



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH WEBOVÉ APLIKACE PRO CELOSVĚTOVOU RADIOAMATÉRSKOU SOUTĚŽ

DESIGN OF A WEB APPLICATION FOR THE WORLDWIDE AMATEUR RADIO CONTEST

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ TOVARYŠ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ALEŠ KLUSÁK, Ph.D.

BRNO 2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Tovaryš Ondřej

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh webové aplikace pro celosvětovou radioamatérskou soutěž

v anglickém jazyce:

Design of a Web Application for the Worldwide Amateur Radio Contest

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

KOCH, M. Datové a funkční modelování. Brno : VUT FP, 2004. ISBN 80-214-2724-8.

LEISS, O. a J. SCHMIDT. PHP v praxi: pro začátečníky a mírně pokročilé. 1. vyd. Praha: Grada, 2010, s. 13-110. Průvodce (Grada). ISBN 978-80-247-3060-8.

PÍSEK, S. HTML: začínáme programovat. 3., aktualiz. vyd. [i.e.] 1. vyd. Praha: Grada, 2010. ISBN 978-80-247-3117-9.

PONKRÁC, M. PHP a MySQL: bez předchozích znalostí : [průvodce pro samouky]. Vyd. 1. Brno: Computer Press, 2007. ISBN 978-80-251-1758-3.

VRÁNA, Jakub. 1001 tipů a triků pro PHP. Vyd. 1. Brno: Computer Press, 2010. ISBN 978-80-251-2940-1.

Vedoucí bakalářské práce: Ing. Aleš Klusák, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2014/2015.

L.S.

doc. RNDr. Bedřich Půža, CSc.
Ředitel ústavu

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
Děkan fakulty

V Brně, dne 28.2.2015

ABSTRAKT

Bakalářská práce se zaměřuje na návrh a implementaci webové aplikace na základě požadavků a analýzy současného stavu pomocí technologie PHP a MySQL. Jedná se o specifický systém pro analýzu a prezentaci dat.

ABSTRACT

This thesis is focused on design and implementation web application it based on the requirements and analysis of the current situation using technologies PHP and MySQL. It is specification system for analysis and data presentation.

KLÍČOVÁ SLOVA

Webová aplikace, informační systém, databáze uživatelů, PHP, MySQL, HTML, CSS, analýza souborů.

KEY WORDS

Web application, information system, database of users, PHP, MySQL, HTML, CSS, analysis of files.

BIBLIOGRAFICKÁ CITACE

TOVARYŠ, O. *Návrh webové aplikace pro celosvětovou radioamatérskou soutěž*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2015. 56 s. Vedoucí bakalářské práce Ing. Aleš Klusák, Ph.D.

ČESTNÉ PROHLÁŠENÍ

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

V Brně dne 31. května 2015

.....

podpis

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu mé bakalářské práce panu Ing. Aleši Klusákovi, Ph.D. a panu Rudolfu Sedlákovu za cenné rady, připomínky a vstřícnost při konzultacích.

OBSAH

ÚVOD	11
CÍL PRÁCE A METODIKA	12
1 TEORETICKÁ VÝCHODISKA PRÁCE	13
1.1 Internet	13
1.1.1 World Wide Web	13
1.2 Technologie.....	14
1.2.1 PHP	14
1.2.2 SQL	15
1.2.3 MySQL	16
1.2.4 PhpMyAdmin.....	17
1.2.5 HTML	17
1.2.6 CSS	18
1.2.7 JavaScript.....	19
1.2.8 Cron	20
1.2.9 Apache server	21
1.3 Vývoj softwaru.....	21
1.3.1 Životní cyklus softwaru	22
2 ANALÝZA SOUČASNÉHO STAVU	24
2.1 Český radioklub	24
2.1.1 Základní informace	24
2.1.2 Organizační struktura.....	24
2.2 Obecné informace	25
2.3 Požadavky na aplikaci.....	26
2.3.1 Přístupová práva	26
2.3.2 Přehled QSO	27
2.3.3 Správa stanic	29
2.3.4 Správa dat	29
2.3.5 Správa uživatelů.....	30

2.3.6	Evidence spojení	30
2.3.7	Správci systému	31
2.3.8	Správa diplomů	32
2.3.9	Evidence aktivity	32
2.4	Shrnutí analýzy.....	33
3	VLASTNÍ NÁVRH ŘEŠENÍ	34
3.1	Souhrnný pohled na aplikaci	34
3.2	Nastavení serveru a php.ini	35
3.3	Grafický návrh	36
3.4	Přehled QSO.....	38
3.4.1	Mapa spojení.....	38
3.4.2	Tisk diplomů	40
3.4.3	Žebříčky	41
3.5	Přihlašování do administrace	42
3.6	Správa stanic	43
3.6.1	Přidávání a přehled	43
3.6.2	Vysílání	44
3.7	Nahrávání dat	44
3.7.1	Přehled souborů	45
3.7.2	Nahrát soubor.....	45
3.8	Správa uživatelů	47
3.9	Evidence spojení	48
3.9.1	Přehled spojení.....	48
3.9.2	Počet spojení	48
3.10	Evidence správců systému.....	49
3.11	Správa diplomů.....	49
3.11.1	Přehledy diplomů.....	49
3.11.2	Vytvořit diplom.....	50
3.12	Evidence aktivity	50

3.12.1	Přihlašovací logy.....	50
3.12.2	Vysílací logy	51
3.13	Ekonomické zhodnocení a přínosy.....	51
3.14	Ostrý provoz	52
ZÁVĚR		53
SEZNAM POUŽITÝCH ZDROJŮ.....		54
SEZNAM OBRÁZKŮ		56
SEZNAM TABULEK A GRAFŮ.....		56
SEZNAM PŘÍLOH.....		56

ÚVOD

Webový prostor je nejsnadnější a nejefektivnější cesta, jak dopřát široké mase uživatelů příležitost dostat se k potřebným informacím. Při Mistrovství světa v hokeji 2015 v České republice zároveň proběhne celosvětová akce (soutěž) radioamatérů. Tato práce se zabývá návrhem a tvorbou webové aplikace (informačního systému) pro potřeby radioamatérů.

Na úvod, v teoretické části, budou zmíněny veškeré technologie, které budou v praxi použity. V analýze současného stavu budou pro správné pochopení uvedeny požadavky na systém a nynější situace bez této aplikace. Praktická část bude obsahovat samotnou tvorbu a implementaci do chodu. Výsledný systém bude mimo jiné sloužit k analýze získaných dat od radiostanic a prezentaci výsledků uživatelům.

CÍL PRÁCE A METODIKA

Cílem práce je navrhnout a implementovat do praxe specifický informační systém. Radioamatéři potřebují na veřejně dostupný webový server nahrávat a vyhodnocovat různá data z radiostanic, která se budou dále zpracovávat a prezentovat. Celý systém je obsáhlý, jednou z mnoha funkcí např. bude možnost ukázat, ve kterém pásmu a jakém módu právě každá stanice vysílá, nahrávání souborů se spojeními na server nebo stahování získaných certifikátů (diplomů).

V teoretické části budou vysvětleny pojmy z okruhu tvorby webových aplikací a detailně popsány všechny technologie pro samotnou tvorbu. Následně v analýze bude zhodnocena současná situace, jaké probíhají procesy, požadavky na aplikaci a samotná analýza problémů spojených s návrhem. V praktické části přistoupíme k samotné realizaci.

Na závěr budou zhodnoceny celkové přínosy pro uživatele, ekonomické zhodnocení a především ohlasy z praxe, protože celá akce je naplánována na květen 2015. Výstupem celé práce bude plně provozu schopný, snadně ovladatelný systém s administrátorskými i uživatelskými právy pro přístup, který budou reálně používat tisíce až desetitisíce lidí po celém světě.

1 TEORETICKÁ VÝCHODISKA PRÁCE

V této části práce jsou shrnuty teoretické poznatky a pojmy, které jsou důležité pro správné pochopení analytické části práce i části praktické.

1.1 Internet

Internet je celosvětová globální počítačová síť. Zprvu, v 60. letech 20. století, se jednalo pouze o propojení několika počítačů americké armády. Následně, při dalším rozvoji sítě, se připojily americké university. Jako o zlomu můžeme mluvit o 90. letech, kdy došlo k velkému rozmachu počítačů ve firmách i domácnostech a především uvedení do provozu služby WWW. V roce 1992 se internet dostal také do České republiky – síť CESNET propojila vysoké školy a výzkumné organizace. [1]

1.1.1 World Wide Web

World Wide Web je nejpoužívanější služba internetu, známější spíše pod zkratkou WWW. Skutečný rozvoj internetu započal až se spuštěním této služby. V roce 1989 byla vyvinuta v laboratořích Evropského centra pro jaderný výzkum (CERN). Zprvu bylo cílem vytvořit službu pro sdílení informací týkajících se výzkumných projektů. Pro širokou veřejnost byla služba uvolněna v roce 1992, v roce 1993 CERN předává technologii nekomerční organizaci W3C, která má za cíl další vývoj a především standardizaci WWW pro široké použití. [1]

World Wide Web je založen na principu klient-server, kteří spolu komunikují pomocí protokolu HTTP. Klient je webový prohlížeč, který komunikuje se vzdáleným serverem a zobrazuje požadované dokumenty. Server přijímá požadavky od klientů (webových prohlížečů) a doručuje požadované dokumenty, které identifikuje pomocí adresy URL. [1]

1.2 Technologie

Tato kapitola detailně popisuje všechny použité technologie pro tvorbu webové aplikace – informační systém pro radioamatéry.

1.2.1 PHP

PHP je skriptovací programovací jazyk. Zkratka PHP znamená „Hypertext preprocessor“. Vytvořil ho Kanadčan Rasmus Lerdorf. Chtěl vytvořit skript na sledování počtu návštěvníků, kteří přišli na jeho online životopis. Když viděl, jaký má skript úspěch, pokračoval ve vývoji jazyka. Postupem času se přidávali další vývojáři a rozvíjeli jeho možnosti. V dnešní době je PHP nejpoužívanějším programovacím jazykem na internetu. [2]

S pomocí PHP můžeme vytvářet dynamické webové stránky – webové stránky, které se mění na základě různých podmínek. Například: chci se přihlásit na svůj profil na Facebooku a sledovat nový obsah. Po přihlášení budu přesměrován na stále stejnou stránku, ale vrácen mi bude pokaždé jiný obsah. Protikladem je čistý HTML dokument. Zde se jedná o čistě statické stránky – neprobíhají žádné změny na základě podmínek, vytvořený dokument je kdykoliv stále stejný. [2]

Skripty mohou běžet na kterémkoliv počítači, kde je PHP nainstalováno, a to i pouze v příkazovém řádku. V případě, že nechceme skripty spouštět pouze lokálně, je potřeba mít server, kde bude nainstalováno běhové prostředí (např. Apache). Apache je nejoblíbenější softwarový webový server na internetu. V současné době na něm „běží“ většina webových aplikací. Jedná se o projekt s otevřeným kódem, který dokáže virtuálně „běžet“ na všech operačních systémech. Zde se skripty vykonávají. Uživatelé je do prohlížeče posílají pouze výsledek operace. [2]

Skriptovací jazyk podporuje rozhraní pro komunikaci s databázovými systémy, jako například MySQL, který bude v této práci použit.

1.2.2 SQL

Structured Query Language (zkráceně „SQL“) je standardizovaný databázový dotazovací jazyk, který podporuje většina systémů pro správu databáze, jako například MySQL nebo Oracle. První verze tohoto jazyka se jmenovala SEQUEL, vznikla v polovině 70. let a vyvinula ji firma IBM jako standardní jazyk pro přístup k vůbec první relační databázi, kterou taktéž firma IBM provozovala. Na konci 70. let byl původní název zkrácen na dodnes používané SQL. [3]

Většinu základních příkazů, jako například „SELECT“ pro výběr, „INSERT“ pro vkládání, „DELETE“ pro mazání nebo „UPDATE“ pro upravování, můžeme použít ve všech databázových systémech stejně nebo velice podobně. Většina programovacích jazyků si s SQL rozumí, jeho příkazy jsou téměř vždy použitelné. [4]

Ukázka použití příkazu INSERT, příklad vloží uživatele do tabulky „uzivatele“: [4]

```
INSERT INTO Uzivatele (login, heslo) VALUES ('Marek', 'pass123')
```

Ukázka použití příkazu SELECT, příklad vybere z tabulky „uzivatele“ uživatele s přihlašovacím jménem „Marek“: [4]

```
SELECT login FROM Uzivatele WHERE login = 'Marek'
```

Ukázka použití příkazu UPDATE, příklad změní přihlašovací jméno uživateli s unikátním identifikátorem 3 na „Pavel“: [4]

```
UPDATE Uzivatele SET login = 'Pavel' WHERE id_uzivatele = 3
```

Ukázka použití příkazu DELETE, příklad odstraní uživatele s unikátním identifikátorem 46: [4]

```
DELETE FROM Uzivatele WHERE id_uzivatele = 46
```

SQL je nejvhodnější použít u relačních databází, tj. databází, které ukládají data do tabulek neboli relací. Jeden řádek v tabulce tvoří jednu datovou větu. Tabulky se skrze příkazy SQL mohou všemožně spojovat, čehož se využívá, aby nedocházelo ke zbytečné redundanci dat. [4]

1.2.3 MySQL

Téměř všechny webové aplikace, které jsou vystavěny na bázi skriptovacího jazyka PHP využívají nějaké databáze. Tím nejpoužívanějším je databázový systém MySQL. Je to produkt vytvořený komerční švédskou firmou MySQL AB. Ve srovnání s jinými databázovými systémy patří spíše k těm jednoduchým, ale zase nenáročným na hardware. U některých operací nabízí zvýšení rychlosti. Jednoduché shrnutí: MySQL je rychlý, malý, jednoduchý a nenáročný databázový systém. [5]

Základním jazykem pro práci s MySQL je jazyk SQL. Jeho hlavními úkoly jsou navázání spojení s databázovým serverem, zaslání příkazu pro práci s daty, vyzvednutí výsledných dat a následné ukončení spojení s databázovým serverem. [5]

Databáze v MySQL se skládá z databázových tabulek, které jsou vzájemně propojeny relačními vztahy. V tabulkách jsou uložena veškerá data. Tabulka se skládá ze schéma relace a těla relace. Schéma se skládá z jednotlivých domén, které mají daný svůj datový typ. Tělo tvoří data tabulky – každý řádek je jeden záznam, který má svůj unikátní identifikátor tzv. primární klíč. [5]

„Primární klíč je speciálním typem indexu, který jednoznačně identifikuje každý řádek v tabulce. Jde tedy o unikátní klíč, který nemůže obsahovat hodnoty NULL (jde definovat jen nad NOT NULL sloupce).“ [6]

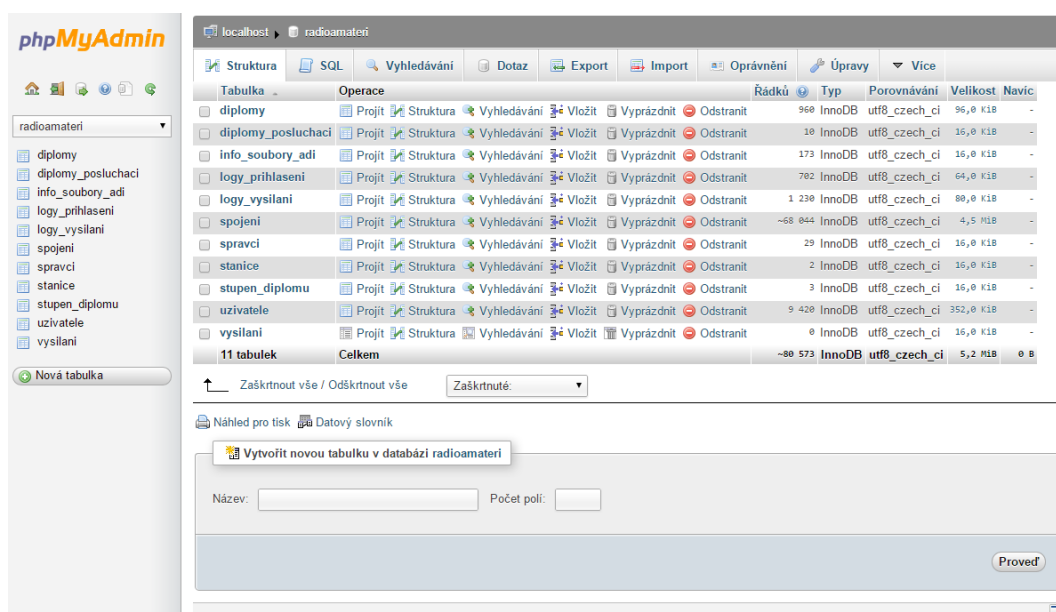
Primární klíč by měl být definován nad všemi tabulkami. Výjimkou mohou být např. jednořádkové tabulky. Každá tabulka může mít pouze jeden primární klíč. Primární klíč se může vybrat přímo z dat tabulky, tomuto klíči se říká přirozený. Mnohem častější je ovšem jev, kdy se vytváří tzv. klíč umělý, který s daty nesouvisí – pouze se na ně odkazuje (nejčastěji sloupec „id“). [6]

1.2.4 PhpMyAdmin

PhpMyAdmin je volně šiřitelný software napsaný ve skriptovacím jazyku PHP. Používá se především ke správě databázového systému MySQL přes webové rozhraní – musí běžet na webovém serveru. K MySQL serveru je možno přistupovat z jakéhokoliv počítače přes webový prohlížeč, což může být i nevýhoda. Pomocí phpMyAdmin můžeme spravovat celý MySQL server nebo pouze jednotlivé databáze. [7]

Pomocí tohoto softwaru můžeme mimo jiné jednoduše:

- Vytvořit databázi
- Vytvořit a spravovat databázové tabulky
- Vkládat data do tabulek
- Zálohovat databázi [7]



Obrázek 1 - Uživatelské prostředí phpMyAdmin (zdroj: vlastní)

1.2.5 HTML

HyperText Markup Language (zkráceně HTML) je značkovací jazyk určený k vytváření dokumentů, které obsahují hypertextové odkazy a pokročilejší formátování textu. Příklady použití HTML v dokumentu:

- Vložení obrázku a obrázkových map různých tvarů a velikostí, možnost vložení i animovaných GIFů
- Vytváření formulářů, tabulek, rámců
- Strukturování textu do odstavců, vytváření hypertextových odkazů [8]

HTML dokument se od obyčejného textového dokumentu odlišuje tím, že v obsahu navíc zahrnuje i informace o formátování stránky. Tyto informace jsou určeny pouze pro prohlížeč, uživatel je při prohlížení dokumentu nevidí. Nazývají se značky (tzv. tagy) a zapisují se do ostrých závorek „< >“. Většina těchto značek jsou párové, to znamená, že existuje stejná značka s lomítkem na začátku, která ukončuje platnost značky. Například:

<code>tento text bude tučně </code>
--

Samotné používání značek nedělá z obyčejného textu HTML dokument. Každá webová stránka musí mít určitou strukturu. Její celý obsah je uzavřen mezi značkami `<html>` a `</html>`. Dvěma důležitými částmi dokumentu je hlavička uzavřená mezi značky `<head>` a `</head>`, kde se nachází např. nadpisek a údaje o použitém kódování. Druhou částí je samotné tělo dokumentu mezi značkami `<body>` a `</body>`, tady se nachází veškerý obsah HTML dokumentu. [8]

Se značkovacím jazykem HTML úzce souvisí kaskádové styly (CSS).

1.2.6 CSS

Kaskádové styly (Cascade style sheets – zkráceně CSS) slouží k formátování webových dokumentů. Dříve se data na stránce uspořádávala pouze pomocí HTML tabulek. Někdy okolo roku 1997 vznikly právě kaskádové styly. Díky nim můžeme tvořit efektivní webové prezentace. CSS například umí:

- Jakkoliv formátovat písmo
- Zvýrazňovat odkazy po přejetí myši
- Nastavit pozadí čehokoliv
- Jakékoliv pozicování objektů

- Téměř jakoukoliv část dokumentu zneviditelnit, zprůhlednit nebo nezobrazit
- Předefinovat grafický význam běžných značek a mnoho dalšího [9]

„Hlavní význam CSS spočívá v tom, že fungují hodně automaticky, přičemž se vzhled celého webu deklaruje jedním souborem.“ [9] S tímto souvisí trojí možnost použití CSS:

- **Přímý styl** – formátování elementu přímo v textu dokumentu pomocí atributu *style*=“... “. Je to nešikovné, ale občas se používá.
- **Pomocí stylopisu v hlavičce dokumentu** – styly jsou nadefinovány v hlavičce každého dokumentu mezi značkami `<style>` a `</style>`.
- **Externí stylopis** – styly jsou nadefinovány v souboru s příponou `.css`, na který se v hlavičce dokumentu odkazujeme značkou `<link>`. Kóderům tato funkce neuvěřitelně usnadňuje práci. Vzhled webových stránek je možno měnit v jednom souboru a změna proběhne všude, kde je použit. [9]

Příklad použití CSS v HTML dokumentu:

```
<p style="color: red; font-family: Arial;">Relativně krátký text pro ukázkou použití CSS, pomocí přímého stylu zápisu</p>
```

Tato ukázka nastaví barvu odstavce na červenou a font písma na Arial. [9]

1.2.7 JavaScript

JavaScript je programovací jazyk vestavěný do webového prohlížeče. Lze ho použít ve většině prohlížečů, což ho dělá unikátním. Pracuje přímo s prohlížečem, tj. kód se zpracovává na straně uživatele (ne na straně serveru jako je tomu např. u PHP). Pomocí JavaScriptu rychle a jednoduše dosáhneme způsobu jak do statických stránek přidat dialogy, dynamiku nebo usnadnit uživateli práci s dokumentem. K vývoji skriptů v tomto jazyce je potřeba pouze libovolný textový editor a webový prohlížeč. Přes veškeré výhody má však jazyk i svá omezení: [10]

- **Kód v JavaScriptu nekomunikuje se servery** – skript nemůže komunikovat s ostatními zařízeními, pracuje výhradně v prohlížeči uživatele. Pokud bychom chtěli vytvořit formulář a nashromážděná data uložit na server do databáze, musíme využít služby jiného jazyka. JavaScript nám může například umožnit plynulý průběh vyplnění formuláře.
- **JavaScript nemůže vytvořit grafiku** – složitější jazyky dokáží kreslit obrázky, kód JavaScriptu může pracovat s jen už existujícími soubory (GIF nebo JPEG).
- **Kód v JavaScriptu se chová různě v různých prohlížečích** – tento problém se týká dnes už pouze starších prohlížečů např. IE 5.4 a nižší nebo Opera 4. Od roku 1990 vydává ECMA standardy pro JavaScript, které se nazývají ECMAScript. 99 % prohlížečů, dnes používaných, splňuje minimálně verzi 3 toho standardu. [10]

1.2.8 Cron

Cron je systémový nástroj z prostředí Linuxu/Unixu. Jeho hlavní činností je spouštět různé programy (skripty) v předem danou dobu a předem stanoveném intervalu. Téměř každý vyspělejší webový projekt se bez tohoto systémového nástroje neobejde. Nejčastěji slouží k:

- **Generování databázově velmi náročných stránek** – data z databáze se mohou vytahovat jen vždy v určitém intervalu, ukládat do souboru, z tohoto souboru číst bez neustálého přístupu k databázi a tento soubor i v daném intervalu přepisovat.
- **Získávání aktuálních informací z různých zdrojů na internetu** – například aktuální kurzovní lístek, výsledky sportovních utkání nebo reportáže
- **Promazávání nepotřebných dat v databázi** [11]

Pokud víme, co chceme s pomocí Cronu spouštět, musí se nastavit i pravidelný čas spouštění. Tohoto systémového démona nelze použít k jednorázovému spouštění aplikací či skriptů, spouští námi definované úlohy v předem daném intervalu. Nejmenší interval, který lze nastavit, je 1 minuta. [11]

Cron sám o sobě může spouštět skripty denně, každou hodinu, týdně nebo měsíčně. Pokud bychom chtěli časovat například v přesně daný čas 01:15 atp., musí se využít utilita crontab. Tato utilita je určena ke správě seznamů úloh, které Cron vykonává. [11]

1.2.9 Apache server

Důležitý prvek ve stavbě webové aplikace je webový server, na kterém aplikace běží a je dostupná uživatelům i po celém světě. Problematika webových serverů je velice obsáhlá a složitá, z tohoto důvodu si uděláme pouze obecný přehled o fungování této důležité součásti.

Webový server je prvek připojený k počítačové síti a přijímá požadavky protokolu HTTP. Tyto požadavky vyřizuje (obsluhuje) a uzlu, který požadavek odeslal, vrací odpověď. Odpovědí je většinou nějaký HTML dokument, který se pomocí webového prohlížeče zobrazí v libovolném zařízení. [12]

Apache server je softwarový webový server s otevřeným kódem pro většinu platforem nevyjímaje Microsoft Windows a Linux. Mimo jiné slouží k publikování webových stránek i provoz velkých (náročných) webových aplikací. Dokumenty jsou na serveru uloženy ve formě souborů obsahující kód (např. PHP), při požadavku serverem zpracovány a odeslány ve formě HTML dokumentů prohlížeči koncového uživatele. [12]

Podle serveru Živě.cz je Apache stále nejpopulárnějším webovým serverem. Apache běží zhruba na 65 - 70 % všech webových serverů. Zpovzdálí ho sleduje ISS server od Microsoftu. [13]

1.3 Vývoj softwaru

Vývoj softwaru ovlivňují 3 strany. První je zákazník, který vývoj platí a specifikuje požadavky na výsledný produkt. Druhou stranou je dodavatel, který software vyvíjí a třetí jsou uživatelé, kteří budou systém reálně používat, tudíž upřesňují požadavky nejbližší. Ne vždy mají uživatelé zájem na zavedení nového softwaru (na rozdíl od zákazníka, tj. jejich nadřízený). Typickým důvodem je strach z nového, neochota učit se a také obava,

že se ukáže, že už není potřeba tolik lidí – propouštění. Tento důvod nebývá jen planou obavou, skutečnou motivací managementu pro vytvoření nového systému může být snížení mzdových nákladů. [14]

1.3.1 Životní cyklus softwaru

Existuje mnoho pohledů na vývoj softwaru, nicméně velká většina se liší pouze v posloupnosti jednotlivých etap nebo jejich spojování.

„Model životního cyklu softwaru definuje etapy vývoje softwaru a pro každou etapu dále definuje nutné činnosti a její vstupy a výstupy.“ [14]

Analýza a specifikace požadavků

Úvodní etapa, ve které se zabýváme požadavky zadavatele (zákazníka). Často jsou požadavky nejasné, neúplné, zákazník přesně ani nemusí vědět, co chce. My se snažíme získat co nejjasnější informace, musíme vyhodnotit, zda jsou definované požadavky vůbec proveditelné, což nám odpoví na otázku, jestli se do projektu vůbec pouštět. V této etapě by měla být pozornost věnována jen teoretickým požadavkům – případné realizaci se budeme věnovat v etapách následujících. Dalšími výstupy této etapy by měla být identifikace a analýza možných rizik a plán akceptačního testování (testy, na základě kterých zákazník software převezme). [14]

Architektonický návrh

Etapa navazující na analýzu. Je potřebná pro ujasnění celé koncepce systému a hlavně k jeho dekompozici. Dekompozice je rozložení celého problému na menší části, čímž se usnadní jeho řešení. Mezi jednotlivými menšími částmi se musí definovat vztahy tak, aby vše správně zapadlo do funkčního celku. [14]

Podrobný návrh

V této etapě se řeší podrobná specifikace jednotlivých softwarových součástí. Vybírají se správné algoritmy pro realizaci požadovaných funkcí, stanovují se logické a fyzické struktury dat a to, jak se budou ošetřovat chybové nebo neočekávané, nestandardní stavy. Výstupem by měl být odhad doby trvání a nákladů na projekt a také návrh testů součástí, včetně testovacích dat. [14]

Implementace a testování součástí

Zde se realizuje programová část jednotlivých softwarových součástí, vypracovává se k nim dokumentace a implementované části se testují. [14]

Integrace a testování systému

Poté, co jsou otestovány jednotlivé součásti, se spojí do jednoho celku (celého systému). Testuje se jeho provoz, odhalují se chyby, které nebylo možné objevit při testování jednotlivých komponent. Pokud jsou objeveny nějaké chyby, vše se musí otestovat ještě jako jednotlivé součásti, aby se vyloučil fakt, že se neobjevily chyby nové – čímž se vracíme do předchozí etapy. [14]

Akceptační testování a instalace

V této fázi testuje systém uživatel. Na základě tohoto testování se zákazník rozhodne, jestli systém převezme nebo jestli je potřeba ještě něco doladit. Po kladném přijetí systému uživatelem začíná instalace softwarového systému u zákazníka a důkladné školení uživatelů. „*Akceptační testování tak vlastně rozhoduje o úspěchu či neúspěchu softwaru.*“ [14]

Provoz a údržba

S nasazením do reálného provozu souvisí průběžné řešení problémů, případné opravy chyb, ale hlavně rozšiřování systému o nové funkce či přizpůsobování měnícím se podmínkám okolí. [14]

2 ANALÝZA SOUČASNÉHO STAVU

Tato část detailně popisuje požadavky na aplikaci a účel, ke kterému bude využita. Částečně popisuje funkčnost systému. Detailněji bude vše popsáno ve vlastním návrhu řešení. Požadavky jsou roztrženy podle jednotlivých částí celého systému.

2.1 Český radioklub

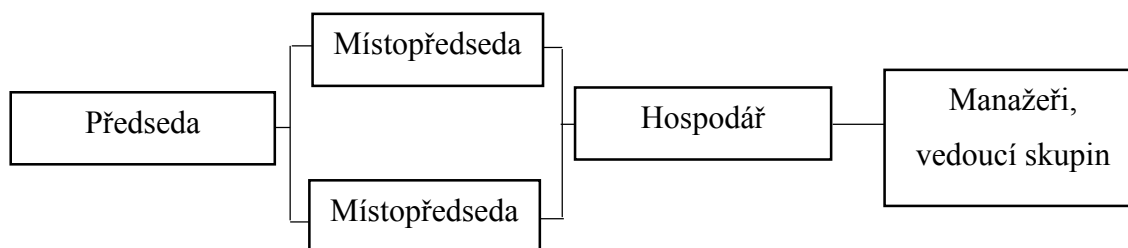
V této kapitole jsou shrnuty základní informace o Českém radioklubu, pro který je webová aplikace pro soutěž tvořena.

2.1.1 Základní informace

Český radioklub je občanské sdružení sdružující největší počet radioamatérů v České republice. Vznikl 10. května 1990. Náplň jeho činností tvoří:

- Rozšiřování a hájení zájmů všech lidí zabývajících se radioamatérstvím v ČR
- Reprezentace radioamatérů vůči úřadům, zejména Českému telekomunikačnímu úřadu
- Reprezentace zájmů členů vůči Mezinárodní radioamatérské unii
- Spolupráce s radioamatérskými organizacemi sousedních států
- Podpora všech činností podporující radioamatérství a přidružené aktivity

2.1.2 Organizační struktura



Graf 1 – Organizační struktura ČRK (zdroj: vlastní)

2.2 Obecné informace

Při Mistrovství světa v ledním hokeji 2015, které se koná v květnu v České republice, zároveň proběhne celosvětová akce radioamatérů. Pořádá ji Český radioklub. Podobné akce probíhají po celém světě při významných prestižních, nejen sportovních, událostech. Oficiální název soutěže je Czech Hamradio ice cup 2015.

V České republice budou zaregistrovány dvě volací značky (stanice) **OL2015O** a **OL2015P**. Soutěží se na krátkých i velmi krátkých vlnách. Každá ze stanic má stejná pásma (160M, 80M, 40M...). Každé pásmo má 3 – 4 módy (CW, PH, RTTY, DG). Radioamatéři z celého světa se budou prostřednictvím svého vysílacího zařízení spojit s těmito dvěma registrovanými stanicemi. Úspěšné spojení na jeden z módů v jednom z pásem = 1 QSO (1 bod do soutěže). Pro snadnější pochopení je přiložena následující tabulka (jedno vybarvené políčko znamená 1 QSO):

	160M				80M				40M				30M				20M				17M				15M				12M				10M			
	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG				
OL2015O																																				
OL2015P																																				

Obrázek 2 - Tabulka spojení (zdroj: vlastní)

Na předchozím obrázku nejsou zobrazena všechna pásma, pouze krátké vlny. Záhloví tabulky je pevně dáno a nemění se. Za určitý počet QSO se budou generovat diplomy ve třech stupních (bronzový, stříbrný, zlatý).

Data o jednotlivých spojení, která se ze stanic generují do speciálních souborů ve formátu .adi, se budou do systému vkládat skrze formulář pro nahrávání souborů. Jelikož soubory nemají přesně danou strukturu a v jednotlivých verzích se dost liší, je potřeba vytvořit algoritmus na jejich univerzální zpracování.

Jedná se o soutěž. Radioamatéři z celého světa mezi sebou soutěží v tom, aby měli co nejvyšší počet QSO. Z tohoto důvodu se musí udělat žebříčky top 10 a top 100 nejlepších. Z časových důvodů pouze v kategorii Češi a zbytek světa.

Celý systém bude uživatelům sloužit pro:

- přehled, která stanice v jakém pásmu a módu právě vysílá
- přehled úspěšně uskutečněných spojení
- přehled celkového počtu vykonaných spojení
- generování diplomů
- přehled žebříčkového umístění

Ke všem těmto funkcím se váže veškerá administrace, která je popsána v další podkapitole.

Celý systém bude tvořen od začátku na „čisté lince“. Je to specifický systém, není z čeho vycházet, žádná podobná aplikace se komerčně neprodává – bude „ušita“ na míru potřebám radioamatérů. Úspěch celé akce stojí na zdárném vytvoření a implementaci systému. Bez této aplikace se soutěž nemůže uskutečnit.

2.3 Požadavky na aplikaci

Tato kapitola detailně popisuje jednotlivé funkční části systému, k čemu budou sloužit a podmínky, za jakých by měly fungovat.

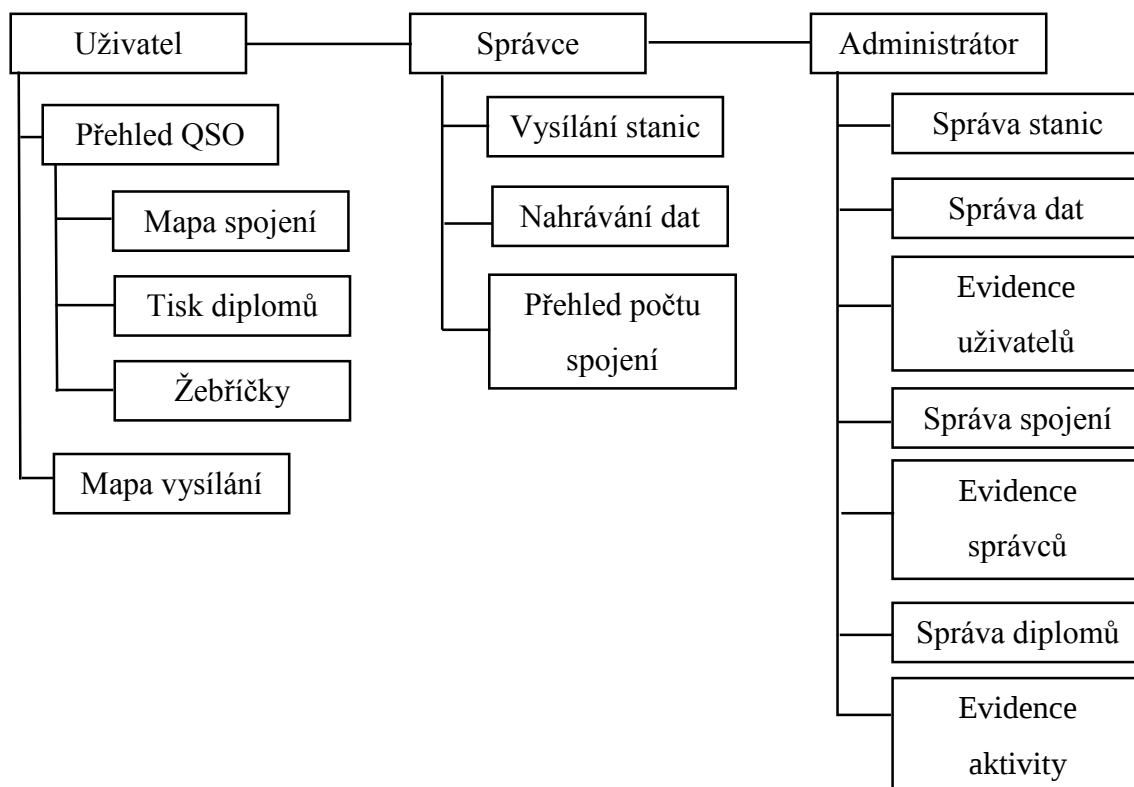
2.3.1 Přístupová práva

Přístupových práv jsou 3 druhy:

- **Uživatel**, pro kterého je aplikace primárně určena. Pro přístup se nemusí nijak přihlašovat, může to být kdokoliv s přístupem na internet, komu aplikace přinese nějaký užitek.
- **Správce**, kterých bude jen omezené množství. Bude se přihlašovat na základě přihlašovacího jména a hesla. Může nahrávat soubory s daty, určovat, v kterých pásmech a módech bude která stanice vysílat a má přístup do detailního přehledu uskutečněných spojení.

- **Administrátor** má na starosti kompletně celý systém, má přístup ke správě všech částí. Mimo jiné přiděluje přihlašovací údaje správcům, může vytvářet diplomy, přidávat stanice, má k dispozici veškeré přehledy atd.

Následující schéma ukazuje, kam má kdo přístup:



Graf 2 – Přístupová práva (zdroj: vlastní)

Přístupová práva jsou hierarchicky uspořádána. K čemu má přístup uživatel, má přístup i správce a správce má přístup jen k malé podmnožině toho co administrátor. Administrátor může vše.

2.3.2 Přehled QSO

Mapa spojení

Základním pilířem pro uživatele je jednoduchý a jasný přehled uskutečněných spojení se stanicemi. K tomuto účelu se jako ideální jeví prezentace dat pomocí tabulky (viz „Obecné informace“ – obrázek „Tabulka spojení“). Každý mód má svoji barvu:

- CW - červená
- PH – modrá
- RTTY – tmavě zelená
- DG – světle zelená

Při vícero spojení tvoří tyto kombinace „mapu spojení“.

Důležitou podmínkou je, aby se spojení v přehledu zobrazilo (započítalo) pouze jednou, a to i tehdy, pokud bude uskutečněno na stejném pásmu ve stejném módu x-krát. To samé platí i pro celkový počet QSO, který se bude mimo jiné zanášet i do žebříčků. Každý radioamatér má svou unikátní značku, kterou používá při vysílání a zároveň pomocí ní může vyhledávat své údaje o spojení v systému. Jednoznačně se tak identifikuje jednotlivý uživatel.

Žebříčky

Celkové žebříčky o počtu QSO – Top 10 nejlepších, by měly být dostupné z každé stránky. Po rozkliknutí je potřeba se dostat na Top 100 nejlepších z každé evidované skupiny. Skupiny budou dvě Češi (zkr. OK) a ostatní světové země (zkr. DX).

Tisk diplomů

Při dosažení určitého počtu QSO si bude moci daný uživatel, pod svou značkou, stáhnout dynamicky vygenerovaný diplom s pořadovým číslem. Do formuláře pouze zadá jméno, které chce mít na diplomu. Stažení diplomu není nijak početně omezené, podmínka je, aby při opakovaném stažení zůstávalo pořadové číslo 1. stažení.

Na diplomu tedy bude jméno a příjmení, unikátní značka radioamatéra, dosažený počet QSO a stupeň diplomu (zlatý, stříbrný nebo bronzový). Stáhnout diplom pro danou značku si může jakákoli osoba s uvedením svého jména a příjmení. Unikátní značka radioamatéra je na diplomu dána pevně, tzn. diplom pro takovou osobu nebude mít opodstatnění.

2.3.3 Správa stanic

Správa volacích značek (stanic) zahrnuje tyto funkce:

- **Přehled stanic** – jednoduchý přehled stanic zanesených v systému, na které je možno se připojit. Zahrnuje i základní administraci.
- **Přidat stanici** – možnost zanést do systému novou stanici
- **Vysílání stanic** – jedna ze dvou důležitých funkcí, které budou denně používat jak správci, tak administrátor. Slouží k tomu, aby uživatelé (radioamatéři) z celého světa měli informaci o tom, v kterém pásmu a na jakém módu která stanice právě vysílá. Při zahájení vysílání správce (nebo administrátor) zatrhne příslušné „políčko“ v tabulce, do databáze se uloží informace o vysílání a na hlavní stránce, přístupné pro kohokoliv, se tyto informace zobrazí.

2.3.4 Správa dat

Nejdůležitější ze všech funkčních prvků systému je správa dat. Bez dat (informací o provedených spojení) je celá aplikace k ničemu. Skládá se ze dvou důležitých funkčních prvků:

- **Přehled souborů** – zde se budou nacházet informace o všech nahraných souborech s daty na server. Shromažďovat se bude název souboru, datum a čas nahrání na server, kdo (který správce) soubor nahrál a počet nahraných spojení ze souboru. Po kliknutí na název souboru se musí zobrazit obsah souboru tak, jak vstupoval do skriptu pro zpracování.
- **Nahrávání dat** – druhá funkce správců. Požadavek je, aby samotné nahrávání souborů a jejich zpracování bylo uživatelsky co nejjednodušší. Pouze je třeba vybrat stanici ze seznamu, pro kterou se daná spojení nahrávají a přiložit soubor s daty (spojeními). Vše ostatní, zpracování a uložení důležitých informací do databáze, zajistí skript. To, jak jeden (zkrácený) soubor může vypadat, je přiloženo v přílohách. Jedno typické spojení ze souboru .adi vypadá může vypadat takto:

```
<QSO_DATE:8:d>20150208 <TIME_ON:6>051228 <CALL:3>K1N <FREQ:7>7.00358 <PFX:2>K1  
<DXCC:3>182 <CONT:2>NA <MODE:2>CW <BAND:3>40m <RST_SENT:3>599 <RST_RCVD:3>599  
<CQZ:1>8 <IOTA:6>NA-098 <QSL_RCVD:1>N <LOTW_QSL_SENT:1>Y  
<LOTW_QSLSDATE:8:d>20150213 <EQSL_QSL_SENT:1>Q <QSL_VIA:4>WN1F <EOR>
```

Obrázek 3 - Ukázka 1 spojení ze souboru (zdroj: vlastní)

Takových spojení může být v jednom nahrávaném souboru tisíce. Samozřejmě ne vždy má přesně takovou strukturu, ta je proměnlivá. Důležitá jsou pole **QSO_DATE**, **TIME_ON** – což jsou datum a čas spojení, **CALL** – stanice (uživatel), která spojení navazovala, **BAND** – pásmo, **MODE** – mód spojení. Těchto 5 údajů musí mít každé spojení, jinak je špatné a zahazuje se.

2.3.5 Správa uživatelů

Správa uživatelů zahrnuje pouze přehled a případnou úpravu uživatelů (stanic), kteří navázali alespoň jedno spojení se stanicemi zanesenými v systému. Eviduje se značka stanice a dosavadní počet uskutečněných QSO.

Pro snadnější orientaci v uživatelích, kterých budou zřejmě tisíce, je třeba vyřešit jejich snadné filtrování tj. vyhledávání na základě značky.

Plánovaný počet uživatelů, kteří budou aplikaci reálně používat, jsou stovky až tisíce. V databázi budou ovšem uloženy tisíce uživatelů. Při vývoji je na tuto skutečnost potřeba myslet a vše navrhovat, tak aby se zbytečně databáze nepřetěžovala.

2.3.6 Evidence spojení

Tato část bude sloužit pro přehledy jednotlivých uskutečněných spojení nahraných do systému. O spojeních je nutné z hlediska soutěže evidovat pouze datum, čas, pásmo, mód a stanici, která spojení vykonala. Nejpřehlednější bude prezentace pomocí tabulky. Jeden řádek = 1 spojení. Zobrazovat se musí všechna spojení, tedy i veškeré duplicity. Z funkčního hlediska bude moci administrátor jednotlivá spojení pouze smazat, více zásahů není třeba.

Filtrovat spojení lze pomocí stanice, na kterou bylo příslušné spojení provedeno. Počet spojení nahraných v databázi se odhaduje, i díky duplicitám, na desítky tisíc. Aby se předešlo zdlouhavému načítání stránky a v neposlední řadě zpřehlednila práce v tomto obrovském množství záznamů, je třeba efektivně vyřešit stránkování.

Druhou funkcionalitou je možnost zjistit přesný počet uskutečněných spojení v každém módu pro každou vysílací stanici. Přehled bude sloužit správcům pro informaci, na kterém pásmu a v jakém módu se vysílá více nebo naopak méně – na které módy se více zaměřit, aby byl poměr vyvážený.

2.3.7 Správci systému

Správci systému mají na starosti, jak již bylo uvedeno, dvě důležité funkce – **nahrávání dat** a **vysílání stanic**. Přístup do aplikace budou mít prostřednictvím přihlašovacího jména a vygenerovaného hesla, které jim může přidělit pouze administrátor. Z administrátorského hlediska zahrnuje tato položka dvě části:

- **Přehled správců** – u správců se bude evidovat pouze jméno (příjmení), přihlašovací jméno (login), heslo a datum registrace. Administrátor bude mít možnost každého správce smazat (znemožnit mu přístup do systému) nebo mu vygenerovat nové heslo v případě ztráty či zapomnění. Přehled správců bude řešen pomocí tabulky.
- **Přidat správce** – jednoduchý formulář, administrátor vyplní přihlašovací jméno (které musí být unikátní) a jméno (popřípadě příjmení) budoucího správce. Systém musí vygenerovat heslo pro přístup a vše uložit do databáze. Přihlašovací údaje, které musí administrátor správci předat, se zobrazí po úspěšném uložení.

2.3.8 Správa diplomů

Za určitý počet uskutečněných QSO soutěžící radioamatéři získají možnost stažení diplomu. Diplomy se dělí na dvě hlavní skupiny:

- **Vysílači** – skupina, která sama aktivně navazuje spojení se stanicemi zařazenými do soutěže. Těchto soutěžících je nepoměrně více než posluchačů.
- **Posluchači** – skupina, která jen „naslouchá“ uskutečňovaná spojení vysílačů.

Obě skupiny mohou získat 3 druhy diplomů – bronzový, stříbrný, zlatý. Administrátor musí mít přehled o jednotlivých vygenerovaných diplomech, které se budou filtrovat na základě druhu a také značky uživatele. Je třeba oddělit přehled diplomů vysílačů a posluchačů. Přehledy budou dva, každý pro svou skupinu. O diplomu se bude evidovat jeho pořadové číslo, značka uživatele a jeho jméno s příjmením, počet QSO v době stažení, datum a čas stažení.

Data posluchačů, o tom, kolik spojení „odposlechli“, nikde nejsou zanesena. Posluchači budou sami posílat lístky s počtem spojení administrátorovi, který na jejich základě bude **vytvářet diplomy**. Do formuláře administrátor zadá jméno a příjmení, unikátní značku posluchače a dosažený počet QSO. Na základě počtu QSO skript vygeneruje hotový diplom – bronzový, stříbrný nebo zlatý, který se musí posluchači doručit.

2.3.9 Evidence aktivity

Správce, nějakým způsobem zasahujících do systému, bude poměrně dost (kolem 30). Administrátor musí mít alespoň základní informace o jejich aktivitě a jejich zásadách do aplikace. Kromě sledování toho kdo a kdy nahrál jaký soubor, se budou sledovat tyto aktivity:

- **Logy o přihlášení** – při přihlášení do systému se bude ukládat datum, čas, IP adresa, unikátní značka a jméno správce.
- **Logy o vysílání** – tato funkcionalita se přímo váže na **Vysílání stanic**. Při jakékoliv změně ve vysílání (zahájení či ukončení vysílání na určitém pásmu v konkrétním módu) se uloží log o této aktivitě a to ve formě značky a jména správce, pásma a módu, ve kterém došlo k zahájení či ukončení vysílání a také

samozřejmě času a data. Logy se budou zobrazovat (filtrovat) podle jednotlivých stanic. Bude jich velké množství, stejně jako jinde se musí efektivně vyřešit stránkování.

2.4 Shrnutí analýzy

Z analýzy vyplývá, že Český radioklub požaduje velice specifickou webovou aplikaci, která bude vytvořena na míru konkrétním potřebám radioamatérů. Žádný podobný systém se neprodává ani není volně dostupný – musí se navrhnout, vytvořit a implementovat do praxe nová aplikace.

Podle celosvětových akcí podobného druhu, pořádaných zahraničními radiokluby, se dá usuzovat, že webová aplikace bude hojně navštěvována. Databáze bude vytěžována, s čímž se při návrhu a samotné realizaci musí počítat. Zároveň se klade důraz na jednoduchost, rychlost, efektivitu při práci a také univerzálnost. Systém by se měl dát snadno přizpůsobit, pokud by například v budoucnu radioklub pořádal další podobnou akci.

3 VLASTNÍ NÁVRH ŘEŠENÍ

Tato část se zabývá návrhem webové aplikace pro potřeby radioamatérské soutěže. Vše bude tvořeno na základě požadavků specifikovaných v předchozí části.

3.1 Souhrnný pohled na aplikaci

Vzhledem k požadavkům bude aplikace naprogramována v jazyce PHP s využitím databázového systému MySQL. Tyto technologie jsou vybrány vzhledem k rozšířenosti a dostupnosti. Na serveru, na kterém celá aplikace „poběží“, jsou právě tyto technologie, v novějších verzích, nainstalované.

Na aplikaci se musíme dívat ze tří pohledů podle přístupových práv:

- Uživatel
- Správce
- Administrátor

Každá z funkčních částí podle přístupových práv (viz Graf 2 – Přístupová práva) je popsána i s ukázkami programového kódu v následujících kapitolách. Vzhledem k omezenému prostoru této práce budou ukázky zjednodušeny, což ovšem nebude mít vliv na funkčnost. Ukázkové části kódu budou „osekány“ o bezpečnostní prvky, zdlouhavé syntaktické pasáže kódu, a veškerou „omáčku“, která najde opodstatnění jen v reálném provozu. Pro potřeby ilustrace funkčního hlediska jsou tyto ukázky plně postačující.

Aplikace využívá jednu databázi, která se ovšem skládá z více databázových tabulek. Nejdůležitější je tabulka „spojeni“, do které se ukládají vyextrahovaná data z nahrávaných souborů. Tato tabulka je největší a nejvytěžovanější. Ostatní tabulky týkající se diplomů, informací o ložích, správců, uživatelů, stanic atd. tak vytěžované nebudou.

Vzhledem k předpokládané vytiženosti se žebříčky budou přepočítávat 2 – 3x denně pomocí skriptů, které bude pravidelně spouštět Cron. Aby se databáze zbytečně

nevytěžovala, budou se informace o žebříčkovém umístění nebo celkový počet QSO, uvedený v hlavičce každé stránky, ukládat do textových souborů, ze kterých se budou číst. Předejde se tak neustálým (zbytečným) dotazům do databáze, které by každý návštěvník, při změně jakékoli stránky, generoval. Informace v mapě spojení každého jednotlivého uživatele budou samozřejmě aktuální, budou se generovat při každé návštěvě z databáze znova.

3.2 Nastavení serveru a php.ini

Server, na kterém aplikace „poběží“ má následující parametry:

- **Operační systém:** Debian Linux 7
- **Procesor:** Intel® Xeon™ CPU 3.40GHz, 4 jádra
- **RAM paměť:** 8GB
- **Pevný disk:** 260 GB
- **Verze MySQL:** 5.5.43
- **Verze PHP:** 5.5

Tyto parametry jsou naprosto dostačující pro bezproblémový chod aplikace.

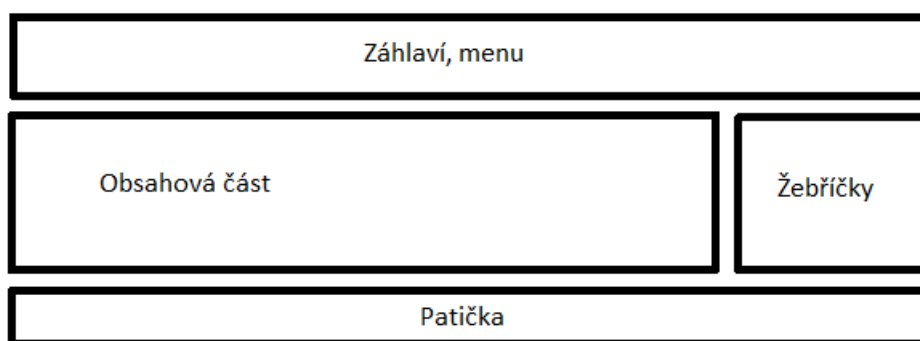
Než začneme vytvářet dílčí skripty aplikace, musíme pro správný chod nastavit PHP. K tomuto účelu slouží konfigurační soubor php.ini. Můžeme v něm nastavit volby jazyka, zpracování dat, chyb, sessions atd. Uvedu zde nejdůležitější volby nastavení:

- `Max_execution_time = 3600` – čas v sekundách po který se bude skript maximálně vykonávat. Při zpracování hodně velkých nahraných souborů nebo přepočítávání databáze může být tento čas opravdu velký.
- `Max_input_time = 400` – čas v sekundách, kterou skript může strávit při přijímání nějakého vstupu.
- `File_uploads = ON` – povolení nahrávat soubory
- `Upload_max_filesize = 14M` – maximální velikost nahrávaných souborů s daty
- `Session.auto_start = 1` – Superglobální proměnné session se nemusí ručně „startovat“

Možností nastavení je samozřejmě více, pro ukázkou jsem uvedl ty nejzákladnější. Všechny hodnoty v nastavení php.ini jsou mírně nadsazené. Pro potřeby této aplikace jsou plně dostačující, což se ukázalo i při testování.

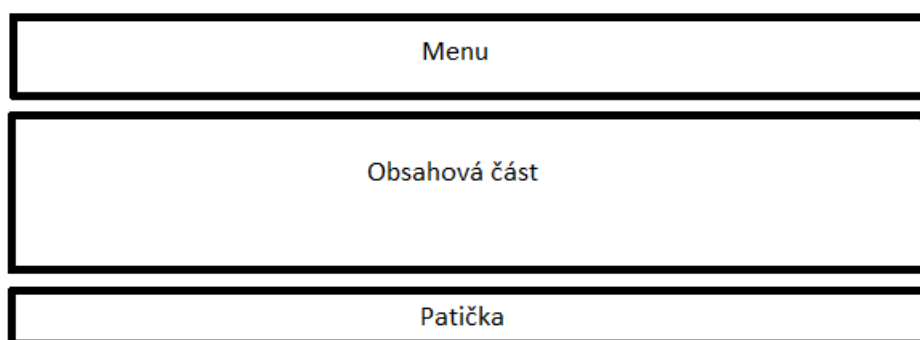
3.3 Grafický návrh

Grafický návrh se při vývoji aplikace, vzhledem k měnícím se požadavkům, poměrně dost měnil. Jako nejlepší, výsledný layout, se pro uživatelskou část aplikace ukázal dvousloupcový se záhlavím a vrchním, horizontálním menu vedle loga.



Obrázek 4 - Layout pro uživatele (zdroj: vlastní)

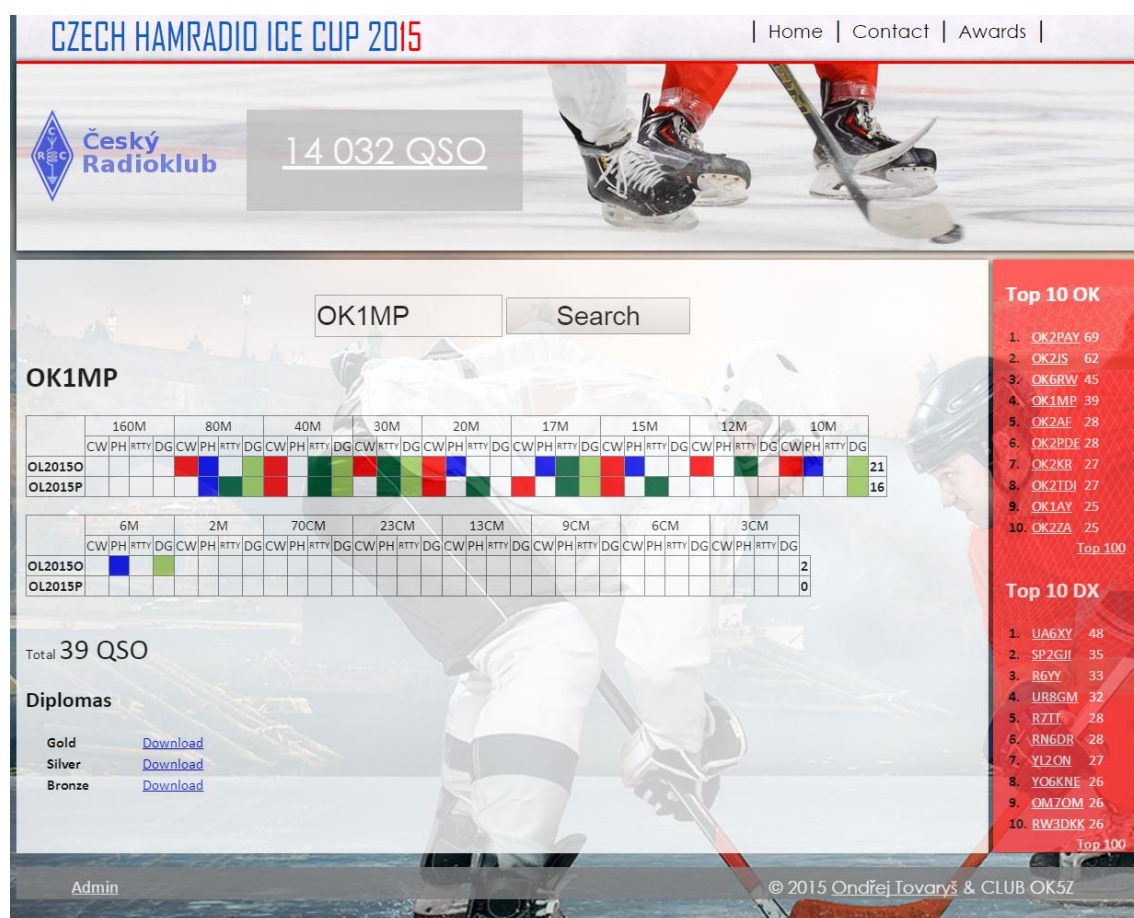
Pro administrátorskou část je nejpraktičtější jednoduché horizontální menu s pod ním umístěnou obsahovou částí.



Obrázek 5 - Layout pro administraci (zdroj: vlastní)

Při návrhu se počítalo i se zobrazováním aplikace na mobilních telefonech či tabletech. Šířka jednotlivých oddílů je proměnlivá (v procentech). Stránka se zobrazuje přes celou obrazovku (display) – 100% šířka.

Soutěž se koná speciálně při Mistrovství světa v hokeji, byla tedy zvolena hokejová tematika. Vše bylo tvořeno s ohledem na to, že systém se může využívat i při příštích akcích, které s hokejem nemusí mít nic společného. Není tedy problém relativně rychle grafický návrh změnit, tj. přizpůsobit novým podmínkám. Reálný vzhled ilustruje následující obrázek:



Obrázek 6 - Vzhled uživatelské mapy spojení (zdroj: vlastní)

Pro průhlednost jednotlivých oddílů, tak aby vystoupil obrázek na pozadí, se využívá CSS vlastnosti *opacity*. V některých hodně starých prohlížečích ještě tato vlastnost není podporována. Naštěstí v dnešní době je již minimum uživatelů, kteří tyto prohlížeče

využívají (což dokládají i statistiky uvedené v přílohách). Špatné zobrazení (neprůhlednost) ve starých prohlížečích nemá žádný vliv na funkčnost.

Grafický návrh byla „vedlejší kolej“ celé aplikace. Daleko důležitější byla pro potřeby radioamatérů správná a bezchybná funkčnost systému.

3.4 Přehled QSO

Tato kapitola popisuje tři základní funkce pro uživatele. Společně s přehledem vysílání stanic tvoří uživatelské rozhraní aplikace.

3.4.1 Mapa spojení

Mapu spojení tvoří dvě na sobě nezávislé HTML tabulky se stálým záhlavím. Jedna je pro krátké vlny (160M – 10M) a druhá pro velmi krátké vlny (6M – 3CM). Tabulku nejdříve naplníme stanicemi, každý řádek je jedna stanice, kterou jsme získali z databázové tabulky „stanice“.

Následně z tabulky „spojení“ vybereme číslo sloupce. V této tabulce je mimo informací o spojení i číslo sloupce, pro který je v HTML tabulce dané spojení určeno. Díky tomu, že záhlaví je v těchto dvou tabulkách vždy stejné, číslo sloupce má každý mód jednoznačné. Příklad: spojení v pásmu 17M na mód RTTY má číslo sloupce 22 a bude ho mít vždy stejné. To jak se toto číslo z .adi souborů do databáze získává, je vysvětleno v kapitole „Nahrávání dat“.

Algoritmus pro „vybarvení“ jednotlivých správných políček tabulky je následující. Pomocí dotazu:

```
SELECT DISTINCTROW s.sloupec FROM spojeni s WHERE
    s.stanice = ".$stanice." AND
    s.uzivatel = ".$uzivatel." AND
    s.sloupec >= 1 AND s.sloupec <= 35
ORDER BY s.sloupec
```

získáme pro danou stanici čísla sloupců, seřazená podle velikosti, takže ve „vybarvování“ políček se postupuje zleva doprava. V tomto dotazu určeném pro krátké vlny vybíráme pouze čísla sloupců 1 – 35, protože více možností není (módů v pásmech je 35). Dotaz pro velmi krátké vlny (druhá tabulka) je stejný, ale možností je pouze 32. Aby se to nepletlo, v číslování sloupců pokračujeme 36 – 67.

Pokud dotaz pro danou stanici vrátí nulový počet řádků, pouze pomocí *for* cyklu vykreslíme 35 prázdných políček. V opačném případě postupujeme takto:

- Cyklem *while* budeme postupně procházet všechny čísla sloupců. Číslo sloupce si přiřadíme do proměnné *sloupec*.
- Pomocí konstrukce vícenásobného větvení *switch* zjistíme barvu políčka pro daný sloupec. Každé políčko má přesně danou barvu, například:

```
switch($sloupec){  
    case 1: $barva = "#ff0000"; break;  
    case 19: $barva = "#a2cd5a "; break;  
    ...  
}
```

- Barvu pro vybarvení máme. Nyní pomocí cyklu *for* procházíme od 1 do 35. Pokud počet dosavadních průchodů = *sloupec*, vykreslíme políčko s danou barvou pozadí, proměnnou *\$i* zvýšíme o 1 a vyskočíme z *for* cyklu pomocí *break*. Pokud počet dosavadních průchodů != *sloupec*, vykreslíme prázdné políčko a cyklus pokračuje dál.
- Poslední buňka každého řádku je celkový počet spojení na danou stanici, které získáme jednoduše – počtem řádků vrácených SQL dotazem uvedeným výše.

Stejný algoritmus je použit i pro druhou tabulku VKV, pouze s jinými hodnotami. Celkový počet QSO pro daného uživatele je součtem všech QSO pro danou stanici na KV i VKV.

Vyhledávání pomocí uživatelské značky je realizováno prostřednictvím formuláře a speciální proměnné `$_GET[]`, která předá data skriptu, potažmo SQL dotazu. Všechny

vstupy jsou samozřejmě ošetřeny k tomu určenými funkcemi tak aby potenciální útočník nemohl skriptu podstrčit něco, co neočekává. Mapu spojení ilustruje obrázek „Vzhled uživatelské mapy spojení“ v předchozí kapitole „Grafický návrh“.

3.4.2 Tisk diplomů

Po dosažení určitého počtu QSO se na stránce s mapou spojení pro daného uživatele zpřístupní možnost stažení diplomu. Po kliknutí na „Download“ se zobrazí formulář pro zadání jména a příjmení, které bude na diplomu. Značka radioamatéra je na diplomu napevno. Údaje se předají skriptu, který vygeneruje diplom pro stažení. Skript funguje zhruba takto:

- Nejprve proběhne kontrola, jestli si daný uživatel vůbec může diplom vygenerovat, pomocí dotazu, který vrátí počet jednotlivých uskutečněných spojení:

```
SELECT COUNT(DISTINCTROW sloupec, stanice) as kolik FROM spojeni
WHERE uzivatel = ".$uzivatel." AND sloupec <> 0
```

- Pokud kontrola neprojde, skončíme. Pokud projde, zjistíme pořadí posledního vydaného diplomu. Když žádný diplom vydán ještě nebyl, aktuální pořadí nastavíme na 1. Pokud už vydán byl, zkontrolujeme, jestli pro danou značku a druh diplomu nebyl diplom generován. Pokud ano, pořadové číslo zůstává, pokud ne, k poslednímu pořadovému číslu, které si z databáze vytáhneme, přičteme 1.
- Poté skript všechny informace o diplomu, včetně data a času, uloží do databáze.

Následně přichází na řadu samotné vytvoření diplomu. Grafické rozvržení diplomu je pro všechny tři druhy diplomu stejné. Na základě podmínek se mění pouze texty, které se do diplomu „dokreslují“ pomocí jazyka PHP a jeho funkcí:

- Nejprve si pomocí funkce *ImageCreateFromPNG()* do libovolné proměnné uložíme obrázek prázdného diplomu. Do proměnných si pomocí *ImageColorAllocate()* pro daný obrázek uložíme barvy, se kterými budeme pracovat. Vybereme font písma, který se hodí a je potřeba ho stáhnout ve formátu .ttf (tento soubor musí být uložen na serveru).

- Nyní následuje přesné pozicování textů do diplomu – využívá se funkcí *ImageTtfText()*, *ImageTtfBox()*, *ImageSX()* a další. Centrování libovolně dlouhého jména či značky přesně na střed vychází z úvahy: (šířka obrázku – (x pozice levého horního rohu textu + x pozice pravého horního rohu textu)) / 2.
- Před samotným výstupem obrázku (prohlížeč nám nabídne stažení hotového diplomu) musíme odeslat 2 hlavičky, aby prohlížeč věděl, co má očekávat a vykonat:

```
header('Content-Type: image/png');
header('Content-disposition: attachment; filename="diploma_mshockey_2015_cz.png";');
```

- Na závěr už jen vykreslíme hotový diplom pomocí funkce *ImagePng()* a odstraníme obrázek z paměti *ImageDestroy()*.

3.4.3 Žebříčky

Top 10

Žebříček Top 10 se nahrává z textového souboru, do kterého v určitých intervalech skript spouštěný Cronem aktualizuje údaje z databáze. Tento skript ukládá informace o pořadí prvních 10 lidí do textového souboru sestupně ve formátu:

značka_radioamatéra počet_QSO značka_radioamatéra2 počet_QSO2

Mezi jednotlivými textovými řetězci je jedna mezera. Pro zpracování tohoto textového souboru, je potřeba ho pomocí funkce *File_get_contents()* nejprve načíst. Potom ho zpracuje algoritmus:

- Funkcí *explode()* řádek rozdělíme na základě mezery a načteme do pole.
- Cyklem *for* procházíme jednotlivé textové řetězce. Proměnnou *\$i* nastavíme na 0, protože číslování hodnot pole začíná od 0. Jeden řádek žebříčku tvoří 2 hodnoty pole nacházející se bezprostředně po sobě. Na konci průchodu každého cyklu se musí zvednout hodnota *\$i* o 1, aby nový řádek začínal na správné hodnotě.

```
for($i = 0; $i <= count($pole_hodnot) - 1; $i++){  
    echo "<tr>  
        <td>$pole_hodnot[$i]</td>  
        <td>$pole_hodnot[$i + 1]</td>  
    </tr>";  
    $i++;  
}
```

Top 100

Žebříček Top 100 se nenačítá z textového souboru jako Top 10, protože frekvence návštěv zde nebude taková. Při každé návštěvě této stránky se vykoná nový SQL dotaz (pozn. login začínající „OK“ nebo „OL“ značí radioamatéra z České republiky):

```
SELECT login, pocet FROM uzivatele  
WHERE (login like 'ol%' OR login like 'ok%')  
ORDER BY pocet desc limit 100
```

Pro lepší přehlednost se zobrazí těchto 100 uživatelů v pěti tabulkách po dvaceti vedle sebe. Tabulky se chovají intuitivně. Pokud je v databázi například jen 68 uživatelů, nezobrazí se 5 tabulek, ale jen 4 a poslední z tabulek nemá všech 20 řádků, ale jen 8. Tohoto výsledku je dosaženo pomocí pole, do kterého jsou všechny řádky z databáze nahrány. Pole prochází *for* cyklus, omezený jeho velikostí, společně se stromovým rozhodováním.

3.5 Přihlašování do administrace

Na přihlašovací formulář, kde se musí zadat login a heslo, je možno se dostat z patičky každé stránky. Administrátor i správci mají přihlašovací formulář stejný. Práva se jim nastaví po přihlášení na základě loginu a hesla.

Celé přihlašování a správa administrace funguje na základě tzv. superglobální proměnné `$_SESSION[]`. Na základě vložených přihlašovacích údajů se v databázi zkontroluje, jestli se tam daný správce či administrátor nachází. Pokud ne, vypíše se hláška. Pokud

ano, vytvoří se `$_SESSION[id]` a přesměruje se pomocí hlavičky `header(„Location:“)` do administrace. Na začátku skriptu každé stránky se pomocí funkce:

```
function jeSpravce(){
    if (!isset($_SESSION["id"])) {return false;}
    $zjisteni = mysqli_query($mysqli, "SELECT id FROM spravci
    WHERE je_admin = 0 AND id = " . $_SESSION["id"]."");
    if(mysqli_num_rows($zjisteni) == 1) {return true;} else {return false;}
}
```

zjišťuje, jestli je vůbec uživatel přihlášen a má oprávnění k tomuto skriptu přistupovat. Obdobně vypadá i funkce pro kontrolu jestli je administrátor.

Všechny bezpečnostní prvky ohledně přihlašování a kontroly zde nebudou uvedeny. Pro příklad: heslo se neukládá v čisté formě, ukládá se jeho hash sha-512 se solí. Při přihlašování se porovnávají dva textové řetězce o délce 128 znaků. Nevýhodou tohoto postupu je, že při zapomenutí hesla toto heslo již nikdo nezjistí, musí se generovat heslo nové.

3.6 Správa stanic

Tato kapitola se věnuje správě stanic (volacích značek). Jedná se o stanice, které budou vysílat v České republice a radioamatéři z celého světa se s nimi budou spojovat.

3.6.1 Přidávání a přehled

Přidávání stanic do systému je přes formulář o jednom poli, kde se zadá název stanice. Skript název zkontroluje a pomocí SQL příkazu `INSERT` uloží do tabulky „stanice“. Takto vložená stanice se objeví v přehledu stanic.

Přehled ukazuje aktivní stanice, které jsou přidány v systému. Do HTML tabulky se jen pod sebe, pomocí SQL dotazu `SELECT`, vypíše názvy stanic. Každá stanice je zároveň odkaz na přehledy jednotlivých spojení, které na tuto stanici proběhly (viz „Evidence spojení“). U každé stanice je odkaz na upravení názvu – pro případné překlepy. Je to

řešeno pomocí formuláře, do kterého se z databáze vypíše název stanice. Po úpravě a uložení změn skript pomocí SQL příkazu *UPDATE* přímo upraví záznam v databázi.

3.6.2 Vysílání

Využívají se, co se záhlaví a názvů řádků týče, naprosto stejné HTML tabulky jako v mapě spojení. Rozdíl v nich je, že se příslušná políčka nevybarvují, ale obsahují formulářové checkboxy – každé políčko jeden. Při zaškrtnutí a uložení se do databázové tabulky „vysilani“ uloží informace o stanici a číslu sloupce, který byl zaškrtnut – vysílá se v daném pásmu a módu na dané stanici.

	160M				80M				40M				30M				20M				17M				15M				12M				10M			
	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	
OL2015O	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	
OL2015P	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	

	6M				2M				70CM				23CM				13CM				9CM				6CM				3CM			
	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG	CW	PH	RTTY	DG
OL2015O	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
OL2015P	☐	☐	☐	☐	☐	☐	☒	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Uložit

Obrázek 7 - Vysílání stanic (zdroj: vlastní)

Při kliknutí na červeno černý obrázek kříže se spustí skript, který na základě id smaže daný záznam z databázové tabulky „vysilani“ a přesměruje zpátky na tyto dvě tabulky. Princip vykreslování jednotlivých políček tabulek je z části podobný jako u mapy spojení. Funkčně se děje něco jiného a obě tabulky tvoří jeden HTML formulář. Algoritmus je oproti mapě spojení obohacený o pár podmínek, *for* a *while* cyklů, pro správné dopočítávání.

3.7 Nahrávání dat

Kapitola zabývající se jednou z nejdůležitější částí aplikace, a to dostávání dat do systému. Popisuje jak, čím, z čeho a v jakém formátu jsou tolik potřebná data získávána. Vše je popisováno velice stručně, protože tento proces je obsáhlý.

3.7.1 Přehled souborů

Přehled souborů tvoří tabulka, ve které jsou všechny nahrané a systémem zpracované soubory. O souborech jsou v tabulce zobrazeny i informace, které se ukládají do databáze při úspěšném nahrání souboru. Při kliku na název souboru se přímo v prohlížeči zobrazí jeho obsah v plné podobě. Z praktického hlediska jde pouze o spojení třech databázových tabulek a jejich výpis s příslušným stylováním.

3.7.2 Nahrát soubor

Formulář, prostřednictvím kterého, se soubory nahrávají na server, má dvě položky: select box pro výběr stanice, pro kterou se daný soubor se spojeními nahrává, a tlačítko pro výběr souboru z disku. V definici formuláře musíme pro správnou funkci odeslání souboru na serveru nastavit:

```
<form ENCTYPE="multipart/form-data" ...>
```

Položky (jednotlivé stanice) se do select boxu vypisují z databáze. Aby nedocházelo k nechtěnému vícenásobnému odeslání při kliku na tlačítko „Odeslat“, se toto tlačítko, po prvním stisku, pomocí Javascriptu stane neaktivní. V případě odeslání souboru na server, dochází k jeho zpracování následujícím způsobem:

- Kontrola, zda byl soubor odeslán a nachází se na serveru pomocí funkce *is_file()*.
- Kontrola, zda má soubor požadovanou příponu (.adi), jiná není možná.
- Kontrola všech souborů na serveru, jestli se jejich název neshoduje s nahrávaným souborem. Dva stejné názvy nejsou možné.
- Soubor se při nahrání na server uloží do nějaké dočasné složky, pod dočasným názvem. Musí být pomocí funkce *Move_uploaded_file()* přesunut do námi požadované složky. Zároveň musí být uložen ne s dočasným názvem, ale s názvem, s kterým byl nahrán na server. Tento název je uložen v asociativním poli *\$_FILES['nazev_položky']['name']*.
- Nyní se začíná pracovat přímo s obsahem souboru. Otevře se v módu pro čtení skrze funkci *fopen()* s parametrem „r“.

- Dokud nenastane konec souboru, prochází se cyklem *while* každý řádek souboru pomocí funkce *fgets()*. Pokud je řádek prázdný, pokračuje se v cyklu od začátku (příkaz *continue*).
- Soubor .adi se skládá ze dvou částí. Z hlavičky, kde jsou různá metadata o stanicích, spojeních atd., a z datové části (samotná vykonaná spojení), která začíná za značkou <eoh>. V každém procházeném řádku se značka hledá. Dokud není nalezena, všechny řádky se zahazují.
- Pokud je značka nalezena, testuje se, zda se datová část nachází na tomtéž řádku nebo až na řádku následujícím.
- Jednotlivá spojení jsou oddělena značkou <eor>, mohou být zaznamenána na libovolném počtu řádků, ale každé platné spojení musí mít parametry začínající těmito znaky: <QSO DATE, <TIME_ON, <CALL, <MODE, <BAND. Řádky se funkcí *explode()* rozdělí na jednotlivá slova, která se procházejí cyklem s podmínkou na konci a předchozí parametry se v textových řetězcích mezi značkami <eor> hledají.
- Řádky se procházejí tak dlouho, dokud není všech pět parametrů nalezeno a zároveň dokud není nalezena následující značka <eor>, značící konec jednoho spojení. Pokud nejsou tyto dvě podmínky splněny, spojení se zahazuje a začíná se nanovo.
- Když je všech pět parametrů v daném spojení nalezeno, do příslušných proměnných se uloží hodnoty nacházející se přímo za konečným znakem „>“ každého parametru (parametry nemají stálou formu, např. parametr „<TIME_ON>“ reálně vypadá např. „<TIME_ON:6>“ a může se různě měnit; prvních 8 znaků u tohoto konkrétního parametru je vždy stejných).
- Se získanými daty v proměnných se pracuje dále. Na základě pásma a módu se musí zjistit číslo sloupce pro potřeby vykreslení spojení do mapy spojení. Pomocí vícevláknového rozhodování *switch* testujeme pásmo (160M, 80M, ...), každá z větví *case* obsahuje další *switch* pro rozhodování o módu. Příklad: pro pásmo 80M mohou nastat 4 módy spojení a každý má jiné číslo sloupce, ale zároveň jeden mód může mít vícero označení (DG = BPSK31 = PSK63 = PSK125 atd.), ovšem číslo sloupce musí být stejné. Tuto situaci ilustruje následující část kódu:

```

case "80M": {switch(strtoupper($mod)){
    case "BPSK31": $sloupec = 8; break;
    case "PSK31": $sloupec = 8; break;
    case "PSK63": $sloupec = 8; break;
    case "PSK125": $sloupec = 8; break;
    .....
}

```

- Poté, co pro všechny možné kombinace získáme správné číslo sloupce, získaný datum a čas se upraví do správného formátu pro uložení do databáze a všechny informace se uloží jako jeden záznam do databázové tabulky „spojeni“. Cyklus pokračuje znovu od začátku, takto se uloží všechna správná spojení a po posledním se do databázové tabulky „info_soubory_adi“ uloží záznam o tom, co, kdo, kdy, kam a kolik nahrál pro potřeby „Přehledu souborů“.
- Zpracovávaný soubor se prostřednictvím *fclose()* uzavře a zobrazí se hláška o úspěšném nahrání souboru s počtem spojení uložených do databáze.

3.8 Správa uživatelů

Tento oddíl administrace slouží pro úplný přehled o uživateli. Jsou zde všichni, kteří provedli alespoň jedno spojení. V tabulce „uzivatele“ se společně se značkou eviduje i dosažený počet QSO. Údaje se několikrát denně Cronem „přepočítávají“ aby byly aktuální. Při kliku na značku radioamatéra je možno se dostat na jeho mapu spojení.

Filtrování na základě značky (při hledání konkrétního uživatele) je použito HTML formuláře s předáváním hodnot SQL dotazu pomocí metody *GET*. Stránkování, vzhledem k velkému množství záznamů, je vyřešeno poměrně složitě, ale intuitivně. Možnosti přesunu jsou: začátek – předchozí – následující – konec. Pracuje se s asociativním polem *\$_GET[]* a omezujícím prvkem v SQL dotazu – *limit*. Stránkuje se po padesáti záznamech. Skript se chová „chytře“, při přesunu na poslední záznamy vrátí ne posledních padesát, ale správně dopočítaný počet záznamů. Stránkování se zobrazí, až pokud počet záznamů překročí padesátku.

3.9 Evidence spojení

Evidence spojení je důležitá dílčí část administrace. Přehled spojení slouží pouze administrátorovi, položka „Počet spojení“ je přístupná i správcům.

3.9.1 Přehled spojení

V přehledu spojení se nachází všechna (i duplicitní) spojení vyextrahovaná z nahrávaných souborů. Jsou seřazena na základě data a času. Filtrují se podle jednotlivých stanic, na které byla provedena. Administrátor může každé spojení smazat, což má na starosti skript, kterému se předá id spojení.

3.9.2 Počet spojení

Statistika počet spojení poskytuje přímý „živý“ pohled na spojení v databázi. Při každém vstupu (nebo aktualizaci) na stránku s tímto skriptem se celá tabulka přepočítává znovu. Algoritmus pro počítání spojení pro jednotlivé módy každého pásma zjednodušeně funguje takto:

- Záhloví obou tabulek pro KV i VKV je stejné jako jinde. Cyklem *for* se prochází od 1 do 35 (pro VKV od 36 do 67) a v každém průchodu mj. proběhne tento dotaz (proměnná *\$i* je počet dosavadních průchodů):

```
SELECT COUNT(DISTINCTROW sloupec, uzivatel, stanice)
FROM spojeni where stanice = ".$stanice." AND sloupec = $i
```

- Z databáze se „vytáhne“ počet spojení pro daný mód, správně zformátuje a vypíše do buňky tabulky.
- K dosavadnímu celkovému počtu spojení se přičte hodnota této buňky a cyklus jde od začátku.
- Až projde poslední průchod, vypíše se do poslední buňky řádku celkový počet uskutečněných spojení pro danou stanici a vše jde od začátku, ale pro další stanici.

- Takto algoritmus „přepočítá“ obě tabulky a nakonec se sečte počet spojení pro KV a VKV, což dohromady dá celkový počet spojení v soutěži – ten se taktéž vypíše.

3.10 Evidence správců systému

Tato část administrace se skládá z přehledu všech správců a možnosti přidat správce do systému. Přehled je tvořen, podobně jako jiné přehledy, tabulkou. U každého správce jsou dvě možnosti – smazat a vygenerovat nové heslo.

Při kliku na červený kříž (ikona smazat) se objeví potvrzovací tabulka, po potvrzení je správce nenávratně smazán a do systému (administrace) již nemá přístup. Možnost vygenerovat nové heslo je zde z důvodu ztráty či zapomnění hesla. V této situaci musí správce kontaktovat administrátora, který mu nové heslo vygeneruje a předá.

K přidávání nových správců do systému slouží formulář. Vyplní se pouze přihlašovací jméno (značka správce) a jeho jméno s příjmením. Skript vygeneruje heslo a vše uloží do databáze. Heslo se generuje poměrně zajímavým algoritmem i se solí, tak aby byla zajištěna co největší bezpečnost. Přidělené heslo nelze změnit. Při úspěšném uložení nového správce se zobrazí přihlašovací údaje, které musí administrátor předat správci.

3.11 Správa diplomů

Správa diplomů soustřeďuje veškerou agendu týkající se „odměň“ pro radioamatéry za určitý počet úspěšných spojení.

3.11.1 Přehledy diplomů

Přehledy diplomů pro vysílače i posluchače jsou technicky řešeny stejně. Rozdíl je v tom, že diplomy pro posluchače „vytváří“ administrátor (viz následující podkapitola), zatímco diplomy pro vysílače si uživatelé, při dosažení určitého počtu QSO, generují sami.

V přehledu se zobrazují všechny vygenerované diplomy i s duplicitami. Pokud si někdo svůj diplom vygeneruje 20x, bude v přehledu tento počet. Diplomy jsou tříděny podle druhu (zlato, stříbro, bronz) a dají se filtrovat na základě značky, což je řešeno pomocí formulářového textového pole, které předává data do asociativního pole `$_GET[]` a to do SQL dotazu. Do databáze se neukládají celé hotové diplomy jako obrázky, ale pouze textové údaje daného diplomu.

3.11.2 Vytvořit diplom

Tvorba diplomu pro posluchače technicky vypadá podobně jako tisk diplomů pro uživatele, s tím rozdílem, že všechny hodnoty musí administrátor zadat na základě QSO lístku do formuláře ručně. Rozdíly v algoritmu na generování diplomu pro posluchače oproti vysílačům jsou tyto:

- Na základě zadaného počtu spojení se zjistí, jaký druh diplomu se bude tvořit.
- Pro každý druh diplomu jsou souřadnice pro textové řetězce jiné. Ty se musí nastavit.
- Posluchačské diplomy se ukládají do jiné databázové tabulky.
- Používá se jiný obrázkový podklad.

3.12 Evidence aktivity

Tato část administrace slouží k tomu, aby měl administrátor alespoň rámcový přehled o tom, co který ze správců a kdy v systému dělal.

3.12.1 Přihlašovací logy

V přihlašovacím skriptu je část programu, která při úspěšném přihlášení uloží do databáze o tomto přihlášení informace. Ukládá se čas, datum, login a IP adresa osoby, která se přihlašuje. IP adresa se získává pomocí superglobální proměnné `$_SERVER[]` – konkrétně `$_SERVER[„REMOTE_ADDR“]`. Datum a čas prostřednictvím PHP funkce `date()`.

Tyto informace se pro administrátora vypisují do přehledné tabulky, řazené podle data a času – od nejnovějších záznamů po nejstarší.

3.12.2 Vysílací logy

Pokud správce ve „Vysílání stanic“ zaškrtně políčko pro daný mód vysílání stanice a uloží tuto informaci do databáze, zároveň s tímto se provede SQL příkaz *INSERT*, který vloží do speciální databázové tabulky o této události záznam. Stejně tak se do této tabulky uloží záznam o tom, pokud správce klikem na červený křížek, vysílání pro tento mód ukončí. Tímto postupem se zabrání tomu, že jeden správce bude v tomto módu vysílat a jiný mu toto vysílání ukončí – jednoznačně je tedy určen ten, kdo vysílání začal a i ukončil.

V přehledu těchto logů jsou začátky vysílání zobrazeny zelenou barvou, ukončení vysílání znázorňuje barva červená. Logy jsou tříděny podle stanic, každá má svůj přehled. Řazení jsou od nejnovějších po nejstarší.

3.13 Ekonomické zhodnocení a přínosy

Celý systém je stavěn tak, aby se dal v případě potřeby poměrně jednoduše přizpůsobit na další podobnou akci – možnost opakovaného využití. Faktem je, že bez podobného systému se soutěž těchto rozměrů uskutečnit nedá.

Ve stavu v jakém systém fungoval po celou dobu soutěže (do budoucna se bude rozšiřovat o nové funkce), se doba potřebná pro naprogramování pohybuje okolo 140 člověkohodin. Náklady na vývoj aplikace se při hodinové sazbě 400,- Kč pohybují okolo 55 000,- Kč. Investice vložená do systému se při opakovaném využití „rozmělní“ na více částí.

Vzhledem k tomu, že tyto akce nejsou komerčního rázu, z finančního hlediska se nebude částka vložená do vývoje vracet dobře. Zčásti by se dalo, díky návštěvnosti, využít cílených reklam na úzký segment lidí – radioamatérů, popř. by se dalo využít sponzorských darů od pořadatelů velkých kulturních nebo sportovních akcí, při kterých se radioamatérské soutěže odehrávají.

3.14 Ostrý provoz

V ostrém provozu systém běžel 3. 5. – 17. 5. 2015. Zhruba 14 dní před tímto datem probíhalo finální testování, přidávání správců a zaškolování do obsluhování systému. Za dobu soutěže bylo v databázi evidováno **9 420 lidí** z celého světa, kteří provedli alespoň jedno spojení na jednu z registrovaných stanic. Celkem bylo provedeno **18 587 QSO** - unikátních spojení. Nejvíce spojení bylo na stanici OL2015P, v pásmu 15M, na módu PH a to **1 175 QSO**. Diplomů všech druhů bylo vygenerováno **992**. Systém obsluhovalo **29 lidí**.

Aplikaci navštívili lidé z 60 zemí světa. Nejvíce návštěvníků bylo z České republiky, Ruska, Ukrajiny, USA, Německa a Japonska, dále jsou statistiky velice vyrovnané. Podrobnější informace a statistiky ohledně návštěvnosti přikládám v přílohách.

Po celou dobu soutěže s aplikací nenastal žádný technický problém. Kladné ohlasy přicházely i od uživatelů ze světa. Do budoucna se bude aplikace rozšiřovat o nové funkce, její vývoj tedy nekončí.

ZÁVĚR

V teoretické části byly představeny technologie použité na tvorbu webové aplikace včetně základních informací o internetu a především jeho nejpoužívanější služby World Wide Web. Taktéž byl obecně popsán vývoj softwaru a jeho životní cyklus od analýzy po reálný provoz.

Analytická část byla zaměřena na požadavky na aplikaci. Ve zkratce byl představen Český radioklub a obecné informace o soutěži, pro kterou byla aplikace tvořena. Požadavky vycházely z potřeb organizátorů soutěže, vzhledem k tomu, že žádná podobná aplikace se neprodává, při návrhu se vycházelo z bodu 0.

Poslední část, vlastní návrh řešení, se zabývá tvorbou a implementací aplikace do reálného provozu. V jednotlivých kapitolách byly popsány funkční prvky systému. Rovněž byly popsány úvodní etapy tvorby, jako základní nastavení serveru nebo grafické rozvržení uživatelského rozhraní. Závěr této části byl věnován ekonomickému zhodnocení a přínosům aplikace nyní i do budoucna, taktéž bylo uvedeno pár poznatků z ostrého provozu systému.

Takto navržená a vytvořená aplikace může být v budoucnu použita pro ještě větší akce. Díky ní, mohou nyní i čeští radioamatéři pořádat akce světového rázu. Pro potřeby této soutěže byla aplikace naddimenzovaná, při návrhu se počítalo s ještě větším zatížením, než tomu potom ve skutečnosti bylo. Budoucí vývoj bude směřovat ke zpracovávání více formátů souborů či zkvalitňování administrátorského rozhraní.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] RENDA, M. *Český internet a MS Internet Explorer*. 1. vyd. Praha: Grada Publishing, 1999. 264 s. ISBN 80-7169-846-6.
- [2] LENGSTORF, J. a T. HANSEN. *PHP for Absolute Beginners*. 2.vyd. Berkeley, CA: Apress, 2014. ISBN: 978-1-4302-6814-7. Dostupné z: <http://books.google.cz/books?id=vhM7BAAAQBAJ&printsec=frontcover&hl=cs#v=onepage&q&f=false>
- [3] KOCH, M. *Datové a funkční modelování*. Brno : VUT FP, 2004. ISBN 80-214-2724-8.
- [4] LEISS, O. a J. SCHMIDT. *PHP v praxi: pro začátečníky a mírně pokročilé*. 1. vyd. Praha: Grada, 2010, s. 13-110. Průvodce (Grada). ISBN 978-80-247-3060-8.
- [5] PONKRÁC, M. *PHP a MySQL: bez předchozích znalostí : [průvodce pro samouky]*. Vyd. 1. Brno: Computer Press, 2007. ISBN 978-80-251-1758-3.
- [6] VRÁNA, J. *1001 tipů a triků pro PHP*. Vyd. 1. Brno: Computer Press, 2010. ISBN 978-80-251-2940-1.
- [7] CABRAL, K. S. a K. MURPHY. *Mysql administrator's bible*. Hoboken, N.J: Wiley, 2013. ISBN 978-111-8080-580.
- [8] PÍSEK, S. *HTML: začínáme programovat*. 3., aktualiz. vyd. [i.e.] 1. vyd. Praha: Grada, 2010. ISBN 978-80-247-3117-9.
- [9] JANOVSKEJ D. CSS styly – úvod. *Jakpsatweb.cz* [online]. [cit. 2014-11-10]. Dostupné z: <http://www.jakpsatweb.cz/css/css-uvod.html>

- [10] THAU, D. *Velký průvodce JavaScriptem: tvorba interaktivních webových stránek v praxi*. 1. vyd. Praha: Grada, 2009, 516 s. ISBN 978-80-247-2211-5.
- [11] INTERVAL. Jak na démona Cron. *Interval.cz* [online]. Datum vydání: 21. 4. 2002 [cit. 2015-03-12]. Dostupné z: <http://interval.cz/clanky/jak-na-demonu-cron/>
- [12] KABIR, M. J. *Apache server 2 kompletní příručka administrátora*. Praha : ComputerPRESS, 2004. 722 s. ISBN: 80-251-0319-6.
- [13] ŽIVĚ. Vládne webovým serverům Apache, nebo Microsoft? *Živě.cz* [online]. Datum vydání: 19. 9. 2011 [cit. 2015-03-12]. Dostupné z: <http://www.zive.cz/bleskovky/vladne-webovym-serverum-apache-nebo-microsoft/sc-4-a-158814/default.aspx>
- [14] KŘENA, B. a R. KOČÍ. *Úvod do softwarového inženýrství IUS: Studijní opora* [PDF]. Brno: VUT FIT, 28. února 2006.

SEZNAM OBRÁZKŮ

Obrázek 1 - Uživatelské prostředí phpMyAdmin (zdroj: vlastní)	17
Obrázek 2 - Tabulka spojení (zdroj: vlastní)	25
Obrázek 3 - Ukázka 1 spojení ze souboru (zdroj: vlastní).....	30
Obrázek 4 - Layout pro uživatele (zdroj: vlastní).....	36
Obrázek 5 - Layout pro administraci (zdroj: vlastní)	36
Obrázek 6 - Vzhled uživatelské mapy spojení (zdroj: vlastní).....	37
Obrázek 7 - Vysílání stanic (zdroj: vlastní)	44

SEZNAM TABULEK A GRAFŮ

Graf 1 – Organizační struktura ČRK (zdroj: vlastní)	24
Graf 2 – Přístupová práva (zdroj: vlastní)	27

SEZNAM PŘÍLOH

Příloha 1: Ukázkový soubor formátu .adi.....	I
Příloha 2: Statistika návštěvnosti po dobu soutěže.....	II

Příloha 1: Ukázkový soubor formátu .adi

ADIF export from WT version 4.7.0 (c) F5MZN (<http://www.win-test.com>)

Callsign : OL2015P SECTION/ZONE: Grid square :

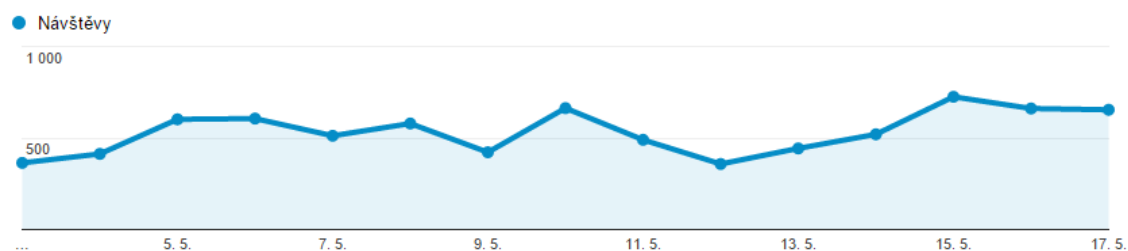
Name : Valentino

<CONTEST_ID:24>DX expedition (HF+50MHz)

<EOH>

<QSO_DATE:8:d>20101004 <TIME_ON:6>193139 <CALL:5>OK2TS <FREQ:7>1.80000
<PFX:3>OK2 <DXCC:3>503 <CONT:2>EU <MODE:3>SSB <BAND:4>160m <RST_SENT:2>59
<RST_RCVD:2>59 <SRX:1>1 <CQZ:2>15 <QSL_RCVD:1>Y <LOTW_QSL_SENT:1>Q
<EQSL_QSL_SENT:1>Q <COMMENT:3>gkr <NAME:8>rossi
<NOTES:19>Rx=1 800kHz|SerNr=1 <EOR>
<QSO_DATE:8:d>20101004 <TIME_ON:6>193214 <CALL:6>OK2PAQ <FREQ:7>1.80000
<PFX:3>OK2 <DXCC:3>503 <CONT:2>EU <MODE:3>SSB <BAND:4>160m <RST_SENT:2>59
<RST_RCVD:2>59 <SRX:1>2 <CQZ:2>15 <QSL_RCVD:1>N <LOTW_QSL_SENT:1>Q
<EQSL_QSL_SENT:1>Q <COMMENT:3>hop <NOTES:19>Rx=1 800kHz|SerNr=2 <EOR>
<QSO_DATE:8:d>20101004 <TIME_ON:6>193257 <CALL:5>OK1XQ <FREQ:7>1.80000
<PFX:3>OK1 <DXCC:3>503 <CONT:2>EU <MODE:3>SSB <BAND:4>160m <RST_SENT:2>59
<RST_RCVD:2>59 <SRX:1>3 <CQZ:2>15 <QSL_RCVD:1>N <LOTW_QSL_SENT:1>Q
<EQSL_QSL_SENT:1>Q <COMMENT:3>bhn <NOTES:19>Rx=1 800kHz|SerNr=3 <EOR>
<QSO_DATE:8:d>20101004 <TIME_ON:6>193314 <CALL:6>OK1MST <FREQ:7>1.80000
<PFX:3>OK1 <DXCC:3>503 <CONT:2>EU <MODE:3>SSB <BAND:4>160m <RST_SENT:2>59
<RST_RCVD:2>59 <SRX:1>2 <CQZ:2>15 <QSL_RCVD:1>N <LOTW_QSL_SENT:1>Q
<EQSL_QSL_SENT:1>Q <COMMENT:3>fhh <NOTES:19>Rx=1 800kHz|SerNr=4 <EOR>
<QSO_DATE:8:d>20101004 <TIME_ON:6>193633 <CALL:6>OK2BTT <FREQ:7>1.80000
<PFX:3>OK2 <DXCC:3>503 <CONT:2>EU <MODE:3>SSB <BAND:4>160m <RST_SENT:2>59
<RST_RCVD:2>59 <SRX:1>6 <CQZ:2>15 <QSL_RCVD:1>Y <LOTW_QSL_SENT:1>Q
<EQSL_QSL_SENT:1>Q <COMMENT:3>hsu <NOTES:19>Rx=1 800kHz|SerNr=5 <EOR>
<QSO_DATE:8:d>20101004 <TIME_ON:6>193712 <CALL:6>OK2ZLD <FREQ:7>1.80000
<PFX:3>OK2 <DXCC:3>503 <CONT:2>EU <MODE:3>SSB <BAND:4>160m <RST_SENT:2>59
<RST_RCVD:2>59 <SRX:1>3 <CQZ:2>15 <QSL_RCVD:1>N <LOTW_QSL_SENT:1>Q
<EQSL_QSL_SENT:1>Q <COMMENT:3>hol <NOTES:19>Rx=1 800kHz|SerNr=6 <EOR>

Příloha 2: Statistika návštěvnosti po dobu soutěže



Země	Návštěvy	Návštěvy v %
1. Czech Republic	2 667	33,29 %
2. Russia	1 316	16,43 %
3. Ukraine	584	7,29 %
4. United States	437	5,45 %
5. Germany	413	5,15 %
6. Japan	344	4,29 %
7. Latvia	244	3,05 %
8. Poland	240	3,00 %
9. Slovakia	186	2,32 %
10. United Kingdom	163	2,03 %

Operační systém	Návštěvy	Návštěvy v %
1. Windows	7 048	87,97 %
2. Macintosh	416	5,19 %
3. Android	291	3,63 %
4. iOS	107	1,34 %
5. Linux	74	0,92 %
6. Windows Phone	55	0,69 %

Prohlížeč	Návštěvy	Návštěvy v %
1. Firefox	2 639	 32,94 %
2. Chrome	2 381	 29,72 %
3. Internet Explorer	1 599	 19,96 %
4. Opera	931	 11,62 %
5. Safari	166	 2,07 %
6. YaBrowser	128	 1,60 %
7. Android Browser	57	 0,71 %
8. Maxthon	38	 0,47 %
9. MRCHROME	23	 0,29 %