

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Aplikace pro podporu činnosti modelářského klubu
(s využitím desktopu a webu)
Bakalářská práce

Autor: Martin Kupka

Studijní obor: Aplikovaná informatika, prezenční forma

Vedoucí práce: doc. Ing. Filip Malý, Ph.D.

Hradec Králové

21.února 2015

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 21.2.2015

Martin Kupka

Poděkování:

Děkuji vedoucímu bakalářské práce doc. Ing. Filipu Malému, Ph.D. za metodické, pedagogické a odborné vedení mé práce. Jeho rady mi velmi pomohly při jejím zpracování. Zároveň děkuji své přítelkyni Markétě Provazníkové a rodině za podporu po celou dobu studia.

Anotace

Bakalářská práce se zabývá problematikou podpory modelářského klubu. Zaměřuje se především na návrh nové aplikace včetně její implementace, která zjednoduší průběh modelářských závodů. Úvodní část je věnována popisu použitých technologií PHP Nette a Java. Následně se zabývá analýzou současného stavu elektronické podpory, na jejímž základě se pak dále věnuje návrhu vhodné aplikace. Další část se zabývá samotnou implementací navrhované aplikace včetně jejího popisu.

Hlavním cílem této práce je tak analýza současného stavu elektronické podpory modelářského klubu. Dále na základě této analýzy pak navrhnout vhodnou aplikaci včetně implementace. Díky tomu tak přispět k všeobecnému vylepšení současných podmínek pro pořádání závodů. Zavedením nové aplikace se docílí zrychlení, zjednodušení celého procesu a eliminují se chyby lidských faktorů.

Annotation

**Title: The application supporting the activity of model club
(with using desktop and web)**

The bachelor thesis deals with problems in the support of the modelling club. It focuses on designing a new application including the implementation which simplifies the process of the model competitions. In the introduction is the description of used technologies PHP Nette and Java and it is followed by the analysis of a contemporary situation of electronic support. This analysis is the basis for design of the most suitable application. The next part is about the implementation of suggested application with description of the whole application.

The main aim of the bachelor thesis is the analysis of contemporary conditions of electronic support for modelling club. According to this analysis will be designed suitable application including the implementation. It will contribute to the general improvement of contemporary conditions for organizing competitions. With the installation of a new application will be the process quicker, easier and mistakes of human factor will be eliminated.

Obsah

1	Úvod.....	1
2	Popis technologií PHP Nette a Java	2
2.1	Popis Technologie PHP	2
2.2	Popis PHP Frameworku Nette.....	3
2.2.1	Obecné informace.....	3
2.2.2	Ladící nástroje.....	4
2.2.3	Zabezpečení	6
2.3	Popis technologie Java.....	8
2.3.1	Obecné informace.....	8
2.3.2	Klíčové vlastnosti a rozdělení Javy	9
3	Analýza současné podpory činnosti.....	13
4	Návrh aplikace.....	16
4.1	Návrh webové části	17
4.2	Návrh desktopové části.....	19
5	Implementace aplikace	21
5.1	Implementace webové části.....	21
5.1.1	Příprava.....	21
5.1.2	Popis struktury	21
5.1.3	Popis aplikace	24
5.2	Implementace desktopové části	29
5.2.1	Příprava.....	29
5.2.2	Popis struktury	30
5.2.3	Popis aplikace	30
6	Shrnutí výsledků.....	34
7	Závěry a doporučení	36
8	Literatura a zdroje	37

Seznam obrázků

Obr. 1 Debugger bar [7].....	4
Obr. 2 Zvýraznění chyby v kódu [7].....	5
Obr. 3 Vzor přihlášky [12]	13
Obr. 4 Zapisovací arch z programu LModel.....	15
Obr. 5 Chování aplikace z pohledu uživatele.....	17
Obr. 6 Class model využitý ve webové části	19
Obr. 7 Interakce mezi MVP společně s uživatelem a databází [15].....	22
Obr. 8 Layout webové části.....	24
Obr. 9 Historie přístupů uživatelů podle rozlišení monitoru [16].....	25
Obr. 10 Ukázka formátování pro tisk.....	26
Obr. 11 Dialogové okno pro hledání souboru v počítači	31
Obr. 12 Zobrazení načtených dat ze souboru	31
Obr. 13 Tabulka pro zapisování jednotlivých jízd.....	32

Seznam tabulek

Tabulka 1 Identifikace kódu PHP. [1]	3
--	---

1 Úvod

V současné době je podpora pořádání modelářských závodů značně nevyhovující. Bylo proto rozhodnuto situaci zanalyzovat a navrhnout řešení, které by pomohlo zefektivnit agendu spojenou s jejich pořádáním.

Vzhledem k charakteristice daného prostředí bylo v rámci této bakalářské práce rozhodnuto navrhnout řešení v podobě aplikace rozdělené na dvě části. Webová aplikace je vytvořena pomocí technologie PHP Nette, která je popsána ve druhé kapitole práce. Tato část aplikace bude sloužit primárně pro závodníky. Druhá desktopová část, vytvořena technologií Java, bude sloužit naopak pořadatelům. Ti na základě dat poskytnutých od závodníků z webového rozhraní a průběžných výsledků budou dále s daty pracovat. Technologii Java je rovněž věnována část druhé kapitoly.

Třetí kapitolou je analýza současné elektronické podpory, na jejímž základě je pak navrhována aplikace. Analýza vychází z vlastních zkušeností z pořádání závodů naším klubem, ze zkušeností účastníka závodů a zároveň z konzultací s pořadatelem. Cílem analýzy je co nejlépe zachytit současný systém a stanovit klíčové prvky, které nejvíce ovlivňují celý proces.

Ve čtvrté kapitole je popsán návrh aplikace. Ta je navržena tak, aby co nejlépe nahradila současný systém pořádání závodů. Prvotním cílem je maximálně eliminovat chyby lidských faktorů při pořádání a zjednodušit celý proces ze strany pořadatele i závodníka. Její koncepce vychází z objektového návrhu a principu znovupoužitelnosti, aby se v budoucnu dala dále vyvíjet, rozšiřovat a bylo tak snadnější reagovat na vzniklé změny v tomto odvětví.

V pořadí již pátá kapitola obsahuje implementaci vytvořenou na základě návrhu popsaném v předchozí kapitole. Vzhledem k rozdílnostem funkcionalit a použitých technologií jsou obě části aplikace popsány zvlášť. Zároveň je popsána také interakce obou částí.

Cílem práce je vytvořit na základě znalostí o dané problematice a analýzy současného stavu nové a efektivnější řešení. Výsledná aplikace by měla zjednodušit celý proces a otevřít nové možnosti.

2 Popis technologií PHP Nette a Java

2.1 Popis Technologie PHP

„Programovací jazyk PHP byl vyvinut k tomu, aby se statické stránky HTML obohatily o dynamické skriptování. Dokud PHP neexistovalo, používaly se pro tyto operace programy, které se volaly přes rozhraní Common Gateway Interface (CGI). K tomu, aby se mohl nějaký program CGI spustit, musel webový server spustit nový proces. Jedná se však o proces, který je pro operační systém velmi náročný a který pro své zpracování vyžaduje nějaký čas.“[1]

PHP je skriptovací jazyk pro tvorbu dynamického webu. Je to programovací jazyk, který pracuje na straně PHP serveru a jeho funkce se načítají jako jeho součást [2]. Vznik jazyka se v různých zdrojích liší, ale všeobecně by se dal datovat do let 1994 – 1996. Původně tato zkratka znamenala Personal Home Page. Od té doby prošlo PHP značnými změnami a i zkratka změnila svůj význam na Hypertext Preprocessor [3].

V současné době je k dispozici verze 5.5. *„Nejnovější verze programu kupodivu nemusí vždy být tou nejlepší. Ve světě softwaru šířeného pod hlavičkou Open Source není vždy snadné orientovat se v různých verzích programů, protože prakticky každý projekt má svoje vlastní číslování.“* [4] Proto je důležité si předem promyslet, jaká verze bude v projektu použita. Důležitým faktorem pro výběr verze je verze instalovaná na serveru. Ne vždy vlastníme webový server a zároveň i přístupy ke konfiguraci na pronajatých serverech se značně liší. Současně je třeba podotknout, že ne každý server podporuje technologii PHP 5. Dokonce existují i servery bez podpory PHP.

Pro správnou funkci PHP skriptů je zapotřebí mít správně nakonfigurovaný server. Při splnění této podmínky každý soubor, mající příponu .php, zpracovává interpreter (překladač) jazyka PHP. Překladač tento soubor načte a vykoná zadané instrukce. Výsledek vrátí webovému serveru a ten ho odešle zpět ke klientovi. Pro překladač jsou důležité zejména instrukce označující začátek a konec PHP kódu. Pokud tyto značky v souboru nenajde, vrátí obsah serveru nezměněn. Příklady příkazů identifikující úseky kódů PHP jsou uvedeny v následující tabulce.

Tabulka 1 Identifikace kódu PHP. [1]

<code><?php echo \$x; ?></code>	Standardní značky označující začátek a konec kódu PHP
<code><script language="php"> echo \$x; </ script></code>	Dlouhý zápis (důležité zejména pro Frontpage, který má s jinými způsoby zápisu potíže)
<code><? echo \$x; ?></code>	Krátký zápis (lze jej použít v kombinaci s parametrem <code>short_open_tag</code>)
<code><% echo \$x; %></code>	Verze ASP (lze ji použít v kombinaci s parametrem <code>asp_tags</code>)
<code><?= \$x; ?></code>	Krátký zápis bez příkazu <code>echo</code>
<code><%= \$x; %></code>	Verze ASP bez příkazu <code>echo</code>

První dva způsoby fungují vždy. Krátké zápisy a tvary ASP fungují pouze za předpokladu, že je správně nastaven program PHP. Poslední dvě varianty zápisu se nedoporučují. Při jejich použití se stává zdrojový kód nepřenositelným, jelikož ne na všech systémech je povolen parametr `short_open_tag`. Dalším důvodem, proč nepoužívat krátké zápisy, je skutečnost, že je nelze integrovat do platných dokumentů ve formátu XML a XHTML [1].

2.2 Popis PHP Frameworku Nette

Framework neboli aplikační rámec je softwarová struktura, která pomáhá při vývoji aplikací. Frameworky mohou být různě obsáhlé a mohou obsahovat podpůrné programy, knihovny, dokumentaci a také mohou podporovat návrhové vzory. Jejich cílem je převzetí typických problémů, rutinních záležitostí dané oblasti a tím vývojáři usnadnit práci a nechat mu tak větší prostor pro řešení ostatních problémů.

„Opravdoví programátoři nepoužívají frameworky. Píší webové aplikace přes příkazovou řádku rovnou na server. Tímto jim vzdáváme hold. Nám ostatním frameworky značně ulehčují práci. Díky Nette Framework budete méně psát, mít přehlednější kód a radost z práce. Přichází s celou řadou technologických inovací, vytvoříte s ním dokonale zabezpečené aplikace, které bude později radost rozšiřovat. A vyrostla kolem něj nejaktivnější komunita českých PHP vývojářů a používají jej významné společnosti.“ [5]

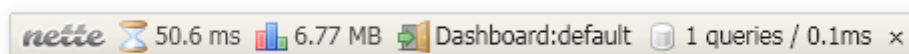
2.2.1 Obecné informace

Nette Framework je ryze český populární moderní nástroj pro vytváření webových aplikací v jazyce PHP verze 5. Při práci vychází vstříc potřebám a trendům, tím

usnadňuje práci a eliminuje bezpečnostní rizika. Nijak neomezuje a lze ho kombinovat i s jinými frameworky. Má velice promyšlený objektový návrh využívající nových vlastností, které poskytuje nová verze PHP 5. Psaný kód je mnohem přehlednější, podporuje Asynchronous JavaScript and XML (AJAX) a díky své neustále se rozšiřující nabídce doplňků podporuje znovupoužitelnost kódu. „*Nette Framework je šířen jako svobodný software, aby ho kdokoliv mohl používat. Můžete si vybrat, zda vám lépe vyhovuje licence New BSD nebo GNU General Public License (GPL) ve verzi 2 nebo 3.*“ [6] Díky tomu je minimalizována závislost uživatele na používaném softwaru. Pokud tvůrce přestane software dále vyvíjet, může se toho ujmout kdokoli jiný. V rámci Open-source licence ho lze bezplatně využívat i v komerčních projektech. K dnešnímu dni 2. ledna 2014 je nástroj k dispozici ve verzi 2.1.0.

2.2.2 Ladící nástroje

Jazyk PHP je jazykem, který dává programátorům velkou volnost, a proto se stal zdrojem těžko odhalitelných chyb. Velmi hodnotným ladícím nástrojem je tzv. Debugger, který patří mezi špičku mezi ladícími nástroji pro jazyk PHP. Pomáhá vývojářům rychle odhalit, opravit a logovat chyby. Při aktivaci debuggeru se zobrazí v pravém dolním rohu tzv. Debugger Bar, který zobrazuje základní informace a lze ho rozšířit o řadu doplňků [7].



Obr. 1 Debugger bar [7]

Dalším pomocníkem při odhalování chyb je jejich vizualizace včetně výjimek. Řádek, na kterém vznikla chyba, je zvýrazněn. Pomáhá k lepší orientaci v kódu a identifikaci chyby než pouze strohé vypsání informace o chybě. Příklad je zobrazen na následujícím obrázku [7].



Obr. 2 Zvýraznění chyby v kódu [7]

V pořadí již třetím pomocníkem je Debugger dump. Je to šikovnější a modernější náhražka funkce `var_dump`, která sice podrobně vypíše obsah proměnné, ale v prostředí HTML výpis není formátován a slije se tak do jednoho řádku. Funkce `Debugger::dump()`, která je součástí Debuggeru, tzv. Laděnky – prostředím pro odhalování chyb, nám vygeneruje podrobný výstup. Například při dumpování seznamu nám vypíše, čím a na jakém místě je seznam naplněn včetně datového typu. To je velice užitečné. Konkrétní případ je demonstrován na následujících ukázkách [7].

```
01 $arr = array(10, 20.0, TRUE, NULL, 'hello');  
02  
03 Debugger::dump($arr);  
04 // včetně jmenného prostoru Nette\Diagnostics\Debugger::dump($arr);
```

Ukázka naplnění seznamu daty [7]

```

01 array(5) {
02     [0]=> int(10)
03     [1]=> float(20.2)
04     [2]=> bool(true)
05     [3]=> NULL
06     [4]=> string(5) "hello"
07 }

```

Ukázka vygenerovaného výpisu hodnot funkcí Debugger::dump()[7]

2.2.3 Zabezpečení

„Co chvíli je hlášena bezpečnostní díra na dalším významném webu nebo je díry zneužito. To je nepříjemné. Pokud vám záleží na zabezpečení vašich webových aplikací, je Nette Framework zcela jistě tou nejlepší volbou.“ [8]

Cross-Site Scripting

První hrozbou, proti které je Nette Framework ochráněn, je metoda narušení webových stránek pomocí neošetřených vstupů tzv. Cross-Site Scripting (XSS). Tato metoda pracuje na principu podstrčení vlastního kódu právě přes neošetřený vstup. Díky tomu lze pozměnit obsah stránky nebo získat citlivé údaje o návštěvnicích. Proti tomu lze bojovat pouze jedním způsobem a to chránit všechny vstupy korektně. Tato skutečnost je velice riziková, protože stačí opomenout jeden jediný vstup a celý web tak může být napaden. Nette Framework proto nabízí technologii Context-Aware-Escaping, která automaticky ošetřuje všechny vstupy a navždy tak oprostí webovou aplikaci od hrozeb Cross-Site Scriptingu [8].

Cross-Site Request Forgery

Druhou hrozbou je Cross-Site Request Forgery (CSRF), což je útok založený na přesvědčení právě přihlášeného uživatele, aby navštívil stránku, která skrytě provede útok na webovou aplikaci, v níž je uživatel přihlášený. Díky tomuto útoku tak lze pozměnit nebo smazat obsah stránky, aniž by si toho uživatel všiml.

V rámci Nette Frameworku lze CSRF útokům předejít velice jednoduše příkazem `$form->add protection()`; Tím se daný formulář ochrání [8].

URL attack, control codes, invalid UTF-8

Tyto názvy útoků mají společný význam a to podstrčit webové aplikaci škodlivý vstup. Následky mohou být velice rozsáhlé a to od poškozených Extensible markup Language (XML) souborů přes nefunkční Really Simple Syndication (RSS) kanály až po získání citlivých informací z databáze. I proti těmto útokům Nette Framework automaticky chrání a programátor se tak nemusí o nic starat [8].

Session hijacking, session steling, session fixation

V útocích spojených se správou session útočník zcizí nebo podstrčí uživateli své session ID a díky tomu dostane přístup do konkrétní aplikace. Potom může měnit obsah nebo získávat citlivá data. Nemusí přitom znát ani uživatele včetně jeho hesla. Proti tomuto druhu útoku se lze chránit správnou konfigurací PHP serveru. Tuto konfiguraci Nette Framework automaticky zajistí a programátor se tak opět nemusí o nic starat [8].

Kontrola práv a parametrů v URL adrese

Dalším typem útoku je zneužití předávaných parametrů v URL adrese požadavku. Jelikož moje aplikace si předává parametry požadavků pomocí URL adresy, bylo zapotřebí zajistit jejich kontrolu s právy uživatele. Neošetření by znamenalo bezpečnostní hrozbu s vážnými důsledky v oblasti dat v databázi. Tento typ útoku Nette Framework neřeší a proto muselo být zabezpečení vytvořeno. Byl navržen a vytvořen mechanismus, který před provedením jakékoli operace kontroluje, zdali má uživatel práva provést danou operaci. Pokud ano, tak zdali má následně práva ji provést s příslušným parametrem.

Zamezení vícenásobného odeslání formuláře

Na rozdíl od předchozích problémů se nejedná o zabezpečení proti útoku, i když i tak by se dala tato problematika chápat. Jedná se o zabezpečení, které zamezuje uživatelům vícenásobně odeslat údaje z formuláře. V řadě aplikací je tato

skutečnost nežádoucí a to platí i pro tento případ. Pokud by vícenásobné odeslání nebylo ošetřeno, hrozila by celá škála důsledků. Od zkreslených výsledků v anketě přes duplicitní data v databázi různého charakteru až po nefunkční přihlašování uživatelů.

Tento problém nastává ve chvíli, když je server z jakéhokoli důvodu zaneprázdněný a jeho odezva není dostatečně rychlá. Projevem je například stále přístupné tlačítko po odeslání formuláře. Tehdy nastává doba, kdy se může zdát, že aplikace nic nedělá a nutí uživatele opět formulář odeslat.

Řešení, které je vytvořené v této práci, spočívá v přesměrování na jinou stránku ihned po stisknutí tlačítka. Teprve až potom dojde k odeslání formuláře. Díky tomu ihned po stisku tlačítka formulář zmizí a případné „zamrznutí aplikace“ tak proběhne mezi jeho zmizením a následným vykreslením přesměrované stránky. Uživatel tak nemá šanci znovu odeslat údaje z formuláře.

2.3 Popis technologie Java

Historie programovacího jazyka Java se datuje k počátku 90. let. Tehdy ve společnosti Sun Microsystems vznikl jednoduchý, ale efektivní jazyk pro spotřební elektroniku nazývaný Oak. Zaměstnanci této společnosti si byli vědomi dobré použitelnosti jazyka na internetu, a tak v roce 1995 přejmenovali jazyk na Java. 23. května 1995 vydali první verzi Java 1.0. Reagovali tím na tehdy rozsáhlý rozvoj internetu [9].

2.3.1 Obecné informace

Java se řadí mezi tzv. hybridní programovací jazyky. Takové jazyky jsou směsicí interpretovaných a překládaných jazyků. *„Hybridní program se nejprve přeloží do jakéhosi mezijazyka, který je vymyšlen tak, aby jej bylo možno co nejrychleji interpretovat. Takto přeložený program je potom interpretován speciálním interpretem označovaným často jako virtuální stroj. Virtuální stroj se mu říká proto, že se vůči programu v mezijazyku chová obdobně, jako procesor vůči programu v čistém strojovém kódu.“* [10]

Programy hybridního typu vznikly proto, aby spojily výhody obou kategorií. *„K tomu, aby v nich napsané programy mohly běžet na různých platformách, stačí pro každou platformu vyvinout potřebný virtuální stroj. Ten pak vytváří vyšší, mnohem univerzálnější platformu.“* [10]

2.3.2 Klíčové vlastnosti a rozdělení Javy

Java je plně objektově orientovaný jazyk. Na rozdíl od jiných jazyků již neumožňuje napsat objektově neorientovaný program, alespoň po formální stránce [10].

Jednoduchost

„Java je velice jednoduchý jazyk. Základy jeho syntaxe může člověk, který umí programovat, zvládnout během několika hodin. Pak už se může soustředit na poznání a pochopení funkce klíčových knihoven a na jejich co nejefektivnější využití.“ [10]

Z jednoduchosti jazyka nevyplývá omezenost jazyka. V současné době patří mezi nejpoužívanější programovací jazyky. Dají se v něm vytvořit aplikace všech možných rozměrů. Od malých programů pro čipové karty, přes mobilní aplikace po rozsáhlé distribuované aplikace [10].

Multiplatformnost

Jak bylo již zmíněno, jazyk Java se řadí mezi hybridní jazyky. Jednoduše řečeno to znamená, že kód v něm napsaný využívá oba způsoby zpracování kódu a je tak překládaný a zároveň interpretovaný.

Program napsaný v Javě se nejprve přeloží do mezijazyka nazvaného bajtkód, se kterým pak dále pracuje virtuální stroj Javy (Java Virtual Machine – zkratka VM). Virtuální stroj nejprve bajtkód zanalyzuje a pak interpretuje. Tím umožňuje, aby jeden program mohl běžet na různých počítačích s různým operačním systémem a konfigurací. Virtuální stroj tak přebírá zodpovědnost za správný běh aplikace. Program ani jeho uživatel nebo vývojář se nemusí z pohledu jeho běhu o nic starat.

Java běží pod operačními systémy Windows, Unix, Linux, MacOS, Solaris a řadou dalších [10].

Jazyk i platforma

„Jedním z klíčových záměrů autorů Javy bylo vytvořit nejenom jazyk, ale celou platformu. Tuto platformu realizuje výše zmíněný virtuální stroj spolu se základní knihovnou nejpoužívanějších funkcí.“ [10]

Rozdělení

Java se rozděluje do tří, resp. čtyř edic dle využití v cílovém zařízení. Navržená aplikace bude postavena na základní edici **J2SE** (Java 2 Standard Edition), která se používá k vývoji aplikací pro stolní počítače a jednoduché verze serverových aplikací.

Druhou edicí je **J2EE** (Java 2 Enterprise Edition), která je nadstavbovou verzí standardní edice. Obsahuje rozšiřující knihovny pro rozsáhlé distribuované aplikace. Jinými slovy má uplatnění na internetu.

V pořadí již třetí edicí, která je zjednodušenou verzí standardní edice je **J2ME** (Java 2 Micro Edition). Ta se používá pro vestavná zařízení jako mobily, domácí elektronika atd..

Poslední verze je označována jako **Java Card** a je to omezenější verze pro tvorbu aplikací pro čipové karty. Někdy bývá řazena pod J2ME [10].

Vývojářská sada

Z pohledu vývojáře je třeba rozlišovat dvě verze programových balíčků, které jsou k dispozici.

Jednodušší verzí je JRE (Java Runtime Environment), která je běhovým prostředím Javy. Jak již název napovídá, poskytuje vše potřebné k spuštění a běhu programů.

Rozsáhlejším balíkem je JDK (Java Development Kit), který slouží pro vývoj programů v jazyce Java. Někdy je tato verze označována jako SDK (Software Development Kit), což je sada pro vývoj softwaru. Jedná se o sadu, která poskytuje vše potřebné pro vývoj programu a v podstatě se jedná o balík JDK, rozšířený o základní vývojové nástroje jako jsou překladač, generátor, dokumentace, ladící program aj. [10].

Ukázka kódu

Poslední záležitostí, kterou se tato kapitola zabývá, je ukázka kódu v jazyce Java. Následuje tak ukázka kódu třídy Menu, na které je popsána syntaxe Javy. Tato třída vytváří panel Menu a zprostředkovává komunikaci mezi položkami menu a danými metodami daných tříd.

```
01 package gui;
02
03 import java.awt.event.ActionEvent;
04 import javax.swing.JMenu;
05 import javax.swing.JMenuBar;
06 ...
07
08 public class Menu extends JFrame {
09     private static final long serialVersionUID = 1L;
10     HlavniOkno hlavniOkno;
11
12     // seznam atributů - akcí
13     Action actKonec, actNactiPrihlasky, actUlozPrezenci;
14
15     public void createActions() {
16         hlavniOkno = new HlavniOkno(this);
17         actKonec = new AbstractAction("Konec", new ImageIcon
18             (getClass().getResource("/icons/CloseIcon.png"))) {
19             public void actionPerformed(ActionEvent e) {
20                 System.exit(0);
21             }
22         };
23         // Nastavíme akcelerátor (klávesovou zkratku)
24         actKonec.putValue(AbstractAction.ACCELERATOR_KEY,
25             KeyStroke.getKeyStroke(KeyEvent.VK_F4,
26                 InputEvent.ALT_MASK));
27     }
28     ...
29
30     public JMenuBar createMenu() {
31         JMenuBar menu = new JMenuBar(); // lišta menu
32
33         // Roleta Soubor
34         JMenu mnSoubor = new JMenu("Soubor");
35         mnSoubor.addSeparator();
36         mnSoubor.add(new JMenuItem(actKonec));
37         menu.add(mnSoubor);
38
39         // Roleta přihlášky
40         JMenu mnPrihlasky = new JMenu("Přihlášky");
```

```

41         mnPrihlasky.add(new JMenuItem(actNactiPrihlasky));
42         mnPrihlasky.add(new JMenuItem(actUlozPrezenci));
43         menu.add(mnPrihlasky);
44         // Roleta Nápověda
45         JMenu mnPomoc = new JMenu("Nápověda");
46
47         menu.add(mnPomoc);
48
49         return menu;
50     }
51 }

```

Řádek č. 01 nás informuje o umístění třídy v projektu. Řádky 03 až 06 nesou informace o importovaných třídách. Jsou to třídy, které jakýmkoli způsobem komunikují s třídou, do které jsou importovány. V tomto případě to je třída Menu.

Každý zdrojový soubor, který má nějaký spustitelný obsah, musí obsahovat třídu, uvedenou klíčovými slovy například `public class`, kde `public` označuje viditelnost třídy – veřejnou. Za klíčovými slovy musí následovat název třídy, shodný s názvem souboru – Menu. Za ním mohou být atributy `extends`, `implements`, které rozšiřují nebo implementují funkcionality do této třídy, včetně názvu třídy, se kterou jsou v daném vztahu.

Kód programu je pak hierarchicky uspořádán v blocích, uzavřených ve složených závorkách `{}`. Nejvyšší úroveň je třída, která kromě svých atributů a deklarací (řádky 09 – 13) obsahuje další bloky (metody), které svým obsahem spolu souvisí např. metoda `createActions` (řádky 15 -27).

Všechny příkazy v kódu jsou odděleny středníkem `;`. Na jednom řádku může být více příkazů, ale z hlediska přehlednosti a ladění se to nedoporučuje. Kromě příkazů zdrojový kód obsahuje volání metod tříd, které může probíhat třemi způsoby.

Prvním z nich se zapisuje ve formátu `metoda(parametry)`, kterým se volají metody uvnitř třídy (řádek 17). Druhým formátem, jak zapisovat volání metody je `objektTridy.metoda(parametry)`. Toto je standardní způsob, avšak objekt musí být předem vytvořen (řádek 36). Posledním formátem, který se využívá je `trida.metoda(parametry)`. Ten je povolen pouze u statických tříd. (řádek 20).

3 Analýza současné podpory činnosti

Současná elektronická podpora modelářské činnosti spojená s pořádáním soutěží je na velice nízké úrovni. Jelikož se jedná o neziskovou činnost fungující pod záštitou Svazu modelářů České republiky, není dostatek financí na dostatečnou podporu a zajištění alespoň základních prostředků, které jsou v dnešní době pro mnohé organizace samozřejmostí. Bylo proto zapotřebí vytvořit program, který by zjednodušil a zefektivnil pořádání závodů lodních modelářů kategorie NS, v rámci této absolventské práce.

„Dálkově řízené lodní modely tříd sekce NS (F-NS) jsou jízdy a plavby schopné modely, které závodníci během závodu ovládají dálkově, bezdrátově. Jsou to předloze věrné, v měřítku postavené modely lodí a člunů.“ [11]

Nedostatky v současném stavu podpory se projevují od samého začátku. Zástupce klubu musí přihlásit závodníky ze svého klubu na předem známý termín soutěže prostřednictvím vytvořeného vzoru v programu MS Excel, jak je vidět na obrázku č. 3.

Přihlašuji se za klub:												
Soutěž "Seriálu Mí ČR - NS" č.:												
LO -												
Kategorie:				F2-A	F2-B	F2-C	F4-A	F4-B	F4-C	DS	DOPLŇKOVÁ	
č. pol.	Závodník	Věk	Č. licence	Kateg.	Parametry modelu							soutěž
	příjmení	jméno	J.S.		název	měřítka	délka [mm]	šířka [mm]	dok [mm]*	kanál č.	0/1	
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
							délka na KVR [mm]	oplachtění [m2]	výtlak [kg]	kanál č.		
16					NSS							
17					NSS							
18					NSS							
19					NSS							
Poznámky: Přihlášku vyplňte prosím čitelně a odpovědně, nejlépe do šablony Excel ve žlutých polích.												
Pro usnadnění automatizovaného zpracování přihlášky používejte znaky uvedené v záhlaví.												
U každého soutěžícího uveďte, zda je J = junior nebo S = senior a v jaké kategorii startuje!												

Obr. 3 Vzor přihlášky [12]

V takto připravené přihlášce musí vyplnit údaje o jednotlivých závodnících a jejich modelech. Po odeslání tohoto souboru na e-mailovou adresu pořadatele, musí pořadatel zadané informace zkontrolovat a zkopírovat do připravených šablon pro prezenci závodníků a do šablony napomáhající k počítání výsledků opět v programu MS Excel. Zároveň je musí ručně zadat do programu LModel, který disponuje pouze funkcí řazení závodníků dle zadaných vysílacích kanálů tak, aby se navzájem neovlivňovali. Je nutné poznamenat, že tato funkce již v dnešní době téměř nemá význam, protože došlo k velkému technickému pokroku. Dříve se v lodním modelářství používal frekvenční kmitočet 40 MHz se škálou kanálů 1–92, kde po sobě startující závodníci museli mít minimálně o 3 kanály jiný kmitočet, aby se navzájem nerušili. Dnes cca 90% závodníků vlastní vysílač s technologií wifi, která v tomto odvětví zaručuje frekvenční nenarušitelnost. U zbylých 10% je vzhledem k známosti výše uvedeného pravidla téměř nepravděpodobné, aby měli 2 závodníci stejný nebo blízký krystal, a proto není nutností řadit závodníky dle krystalů. Druhou funkcí je vytištění archů pro zapisování výsledků jednotlivých jízd.

Zadávání informací z přihlášky do programu LModel si lze ulehčit tím, že si pořadatel načte informace z posledního závodu, nicméně musí všechny informace překontrolovat, protože je běžné, že se jednotlivé informace průběžně mění. Tyto operace jsou velice časově náročné a snadno při nich může dojít k chybě, která se projeví až v průběhu závodu. Konečným výsledkem, kterého lze dosáhnout pomocí programu LModel, jsou vytištěné archy s důležitými informacemi o závodnících a jejich modelech. Ty slouží pro rozhodčí na startovišti a jsou pro ně nezbytné pro průběh závodů. Zároveň se nachází vedle údajů i předtištěná tabulka k zapisování průběhu všech tří jízd jednoho závodu. Ukázka archu je zobrazena na obrázku č. 4. Takto rozvržený arch vyhovuje současným potřebám a není ho třeba měnit. Proto vytištěný arch z nové aplikace by měl mít podobnou úpravu.

ZAPISOVACÍ FORMULÁŘ - JÍZDY
KATEGORIE: F4-A Senior

START. POŘADÍ	START. ČÍSLO	JMÉNO KLUB MODEL	LICENCE DOK KANÁL	I 6	III 9	II 6	I 6	III 9	IV 6	IV 6	V 9	I 6	VI 6	V 9	I 12	DOK 10	CELK. JÍZDA
	1.	Šmejkal Miroslav KLoM Fregata Bakov n. J. Police	316-004 357 57	-2	-3	-2	-2	-3	-2	-2	-3	-2	-2	-3	-4	-5	
	4.	Hlava Petr KLoM Česilko Regatta	189-001 370 00														
	5.	Bilina Jiří KLoM Česilko Whitney	189-019 386 57														
	19.	Jedlička Jan KLoM Maják Borovany Theia	511-009 390 00														
	20.	Jedlička Lubomír KLoM Maják Borovany Police	511-008 370 00														
	21.	Jedlička Pavel KLoM Maják Borovany Leader	511-011 390 00														
	23.	Vlach Jan KLM "Royal Dux" Duchcov Spider	134-022 380 81														

HL. ROZHODČÍ :

ROZHODČÍ :

2. soutěž "Seriálu MiČR - NS", Běleč nad Orlicí

Strana 1/3

Obr. 4 Zapisovací arch z programu LModel

Po vyplnění vytištěných archů musí pořadatel informace zapsat do výše zmiňované připravené šablony v programu MS Excel a ručně vytvořit konečné pořadí závodníků pomocí několikanásobného automatického řazení. K tomu musí znát pravidla a priority u jednotlivých kategorií, a proto tuto činnost mohou provádět jen osoby znalé problematiky. Výsledky si na závěr musí sám zkontrolovat, protože i po automatickém řazení jsou občas závodníci seřazeni špatně. Při zjištění chyby musí pořadatel chybu ručně opravit a následně může výsledkovou listinu vytisknout.

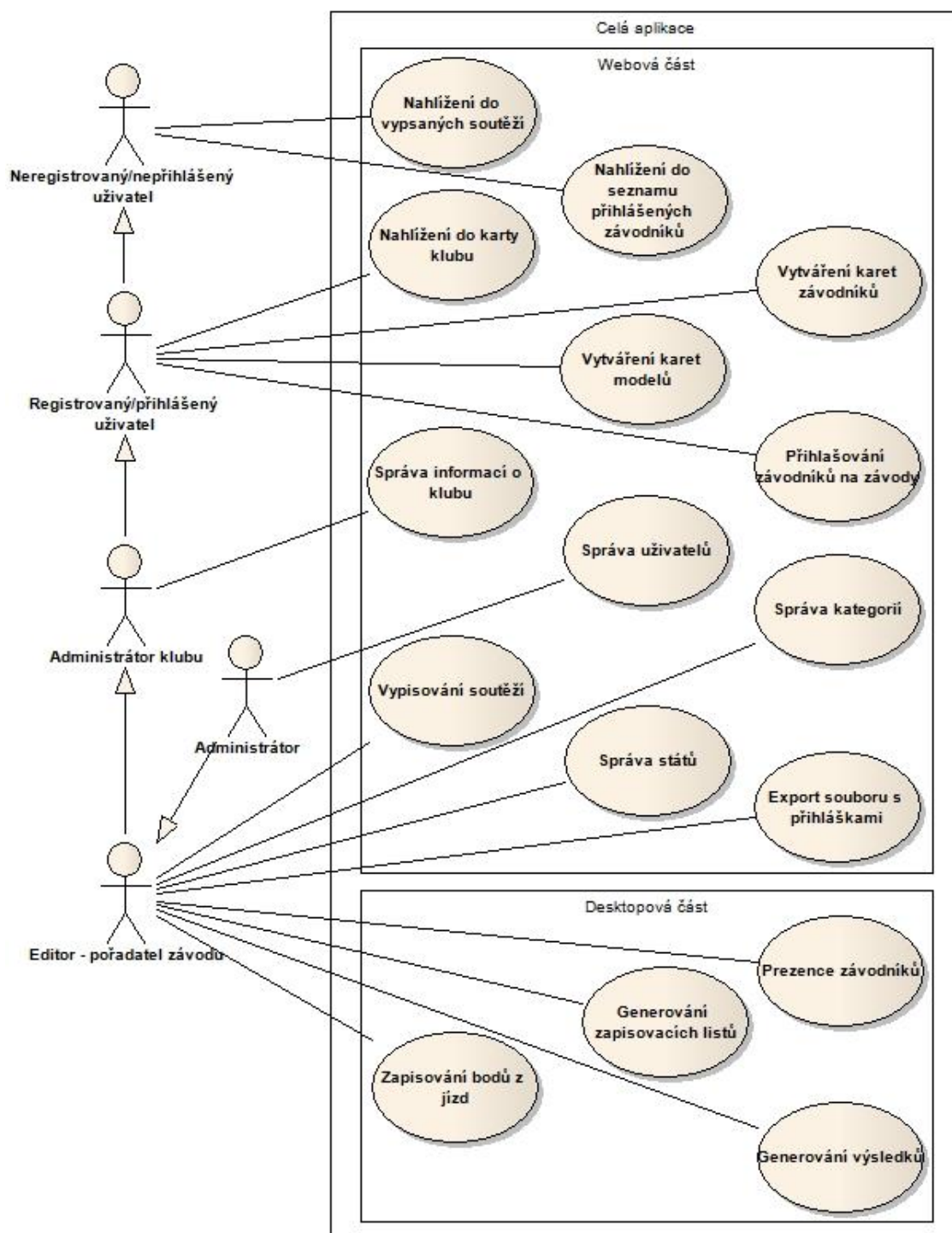
Z výše uvedeného popisu současného průběhu závodů je zřejmá vysoká závislost předávání a vyplňování informací na člověku. Proto je i běžné, že v průběhu závodů nastanou chyby, které vznikly z důvodu časové náročnosti na pověřenou osobu. Tyto chyby se dodatečně musí opravit a zbytečně znepríjemňují jinak zábavnou činnost. Z pohledu osoby znalé problematiky je proto důležité snížit závislost systému na člověku a zautomatizovat většinu procesů, které automatizovat lze.

4 Návrh aplikace

Na základě znalostí získaných při pořádání závodů, účastí a z konzultací s pořadatelem vyplynulo, že bude vhodné navrhnout aplikaci rozdělenou na dvě části v podobě webového rozhraní a desktopové aplikace. Je nutné podotknout, že v dnešní době stále není na všech místech, kde se pořádají závody, možnost připojení k internetu a je potřebné tuto skutečnost respektovat. Proto část aplikace, která bude využívána přímo na místě konání, je navržena v desktopové podobě. Webová část by sloužila pro vypisování termínů závodů, přihlašování závodníků a editaci údajů o závodnících a jejich modelech. Také bude umožňovat náhled na aktuální seznamy přihlášených závodníků do jednotlivých kategorií bez nutnosti přihlášení. Desktopová část by obstarávala prezenci závodníků na základě přihlášení, sestavování zapisovacích archů a počítání výsledků. Chování celé navrhované aplikace z pohledu uživatele je znázorněné na následujícím obrázku č. 5 na další straně pomocí Use case diagramu – diagramu případů užití.

Na diagramu je znázorněna dědičnost práv pomocí vazby generalizace jednotlivých uživatelských vrstev. Z tohoto vztahu tak vyplývá, že například uživatel v roli administrátora klubu může provádět vše, co může jeho předek – registrovaný/přihlášený uživatel a navíc tyto možnosti jsou rozšířeny o správu informací o klubu atd. Uživatel v nejnižší vrstvě – **neregistrovaný/nepřihlášený** uživatel může nahlížet pouze do veřejných informací. **Registrovaný/přihlášený** uživatel má tyto možnosti rozšířené o všeobecnou správu závodníků a modelů včetně přihlašování na závody. **Administrátor klubu** může navíc spravovat informace o daném klubu. Vrstva **editor** je myšlena jako vrstva, ve které budou pracovat pořadatelé závodů. Mohou proto tak spravovat veškeré informace spojené s daným závodem. Nejvyšší vrstva **administrátor** je obohacena o správu uživatelů ve webové části.

Podrobnější popis práv a možností užití je rozepsán v kapitolách 5.1.3 Popis aplikace webové části a v kapitole 5.2.3 Popis aplikace desktopové části.



Obr. 5 Chování aplikace z pohledu uživatele

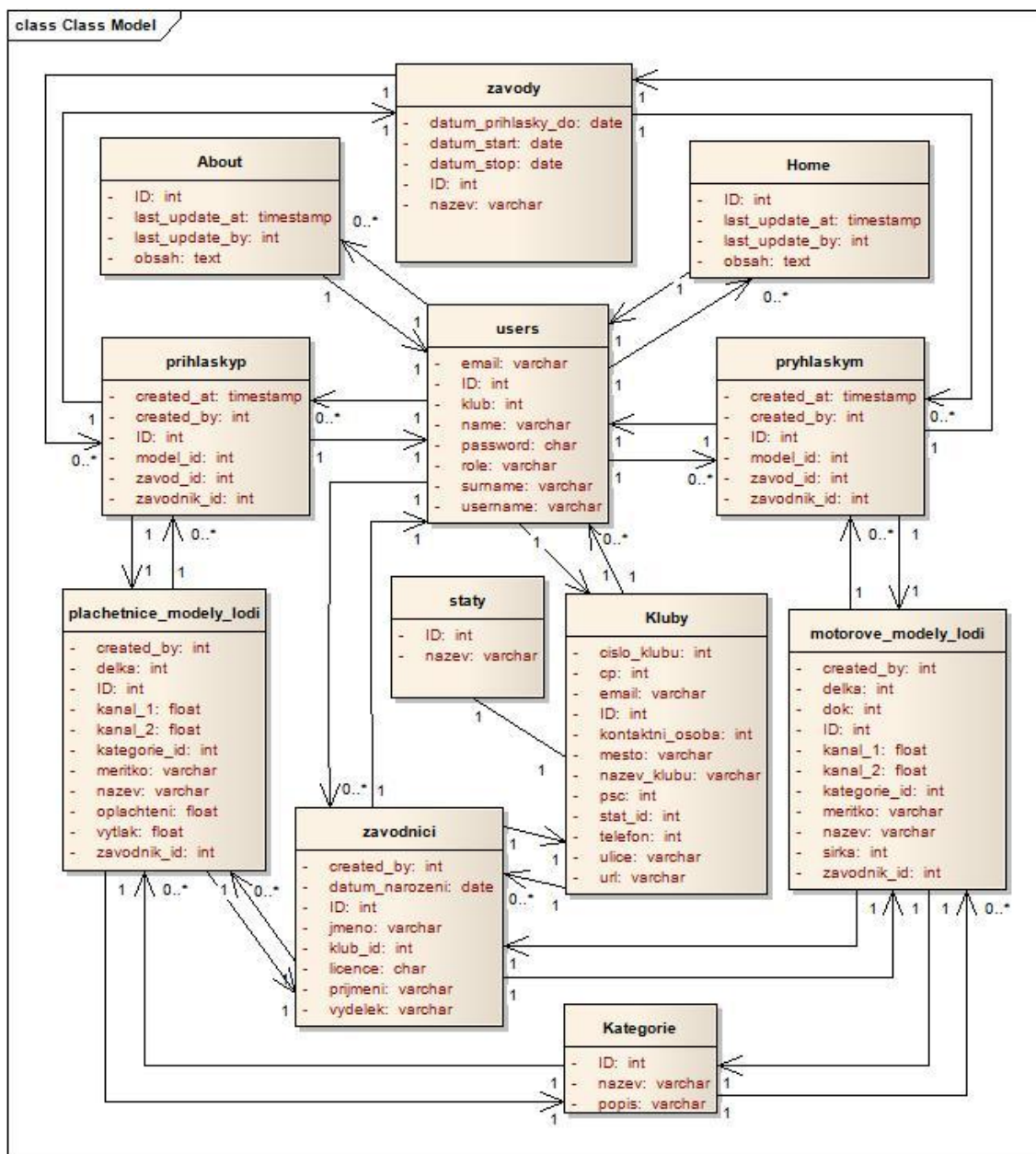
4.1 Návrh webové části

Webové rozhraní bude sloužit pro vytváření karet závodů a přihlašování závodníků na jejich termíny. Přístup do aplikace je volný a může se do ní přihlásit kdokoli, kdo se do systému registruje. K registraci je však zapotřebí mimo jiné

vybrat klub, kterého je uživatel členem. Bude mu tak umožněno vytvářet karty závodníků pouze daného klubu.

Každý klub má přiřazené své registrační číslo, které je sestaveno ze tří číslic a je součástí licence závodníka. Licenční číslo je složeno ze sedmi znaků, z nichž první tři číslice je registrační číslo klubu následováno pomlčkou a za ní pokračuje opět tříciferné číslo, které je pro každého závodníka jedinečné. Číslo je závislé na pořadí registrace v klubu. Například klub Třebechovice pod Orebem má registrační číslo 403 a v pořadí 5. registrovaný člen klubu má licenční číslo 403-005.

Každý uživatel bude mít v systému k dispozici vytváření karet závodníků a jejich modelů. Karty budou uloženy v systému a uživatel bude mít možnost editovat karty závodníků i jejich modelů. Při přihlášení závodníka včetně jeho modelů na daný termín tak budou k dispozici aktuální údaje a díky tomu je pro další použití nikdo nemusí kontrolovat. Při vypršení termínu pro přihlášení na soutěž bude zablokována možnost přihlášení/odhlášení a data z webové části budou připravena na importování do desktopové části, která s nimi bude dále pracovat. Na obrázku č. 6 je class model, vytvořený v programu Enterprise Architect 7.1. Jsou na něm znázorněny třídy včetně svých atributů a vztahy mezi třídami včetně multiplicity.



Obr. 6 Class model využitý ve webové části

4.2 Návrh desktopové části

Přístup do desktopové části bude volný, bez přihlášení. Tato část bude zpracovávat importovaná data z webové části.

K vytvoření karty závodu v desktopové části je zapotřebí nejprve importovat data z webového rozhraní. Dále pak v první fázi závodu bude možnost prezence závodníků, při které bude k dispozici i případná kontrola a editace údajů. Po uzavření prezence budou automaticky generovány seznamy startujících

závodníků. V nich bude pořadí jednotlivých závodníků sestaveno na základě seznamů prezentovaných závodníků v jednotlivých kategoriích a vyplněných vysílacích kanálů tak, aby 2 po sobě jdoucí závodníci se neovlivňovali. Vzhledem k dnešním trendům, popsaných v kapitole o současné podpoře, stačí vytvořit jednoduché pravidlo, aby 2 závodníci bez wifi vysílače nestartovali bezprostředně po sobě. Po vygenerování seznamů budou vytištěny zapisovací archy pro závody. Jelikož stávající forma zapisovacích archů vyhovuje současným potřebám, zapisovací archy z navrhované aplikace budou mít podobnou úpravu.

Druhá fáze aplikace bude obnášet zapisování výsledků do systému ze zapisovacích archů, na jejímž základě budou generovány celkové výsledky. Po vyplnění všech jízd u všech závodníků bude závod uzavřen a systém seřadí závodníky podle jejich výkonů. Po seřazení budou vytištěny výsledkové listiny.

5 Implementace aplikace

Tato kapitola se zabývá vývojem aplikace pro podporu pořádání závodů v modelářské činnosti. Skládá se z webové a desktopové části. Webová část je vytvořena v jazyce PHP ve spojení s Nette Frameworkem. Desktopová část je pak založena na jazyku Java.

Prvním krokem, kterým by měl každý vývojář aplikace začít, je seznámení se s inovacemi ve vybrané technologii, aby ji mohl co nejlépe využít. Ušetří si tak spoustu práce a zvýší se předpoklady ke kvalitě výsledné aplikace. Pokud je technologie pro osobu vyvíjející aplikaci nová, tím spíše je důležité věnovat dostatek času seznámení a pochopení principů, na kterých je postavena.

5.1 Implementace webové části

5.1.1 Příprava

Vzhledem k tomu, že dosud nebyla s Nette Frameworkem žádná zkušenost, bylo zapotřebí se s ním nejprve seznámit. Naštěstí tvůrci Frameworku vycházejí začátečníkům velice vstřícně a připravili pro ně úvodní kurz Seznámení s Nette Frameworkem, v rámci kterého představili základní funkce a principy Frameworku. Ukázky jeho chování a funkce ladících nástrojů pak demonstrují na vznikající aplikaci. Ta je založena na takzvaném Sandboxu, což je stažený balík Nette Frameworku včetně adresářové struktury. Na konci kurzu tak vznikne aplikace, na jejíchž základech může pak nový uživatel Nette Frameworku začít vyvíjet vlastní aplikace.

5.1.2 Popis struktury

Nette Framework je postaven na návrhovém vzoru Model-View-Presenter (MVP). Tato softwarová architektura vznikla z potřeby oddělit u aplikací s grafickým rozhraním kód obsluhy (presenter) od kódu aplikační logiky (model) a zároveň oddělit kód zobrazující data (view). Výsledkem tohoto rozdělení je jednak zpřehlednění aplikace, ale zároveň se umožnilo i testování jednotlivých částí zvlášť [13].

Model

Model je datový a zejména funkční základ celé aplikace. Je v něm obsažena aplikační logika. Jakákoliv akce uživatele (přihlášení, vložení zboží do košíku, změna hodnoty v databázi) představuje akci modelu. Model si spravuje svůj vnitřní stav a ven nabízí pevně dané rozhraní. Voláním funkcí tohoto rozhraní můžeme zjišťovat či měnit jeho stav. Důležité je to, že model o existenci view nebo kontroleru neví [13].

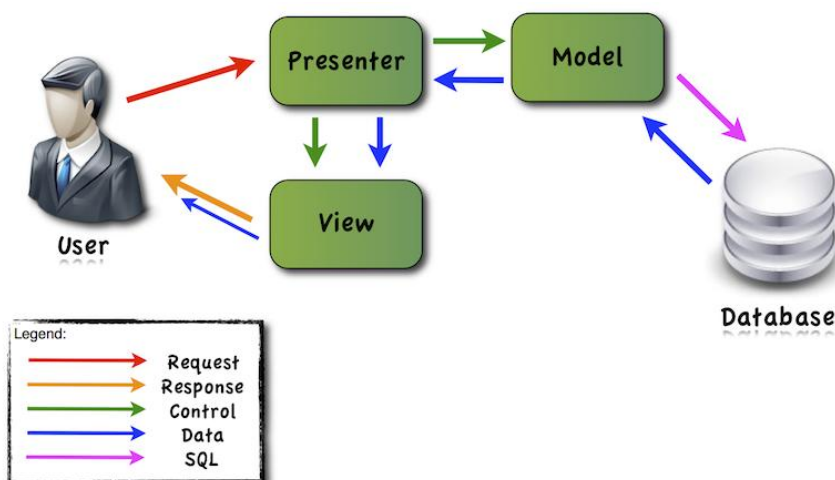
View

Tato vrstva aplikace má na starost zobrazení výsledku požadavku. Obvykle používá šablonovací systém. Nette Framework je přímo založen na šablonovacím systému a tvůrci jej připravili v Sandboxu. Využívá jej tedy i tato aplikace. Vrstva view ví, jak se má zobrazit daná komponenta nebo výsledek získaný z modelu [13].

Presenter

Presenter vychází z kontroleru, což představuje řadič, který zpracovává požadavky od uživatele a na jejich základě pak volá pověřenou aplikační logiku (model). Poté požádá vrstvu view o vykreslení dat [13]. Presenter oproti kontroleru pracuje mnohem více s vrstvou view. Mimo provázání vrstev model a view také může přímo zasahovat a měnit stav vrstvy view[14] .

Interakce všech tří částí spolu s uživatelem a databází je zobrazena na obrázku č. 7.



Obr. 7 Interakce mezi MVP společně s uživatelem a databází [15]

Adresářová struktura

Níže uvedená adresářová struktura je uspořádána tak, aby usnadnila orientaci v aplikaci.

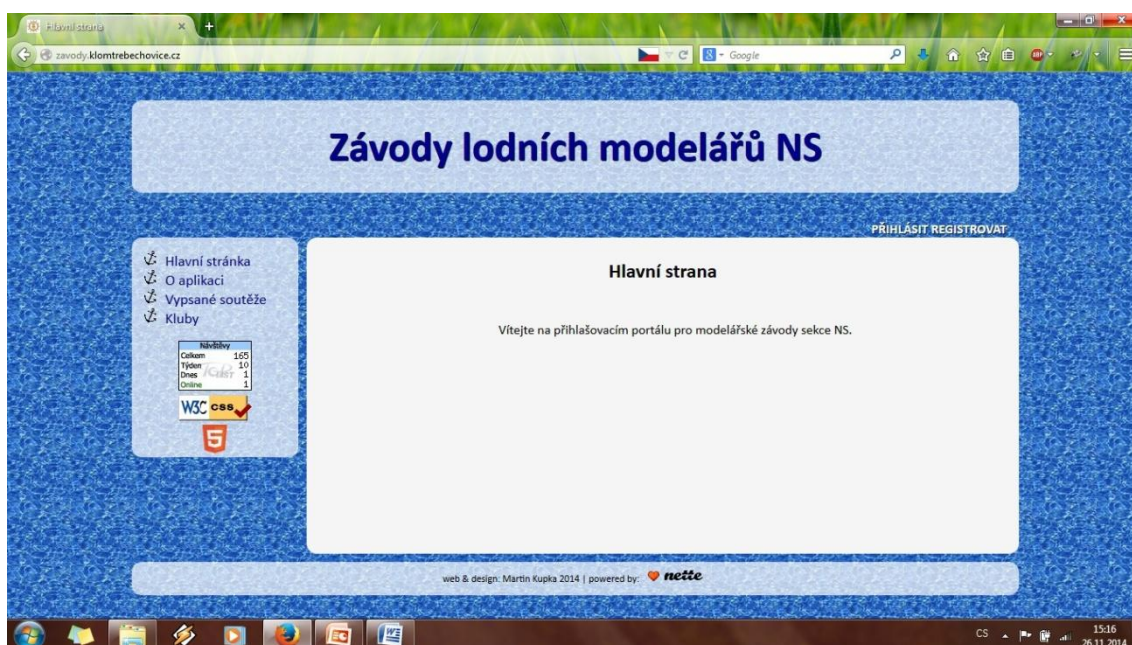
/	← kořenový adresář webu
- app/	← adresář aplikace
- config/	← konfigurační soubory
- Config.neon	← konfigurační soubor
- config.local.neon	
- model/	← modelová vrstva s třídami
- presenters/	← třídy presenterů
- router/	← konfigurace URL adres
- templates/	← šablony
- bootstrap.php	← spouštěcí soubor
- vendor	← knihovny pro aplikaci
- nette/	← adresář Frameworku
- others/	← ostatní knihovny
- log	← chybové logy
- temp	← dočasné soubory
- tests	← adresář pro unit testy
- www	← kořenový adresář

Nejdůležitější částí adresářové struktury je složka `app`, kterou uživatel Nette Frameworku nejvíce používá a vytváří zde soubory své aplikace. Ta se skládá z adresáře `config`, obsahující konfigurační soubory s nastavením připojení k databázi, nastavením přístupových práv aj. Druhou podsložkou je složka `model`, reprezentující jednu ze tří částí architektury Model-View-Presenter. Presenter je adresář obsahující soubory jednotlivých vrstev aplikace. Složka `router` obsahuje soubor sloužící ke konfiguraci URL adres. Adresář `templates` je poslední ze tří částí návrhového vzoru (view), která obsahuje složky analogicky propojené s presentery. Máme-li vytvořený presenter (vrstvu) `Zavodnik`, složka `templates` tak musí obsahovat podsložku `Zavodnik`. V té pak budou vytvořeny jednotlivé stránky této vrstvy. Může to být například formulář pro vytvoření závodníka, stránku s výpisem všech závodníků, detail závodníka aj. Posledním souborem ve složce `app` je `bootstrap.php`, který načítá celý Framework a nastavuje aplikaci. Rovněž slouží i k registraci metod externích knihoven.

Druhou částí Sandboxu je Adresář vendor. Ten obsahuje samotný Nette Framework a je zde připravený adresář i pro další externí knihovny, jako je například DatePicker, textový editor Texyla apod.

5.1.3 Popis aplikace

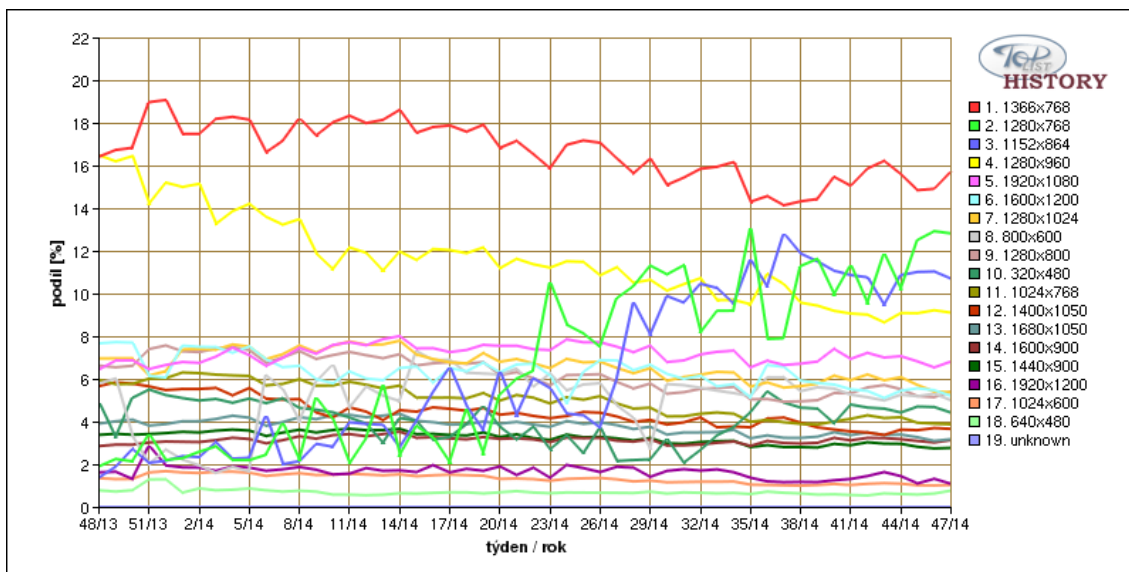
Všeobecně by se dalo říci, že webové aplikace se skládají ze dvou částí a to formy a obsahu. Formou se rozumí vizuální podoba stránek obsahující zejména rozvržení, grafiku, formátování, fonty a barvy písem. Obsah je poté tvořen samotnými daty. Důležitou částí jsou pak práva uživatelů resp. uživatelské vrstvy, které zasahují do obou částí, jak formy, tak obsahu.



Obr. 8 Layout webové části

Forma aplikace

Jak je vidět na obrázku č. 8, webová část je založena na dvousloupcovém layoutu. Rozvržení stránek je responzivní a jejich šířka se tak přizpůsobuje zobrazovacím jednotkám uživatelů v rozmezí od 1200 px do 800 px. Zvolený interval by k dnešním potřebám měl být dostačující. Důkazem je i vygenerovaná historie přístupů uživatelů na webových stránkách zaregistrovaných na serveru TOPlistu, který zprostředkovává audit návštěvnosti webových stránek na další straně.



Obr. 9 Historie přístupů uživatelů podle rozlišení monitoru [16]

Z obrázku číslo 9 je patrné, že za poslední rok mají největší podíl návštěvnosti stránek uživatelé používající zařízení s rozlišením 1366x768px. Jak je zobrazena webová část právě v tomto rozlišení, je znázorněno na obrázku číslo 8 na předchozí straně. Jelikož zkušenosti jsou takové, že monitory s větším rozlišením se spíše používají pro zobrazení více programů vedle sebe z důvodu dynamičtější práce, poskytovat webovou aplikaci ve větším rozlišení by bylo zatím zbytečné. Při zohlednění této skutečnosti se dá z obrázku číslo 9 také vyčíst, že první rozlišení, pro které by zvolené minimum responzivní šířky nebylo úplně vhodné, je rozlišení 320x480 px. To v současné době zaujímá 10. pozici s necelými 5% podílu návštěvnosti.

Další vlastností, které se v rámci formy webové části bylo třeba zabývat, je barevnost a celkové sladění. Všechny bloky, které rozdělují aplikaci na části s jasným obsahem, mají zaoblené rohy. Ty jsou tvořeny vlastností border-radius v CSS3. Vzhledem k využití stránek pro lodní modeláře byla navržena celá aplikaci do odstínů modré barvy a pro pozadí zvolil motiv vody. Bloky hlavička, menu a patička mají bílou barvu s průhledností 0.7. Částečně jimi tak prosvítá voda z pozadí a celkově tak zapadají do konceptu. Obsahová část je zabarvena odstínem „whitesmoke“, aby celek nenarušovala zářící čistě bílá barva. V celé aplikaci je použito písmo pouze černé a tmavě modré barvy. Aktivní odkazy se mění v kontrastující červenou.

Veškeré formátování a určení vzhledu probíhá v externích CSS souborech kaskádových stylů verze 3 tak, aby bylo dodržené oddělení formátové od významové části kódu. Všechny prvky jsou formátovány způsobem, který zaručuje vykreslení ve všech prohlížečích stejné nebo velice podobné. Tím zmizela nutnost vytvářet více souborů s různými formáty pro různé prohlížeče. Za zmínku také stojí formátování pro tisk, ve kterém je vypuštěné pozadí a blok menu. Formátování pro tisk je demonstrováno na výpisu informací o klubu Třebechovice pod Orebem v obrázku číslo 10.

The screenshot shows a web page titled 'Závody lodních modelářů NS'. It includes a header with the page title and a URL. The main content area is divided into two sections: 'Informace o klubu' and 'Závodníci'. The 'Informace o klubu' section contains contact details for the club. The 'Závodníci' section contains a table with competitor information. The page is formatted for print, with a clean layout and no background images or menus.

Informace o klubu

<http://zavody.klomtřebchovice.cz/kluby-page/show?klubid=1>

Závody lodních modelářů NS

Informace o klubu

[← zpět na výpis klubů](#)

Registrační číslo: 403
Název klubu: KLoM Třebechovice pod Orebem
Kontaktní osoba: Pavel Darakev
Telefon: 604855257
Web: <http://www.klomtřebchovice.cz>
E-mail klubu: klomtpo@seznam.cz
Stát: CZ
Město: Třebechovice pod Orebem
Ulice: Moravská
ČP: 94
PSČ: 50346

Závodníci

Počet	Jméno	Příjmení	Licence
1	Pavel	Darakev	403-001
2	Pavel	Darakev ml.	403-004
3	Martin	Kupka	403-005
4	Jakub	Janko	403-008
5	Zdeněk	Janko	403-009
6	Markéta	Provazníková	403-010
7	Jiří	Špínar	403-002

[← zpět na výpis klubů](#)

web & design: Martin Kupka 2014 | powered by: nette

1 z 1

26.11.2014 16:25

Obr. 10 Ukázka formátování pro tisk

Obsah aplikace

Obsahová část má především informační charakter a slouží jako předání informací mezi závodníky a pořadateli z obou pohledů.

Z pohledu pořadatele tato data tvoří především přihlášky s informacemi od závodníků. Ta jsou tvořena údaji o závodnících, jejich modelech a klubu, za který startují. Výše uvedené informace jsou generovány z karet klubů, závodníků a modelů, které vytvoří daný uživatel. Po vyplnění těchto karet se uživateli nabídnou tato data při vytváření přihlášky a on jednoduše prostřednictvím zaškrtačacího políčka vybere konkrétní závodníky a modely, které na daný termín chce přihlásit. Výsledkem, resp. konečným bodem, pro pořadatele je exportovaný soubor s přihláškami ve formátu csv, se kterým dále pracuje desktopová aplikace.

„CSV (comma-separated values neboli čárkou oddělené hodnoty) je soubor, který obsahuje tabulková data v textové podobě. Jednotlivé buňky tabulky jsou od sebe odděleny čárkou. Řádky tabulky jsou odděleny enterem (speciálním znakem CRLF). Hodnoty mohou být uzavřeny v uvozovkách, což dovoluje, aby textové hodnoty obsahovaly čárky. Pokud má textová hodnota obsahovat uvozovku, tak se uvozovka zdvojuje.

Struktura CSV souboru je podrobně popsána v RFC 4180. Avšak jednotná specifikace pro tento formát neexistuje.“ [17]

Z pohledu závodníka se jedná o veškeré informace o závodech poskytnuté od pořadatelů a zároveň možnost nahlédnutí do stavu přihlášených na závody. Níže je zobrazen obsah souboru csv, vygenerovaný z testovací soutěže.

2. testovací soutěž

Export v 2014-10-07 13:59:08

Export pro MKupka

```
1,F4C-S,Jiša,Petr,409-001,ANO,"KLoM Plzeň Letkov",Snowberry,343,84,0
2,F2B-S,Jiša,Stanislav,409-002,ANO,"KLoM Plzeň Letkov","K. D.
Perkasa",455,2.4,0
3,F2A-S,Grňa,Ivan,135-012,NE,"KLoM Kroměříž","Stražak 14",440,2.4,0
4,F2A-S,Šesták,Miloslav,135-007,NE,"KLoM Kroměříž",ZEEAREND,348,9,0
5,F4C-S,Darakev,Pavel,403-001,ANO,"KLoM Třebechovice pod
Orebem",Snowberry,443,56,85
6,F4A-S,"Darakev ml.",Pavel,403-004,NE,"KLoM Třebechovice pod
Orebem",Cyclop,370,81,58
7,F4A-S,Kupka,Martin,403-005,NE,"KLoM Třebechovice pod Orebem",Danča,335,84,0
8,F4A-S,Janko,Jakub,403-008,ANO,"KLoM Třebechovice pod Orebem",GMH,350,81,0
9,F4B-S,Janko,Jakub,403-008,ANO,"KLoM Třebechovice pod Orebem","PT
109",467,4,0
10,F4A-S,Janko,Zdeněk,403-009,ANO,"KLoM Třebechovice pod
Orebem",Classic,378,2.4,0
11,F4A-S,Provazníková,Markéta,403-010,NE,"KLoM Třebechovice pod
```

Orebem", "Danča II", 335, 84, 0
 12, F4A-S, Darakev, Pavel, 403-001, ANO, "KLoM Třebechovice pod
 Orebem", Saucy, 443, 81, 56
 13, F4B-S, Darakev, Pavel, 403-001, ANO, "KLoM Třebechovice pod Orebem", "DGzRS
 Hermann Marwede SAR", 338, 56, 0
 14, FDS, Špinar, Jiří, 403-002, NE, "KLoM Třebechovice pod
 Orebem", D.S.Schaarhorn, 565, 85, 0
 &
 1, NSS-A, Janoš, Milan, 535-002, ANO, "KLoM Brandýs nad
 Labem", Gata, 1:11, 895, 0.487, 10.4, 2.4, 0
 2, NSS-C, Chmelka, František, 535-002, NE, "MK Slezsko Český Těšín", "No
 Name", 1:7, 960, 0.52, 9.5, 2.4, 0
 3, NSS-A, Špinar, Jiří, 535-002, NE, "KLoM Třebechovice pod
 Orebem", Endeavour, , 920, 0.71, 8.8, 2.4,
 4, NSS-B, Špinar, Jiří, 535-002, NE, "KLoM Třebechovice pod
 Orebem", America, , 795, 0.38, 4.74, 2.4,

Uživatelské vrstvy

Důležitou součástí aplikace jsou vrstvy uživatelů, kteří s aplikací pracují. Vrstvy zasahují jak do formy, tak do obsahu aplikace. Z pohledu formy se jedná především o zviditelnění/zmizení odkazů na jednotlivé moduly a operace včetně rozdílných náhledů na data. Z obsahové formy to jsou pak samotná data, resp. rozsáhlost zveřejněných dat a operací s nimi, která roste úměrně s výší uživatelské vrstvy.

Tato aplikace je založena na pěti uživatelských vrstvách. Nejnižší z nich je **neregistrovaný/nepřihlášený uživatel**, který má k dispozici pouze informace o aplikaci, vypsaných soutěžích a informacích o nich, včetně seznamu přihlášených závodníků v jednotlivých kategoriích. Dále to je pak adresář klubů, ve kterém nalezne kromě informací o klubu i seznam členů.

Druhou vrstvou je **registrovaný/přihlášený uživatel**. Jeho práva jsou rozšířena o sekci Můj klub, kde kromě náhledu na informace o svém klubu má uživatel přístupné možnosti vytvořit kartu závodníka případně i kartu jeho model včetně editace a mazání těchto údajů. Zároveň jsou pak přístupné sekce přihlášení závodníků s motorovými modely i plachetnicemi na závody. Obě tyto možnosti jsou přístupné, pokud má uživatel koho přihlásit. Pokud uživatel bude mít pouze závodníky s motorovými modely, nebude mít možnost přihlásit závodníky s plachetnicemi, protože tato funkce nebude dostupná. To platí i v opačném případě. Vzhledem k tomu, že v rámci jednoho klubu může být více uživatelů, každý si může editovat, mazat a přihlašovat/odhlašovat pouze závodníky, které si sám vytvořil.

V pořadí již třetí vrstvou je **administrátor klubu**. Kromě práv registrovaného uživatele může editovat informace o klubu. V rámci jednoho klubu může být pouze jeden administrátor.

Druhou nejvyšší uživatelskou vrstvou je vrstva **editor**. Práva editora rozšiřují práva registrovaného uživatele o možnost editace závodů. Tato vrstva je vytvořena pro pořadatele závodů. Může zasahovat do údajů o závodnících a modelech, může editovat závodní kategorie popřípadě přidat stát. Zároveň upravuje informace o závodu a má možnost exportování přihlášek na závody v souboru csv, s kterým dále pracuje desktopová část. Práva editora jsou dočasná a po uzavření soutěže včetně zveřejnění výsledků mu jsou práva odebrána. Vzhledem k tomu, že tato doba může být proměnlivá a musí být potvrzena písemně, odebrání práv nemůže být automatické a může je odebrat pouze administrátor.

Poslední vrstvou s nejvyššími právy je vrstva **administrátor**. Kromě všech výše zmíněných práv má k dispozici modul uživatelů, kde může vytvářet, editovat a rušit uživatele, včetně editace práv. Tato práva může mít z bezpečnostních důvodů pouze jeden uživatel v celé aplikaci.

5.2 Implementace desktopové části

5.2.1 Příprava

Vzhledem k tomu, že s jazykem Java se pracovalo průběžně během celého studia, příprava probíhala odlišným způsobem oproti přípravě v rámci implementace webové části. Znalosti Javy však bylo nutné hodně oživit. Nápomocné byly zejména projekty z předmětů Programování 1 i 2 a dostupné přednášky s poznámkami. Zároveň byly nápomocné projekty z předmětů Počítačová grafika 1 a 2.

Po oživení znalostí bylo zapotřebí v nově založeném projektu připravit nejvhodnější strukturu, aby byla zachována přehlednost i v této části, oddělení jednotlivých částí, znovupoužitelnost a zejména aby výsledná aplikace byla objektová nejen po formální stránce. Oproti webové části, kde v rámci Frameworku již byla tato struktura připravena a stačilo ji respektovat, bylo nutné ji v této části vytvořit.

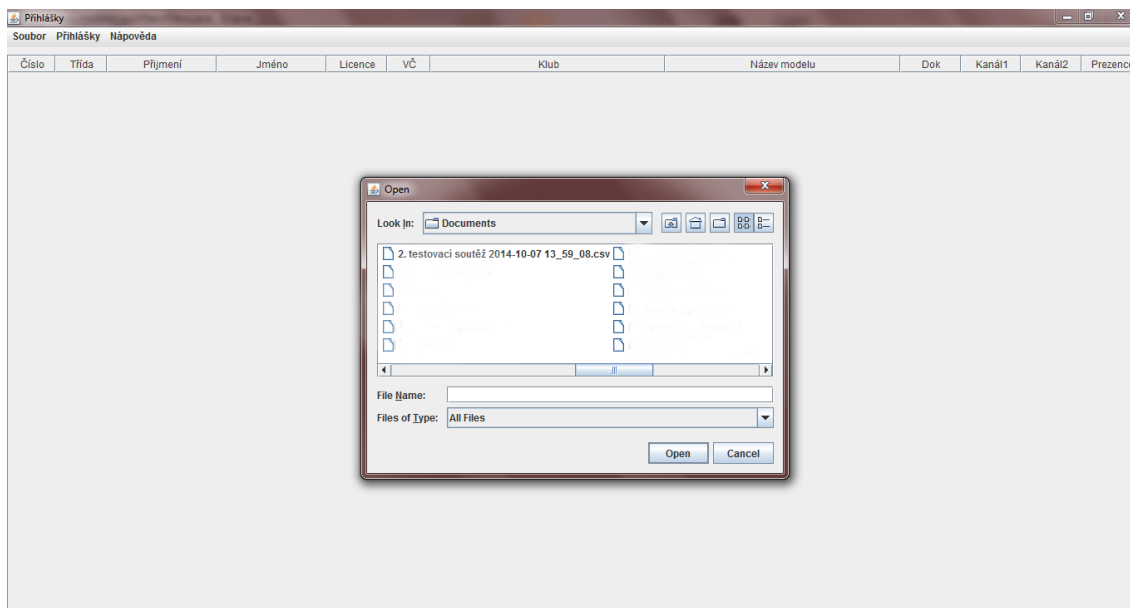
5.2.2 Popis struktury

Jelikož ve webové části byl použit návrhový vzor Model-View-Presenter, kde základní znalosti tohoto návrhového vzoru byly rozšířeny, bylo rozhodnuto podobnou strukturu využít i v desktopové části. Byl proto vybrán velice podobný návrhový vzor Model-View-Controller, dále MVC, kde Controller je obdobou Presenteru v Nette Frameworku s jistými odlišnostmi popsány v kapitole 5.1.3 Popis aplikace webové části. Je to velice populární návrhový vzor, avšak má svá úskalí a jeho popis se dá vyložit více způsoby.

„Architektura MVC má jednu zajímavou vlastnost: základní ideje člověk pochopí během pár minut, ale plně proniknout do všech detailů může trvat měsíce i roky. Jak se stává MVC populárnějším a populárnějším, na webu i v knihách se objevují jeho popisy, které se však podle mé zkušenosti spíše zaměřují na onu několikaminutovou část. Ucelených úvodů do MVC je málo a důsledkem je, že o MVC mají různí vývojáři různé představy a často narazíte na „MVC aplikaci“, která vlastně nemá s tímto vzorem vůbec nic společného.“[18]

5.2.3 Popis aplikace

Jak bylo již zmíněno, desktopová aplikace navazuje na webovou část. Prvním krokem, který je třeba provést, je načtení souboru s příponou .csv, obsahující přihlášky. K tomu slouží klasické dialogové okno pro hledání v počítači. To je implementováno pomocí komponenty JFileChooser a jeho vzhled je zobrazen níže na obrázku č. 11 na další stránce.



Obr. 11 Dialogové okno pro hledání souboru v počítači

Po načtení souboru se zobrazí importovaná data v tabulce, kde se s nimi bude dále pracovat. Načítání jednotlivých dat je prováděno pomocí třídy `BufferedReader`, která pouze načítá ze zdrojového souboru text znak po znaku a ukládá jej do mezipaměti. Tomuto procesu se také může říkat streamování. Vzhledem k danému charakteru vstupního souboru se následně pomocí tokenizerů rozdělí text do jednotlivých řádků a sloupců. Výsledné zobrazení zpracovaných přihlášek ze souboru je znázorněno na následujícím obrázku č. 12.

2. testovací soutěž											
Číslo	Třída	Přijmení	Jméno	Licence	VČ	Klub	Název modelu	Dok	Kanál1	Kanál2	Prezence
1	F4C-S	Jiša	Petr	409-001	ANO	KLoM Píseň Letkov	Snowberry	343	84	0	false
2	F2B-S	Jiša	Stanislav	409-002	ANO	KLoM Píseň Letkov	K.D. Perikasa	455	2.4	0	false
3	F2A-S	Grňa	Ivan	135-012	NE	KLoM Kroměříž	Stražák 14	440	2.4	0	false
4	F2A-S	Šesták	Miloslav	135-007	NE	KLoM Kroměříž	ZEEAREND	348	9	0	false
5	F4C-S	Darakev	Pavel	403-001	ANO	KLoM Třebouchovice pod Orebem	Snowberry	443	56	85	false
6	F4A-S	Darakev ml.	Pavel	403-004	ANO	KLoM Třebouchovice pod Orebem	Cyclop	370	81	58	false
7	F4A-S	Kupka	Martin	403-005	NE	KLoM Třebouchovice pod Orebem	Danča	335	84	0	false
8	F4A-S	Janko	Jakub	403-008	ANO	KLoM Třebouchovice pod Orebem	GMH	350	81	0	false
9	F4B-S	Janko	Jakub	403-008	ANO	KLoM Třebouchovice pod Orebem	PT 109	467	4	0	false
10	F4A-S	Janko	Zdeněk	403-009	ANO	KLoM Třebouchovice pod Orebem	Classic	378	2.4	0	false
11	F4A-S	Provažníková	Markéta	403-010	NE	KLoM Třebouchovice pod Orebem	Danča II	335	84	0	false
12	F4A-S	Darakev	Pavel	403-001	ANO	KLoM Třebouchovice pod Orebem	Saucy	443	81	56	false
13	F4B-S	Darakev	Pavel	403-001	ANO	KLoM Třebouchovice pod Orebem	DG&RS Hermann Marwede SAR	338	56	0	false
14	FDS	Špinar	Jiří	403-002	NE	KLoM Třebouchovice pod Orebem	D.S.Schaarhorn	565	85	0	false

Obr. 12 Zobrazení načtených dat ze souboru

Po uzavření prezenze uživatel může vyvolat generování zapisovacích listů, jejichž rozvržení je velice podobné tomu z programu LModel. To je již ukázáno na obrázku č. 4.

[illegible]

Obr. 13 Tabulka pro zapisování jednotlivých jízd

Pomocí výše uvedené tabulky se podle současných pravidel dopočítávají body pro konečné umístění, ale také pro umístění v celém seriálu Mistrovství České republiky. V současné době je princip počítání výsledků takový, že ze tří jízd se vyřadí jízda s nejmenším bodovým ziskem (sloupec Vyřazená jízda). Ze zbylých dvou jízd se vypočítá průměr, podle kterého je stanoveno konečné pořadí (sloupec Celkem) [11].

Po zapsání všech jízd u všech závodníků je stanoveno konečné pořadí, pro které ve většině případů nestačí jen sloupec Celkem. Vzhledem k tomu, že je běžná shoda bodového zisku v tomto sloupci, musí se přihlídnout k dalšímu parametru. Tím je právě vyřazená jízda, kde rozhoduje opět bodový zisk. V dnešní době však je schopnost závodníků získat maximální bodový zisk tj. 3x 100b. na vysoké úrovni. Z toho důvodu není neobvyklé, že i přes vyřazení jízdy není na prvních příčkách rozhodnuto, a proto je nutné konečné pořadí stanovit pomocí rozjížd'ky [11].

Výše popsaný proces určení konečného pořadí je pro všechny třídy stejný, avšak v některých třídách jsou k určení pořadí započítávána i další hodnotící kritéria. Jedná se zejména o stavební zkoušku, která ovlivňuje u většiny tříd zhruba z 50 % konečné pořadí. Stavební zkouška se vykonává ve všech třídách kromě třídy F4-A, kde jediným hodnotícím kritériem jsou jízdy.

Stavební zkouška je hodnocena třemi rozhodčími, z nichž každý může udělit v hodnocení maximálně 100b. *„Každý člen hodnotící komise hodnotí každý model a zaznamenává si jím dané body v hodnotícím listu v souladu s hodnotícími kritérii (udělují se pouze celé body). Součet bodů udělených podle jednotlivých kritérií tvoří výsledek hodnocení.“*[11] Celkové hodnocení stavby je pak tvořeno průměrem těchto tří výsledků hodnocení od jednotlivých rozhodčí. Celkový výsledek je pak součtem celkového hodnocení stavby a celkového bodového zisku z jízd, vypočítaného podle výše uvedeného postupu.

Výjimku tvoří třída F-DS, kde se kromě stavební zkoušky hodnotí i parní stroj. Kromě klasické stavební zkoušky, kde se dá získat maximálně 100b., se v této kategorii dá získat dalších 100b. za parní stroj. Hodnocení parního stroje je obdobou hodnocení celého modelu. Oproti ostatním stavebním zkouškám u parníků stavební zkouška ovlivňuje konečné pořadí zhruba z 66 %, jelikož je součtem celkového hodnocení stavby, celkového hodnocení parního stroje a celkového bodového zisku z jízd.

6 Shrnutí výsledků

Hlavním cílem této práce bylo zanalyzovat současnou podporu pořádání modelářských závodů a na těchto základech naprogramovat aplikaci, rozdělenou na dvě části, která by výrazně zlepšila podmínky pořádání. Při vývoji aplikace bylo využito mých znalostí z jazyka Java. Zároveň jsem si své znalosti obohatil o jazyk PHP, zejména ve spojení s Nette Frameworkem.

Ve webové části bylo využito připraveného šablonovacího systému PHP Nette Frameworku ve spojení s návrhovým vzorem Model-View-Presenter. To bylo základem k vytvoření přehledné, znovupoužitelné, snadno rozšiřitelné, objektově orientované aplikace. Díky propracovanému Frameworku se podařilo vytvořit z pohledu přenosu dat velice úspornou aplikaci. Při načítání stránky probíhá přenos řádově v desítkách kB, což ocení zejména uživatelé mobilních zařízení. Kromě čistého Frameworku bylo využito i zásuvného modulu DateTimePicker, který ulehčuje zadávání dat a zároveň unifikuje jejich formát. Z hlediska využití a ochrany dat byl kladen důraz na propracovaný systém uživatelských práv, aby práce uživatelů jednotlivých vrstev byla pohodlná, udržitelná a v případě problému dohledatelná (logování). Webová část aplikace slouží především pro závodníky, kteří se díky jeho funkčnostem přihlašují na závody a ihned vidí aktuální stav přihlášených. Výstupem webového portálu je soubor CSV, který obsahuje informace o přihláškách, důležité pro druhou, desktopovou část.

Desktopová část je v podstatě rovněž založena na návrhovém vzoru Model-View-Controller, kde Controller je obdobou Presenteru ve webové části. Tato struktura byla ve webové části předpřipravena, avšak v desktopové musela být vytvořena. Desktopová část využívá standardní knihovny, jako jsou např. awt, util a swing. Zejména proto v této práci není jejich podrobnější popis.

Desktopová část na rozdíl od webové slouží pro pořadatele. Na základě exportovaných dat v souboru CSV z webového portálu pak pořadatelé v aplikaci prezentují soutěžící při prezenci. Po ukončení prezence aplikace zajišťuje generování zapisovacích archů. Poslední funkcí je zapisování výsledků jednotlivých jízd závodníků, na jejímž základě je generováno konečné pořadí a

ukončení závodu. Tato funkčnost ukončuje nahrazení současného stavu elektronické podpory, které bylo cílem práce. Dále by se v budoucnu dala aplikace rozšířit například o charakteristiky úspěšností závodníků, hodnocení nejlepšího soutěžícího aj.

7 Závěry a doporučení

V rámci této práce bylo zapotřebí využít získaných znalostí při studiu v praxi. V rámci analýzy současného stavu a návrhu nového řešení bylo využito zkušeností z předmětů OMO1 a SYSP. Při implementaci to pak byly znalosti získané z předmětů DBS1, DBS2, OWE, PGRF1, PGRF2, PRO1, PRO2, PRO3, TNPW1 a TNPW2. Tato práce zachycuje celý proces, od analýzy současného stavu po nasazení nového řešení do provozu.

Díky spuštění celého procesu do praxe byly odhaleny problémy, které na první pohled nebyly viditelné, ale postupem času se ukázaly jako klíčové.

Nasazení celé aplikace do provozu proběhne během jara 2015 a to ve dvou fázích. V první fázi bude zapojen do provozu webový portál, kde by se mělo spustit přihlašování na 1. seriálovou soutěž během měsíce března. V druhé fázi bude zapojena desktopová část a to pravděpodobně až na 2. seriálové soutěži v červnu, z důvodu zpracování změn nových pravidel.

Vzniklá aplikace tak bohužel v současné době není v souladu s novými pravidly a nelze ji ihned zapojit. Vzhledem k tomu, že se pravidla často mění, bylo vhodné dotvořit verzi v souladu s předchozími pravidly. Je pravděpodobné, že některá pravidla budou obnovena a díky tomu na ně bude aplikace předpřipravena.

8 Literatura a zdroje

- [1] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: Průvodce webového programátora*. Brno: Computer Press, 2007, s. 85. ISBN 978-80-251-1813-9.
- [2] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: Průvodce webového programátora*. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.
- [3] TVORBA-WEBU.CZ. *Tvorba-webu.cz: PHP* [online]. 2003 [cit. 2014-01-06]. Dostupné z: <http://www.tvorba-webu.cz/php/>
- [4] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: Průvodce webového programátora*. Brno: Computer Press, 2007, s. 33. ISBN 978-80-251-1813-9.
- [5] NETTE FOUNDATION. *Nette Framework: O frameworku* [online]. 2008 [cit. 2014-01-02]. Dostupné z: <http://nette.org/cs/about>
- [6] NETTE FOUNDATION. *Nette Framework: Licenční politika* [online]. 2008 [cit. 2014-01-02]. Dostupné z: <http://nette.org/cs/license>
- [7] NETTE FOUNDATION. *Nette Framework: Debugování a zpracování chyb* [online]. 2008 [cit. 2014-01-02]. Dostupné z: <http://doc.nette.org/cs/2.1/debugging>
- [8] NETTE FOUNDATION. *Nette Framework: Zabezpečení před zranitelnostmi* [online]. 2008 [cit. 2014-01-02]. Dostupné z: <http://doc.nette.org/cs/2.1/vulnerability-protection>
- [9] MIČKA, Pavel. *Algoritmy.net: Java pro začátečníky* [online]. 2008. [cit. 2015-01-06]. Dostupné z: <http://www.algoritmy.net/article/21340/Uvod-1>
- [10] PECINOVSKÝ, Rudolf. *Myslíme objektivně v jazyku Java: kompletní učebnice pro začátečníky*. 2., aktualiz. a rozš. vyd. Praha: Grada, 2009, 570 s. 39. - 41. Myslíme v--. ISBN 978-80-247-2653-3.
- [11] NAVIGA. *Pravidla sekce NS 2013*. Oberhausen, 2012. Dostupné z: http://svazmodelaru.cz/odbornosti/klom/pravidla/pravidla_ns_51.pdf
- [12] KLUB LODNÍCH MODELÁŘŮ ČR. *VZOR - Přihláška NS (ZT)*. 2004. [cit. 2014-10-01]. Dostupné z: http://svazmodelaru.cz/odbornosti/klom/pravidla/pravidla_ns_13.xls
- [13] NETTE FOUNDATION. *Nette Framework: MVC aplikace & presentery* [online]. 2008. [cit. 2014-10-01]. Dostupné z: <http://doc.nette.org/cs/2.1/presenters>
- [14] BERNARD, Borek. *Zdroják.cz: Prezentační vzory z rodiny MVC* [online]. 2009 [cit. 2015-02-21]. Dostupné z: <http://www.zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc/>

- [15] NETTE FOUNDATION. *Nette Framework: Vytvoření presenteru* [online]. 2008. [cit. 2014-11-01]. Dostupné z: <http://doc.nette.org/cs/0.9/quickstart/vytvoreni-presenteru>
- [16] TOPlist. Historie. 1997. [cit. 2014-11-26]. Dostupné z: <http://www.toplist.cz/stat/?a=history&type=3>
- [17] IT-SLOVNIK.CZ TEAM. IT-Slovník.cz: CSV [online]. 2008. [cit. 2015-01-05]. Dostupné z: <http://it-slovník.cz/pojem/csv>
- [18] BERNARD, Borek. *Zdroják.cz: Úvod do architektury MVC* [online]. 7.5.2009 [cit. 2015-01-19]. Dostupné z: <http://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>

**UNIVERZITA HRADEC KRÁLOVÉ****Fakulta informatiky a managementu**

Rokitanského 62, 500 03 Hradec Králové, tel: 493 331 111, fax: 493 332 235

Zadání k závěrečné práci

Jméno a příjmení studenta:

Martin Kupka

Obor studia:

Aplikovaná informatika

Jméno a příjmení vedoucího práce:

Filip Malý

Název práce:

Aplikace pro podporu činnosti modelářského klubu (s využitím desktopu a webu)

Název práce v AJ:

The application supporting the activity of model club (with using desktop and web)

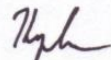
Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Cílem práce je analýza současného stavu elektronické podpory modelářského klubu. Dále na základě této analýzy pak navrhnout vhodnou aplikaci včetně implementace.

Osnova práce:

- 1) Úvod
- 2) Popis technologií PHP Nette a Java
- 3) Analýza současné podpory činnosti
- 4) Návrh aplikace
- 5) Implementace aplikace
- 6) Výsledky a závěr
- 7) Literatura a zdroje

Projednáno dne: **14. 10. 2013**Podpis studenta Podpis vedoucího práce 