehož výstupu je HTML stránka. Framework běží na serveru IIS (Internet Information Services), ale může běžet i na webserveru Apache jako modul (podobně jako například PHP).

Microsoft nabízí 2 hlavní způsoby, jakými lze vytvářet webové stránky. Jsou to WebForms a MVC. Oba způsoby jsou zajímavé a dívají se na webovou aplikaci trochu jinak. WebForms je

snaha převést WinForms (formulářové/okenní aplikace z desktopu) do webové podoby. Myšlenkou je jednoduše poskládat formulář z prvků (tlačítka, popisky, textová pole atd.) a těmto prvkům přiřadit události, které mají vykonávat. MVC je novější návrh a rozděluje webovou aplikaci do třech typů komponent: kontrolérů, modelů a pohledů (Model View Controller).

[10]

1.3.4 JavaScript

JavaScript je interpretovaný, objektově orientovaný programovací jazyk nejčastěji používaný jako součást webových aplikací, jejichž implementace povoluje *client-side* (na straně klienta) skripty k interakci s uživatelem a vytváření dynamických stránek. JavaScript se poprvé objevil v Netscape 2.0 v roce 1995 s názvem LiveScript, ale společnost Netscape časem tento název změnila na JavaScript. V roce 1998 byl standardizován jako ECMAScript. Cesta ke standardizaci jazyka byla poněkud obtížná, tvůrci si nakonec nechali jazyk standardizovat u evropské firmy, protože ani W3C ani ISO jim nechtěli standard schválit. JavaScript trpí jedním podstatným nedostatkem – zdrojový kód se nachází u klienta. Tím pádem si jej může každý zobrazit, popřípadě si někam uložit nebo předělat.

Syntaxe jazyka je *C-like* (podobná jazyku C) a je dynamicky typovaný. Obsahuje jen jeden numerický typ – *number*, *string* pro text, *boolean* pro pravdivostní hodnotu a *Object* pro cokoliv jiného. Jazyk je sice objektově orientovaný, ale je zde velká neobvyklost – neexistuje zde nic jako *třída*. Dále využívá tzv. funkcionální paradigma, které umožňuje do běžné proměnné uložit funkci. Což se hodí například pro tvorbu tzv. lambda funkcí. V JavaScriptu se poslední dobou objevují i hry a díky tomu je vytlačována Flash platforma. Po příchodu HTML5, které poskytuje hardwarově akcelerované plátno, a s podporou SVG je možnost přes JavaScript kreslit vektorovou grafiku. JavaScriptové aplikace jsou plně multiplatformní a fungují na libovolném zařízení, které má internetový prohlížeč. Takže se může jednat o počítač, tablet, telefon nebo třeba i chytré hodinky.

[11][16]

1.3.5 TypeScript

TypeScript je silně typový, objektově orientovaný, kompilovaný jazyk. Návrh vytvořil Anders Hejlsberg (Návrhář jazyka C#) ve společnosti Microsoft v roce 2012. TypeScript je nejen jazyk, ale i sada nástrojů. Jelikož je TypeScript nadstavbou JavaScriptu, je libovolný JavaScriptový kód automaticky validním TypeScript kódem. TypeScript nabízí: statické datové typy, třídy a dědičnost, moduly, rozhraní nebo generické datové typy.

[14][15]

1.3.6 CoffeeScript

CoffeeScript je odlehčený jazyk inspirovaný jazyky Ruby a Python, který překompilovává svůj kód do JavaScriptu. Tím nabízí lepší syntaxi a vyhnutí se bizarním částí JavaScriptu. Byl navržen Jeremy Ashkenasem a poprvé vložen na Git 13. prosince 2009. V roce 2010 byla vydána nová verze CoffeeScriptu 1.0 a v této době byla jednou z nejžádanějších projektů na Git hubu. Mezi jeho hlavní výhody patří srozumitelnější a snáze udržitelný kód, čistší a kratší kód, možnost využití tříd a dědičnosti, vyhnutí se problémovým znakům nebo klíčovému slovu *var* pro vytváření proměnných. Nicméně k výhodám patří i nevýhody. Jednou z takových nevýhod je citlivost na mezery (prázdné znaky). Překladač CoffeeScriptu je obyčejný JavaScriptový program. Ve skutečnosti je napsán v CoffeeScriptu a „*přeložen sám sebou*“ do JavaScriptu.

[12][13]

1.4 XML, JSON

XML a JSON jsou dnes dva nejpoužívanější formáty pro výměnu dat na webu. Přestože jejich účely nejsou stejné, často se používají k provedení stejného úkolu – výměně dat. Oba jsou dobře zdokumentované a oba jsou pro člověka i stroj dobře čitelné. Nedá se s jistotou říci, který z nich je lepší, neboť každý je vhodnější pro jiné druhy použití.

1.4.1 XML

Jazyk XML (Extensible Markup Language) je jednoduchý rozšiřitelný značkovací jazyk odvozený z jazyka SGML (ISO 8879). Standard XML je flexibilní způsob, jak vytvářet informační formáty a elektronicky sdílet strukturovaná data mezi aplikacemi prostřednictvím podnikových sítí nebo Internetu nezávisle na platformě. Aktuální verze je XML 1.1 z roku 2006 a první verze XML 1.0 existuje již v páté revizi. Byl vyvinut a standardizován konsorciem W3C a v mnohém se podobá HTML. Oba jazyky totiž obsahují značky (tagy) pro popis obsahu stránky nebo souboru. XML na rozdíl od HTML neobsahuje informace o způsobu zobrazení, dochází tak k úplnému oddělení formy od obsahu. Tím umožňuje vlastní výběr zobrazení kterékoli aplikaci, která pracuje s XML dokumentem.

Základním stavebním blokem dokumentu XML je *element* (prvek), definovaný párovou značkou. XML podporuje vnořování prvků do jiných prvků. Tím pádem jsou všechny prvky v dokumentu XML obsaženy v nejvzdálenějším prvku – kořenovém prvku (root element). Tato schopnost umožňuje členění do hierarchických struktur. Názvy prvků popisují jejich obsah a struktury popisují vztahy mezi prvky.

[17][18][19]

1.4.2 JSON

JSON (JavaScript Object Notation) neboli JavaScriptový zápis objektů je odlehčený formát pro předávání dat, který se průběhem posledních let začlenil mezi nejpodstatnější formáty webu. Jeho specifikaci najdete v RFC 4627 a vymyslel ho Douglas Crockford v roce 2006. Je snadno čitelný pro člověka, ale i analyzovatelný a generovatelný strojově.

JSON je založen na dvou typech struktur: kolekce párů (název/hodnota) a seřazený seznam hodnot. Jedná se o obecné datové struktury, které všechny moderní programovací jazyky v nějakém tvaru podporují. Logické tedy je, aby na těchto strukturách byl založen na jazyce nezávislý výměnný formát. Do JSON tedy lze uložit nadcházející druhy dat:

- JSONNumber – číslo (celočíselné nebo reálné, včetně zápisu s exponentem),
- JSONString – textový řetězec,
- JSONNull – hodnota null,
- JSONBoolean – logická hodnota,
- JSONArray – pole,
- JSONObject – objekt.

Zbylé datové typy nemůžeme vkládat rovnou, např. datum musíme před vložením do JSON převést na JSONString.

JSON se objevil v době, kdy se na webu pro vyměňování dat využíval hlavně formát XML. V očích javascriptových vývojářů trpěl jazyk XML podstatnými nedostatky. A tak přestože při zápisu celých dokumentů JSON nemůže, a ani nemá, formátu XML konkurovat, v zápisu krátkých strukturovaných dat vyměňovaných webovými aplikacemi konkurenční boj JSON vyhrál.

[20][21]

1.5 Relační databáze

Databázová technologie je unifikovaný soubor pojmů, prostředků a technik, sloužící pro vytváření informačních systémů. Databáze, vyhovující relačnímu modelu dat (RMD) se nazývají relační databáze. RMD je založený na matematické teorii množin a predikátové logice a definuje způsob reprezentace dat, způsob jejich ochrany (integritní omezení) a možné operace nad daty. RMD navrhnul a jeho pravidla publikoval v roce 1970 Edgar Frank Codd (pracovník firmy IBM).

Relační databáze se hojně využívají v aplikacích z důvodu své dobré pochopitelnosti a jednoduchosti. Mezi neznámější zástupce relačních databázových systémů patří MySQL, Oracle,

PostgreSQL, MS SQL a další. Pro webové aplikace menšího rozsahu se nejčastěji používá databáze MySQL ve spojení s PHP.

Základem relačních databází jsou databázové tabulky (entity), které jsou na sebe určitým způsobem závislé. Každá tabulka je tvořena sloupci a řádky, přičemž sloupce reprezentují vlastnosti (atributy) této entity. Každý sloupec musí mít jedinečný název, určený datový typ podle dat v něm uložených (číslo, text, logická hodnota apod.) a doménu, což je množina přípustných hodnot daného atributu. Řádky reprezentují samotné záznamy v databázové tabulce. Každý řádek by měl mít svůj jedinečný identifikátor, podle kterého bude možné určit příslušný záznam tzv. *klíč*. Existují dva základní druhy klíčů:

- Primární klíče – slouží jako jednoznačný (unikátní) identifikátor záznamu (řádku) tabulky.
- Cizí klíče – slouží k vyjádření vztahů mezi jednotlivými tabulkami. Jedná se o atribut nebo skupinu atributů, které umožní identifikovat, které záznamy z různých tabulek spolu souvisejí. Tímto způsobem se tabulky vzájemně propojují.

1.5.1 Normalizace databáze

Normalizací se rozumí proces, ve kterém se snažíme návrh databáze upravovat tak, abychom mohli lépe pracovat a manipulovat s daty, zabránit redundanci dat a zvýšit konzistenci informací napříč databázemi. Pravidla, která by měla být pro určitý stupeň normalizace databáze splněna, se nazývají normální formy. Čím vyšší normální forma, tím lze říci, že je databáze podle teoretických konceptů lépe navržena a mělo by být snadnější s daty pracovat.

- První normální forma (1NF) – každé pole v tabulce by mělo představovat jedinečný typ informace, tedy být atomické – obsahovat pouze jednu nedělitelnou hodnotu.
- Druhá normální forma (2NF) – všechny atributy mají být závislé na primárním klíči. Nelze tedy mít v tabulce atribut, který nesouvisí s primárním klíčem.
- Třetí normální forma (3NF) – všechny neklíčové atributy musí být vzájemně nezávislé.
- Čtvrtá normální forma (4NF) – každý atribut tabulky popisuje pouze jeden fakt nebo souvislost.

[22][23]

1.6 NoSQL databáze

Databáze NoSQL (původně odkazující na „non SQL“ nebo „not only SQL“) poskytuje mechanismus pro ukládání a načítání dat, jinak než pomocí tabulkových schémat použitých v relačních databázích (RDBMS). To zahrnuje širokou škálu různých databázových technologií, které byly vyvinuty díky nárůstu objemu uložených dat, frekvenci přístupů nebo výkonu k potřebnému zpracování. Obecně jsou databáze NoSQL strukturovány jako dvojice klíč/hodnota, grafová databáze, orientovaný dokument nebo sloupcově-orientovanou strukturu.

RDBMS předpokládá dobře definovanou strukturu dat, která jsou do značné míry jednotná. K tomu je zapotřebí předem nadefinovat schéma aplikace i s vlastnostmi (sloupce, datové typy atd.). To nevyhovuje pružnému vývoji vysoce dynamických aplikací. Při větším nárůstu dat budete muset velikost databáze rozšiřovat „svisle“, tzn. přidat větší kapacitu do již existujících serverů, což může být občas dost složitý proces.

Přínosy databází NoSQL oproti RDBMS jsou:

- Bezschématovost – nedefinují přesně danou datovou strukturu.
- Dynamičnost a pružnost – mají dobré tendence se dynamicky vyvíjet dle měnících se požadavků. Tudiž si poradí se strukturovanými, částečně strukturovanými nebo úplně nestrukturovanými daty.
- Změna velikosti „vodorovně“ - přidáním dalších serverů a použití myšlenky „úlomků“ nebo replikací. Toto chování vyhovuje použití s cloudovými službami, které využívají virtuální servery, které lze rozšířit horizontálně kdykoliv je potřeba.
- Vyšší výkon – všechny NoSQL databáze tvrdí, že mají lepší a rychlejší výkonost ve srovnání s tradiční implementací RDBMS.

[24]

2 TRENDY VÝVOJE WEBOVÝCH APLIKACÍ

2.1 Framework

Frameworkem neboli rozhraním je myšlena softwarová knihovna, která poskytuje základní strukturu vývoje aplikací pro konkrétní prostředí. Framework funguje jako kostra pro tvorbu aplikací. Záměrem návrhu frameworků je snížit výskyt běžných problémů, kterým se během vývoje aplikací čelí, a tím usnadnit samotný vývoj aplikace. Toho je dosaženo prostřednictvím již hotového kódu, který může být sdílen mezi různými moduly bez nutnosti řešení vlastní implementace.

2.1.1 Model view controller (MVC)

Hlavní myšlenka MVC architektury je rozdělení výstupu od logiky. Celá aplikace je rozdělena na 3 komponenty: model, view (pohled) a controller (kontrolér), od toho zkratka MVC.

Model obsahuje logiku a vše, co do ní spadá. Mohou to být výpočty, databázové dotazy, validace apod. Jeho funkce spočívá v přijetí parametrů zvenku a vydání dat ven. Nicméně neví, odkud data přišla a ani jak budou výstupní data zformátována a vypsána.

Pohled (View) se stará o zobrazení výstupu uživateli. Nejčastěji se jedná o šablonu, obsahující HTML stránku a tagy nějakého značkovacího jazyka, který umožňuje do šablony vkládat proměnné, případně provádět iterace (cykly) a podmínky. Pohledů je vždy více, pro každou funkcionalitu jeden. Šablony lze samozřejmě vkládat do sebe, a tím vytvářet složitější layout. Pohled není jen šablona, ale zobrazovač výstupu. Obsahuje tedy minimální množství logiky, která je pro výpis nutná. Podobně jako model vůbec neví, odkud mu data přišla, stará se jen o jejich zobrazení uživateli.

Kontrolér je nyní onen chybějící prvek, který osvětlí funkčnost celého vzoru. Jedná se o jakéhosi prostředníka, se kterým komunikuje uživatel, model i pohledy. Drží tedy celý systém pohromadě a komponenty propojuje.

[27]

2.2 Responzivní web design (RWD)

Poprvé tento výraz použil americký programátor Ethan Marcotte v článku na blogu A List Apart 25. května 2010. Během necelých 3 let vstoupil do povědomí většiny internetových vývojářů. Jedná se tedy o způsob stylování HTML dokumentu, které zajistí zobrazení stránky tak, že bude optimalizováno pro všechny druhy nejrůznějších zařízení (mobilní telefony, notebooky, tablety

apod.). To je zajištěno především díky vlastnosti Media Queries (*@media*), zahrnuté ve specifikaci CSS3, jež rozpozná vlastnosti zařízení, na kterém je stránka prohlížena a přizpůsobí tak samotnou stránku a její obsah.

Zásady responzivního designu:

- Šířka objektů v procentech. Jelikož web není vytvořen pro jedno konkrétní rozlišení koncového zařízení, musí být šířka elementů dána procentuální šířkou rodičovského elementu, ve kterém se element nachází.
- Rozložení elementů na stránce. Rozmístění bloků musí reagovat na rozlišení prohlížeče. Bloky, které jsou na počítači zobrazeny vedle sebe, se na mobilním zařízení zobrazí pod sebou.
- Rozdílná velikost obrázků. Používá se rozdílná velikost stejného obrázku s popisem jeho rozlišení a prohlížeč si již sám vybere, který obrázek ze serveru stáhne, aby nedocházelo ke zbytečně nadměrnému přenosu dat. Popřípadě lze uvést procentuální šířku obrázku k danému elementu a výška je dopočítána automaticky.

[28]

2.3 Search Engine Optimization (SEO)

Popisuje řadu technik, které zlepšují viditelnost webu pro webové vyhledávače. Cílem takové optimalizace je získat ve výsledcích vyhledávání lepší pozici. Když se řekne webový vyhledávač, obvykle je tím myšlen Google, jelikož jeho tržní podíl činí 91 %, což z něj dělá nejdůležitější webový vyhledávač.

Internet je neuvěřitelně velký a v současnosti existuje přibližně 50 miliard indexovaných webových stránek a bez vyhledávačů by bylo nemožné najít užitečné informace. Proto vytvořením indexu mohou vyhledávače najít nejbližší webové stránky pro hledaný dotaz. Ale tento index se často mění, protože se každý den přidávají nové webové stránky, starší se přeprocovávají, nahrávají se nové soubory atd. Jelikož je internet dynamická entita, vyhledávače potřebují nástroj, který pomůže v daném indexu držet aktuální informace. Tento nástroj se nazývá „crawler“, což je vyhledávací robot, který prochází webové stránky a shromažďuje informace – indexuje odkazy, obrázky, videa a další soubory. Vyhledávače používají systém hodnocení (rank), aby zobrazili nejrelevantnější výsledky ze svých indexů. Existuje přes 200 faktorů, které ovlivňují dané hodnocení a jsou zásadní pro úspěch SEO. Mezi základní faktory patří:

- Odkazy – tzv. PageRank (algoritmus používaný společností Google) bere v úvahu množství odkazů. Každý odkaz na stránky je viděn jako jeden hlas, tedy čím více odkazů se na webu nachází, tím vyšší bude i hodnocení. Bohužel to není demokratický systém, neboť všechny odkazy nemají stejnou váhu. Odkazy z autoritativních webů mají větší váhu než odkazy z méně oblíbených webů.
- Klíčová slova – Google kontroluje, jak často se vyhledávaný dotaz na stránce nalézá. Pokud se slovo objeví v názvu domény, v URL, ve značkách v záhlaví nebo v obsahu, bude stránka považována za více relevantní.
- Stáří webu – Google považuje starší webové stránky za více autoritativní než nové webové stránky.
- Aktuálnost – Časté aktualizace webu jsou znamením, že webová stránka není „mrtvá“.
- Další faktory – struktura stránek, rychlost webu, AuthorRank a mnoho dalších.

[29]

2.4 Progressive Web Apps (PWA)

PWA jsou kombinací toho nejlepšího z webu a toho nejlepšího z mobilních aplikací. Nemusí se instalovat, tudíž jsou pro uživatele rychle a pohotově dostupné již od první návštěvy v prohlížeči. Jsou to tedy v podstatě webové stránky vytvořené s použitím moderních webových technologií, ale působící a cítící se jako mobilní aplikace. Termín „Progressive Web App“ v roce 2015 vytvořili Alex Russell, inženýr Googlu, a Frances Berriman. Charakteristiky této nové třídy webových aplikací jsou:

- Progresivita. Fungují pro každého uživatele, bez ohledu na volbu prohlížeče.
- Responzivní vzhled. Uživatelské rozhraní sedí pro jakékoliv zařízení: stolní počítač, mobilní zařízení, tablet nebo cokoliv bude následovat.
- Nezávislost připojení. Mohou pracovat offline nebo v místech se slabým internetovým připojením.
- „App-like.“ Pro uživatele působí jako mobilní aplikace díky stejnému stylu ovládání a navigaci, protože jsou postaveny na struktuře mobilních aplikací.
- Aktuálnost. Jsou vždy aktuální po připojení k internetu.
- Bezpečnost. Komunikace prostřednictvím protokolu HTTPS, aby bylo zajištěno, že obsah nebyl nikým zfalšován.
- Zjistitelnost. Identifikovány jsou jako aplikace a tím pádem zjistitelné pomocí vyhledávačů.

- „Znovu-zaujetí.“ Prostřednictvím funkcí, jako jsou například oznámení, nutí uživatele k opětovnému použití aplikace.
- Uchovatelnost. Umožňuje uživatelům uchovat si PWA přímo na své domovské obrazovce bez použití obchodu s aplikacemi.
- Propojenost. Snadné sdílení prostřednictvím adresy URL bez složitých instalací.

[30][31]

3 UNIFIKOVANÝ PROCES VÝVOJE APLIKACÍ

3.1 Unified Modeling Language (UML)

Unifikovaný modelovací jazyk je univerzální jazyk pro vizuální modelování systémů. Byl navržen pro spojení toho nejlepšího z existujících postupů modelovacích technik a softwarového inženýrství. Není vázán na žádnou specifickou metodiku nebo životní cyklus, tudíž jej lze použít společně se všemi existujícími metodami. V roce 1997 sdružení OMG (Object Management Group) jazyk UML přijalo a první průmyslový standard objektově orientovaného jazyka pro vizuální modelování byl na světě. Následně byl jazyk v roce 2000 významně rozšířen. A to především o sémantiku akcí, která poskytuje možnost dokončit specifikaci UML výpočetně, což v konečném důsledku umožňuje modely UML spustit. Okolo roku 2005 se objevuje nová specifikace a to UML 2.0, která přináší mnoho nových prvků vizuální syntaxe. Druhá verze jazyka se dá již považovat za velmi vyspělý modelovací jazyk.

Struktura jazyka UML obsahuje tyto součásti:

- **Stavební bloky** – jsou základními prvky modelu, které se dělí na:
 - **Předměty**: samotné prvky modelu.
 - **Relace**: vztah mezi předměty (určují, jak spolu dva nebo více předmětů významově souvisí). Druhy relací: závislost, asociace, agregace, kompozice, zobecnění, realizace.
 - **Diagramy**: „Pohledy“ na model. Dělí se na diagramy struktury (diagram tříd, objektový diagram, ...) a diagramy chování (diagram aktivit, diagram případů užití, ...).
- **Společné mechanismy**: obecné způsoby k dosažení specifických cílů
 - **Specifikace** – popis funkcí a sémantiky jednotlivých prvků
 - **Ornamenty** – informace o prvku modelu
 - **Podskupiny** – klasifikátor, instance, rozhraní a implementace
 - **Mechanismy rozšíření** – omezení, stereotypy a označené hodnoty
- **Architektura**: zachycuje strategické aspekty systému v architektuře „4-1“:
 - **Logický pohled** – funkce systému
 - **Pohled procesů** – výkon systému
 - **Pohled implementace** – montáž systému
 - **Pohled nasazení** – topologie systému
 - **Pohled případů užití** – požadavky uživatele (sjednocení všech pohledů)

3.2 Unified Process (UP)

Je vyspělý otevřený standart pocházející od autorů jazyka UML z roku 1967. Metodika UP je obecnou metodou tvorby softwaru. Pro každou organizaci stejně tak pro každý jednotlivý projekt je potřeba vytvořit novou instanci metodiky UP, neboť je do ní třeba zapracovat vnitropodnikové standardy, šablony dokumentů, databáze apod.

Software je zde vytvářen v tzv. iteracích. Každá iterace je vypadá jako samostatný „miniprojekt“, který vytváří dílčí součásti do vznikajícího systému. Iterace jsou skládány jedna za druhou, čímž vytváří podobu konečného systému. V každé iteraci existuje pět základních pracovních postupů (workflows):

- Požadavky: zachycují to co by měl systém dělat.
- Analýza: upřesnění požadavků a jejich strukturování.
- Návrh: realizace požadavků v architektuře systému.
- Implementace: tvorba softwaru.
- Testování: ověření, zda implementace funguje tak, jak se do ní očekává.

Samotná metodika UP se skládá ze čtyř fází, z nichž každá končí definovaným milníkem:

- Zahájení: start projektu; milník: záměry životního cyklu.
- Rozpracování: vznik architektury systému; milník: architektura životního cyklu programového díla.
- Konstrukce: vytváření softwaru; milník: počáteční funkční varianta
- Zavedení: nasazení softwaru do uživatelského prostředí; milník: nasazení produktu

Metodika RUP (Rational Unified Process) je komerčním rozšířením metodiky UP. S touto metodou je zcela kompatibilní, nicméně však je úplnější a podrobnější. Obsahuje veškeré standardy, nástroje a další nezbytnosti, které nejsou součástí metodiky UP, ale které by si uživatel stejně musel opatřit. Navíc je dodávána společně s bohatým uživatelským prostředím, doplněným o dokumentaci, a rádcem k jednotlivým nástrojům.

3.3 Funkční a nefunkční požadavky

Požadavky jsou základem všech systémů. Lze je definovat jako „specifikaci toho, co by mělo být implementováno“. Požadavky by měly být vyjádřením toho, co by měl systém dělat, nikoliv toho, jak by to měl dělat.

Existuje mnoho způsobů kategorizace požadavků, ale pro zjednodušení se vyplatí rozdělit požadavky jen na funkční a nefunkční. Funkční požadavek je formulace toho, co by měl systém dělat, tedy popis požadované funkce: *system bude ověřovat validitu zadaného hesla*. Nefunkční požadavek je omezující podmínka uvalená na daný systém: omezující způsob, jímž bude systém implementován: *system ověří validitu hesla z databáze maximálně do tří sekund*.

Každý požadavek může obsahovat určitou množinu atributů, jež zachycují dodatečné informace týkající se daného požadavku. Mezi nejběžnější atribut patří *priorita*. Hodnotou je relativní priorita požadavku vůči ostatním požadavkům. Hodnoty mohou být následovné:

- Nezbytný: povinné požadavky, které jsou základem systému.
- Možný: důležité požadavky, které lze vynechat.
- Eventuální: nepovinné požadavky, které se implementují, pokud zbyde čas.
- Chceme mít: požadavky do dalších verzí systému.

Pro zápis požadavků se doporučuje velmi jednoduchý formát: *<id> <system> bude <funkce>*. Každý požadavek v něm má jednoduchý identifikátor, klíčové slovo *bude*, a příkaz funkce.

3.4 Případy užití

Modelování případů užití je jednou z forem inženýrství požadavků. Skládá se z následujících aktivit:

- Nelezení hranic systému (subjekt): ohraničení kolem případů užití.
- Vyhledání aktérů: role přidělené osobám nebo předmětům používajícím daný systém
- Nalezení případů užití (specifikace, alternativní scénáře): činnosti, které mohou aktéři se systémem vykonávat
- Opakování postupu, dokud nedojde k ustálení případů užití, aktérů a hranic systému.

Případ užití je definován jako specifikace posloupností činností, včetně proměnných a chybových posloupností, které systém může vykonat prostřednictvím interakce s vnějšími aktéry. Je vždy iniciován aktérem a vždy napsán z pohledu aktéra. Pro specifikaci případů užití neexistuje žádný standard UML. Obvykle se však používá šablona obsahující následující informace: název, jedinečný identifikátor, stručný popis, aktéři, vstupní podmínky, hlavní scénář, výstupní podmínky a alternativní scénáře.

3.5 Diagramy aktivit

Diagramy aktivit jsou objektivě orientovanými vývojovými diagramy. Díky nim lze procesy modelovat jako aktivitu, která se skládá z kolekce uzlů spojených hranami. Jsou obvykle připojeny k případům užití, třídám, rozhraním, komponentám nebo operacím. Lze je rovněž použít k modelování obchodních procesů nebo pracovních postupů. Zcela jedinečná je schopnost modelovat proces bez nutnosti specifikace statické struktury tříd a objektu realizujících daný proces. Hodí se zejména v raných fázích analýzy, kdy se snažíme zjistit, o jaký proces vlastně jde.

Aktivita se skládají ze sítí uzlů spojených hranami. Rozlišuje tři kategorie uzlů:

- Akční uzly zastupují samotné jednotky, jež jsou v rámci aktivity nedělitelné.
- Řídící uzly řídí cestu uvnitř aktivity.
- Objektové uzly zastupují objekty použité v rámci dané aktivity.

A dvě kategorie hran:

- Řídící hrany zastupují postup řízení v rámci aktivity.
- Objektové hrany zastupují cestu objektů v rámci aktivity.

[32]

4 T. K. PRŮZKUMNÍK

Táborský klub Průzkumník je nezisková volnočasová organizace pro děti, spadající od roku 1990 pod Českou táborskou unii (ČTU). Klub byl založen v roce 1969, tehdy jako radiotechnický kroužek čítající přibližně 12 členů. Dnes je jeho náplň již úplně jiná a klub čítající okolo 60 dětí a 30 aktivních vedoucích je zaměřen převážně na kolektivní hry, poznávání přírody a pořádání každoročních klubových akcí (N300, Vlajková hra, Uzlařská regata) a táborů. Celkem oddílem prošlo na více jak 450 členů a bylo uspořádáno přes 700 oddílových akcí počítaných od roku 2000.

4.1 Historie webu

První verze webového portálu T. K. Průzkumník se datuje do roku 1997. Web běžel na doméně *kolin.cz/pruzkumnik*, ale pouze jako statická prezentace obsahující jen pár nejdůležitějších informací ohledně fungování klubu (data výprav, schůzek, táborů, kontakty, ...). V roce 1998 byla zakoupena dodnes používaná doména *pruzkumnik.cz*. Web byl uložen na serveru Gymnázia Kolín, který byl ale připojen pomalou linkou, proto se roku 1999 přesunul na hosting *moment.cz*, který vlastnila společnost z Kolína. Po přesunu se zde začaly vyskytovat první věci používající PHP3, takže se ze statické prezentace začala stávat dynamická aplikace. Jako první dynamický objekt zde byly *vzkazy čtenářů*. Tehdy ještě nebyla použita žádná databáze pro ukládání dat, nýbrž se data ukládala do prostého textového souboru. V roce 2000 byla přidána funkcionality *Koná se právě teď*, která informovala o nadcházející akci nebo výpravě. Obsahovala i seznam e-mailů (rodičů, vedoucích...), podle kterého byly rozesílány e-mailová upozornění právě o dané akci nebo výpravě, což byla na tu dobu velice nadčasová záležitost. Prvně takto založená výprava byla v září roku 2000. V pozdější době se všechna data dynamicky zobrazovala z databáze, nicméně administrace vznikla mnohem později, takže *phpmyadmin* byl nepostradatelným nástrojem při vývoji. Pro zjednodušení práce existovaly tabulky v Microsoft Excel, které generovaly SQL kód pro vkládání do databáze. Tudíž v začátcích web vznikal převážně v Excelu. V následujících letech se postupně přidávaly další dynamické funkcionality: *ankety, fotogalerie, slavné výroky, kalendář akcí, kroniky, N300, 300 nanečisto, akce návrat*, a nakonec nástroj pro správu *můjPrůzkumník* a s tím spojený vznik uživatelů, práv, agendy klubu a bodování členů. Okolo roku 2010 byl web dosti promazán od nepotřebných a nevyužívaných zbytečností. V průběhu psaní této práce web přešel na zabezpečenou komunikaci pomocí protokolu HTTPS.

Web začal vznikat jako nástroj pro lepší prezentaci a propagaci klubu. Následně po přidání *můjPrůzkumník* i pro lepší a snadnější organizaci klubu a jako jednotné místo pro ukládání dat (fotky, odkazy, kalendářní data, bodování apod.). Nejdříve byl tedy určen pro veřejnost, hlavně pro rodiče dětí, a následně fungoval také jako nástroj vedení klubu pro lepší a centralizovanou správu.

4.2 Odchylky oproti dnešním trendům

Jelikož web vznikal ještě v minulém tisíciletí a postupem času na něj byly nalepovány další a další funkcionality, navíc nebyl nikdy kompletně předělán, najde se zde mnoho prvků typických pro dílčí historické etapy vývoje webových stránek.

Jednou z nejvíce zastaralých metod, která se v některých částech kódu nachází, je použití stylování přímo v zápisu html tagu (značky). Tím vzniká složitě vypadající a dosti nepřehledný kód. Postupem času se přešlo na použití CSS, ale to jen částečně. Samotný soubor obsahující CSS se zaměřuje spíše na globálnější stylování, nicméně mnoho stylů zůstává právě u jednotlivých tagů. Tím vzniká chaotické a decentralizované stylování webové stránky, ve kterém je obtížné dělat změny. Dnešní snahou je naopak rozdělit obsah od grafického zobrazení pro lepší přehlednost kódu a již zmiňované centralizace stylování pro lepší a snazší úpravy grafiky.

Dalším velkým nešvarem je rozvržení stránky. Celý web je rozdělen na nespočet html tabulek, řádků a sloupců, které si s sebou nesou základní vlastnosti o velikosti, šířce, mezerách mezi buňkami apod. Tím znovu vzniká obrovský a nepřehledný kód, ve kterém se nedá rozumně vyznat. Tato metoda rozvržení do tabulek se používala opravdu dávno, byla vystřídána rámečky (frames), ale i ty jsou dnes již zastaralou a nepoužívanou metodou. Dnes se rozvržení definuje převážně bloky, které jsou následně rozmístěny pomocí CSS. V lepší variantě se tak děje jen procentuálně vůči nadřazeným blokům, abychom docílili responzivního designu, který na tomto webu budeme hledat velice obtížně, protože je vše definováno staticky v jedné velké tabulce.

Velkým plusem je zde naopak viditelnost webu pro vyhledávače, protože ty převážně hodnotí stáří stránek, aktuálnost, odkazy a klíčová slova. Web existuje již 20 let, každý týden se zde aktualizují nové informace o výpravách, akcích, ankety apod., stránky zdaleka nejsou neobsáhlé, takže odkazů na projití je opravdu mnoho a textů stejně tak.

V dnešní době, kdy každý vlastní chytrý mobilní telefon, je trochu na škodu, že webová stránka není přizpůsobivá mobilním zařízením nebo alespoň neexistuje její jednodušší varianta jako mobilní aplikace, popř. PWA.

4.3 Stávající funkcionalita

Nejlepší ukázkou a přiblížením stávající funkcionality bude rozvrhnout si ji na funkční požadavky, jako kdybychom web chtěli teprve vytvářet:

1. Web bude obsahovat základní informace o klubu a kontakty.
2. Web bude obsahovat informace o schůzkách družin.
3. Web bude obsahovat kalendář akcí.
4. Web bude obsahovat informace o nadcházející výpravě.
5. Web bude obsahovat bodování:
 - a. Bodování po družinách,
 - b. Celkové bodování.
6. Web bude obsahovat informace o akcích pořádaných klubem a jejich výsledky,
 - a. Nouzovská třístovka (N300),
 - b. Uzlařská regata,
 - c. Vlajková hra.
7. Web bude umožňovat hrát hru „N300 nanečisto“.
8. Web bude obsahovat ankety čtenářů.
9. Web bude obsahovat vzkazy čtenářů.
10. Web bude obsahovat veřejné dokumenty ke stažení.
11. Web bude obsahovat slavné výroky členů.
12. Web bude obsahovat fotogalerii.
13. Web bude obsahovat upoutávky na titulní straně.
14. Web bude obsahovat administrační nástroj pro správu klubu můjPrůzkumník.
15. můjPrůzkumník bude obsahovat správu osobních údajů a hesla.
16. můjPrůzkumník bude obsahovat správu titulních upoutávek.
17. můjPrůzkumník bude obsahovat přehled, zápis a editaci bodování členů.
18. můjPrůzkumník bude obsahovat správu fotogalerie.
19. můjPrůzkumník bude umožňovat přidání členů, jejich správu a hledání.
20. můjPrůzkumník bude umožňovat odeslání e-mailu pro celé vedení.
21. můjPrůzkumník bude umožňovat přidání akce nebo výpravy.
22. můjPrůzkumník bude umožňovat přidání zápisu do kroniky.
23. můjPrůzkumník bude obsahovat přehled o členech vaší družiny.
24. můjPrůzkumník bude obsahovat skupiny s rozdílnými oprávněními.
25. můjPrůzkumník bude umožňovat přidání registrací na akce.

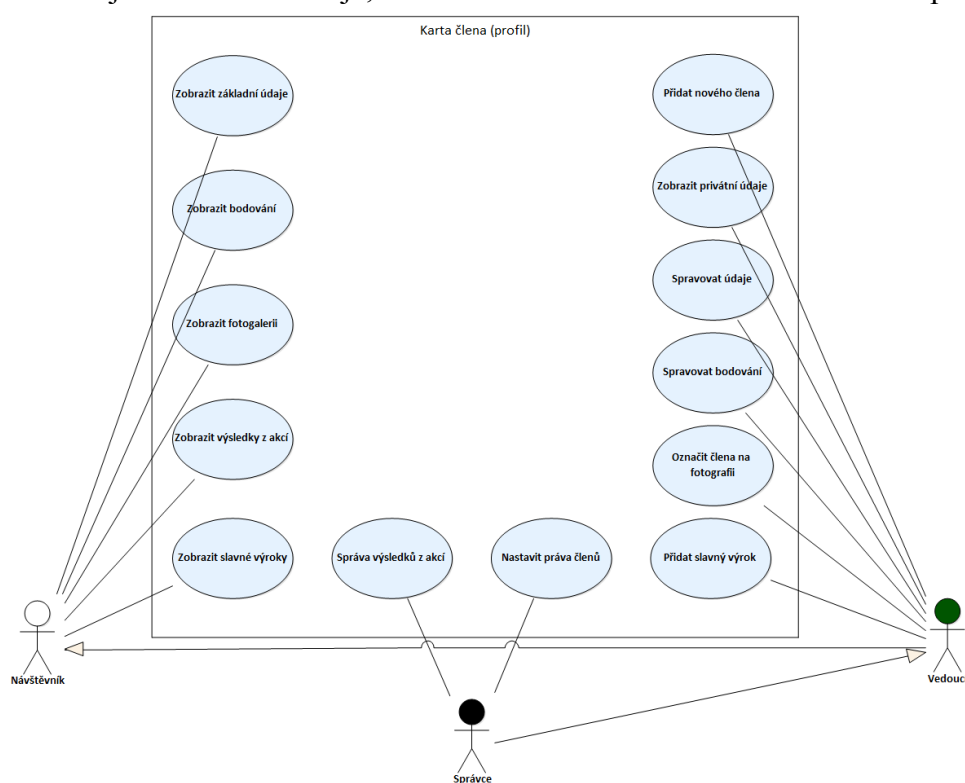
26. můjPrůzkumník bude obsahovat správu interních dokumentů.
27. můjPrůzkumník bude umožňovat rezervaci oddílového srubu.
28. můjPrůzkumník bude umožňovat přidání slavných výroků.
29. můjPrůzkumník bude obsahovat zápisy z porad a akcí a jejich správu.

4.4 Diagramy případů užití a aktivit

Při souhrnu toho, co web obsahuje a jaké má nedostatky nebo odchylky od dnešních trendů, je zapotřebí udělat návrh inovací, které vylepší stávající funkcionalitu a usnadní tak její používání.

Jednou z větších inovací je *Karta člena* neboli profil člena, viz Obrázek 1.

Něco podobného již na webu existuje, nicméně ve více decentralizované a nekompletní formě.



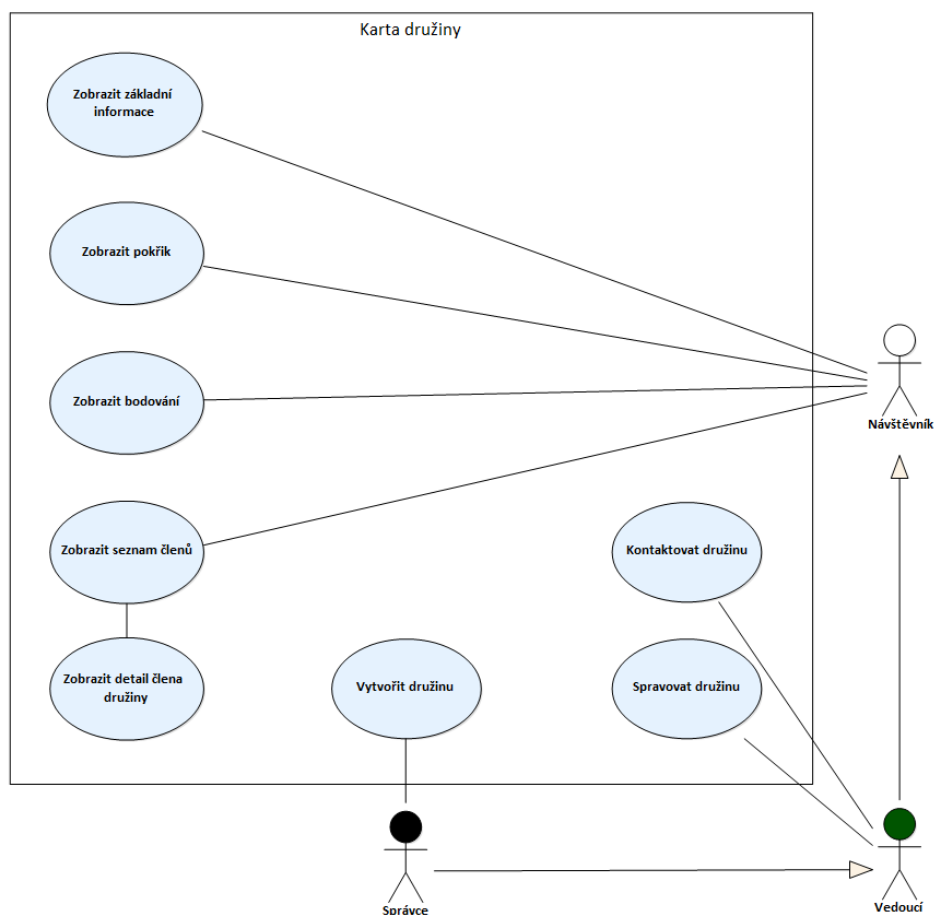
Obrázek 1 – Diagram případu užití (Karta člena)

Karta člena je rozdělena na veřejnou a neveřejnou část, která je přístupná jen přihlášeným uživatelům (vedoucím). Veřejná část obsahuje základní údaje o členovi: jméno, příjmení, přezdívkou, profilová fotografie, současná družina, seznam absolvovaných táborů, seznam absolvovaných výprav. Dále je zde možnost zobrazit bodování, které se dělí na bodování schůzek a bodování výprav, a je seřazeno podle měsíců. Jelikož je profil člena propojen s označováním na fotografiích, lze zobrazit všechny fotografie, na kterých je člen označen, i s informacemi, o jakou výpravu nebo akci se jedná. Další možností je zobrazení osobních výsledků z akcí jako

je N300, Uzlařská regata, Vlajková hra a další. Poslední sekci, kterou lze ve veřejné části zobrazit, jsou slavné výroky právě prohlíženého člena.

Neveřejná (privátní) část obsahuje převážně moduly pro administraci jednotlivých sekcí veřejné části. Jednou z nejdůležitějších funkcí je přidání nového člena. Tato funkce umožňuje tisk přihlášky skládající se z údajů potřebných pro zadání člena do systému i samotné přidání člena. K základním údajům o členovi z veřejné části přibývají ještě privátní údaje obsahující datum narození, rodné číslo, informace o rodičích a kontakty, adresu, informaci o zaplacených příspěvcích a poznámky. Vedoucí může spravovat všechny údaje o členovi, ať už veřejné nebo neveřejné. Stejně tak dodatečně upravit členovi bodování, přidat slavný výrok nebo člena označit na fotografii. Do neveřejné části také patří editace a správa výsledků akcí a nastavení práv členů (dítě, vedoucí, hlavní vedoucí a další). Toto má na starosti uživatel s právy správce.

Další z inovací navazující na *Kartu člena* je *Karta družiny*, viz Obrázek 2. *Karta družiny* se stejně jako *Karta člena* rozděluje na veřejnou a neveřejnou část. Ve veřejné části jsou zobrazeny



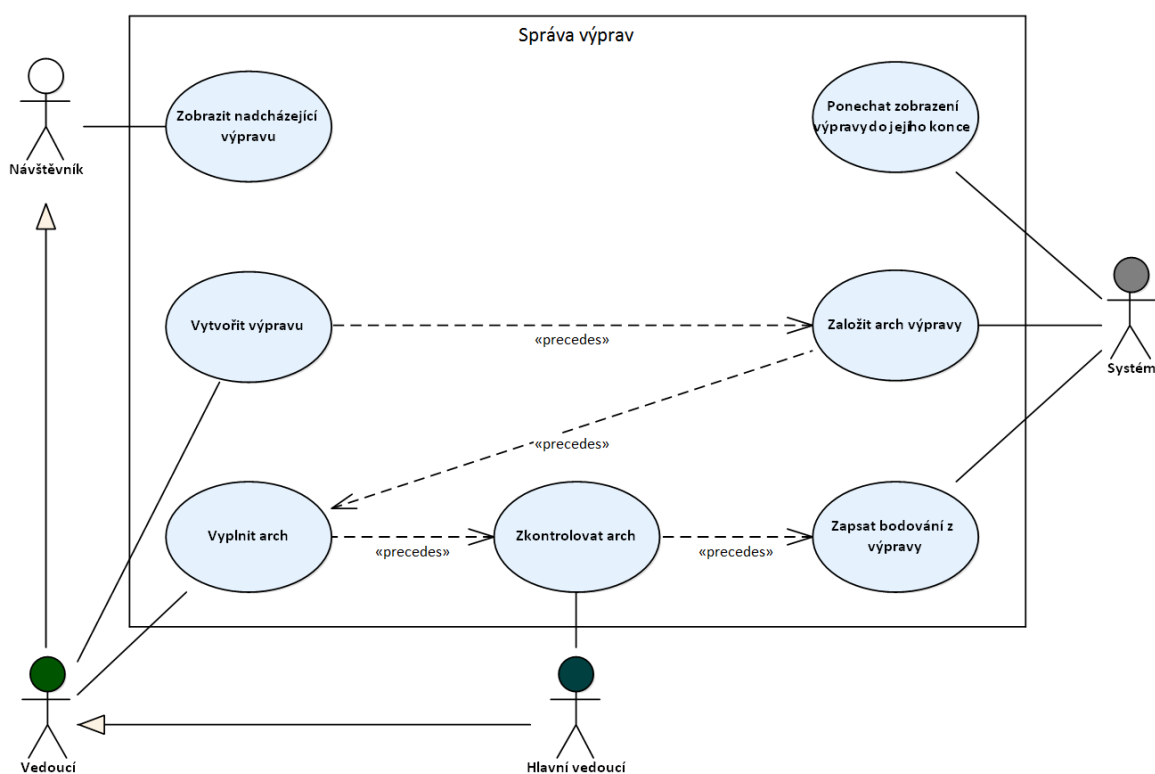
Obrázek 2 – Diagram případu užití (Karta družiny)

základní údaje o družině: název, termíny schůzek a vedoucí družiny. Poté družinový pokřik, bodování v dané družině a seznam členů. Při rozkliknutí člena ze seznamu se zobrazí jeho karta

(profil). Neveřejná část obsahuje správu družiny a hromadné kontaktování členů dle primární e-mailové adresy uvedené u každého z členů. Poslední funkcí v neveřejné části, kterou může použít znovu pouze uživatel s právy správce, je přidání nové družiny.

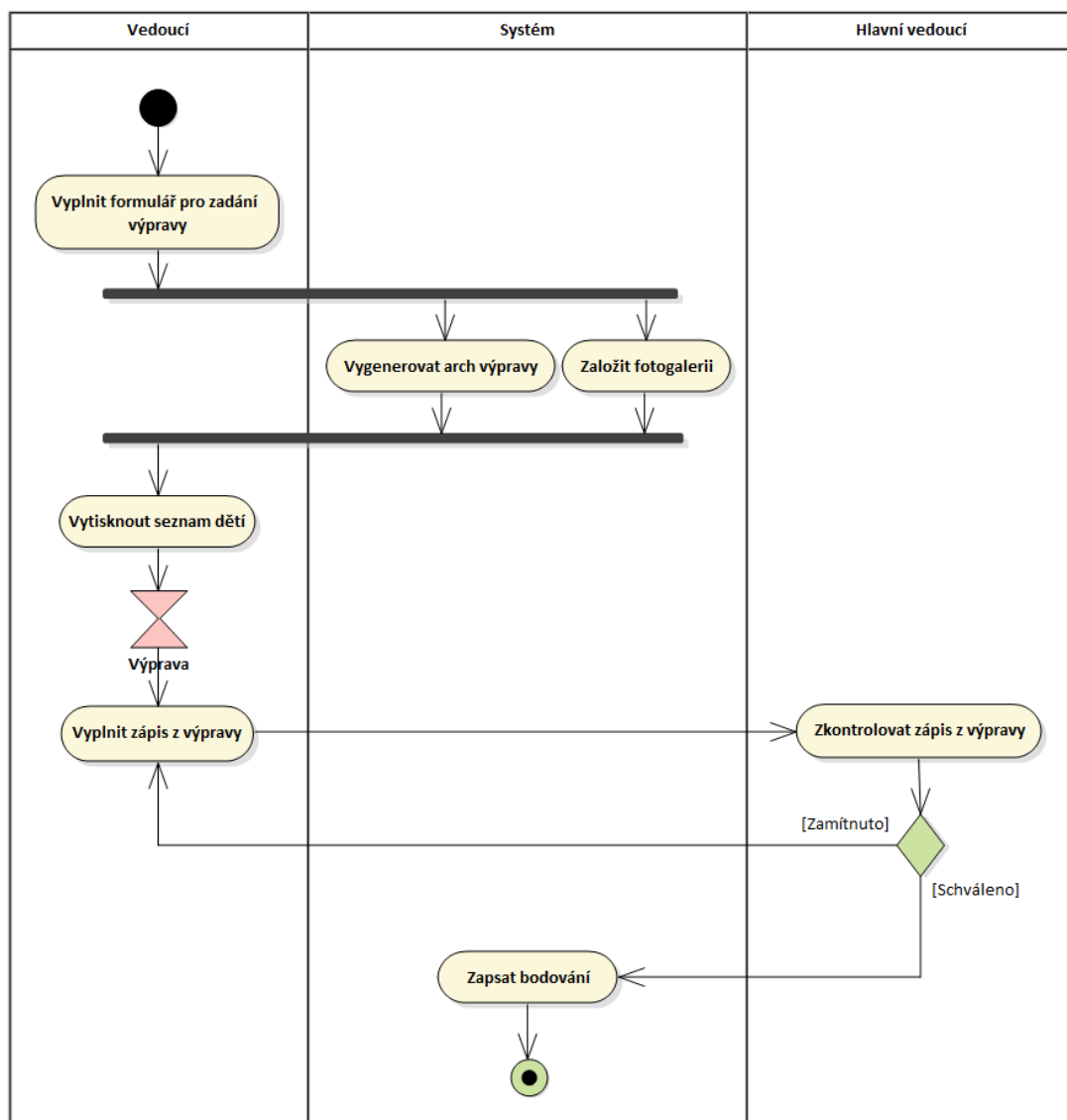
Velkou změnu přinese nový systém procesu organizace výpravy od jejího založení až po přiřazení bodování. Dnešní podoba vypadá zhruba následovně: vedoucí založí výpravu, vytiskne si arch výpravy se seznamem dětí, na výpravě vyplní účast, účast doma přepíše do e-mailu a odešle vedoucím, aby mohli zapsat bodování. Následně předá arch hlavnímu vedoucímu, který arch přepíše do systému. Vedoucí pak ještě zapíše zápis z výpravy, kroniku a přidá fotografie, a to vše na různých místech.

Nový systém výprav (viz Obrázek 3 a Obrázek 4) bude vypadat následovně: vedoucí založí výpravu obdobně jako doposud. Ta se zobrazí na úvodní stránce a bude zde až do konce trvání akce. Automaticky se vygeneruje arch výpravy, ve kterém lze vytisknout seznam dětí pro lepší zapsání účasti přímo na výpravě. Navíc tento arch v elektronické podobě obsahuje rovnou zápis z výpravy (trasa, kilometry, vybraná částka, zbylá částka a další poznámky), kroniku a přímé vložení fotografií, které se přidají do automaticky vygenerované fotogalerie dle názvu výpravy. Následně hlavní vedoucí zkontroluje arch a po odsouhlasení správnosti údajů se účastníkům



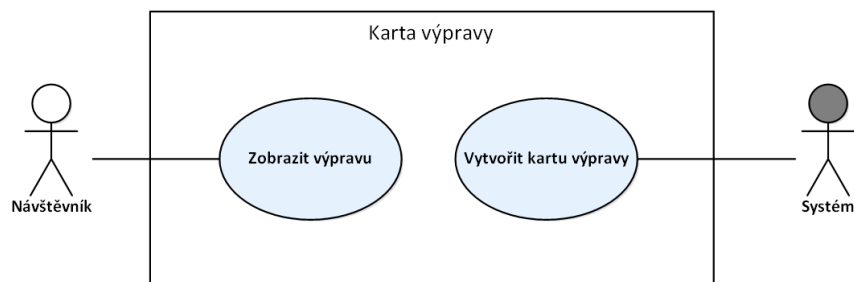
Obrázek 3 – Diagram případu užití (Správa výprav)

automaticky zapíše bodování. Celým tímto systémem se eliminuje komunikace mezi jednotlivými vedoucími a následná chybovost při předávání informací. Celý proces se zjednodušuje a centralizuje.



Obrázek 4 – Diagram aktivit (Systém výpravy)

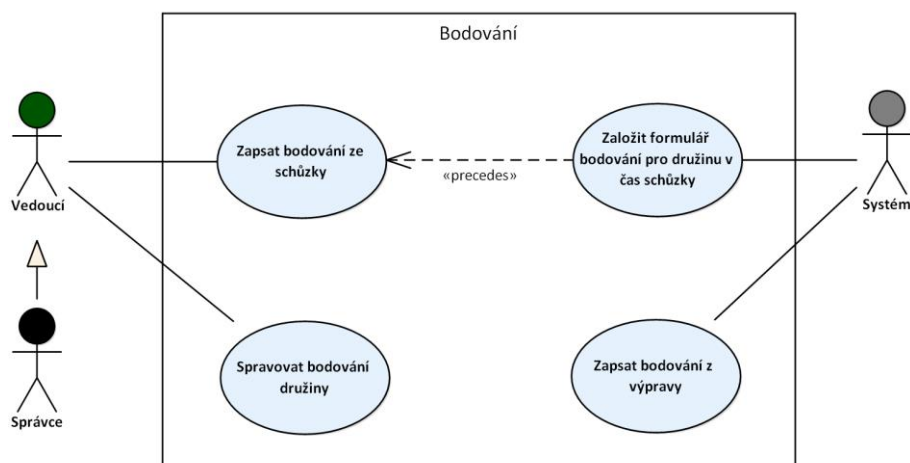
K novému systému výprav bezpochyby patří i *Karta výpravy* (viz Obrázek 5), která bude obdobou karet člena nebo družiny. Systém kartu automaticky vytvoří po akceptovaném archu dané výpravy.



Obrázek 5 – Diagram případu užití (Karta výpravy)

Veřejná část je kombinací základních údajů o výpravě zadaných při vytvoření (název, počet bodů do bodování, místo, čas srazu a návratu, místo srazu a návratu, věci s sebou a další poznámky), kroniky, zápisu z výpravy (účastníci, trasa a kilometry) a fotogalerie.

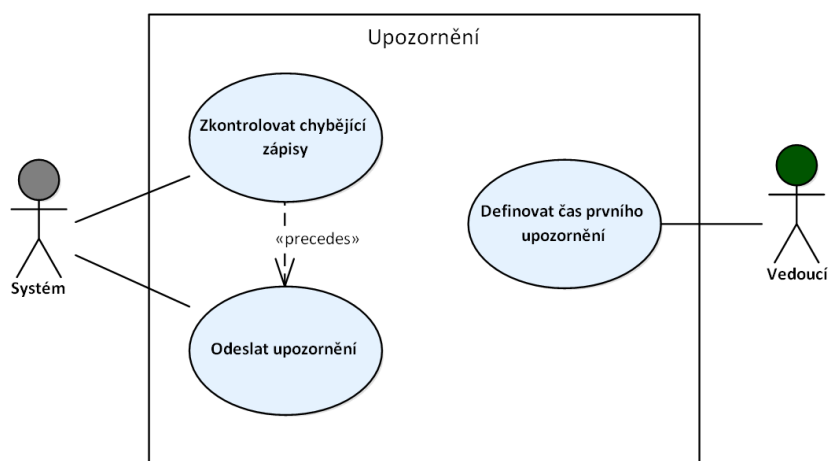
Další podstatnou změnou je předělaný systém bodování (viz Obrázek 6).



Obrázek 6 – Diagram případu užití (Bodování)

Jak už bylo řečeno, body z výprav jsou přidělovány automaticky dle účasti na dané výpravě. Bodování schůzek je na velmi podobném systému. V den a čas schůzky systém automaticky vygeneruje formulář schůzky, který obsahuje datum schůzky a seznam dětí dané družiny. Vedoucí pak následně jen vyplní účast na schůzce, stejně jako u účasti na výpravě, a body jsou automaticky přiděleny systémem. Vedoucí se může kdykoliv k formuláři vrátit a bodování upravit.

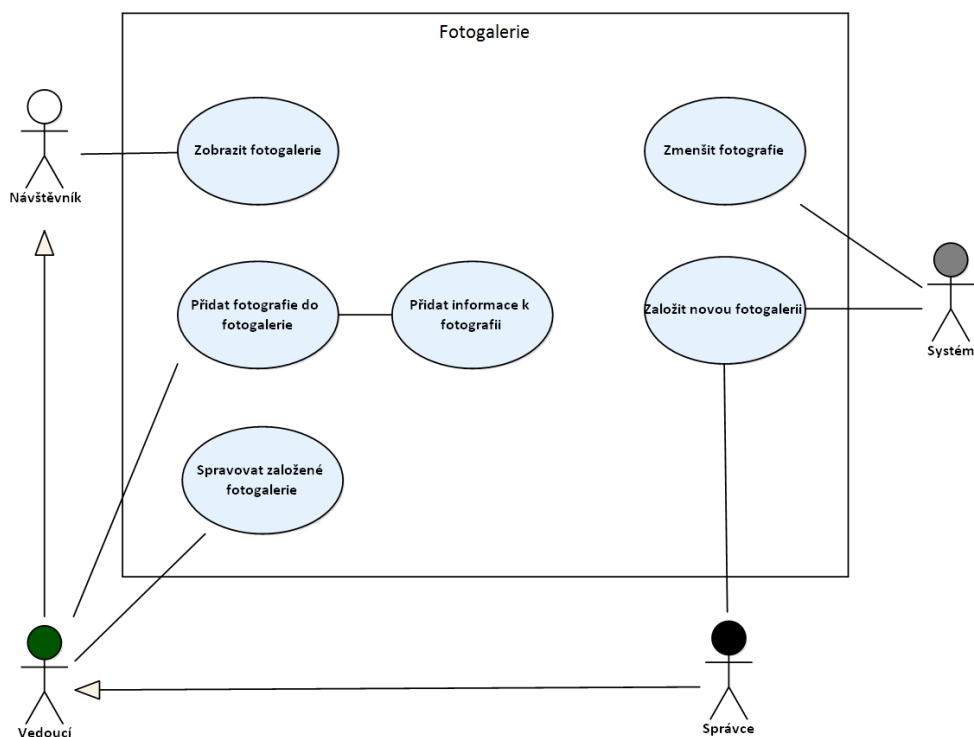
Aby vedoucí na vyplňování formulářů ze schůzek a archů z výprav nezapomínali, je zde nová inovace – *Upozornění* (viz Obrázek 7). Tato funkce automaticky kontroluje po určitém čase,



Obrázek 7 – Diagram případu užití (Upozornění)

zda vedoucí nezapomněl vyplnit některý z formulářů. Čas prvního upozornění je uživatelsky definovatelný, a to v maximálním rozsahu: u schůzek (1 den, 2 dny, 3 dny, týden a měsíc) a u výprav (1 den, 2 dny, 3 dny a týden). Po prvním upozornění je každý následující den odesláno další upozornění, dokud není formulář vyplněn. Upozornění může chodit na e-mail nebo SMS zprávou na mobilní telefon, což závisí na osobních preferencích každého z vedoucích.

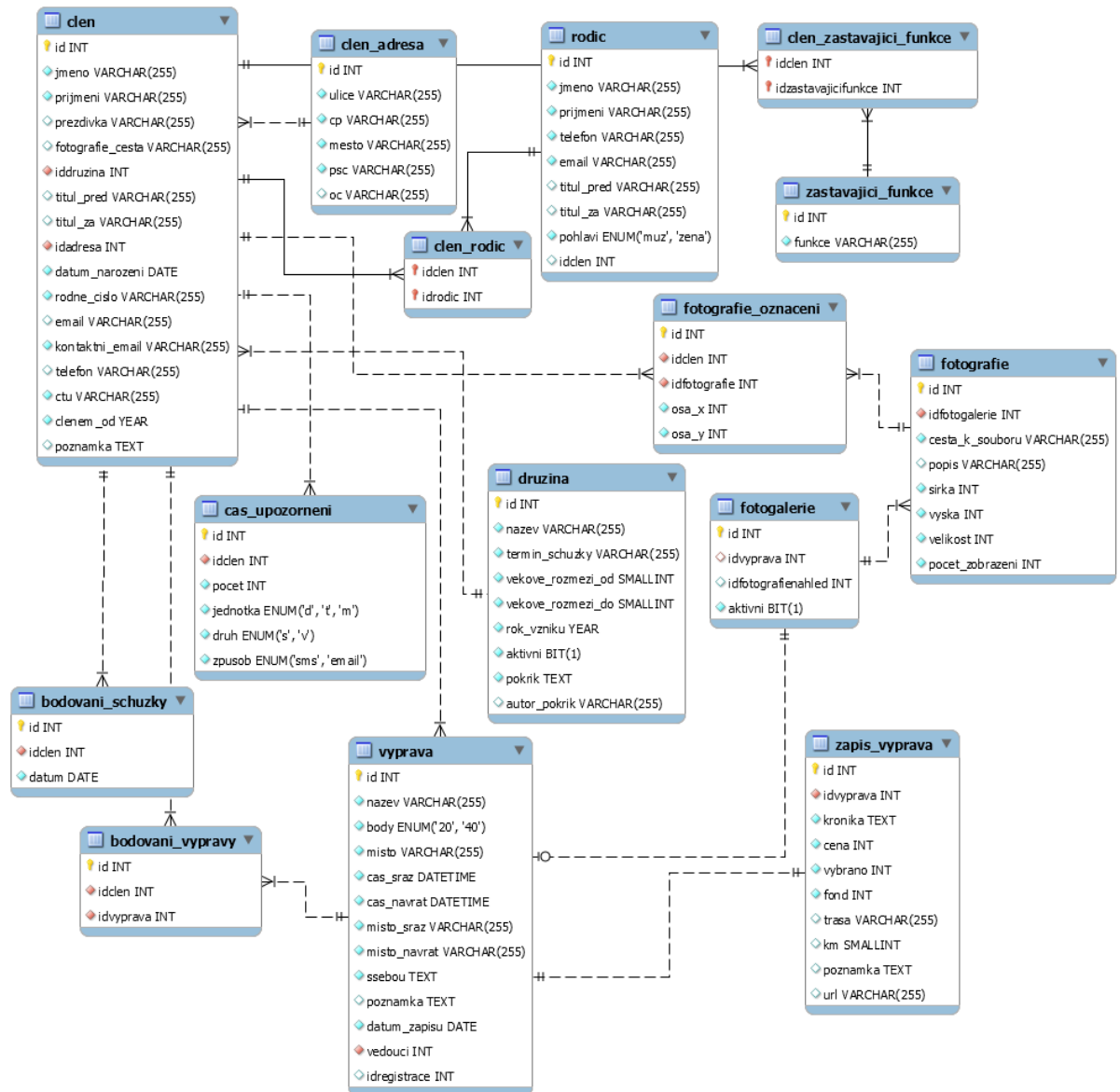
Poslední z větších inovací je předělaná *Fotogalerie* (viz Obrázek 8). Změny jsou následující: nová fotogalerie je zakládána automaticky při zapsání výpravy, a pokud obsahuje alespoň jednu fotografii, je zobrazena jako aktivní a viditelná. Při vkládání nových fotek do fotogalerie systém automaticky zmenšuje větší fotografie na požadované rozlišení, popř. velikost.



Obrázek 8 – Diagram případu užití (Fotogalerie)

Ke každé fotografii je možné přidat popisek, který je nepovinný, označit fotografii jako titulní fotografii dané fotogalerie, a hlavně je zde možnost označit člena na fotografii. Přes takto označenou fotografii se lze kliknutím dostat na kartu daného člena. Nabídka pro označení členů na fotografii je limitována účastí na dané výpravě pro jednodušší dohledání označovaného člena. Fotografie lze přidávat i dodatečně, nikoliv jen při zápisu z výpravy, právě do automaticky vytvořené prázdné fotogalerie. Fotogalerii lze vytvořit i ručně bez nutnosti založení výpravy. Tuto možnost má ale pouze uživatel s právy správce.

4.5 Část upraveného relačního modelu databáze



Obrázek 9 – EER diagram

Jelikož se přidaly nové funkcionality, je zapotřebí změnit i databázový model pro ukládání dat. Měněna je pouze část modelu, která přímo souvisí se změnami moduly (viz Obrázek 9), a zbylá zůstává v původní podobě (cca 72 tabulek).

Nejdůležitější tabulkou, na kterou je napojena většina ostatních, je tabulka *clen*. Vychází ze starého modelu, ale mnoho atributů je upraveno a přizpůsobeno pro potřeby změn z *Karty člena*. Na tabulku *clen*, pomocí relace *1:n*, navazuje tabulka *clen_adresa*, ukládající místo bydliště. Dále pak relací *n:m*, skrze tabulku *clen_rodic*, je připojena tabulka *rodic*, doplňující in-

formace o rodičích člena, a též připojena tabulka *zastavajici_funkce*, skrze *clen_zastavajici_funkce*, ve které jsou uloženy všechny role, kterých člen nabývá (dítě, instruktor, vedoucí, hlavní vedoucí atd.). Díky novému modulu *Karta družiny* přibývá do relačního modelu databáze tabulka *druzina*, obsahující potřebné informace pro zobrazení údajů o dané družině. Tabulka je spojena relací *1:n* s tabulkou *clen* pro identifikaci, do které družiny člen patří.

Další velkou změnou je přepracovaný systém výprav, s čímž souvisí přidání nových tabulek *vyprava* a *zapis_vyprava*. První tabulka obsahuje data převážně pro přidání a zobrazení nadcházející výpravy. Druhá uchovává data pro zápis z výpravy a společně vytváří podklady pro *Kartu výpravy*. K výpravám nedílně patří i bodování. Tabulka *bodovani_vypravy* spojuje člena a výpravu. Lze se tudíž dotázat na bodování člena dle absolvovaných výprav, v daném termínu a součtu jejich bodů. Podobně funguje i *bodovani_schuzek*, kde ale místo záznamu výpravy je pouze datum schůzky pro určení časového období. Následně se bodování spočítá jako počet nalezených záznamů v daném termínu vynásobených počtem bodů za schůzku (10 bodů).

Každý zápis z výpravy obsahuje i připojenou fotogalerii, která se automaticky vytváří při uložení zápisu. Pokud fotogalerie neobsahuje žádné fotografie, je označena jako neaktivní, tzn. není zobrazena. Tabulka uchovávající data o fotogaleriích dále obsahuje ID fotografie, která se bude zobrazovat jako náhledová fotografie dané fotogalerie, a samozřejmě spojení s tabulkou *vyprava* pro určení, ke které výpravě fotogalerie patří. Samotná tabulka obsahující data o fotografiích nese atributy potřebné k zobrazení fotografie a další zajímavé informace. Jelikož lze propojit označení člena na fotografii s jeho *Kartou člena*, je zapotřebí tyto spojení někde uchovat. K tomu slouží tabulka *fotografie_oznaceni*, která spojuje tabulku *clen* s tabulkou *fotografie*. Navíc každý záznam obsahuje informaci o poloze (osa x, osa y) označení v pixelech vůči rozlišení dané fotografie.

Poslední novinkou je modul *Upozornění*. Zde je zapotřebí ke každému členovi uložit jeho nadefinované volby prvního upozornění. Upozornění lze nadefinovat zvlášť, jak ke schůzce, tak k výpravě. V tabulce *cas_upozorneni* je uložen počet, daná jednotka (den, týden měsíc), zda se jedná o výpravu nebo schůzku a způsob odeslání upozornění. Např. první upozornění po 14 dnech, po výpravě, pomocí SMS, bude uloženo v tabulce jako *2 | t | v | sms*, popřípadě jako *14 | d | v | sms*, podle toho, jaké bude zadání člena v uživatelském prostředí.

5 ZÁVĚR

Hlavní myšlenkou celé práce bylo navrhnout nové, popřípadě změnit stávající funkcionality, které by zjednodušily a zpříjemnily používání webového portálu. Tento návrh se později využije jako zadání a návod pro přepracování nynějšího webového portálu. Nápad vytvořit novou podobu portálu se přímo nabízel, jelikož vedoucí práce Ing. Jan Hříděl byl a autor této práce je členem zmiňované organizace T. K. Průzkumník. Následně byla svolána mimořádná porada aktivního vedení klubu a byly vyslechnuty všechny jejich požadavky, připomínky a návrhy k nové podobě a funkcionalitě portálu, z kterých následně vznikaly návrhy inovací popsané v této práci.

V návrzích úprav je převážně myšleno na zpřehlednění informačních částí pro veřejnost a zjednodušení nejpoužívanějších operací, které vedoucí klubu využívají téměř denně. Dnešní podoba portálu sice většinu informací, které veřejnost (rodiče, děti) vyhledává, již obsahuje, nicméně na různých místech a občas i dost skrytě. A jelikož se sociální sítě staly v posledních pár letech velkým trendem, je v nových návrzích zahrnuta i jejich myšlenka. Nikoliv přeměnit stávající webový portál na sociální síť, nýbrž inspirování se v centralizaci informací o vybraném objektu (profil), ať už o členovi, družině nebo výpravě. Na druhou stranu je i zapotřebí zjednodušení systému pro zadávání bodování ze schůzek a výprav a pro vyplnění údajů o uskutečněné výpravě a přidání fotografií. V novém návrhu bude většinu řešit systém automaticky a vedoucí již jen vyplňovat předpřipravené formuláře. Hlavním důvodem pro zautomatizování celého procesu je zkrácení časové náročnosti pro jednotlivé úkony. Jelikož vedoucí občas odkládali např. zápis z výpravy jen pro to, že zabíralo mnoho času, než se vše vyplnilo, vytvořilo a sdělily se potřebné informace o bodování ostatním vedoucím, kteří ho museli též zapsat ručně.

Při analýze v praktické části práce bylo zajímavé zjistit doposud nepublikované informace o tom, jak se webový portál vyvíjel od statické prezentace až do obsáhlé aplikace jako je znám dnes, nebo jakým způsobem a metodami byl v minulém tisíciletí vyvíjen. Naopak méně příjemným bylo zjištění rozsáhlosti stávající databáze, čítající desítky tabulek. Databázi bylo potřeba následně upravit pro možnost aplikování nově navržených inovací.

Práce byla celkově přínosná díky detailnějšímu seznámení se samotným webovým portálem jak z uživatelského prostředí, tak z pohledu databáze a samotného vývoje. Velice přínosné bylo i seznámení se s metodikou unifikovaného procesu vývoje aplikací, díky které bude následující vývoj aplikace mnohem snazší. Dále bylo zajímavé zorientování se v současných vývojových trendech a shrnutí nových verzí používaných standardů.

6 POUŽITÁ LITERATURA

- [1] SMOLA, Martin. HTML5: co přináší a proč se o něj zajímat. *Root.cz* [online]. 2012 [cit. 2017-01-23]. ISSN 1212-8309. Dostupné z: <https://www.root.cz/clanky/html5-co-prinasi-a-proc-se-o-nej-zajimat/>
- [2] HTML5.1. W3C [online]. 2016 [cit. 2017-01-23]. Dostupné z: <https://www.w3.org/TR/html/>
- [3] HTML5 Differences from HTML4. W3C [online]. 2014 [cit. 2017-01-23]. Dostupné z: <https://www.w3.org/TR/html5-diff/>
- [4] HTML5 Migration: Migration from HTML4 to HTML5. *W3schools.com* [online]. [cit. 2017-01-25]. Dostupné z: http://www.w3schools.com/html/html5_migration.asp
- [5] What is CSS? *Tutorials Point* [online]. 2017 [cit. 2017-01-25]. Dostupné z: https://www.tutorialspoint.com/css/what_is_css.htm
- [6] ŠŤASTNÝ, Jiří, ŠIMEČEK, Martin, ed. CSS3 – držte krok s dobou (nové vlastnosti). *Programujte.com* [online]. 2010 [cit. 2017-01-25]. ISSN 1801-1586. Dostupné z: <http://programujte.com/clanek/2010070801-css3-drzte-krok-s-dobou-nove-vlastnosti/>
- [7] History of PHP. *php.net* [online]. c2001-2017 [cit. 2017-01-26]. Dostupné z: <http://php.net/manual/en/history.php.php>
- [8] MONUS, Anna. PHP 7: 10 Things You Need to Know. *HONGKIAT* [online]. 2015 [cit. 2017-01-26]. Dostupné z: <http://www.hongkiat.com/blog/php7/>
- [9] VÁCLAVÍK, Jiří. Perl (1) - Dávka teorie na úvod. *LINUXSOFT.cz* [online]. 2005 [cit. 2017-01-26]. ISSN 1801-3805. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=675
- [10] ČÁPKA, David. 1. díl – Úvod do ASP.NET. *ITnetwork.cz* [online]. 2012 [cit. 2017-01-26]. ISSN 2464-6326. Dostupné z: <http://www.itnetwork.cz/csharp/asp-net/tutorial-uvod-do-asp-dot-net>
- [11] ČÁPKA, David. 1. díl – Úvod do JavaScriptu. *ITnetwork.cz* [online]. 2016 [cit. 2017-01-31]. ISSN 2464-6326. Dostupné z: <http://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>

- [12] CoffeeScript – Overview. *Tutorials Point* [online]. 2017 [cit. 2017-01-31]. Dostupné z: https://www.tutorialspoint.com/coffeescript/coffeescript_overview.htm
- [13] MALÝ, Martin. CoffeeScript: řádně oslazený JavaScript. *Zdroják.cz* [online]. 2010 [cit. 2017-01-31]. ISSN 1803-5620. Dostupné z: <https://www.zdrojak.cz/clanky/coffeescript-radne-oslazeny-javascript/>
- [14] TypeScript – Overview. *Tutorials Point* [online]. 2017 [cit. 2017-01-31]. Dostupné z: https://www.tutorialspoint.com/typescript/typescript_overview.htm
- [15] DAJBÝCH, Václav. K čemu je dobrý TypeScript. *Zdroják.cz* [online]. 2013 [cit. 2017-01-31]. ISSN 1803-5620. Dostupné z: <https://www.zdrojak.cz/clanky/k-cemu-je-dobry-typescript/>
- [16] JavaScript – Overview. *Tutorials Point* [online]. 2017 [cit. 2017-01-31]. Dostupné z: https://www.tutorialspoint.com/javascript/javascript_overview.htm
- [17] Extensible Markup Language (XML). *W3C* [online]. 2016 [cit. 2017-02-02]. Dostupné z: <https://www.w3.org/XML/>
- [18] XML základy. *Tvorba-webu.cz* [online]. 2008 [cit. 2017-02-02]. Dostupné z: <https://www.tvorba-webu.cz/xml/>
- [19] ROUSE, Margaret. XML (Extensible Markup Language). *TechTarget* [online]. 2014 [cit. 2017-02-02]. Dostupné z: <http://searchmicroservices.techtarget.com/definition/XML-Extensible-Markup-Language>
- [20] HASSMAN, Martin. JSON: jednotný formát pro výměnu dat. *Zdroják.cz* [online]. 2008 [cit. 2017-02-02]. ISSN 1803-5620. Dostupné z: <https://www.zdrojak.cz/clanky/json-jednotny-format-pro-vymenu-dat/>
- [21] Introducing JSON. *JSON* [online]. [cit. 2017-02-02]. Dostupné z: <http://www.json.org/>
- [22] Teorie relačních databází: Relační model dat. *Manualy.net* [online]. 2006 [cit. 2017-02-02]. Dostupné z: <http://www.manualy.net/article.php?articleID=9>
- [23] ŠUBRTA, Václav, ed. Základy relačních databází, jejich využití v programování webu. Grafická a multimediální laboratoř: Web design a uživatelská rozhraní [online]. Praha: Fakulta informatiky a statistiky Vysoké školy ekonomické v Praze, 2014 [cit. 2017-02-02]. Dostupné z: <http://gml.vse.cz/data/oppa-webdesign/zaklady-db.html>

- [24] ASHISH. What is NoSQL and is it the next big trend in databases? *Tutorials Point* [online]. 2015 [cit. 2017-02-03]. Dostupné z: <http://www.tutorialspoint.com/articles/what-is-nosql-and-is-it-the-next-big-trend-in-databases>
- [27] ČÁPKA, David. MVC architektura. *ITnetwork.cz* [online]. 2016 [cit. 2017-03-01]. ISSN 2464-6326. Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor/>
- [28] Responzivní design – co je a jak funguje. *WEBSITES.CZ* [online]. 2016 [cit. 2017-02-27]. Dostupné z: <https://www.websites.cz/blog/responzivni-design-co-je-a-jak-funguje/>
- [29] VERTOMMEN, Kevin. An Introduction to SEO. *Envato Tuts+* [online]. 2012 [cit. 2017-02-27]. Dostupné z: <https://webdesign.tutsplus.com/articles/an-introduction-to-seo--webdesign-9408>
- [30] LEPAGE, Pete. Your First Progressive Web App. *Google Developers* [online]. 2017 [cit. 2017-03-01]. Dostupné z: <https://developers.google.com/web/fundamentals/getting-started/codelabs/your-first-pwapp/>
- [31] OTEMUYIWA, Prosper. Introduction to Progressive Web Apps (Offline First) - Part 1. *Auth0* [online]. 2016 [cit. 2017-03-01]. Dostupné z: <https://auth0.com/blog/introduction-to-progressive-apps-part-one/>
- [32] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

7 SEZNAM PŘÍLOH

Příloha A – Přiložené CD	48
--------------------------------	----

Příloha A – Přiložené CD

Obsahující:

- text práce (PDF),
- analýzu zpracovanou v programu Enterprise Architect,
- úpravy databázového modelu v programu MySQL Workbench.