Katedra informatiky Přírodovědecká fakulta Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Webová agenda autoškoly



2015

Vedoucí práce: RNDr. Arnošt Večerka

Dominik Veselý

Studijní obor: Aplikovaná informatika, prezenční forma

Bibliografické údaje

Autor:	Dominik Veselý
Název práce:	Webová agenda autoškoly
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2015
Studijní obor:	Aplikovaná informatika, prezenční forma
Vedoucí práce:	RNDr. Arnošt Večerka
Počet stran:	47
Přílohy:	1 CD/DVD
Jazyk práce:	český

Bibliograhic info

Dominik Veselý
Driving school agenda online
bachelor thesis
Department of Computer Science, Faculty of Science, Palacký University Olomouc
2015
Applied Computer Science, full-time form
RNDr. Arnošt Večerka
47
1 CD/DVD
Czech

Anotace

Snadné vedení agendy autoškoly a maximální možné šetření času byly hlavní cíle vyvinuté aplikace. Systém obsahuje dostatek prostředků pro vedení všech typů autoškol, kterými jsou Autoškola, Akreditované školící středisko a Školení referentů. K dispozici jsou nástroje pro jednoduché vyhledávání a manipulaci se všemi záznamy agendy. Dále systém obsahuje všechny potřebné tiskové výstupy pro vedení autoškoly. Aplikace také kontroluje správnost uživatelem zadávaných dat a snaží se co nejvíce upozorňovat uživatele na nadcházející události.

Synopsis

Easy management of a driving school agenda and maximal possible time saving, those were the main goals of developed application. System contains enough tools to manage all three types of driving schools which are Autoškola, Akreditované školící středisko and Školení referentů. Application offers tools for easy searching and manipulation of all records in the program. System also contains all printouts necessary to manage driving school. Application checks all user inputs for correct data and system warns user about all upcoming events.

Klíčová slova: agenda; autoškola; PHP; Nette; MySQL; Javascript

Keywords: agenda; driving school; PHP; Nette; MySQL; Javascript

Děkuji RNDr. Arnoštu Večerkovi za odborné vedení bakalářské práce.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvo	od	9
	1.1	Úloha agendy autoškoly	9
	1.2	Analýza požadavků	9
		1.2.1 Funkce nad rámec zákonných požadavků	9
		1.2.2 Specifické požadavky na aplikaci	10
		1.2.3 Navigace ve formuláři pomocí klávesy Enter	10
		1.2.4 Změna barvy pozadí	10
		1.2.5 Rychlé získání telefonních čísel žáků	10
	1.3	Rozdělení činnosti autoškol do tří oblastí	11
		1.3.1 Autoškola	11
		1.3.2 Akreditované školicí středisko (Profi)	11
		1.3.3 Školení referentů (Referenti)	11
	1.4	Hlavní požadavky na aplikaci	11
	1.5	Výběr platformy pro implementaci aplikace	13
2	Ρου	ıžité knihovny	13
	2.1	Nette Framework	13
		2.1.1 Architektura frameworku	14
		2.1.2 Presenter (Controller) $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	14
		2.1.3 View	15
		2.1.4 Model	16
		2.1.5 Komponenty \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots	16
		2.1.6 Moduly	17
		2.1.7 Neprůstřelný framework	17
	2.2	Lean Mapper	18
		2.2.1 Rychlost Lean Mapperu	18
		2.2.2 Řešení problému s n+1 dotazy	19
		2.2.3 Jednoduchý zápis entit	19
		2.2.4 Bezpečné obstarávání výsledků	20
	2.3	dibi	21
	2.4	mPDF	22
	2.5	Výčet knihoven mimo programovací jazyk PHP	22
		$2.5.1 Bootstrap \dots \dots \dots \dots \dots \dots \dots \dots \dots $	22
		$2.5.2 jQuery \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots $	22
3	Výv	voj aplikace	23
	3.1	Vzhled	23
		3.1.1 Styly pro tisk \ldots	23
	3.2	Autentizace a oprávnění	23
		3.2.1 Superadministrátor	24
		3.2.2 Admin	24
		3.2.3 Učitel	24
	3.3	Učty	25

3.4.1 Modul pro učitele 2 3.4.2 Modul pro žáky 2 3.4.3 Modul pro žáky 2 3.4.4 Modul pro věrejnost 2 3.5 Sekce modulu pro učitele 2 3.5.1 Obecné vlastnosti sekcí 2 3.5.1 Obecné vlastnosti sekcí 2 3.5.1 Obecné vlastnosti sekcí 2 3.5.1 Mstěnka 2 3.5.4 Firmy 2 3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečné odebírání dat 3 4.4 Bezpečné odebírání dat 3		3.4	Moduly – části webu	25
3.4.2 Modul administrace aplikace 2 3.4.3 Modul pro žáky 2 3.4.4 Modul pro věřejnost 2 3.5 Sekce modulu pro učitele 2 3.5.1 Obecné vlastnosti sekcí 2 3.5.2 Hlavní menu 2 3.5.3 Nástěnka 2 3.5.4 Firmy 2 3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 5.1 Vkládání stránek přes iframe 3 5.2 Přildání účtu 3 <th></th> <th></th> <th>3.4.1 Modul pro učitele</th> <th>25</th>			3.4.1 Modul pro učitele	25
3.4.3 Modul pro věřejnost 2 3.5.4 Modul pro veřejnost 2 3.5.5 Sekce modulu pro učitele 2 3.5.1 Obecné vlastnosti sekcí 2 3.5.2 Hlavní menu 2 3.5.3 Nástěnka 2 3.5.4 Firmy 2 3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpecňení přístupu k datům 3 4.3 Zabezpečení přístupu k datům 3 5.1 Přidání účtu 3 5.			3.4.2 Modul administrace aplikace	25
3.4.4 Modul pro veřejnost 2 3.5 Sekce modulu pro učítele 2 3.5.1 Obecné vlastnosti sekcí 2 3.5.2 Hlavní menu 2 3.5.3 Nástěnka 2 3.5.4 Firmy 2 3.5.5 Žáci 2 3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5.1 Přidání účtu 3 5.2 Přihlášsení do aplikace 4			3.4.3 Modul pro žáky	25
3.5 Sekce modulu pro učitele 2 3.5.1 Obecné vlastnosti sekcí 2 3.5.2 Hlavní menu 2 3.5.3 Nástěnka 2 3.5.4 Firmy 2 3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 3.5.12 Nastavení 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání účthy, vozidla nebo učitele do autoškoly			3.4.4 Modul pro veřejnost	26
3.5.1 Obecné vlastnosti sekcí 2 3.5.2 Hlavní menu 2 3.5.3 Nástěnka 2 3.5.4 Firmy 2 3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 5.1 Vřidání stránek přes iframe 3 5.2 Přihlášení do aplikace 4 5.3 Výtvoření nové autoškoly 4 5.4 Přidání účtu 3 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4		3.5	Sekce modulu pro učitele	26
3.5.2 Hlavní menu 2 3.5.3 Nástěnka 2 3.5.4 Firmy 2 3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání účtu 3 5.5 Výcvik žáků 4 5.6 Přidání žůků do kurzu 4 <td< th=""><th></th><th></th><th>3.5.1 Obecné vlastnosti sekcí</th><th>26</th></td<>			3.5.1 Obecné vlastnosti sekcí	26
3.5.3 Nástěnka 2 3.5.4 Firmy 2 3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jíždy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastvení 3 3.5.12 Nastvení 3 3.5.12 Nastvení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání účků do kurzu 4 5.6			3.5.2 Hlavní menu	27
3.5.4 Firmy 2 3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání účtu 3 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4			3.5.3 Nástěnka	27
3.5.5 Žáci 2 3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání účtu 3 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání řáků a na zkoušku 4			3.5.4 Firmy	28
3.5.6 Kurzy 3 3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpecčné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.6 Výcvik žáků 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku			3.5.5 Žáci	28
3.5.7 Jízdy 3 3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.6 Výcvik žáků 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická ch			3.5.6 Kurzy	30
3.5.8 Zkoušky 3 3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 4 Porovnání dvou nesetříděných polí 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení odebírání dat 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání učtu 3 5.2			3.5.7 Jízdy	31
3.5.9 Inventář 3 3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáka na zkoušku 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4			3.5.8 Zkoušky \ldots	32
3.5.10 Tisk 3 3.5.11 Sklad 3 3.5.12 Nastavení 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 <			3.5.9 Inventář \ldots	32
3.5.11 Sklad 3 3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4			3.5.10 Tisk	33
3.5.12 Nastavení 3 4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4			3.5.11 Sklad	33
4 Implementační aspekty 3 4.1 Porovnání dvou nesetříděných polí 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáku do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4			3.5.12 Nastavení \ldots	33
4 Implementacini aspekty 3 4.1 Porovnání dvou nesetříděných polí	4	Imp	lemente žní egy elity	94
4.1 Folovnám uvou nesetnuenych politivní v částí webu 3 4.2 Zajištění rychlejšího výpisu některých částí webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4	4	1111p	Porovnání dvou posotříděných polí	- 34 - 24
4.2 Zajistem fytnlejšno vypisu nekterých časti webu 3 4.3 Zabezpečení přístupu k datům 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4 5.9.3 Referenti 4		4.1	Zajištění rychlojšího výpisu pěktorých částí wohu	25
4.3 Zabezpecieli přístupu k datum 3 4.4 Bezpečné odebírání dat 3 4.5 Vkládání stránek přes iframe 3 5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4 5.9.3 Referenti 4		4.2	Zahrzen Tychiejsnio vypisu nekterych casti webu	- 30 - 36
4.4 Dezpeche odeonani dat		4.0	Bozpočné odobírání dot	30
5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4		4.4	Vkládání stránok přos iframo	38
5 Uživatelská příručka 3 5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4		4.0		00
5.1 Přidání účtu 3 5.2 Přihlášení do aplikace 4 5.3 Vytvoření nové autoškoly 4 5.3 Výtvoření nové autoškoly 4 5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4	5	Uži	vatelská příručka	39
5.2 Přihlášení do aplikace		5.1	Přidání účtu	39
5.3 Vytvoření nové autoškoly		5.2	Přihlášení do aplikace	40
5.4 Přidání učebny, vozidla nebo učitele do autoškoly 4 5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4		5.3	Vytvoření nové autoškoly	40
5.5 Výcvik žáků 4 5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4		5.4	Přidání učebny, vozidla nebo učitele do autoškoly	40
5.6 Přidání žáků do kurzu 4 5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4		5.5	Výcvik žáků	40
5.7 Stavy kurzu 4 5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4		5.6	Přidání žáků do kurzu	41
5.8 Přidání žáka na zkoušku 4 5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4		5.7	Stavy kurzu	41
5.9 Specifická chování různých typů autoškol 4 5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4 7 ávěn 4		5.8	Přidání žáka na zkoušku	42
5.9.1 Autoškola 4 5.9.2 Profi 4 5.9.3 Referenti 4 7 6 7		5.9	Specifická chování různých typů autoškol	42
5.9.2 Profi 4 5.9.3 Referenti 4 7 ávěn 4			5.9.1 Autoškola	42
5.9.3 Referenti			5.9.2 Profi	42
7 ótrón			5.9.3 Referenti	42
T AVEL 1	7.4	ávěr		<u>∕</u> 1२

\mathbf{A}	Příloha: Návod ke zprovoznění aplikace														
	A.1 Požadavky na zprovoznění aplikace														
	A.2	Instalace aplikace													
	A.3	Spuště	éní aplikace	. 44											
		A.3.1	Administrace aplikace	. 44											
		A.3.2	Předpřipravená data	. 45											
		A.3.3	Testovací verze aplikace	. 45											
в	Obs	ah při	loženého CD	45											
\mathbf{Li}	Literatura														

Seznam obrázků

1	Sekce Nástěnka zobrazující vzhled aplikace.	9
2	Zápis entity bez pomoci anotací (redukované formátování)	21
3	Stránka detailu učitele zobrazená na zařízení s běžným rozlišením.	23
4	Stránka detailu učitele zobrazená na zařízení se středním rozliše-	
	ním (typický je například tablet) vlevo a na zařízení s nejnižším	
	rozlišením (např. mobilní telefon) vpravo	24
5	Hlavní menu s otevřenou nabídkou	27
5	Hlavní menu s otevřenou nabídkou	2

Seznam tabulek

1	Porovnání rvchlosť	í databázových m	$\operatorname{nodelu}\left[4\right]$.	 19
-				

Seznam zdrojových kódů

1	Presenter s akcí a předáním parametrů do View	14
2	Ukázka latte souboru a jeho výhod	15
3	Ukázka přeloženého latte souboru do PHP	16
4	Vytvoření třídy komponenty	17
5	Vytvoření instance komponenty přes továrnu	17
6	Přidání formulářového prvku s omezením na zadání pouze valid-	
	ního emailu	18
7	Zápis entity s pomocí anotací	20
8	SQL dotaz zapsaný pomocí knihovny dibi	21
9	Ukázka tvorby responsivního designu s pomocí Bootstrapu	22
10	Porovnání obsahu dvou nesetříděných polí	35
11	Ukázka automaticky zabezpečeného dotazu do databáze	36
12	Ukázka dotazu získávajícího všechny autoškoly účtu, které nejsou	
	označeny jako odstraněné.	38
13	Definice předka všech presenterů v modulu pro veřejnost, který	
	automaticky povoluje používat stránky jako iframy.	39

1 Úvod

1.1 Úloha agendy autoškoly

Agenda autoškoly slouží k vedení záznamů o autoškole. Obsahuje informace jako seznam žáků, přehled jejich plateb, seznam kurzů, přítomnost žáků v kurzech, účast žáků na lekcích teoretického výcviku, jednotlivé jízdy žáků a výsledky zkoušek.

Jedním z hlavních předpokladů kvalitní agendy je také přehledný vzhled, jednoduchost a rychlost používání.

Menu	Nástěnka	Žáci	Kurzy	Jízdy	Zkoušky	Inventář	Tisk	Statistiky	Nastavení			Test Testik 🕶
Přida	it F	řidat	Obnovit									
Zprá	ry Upo	zornění										
Zprávy Upozornění												
Ne	zapomeňte							04.0	5.2015 01:02:01	Automat	ická upozornění	
Ne:	apomeňte za	apsat jízdy	žáků.							Blížící	se konec zdravotní způsobilos	ti
Up	Upravit (zpráva se bude zobrazovat všem pořád) Odebrat							• H	ájek Josef - zbývá 11 dní			
										Nezaha	ájené kurzy	
										• 0:	2.05.2015 (Autoškola)	

Obrázek 1: Sekce Nástěnka zobrazující vzhled aplikace.

1.2 Analýza požadavků

Analýza byla provedena ve firmě **Autoškola Morava s.r.o.** pod vedením jejího provozovatele **Josefa Lakomého**. Především bylo zjišťováno, jakým způsobem probíhá stávající vedení administrativní činnosti autoškoly a co vše musí aplikace implementující agendu autoškoly obsahovat po stránce naplnění dikce zákona. Dále byly zjišťovány konkrétní požadavky pana Lakomého a jeho zaměstnanců, tedy pro usnadnění práce nad rámec zákonných požadavků.

1.2.1 Funkce nad rámec zákonných požadavků

Pro maximální usnadnění bylo naimplementováno i několik nadstandardních funkcí šetřících čas uživatelům. Mezi tyto funkce patří:

- kniha jízd
- výkony učitelů
- evidence plateb

 $\bullet\,$ tiskové výstupy příjmových dokladů a faktur

1.2.2 Specifické požadavky na aplikaci

Byla zjištěna spousta zajímavých a specifických požadavků na možnosti aplikace, které byly posléze implementovány v rámci jejího vývoje. Většina z těchto požadavků vyplývala z faktu, že zaměstnanci autoškoly jsou zvyklí pracovat s programem pro vedení autoškoly vyrobeným v roce 2000. V tomto programu se nachází spousta funkcí, na které si uživatelé zvykli a rádi by je tedy využívali i v námi vytvářené aplikaci. Protože pro vedení autoškoly existuje jen malé množství programů, je velmi pravděpodobné, že spousta majitelů autoškol využívá právě zmíněné aplikace. Tím pádem i pro tyto potencionální uživatele budou ony specifické funkce přínosné.

Implementace funkcionalit požadovaných uživateli je nezbytná především z toho důvodu, že spousta z nich nejsou odborníky v oblasti práce s počítačem. V mnoha případech tito uživatelé využívají počítač pouze pro práci s tímto softwarem. Případná potřeba jejich přeučování by tedy často byla velmi náročná nebo by uživatelé dokonce odmítli takovou aplikaci používat.

Při vývoji tak bylo ve všech těchto případech dbáno na požadavky uživatelů a byly implementovány i všechny nestandardní funkce.

1.2.3 Navigace ve formuláři pomocí klávesy Enter

Jedním z požadavků bylo, že při vyplňování jakéhokoliv formuláře bude možné se posunout na další pole formuláře pomocí klávesy Enter. Tato klávesa ve výchozím stavu formulář odesílá. Kvůli zvyklostem uživatelů bylo potřeba toto chování upravit. V aplikaci se však nachází možnost, jak lze tuto funkci vypnout.

1.2.4 Změna barvy pozadí

Další z poměrně velmi požadovaných funkcí byla možnost změny barvy pozadí aplikace pro každou autoškolu. Tímto způsobem je uživatelům na první pohled jasné, ve které autoškole se nacházejí. Přestože se v pravém horním rohu aplikace nachází název právě aktivní autoškoly, uživatelé trvali i na možnosti nastavení barvy pozadí, které vycházelo opět z jejich zvyklostí ze staršího programu.

1.2.5 Rychlé získání telefonních čísel žáků

Některá internetová rozhraní pro odesílání SMS zpráv podporují možnost rozeslání jedné SMS zprávy na několik telefonních čísel. Bylo tedy žádáno, aby na určitých místech, kde se nachází seznam žáků, byla také možnost, jak rychlým způsobem přistoupit k seznamu jejich telefonních čísel. Řešením bylo přidání do menu tlačítka, které zobrazí vyskakovací (modální) okno se všemi telefonními čísly. Zobrazená čísla jsou také ihned označena a stačí je tedy pouze zkopírovat a vložit na libovolné místo.

1.3 Rozdělení činnosti autoškol do tří oblastí

Činnost provozovatelů autoškol spočívá ve výcviku nových řidičů, dále ve zdokonalování odborné způsobilosti profesionálních řidičů a také školení řidičů, kteří využívají osobních firemních vozidel při výkonu svého povolání. Jde o tři samostatné oblasti vedení administrativních agend, které jsou od sebe do značné míry odlišné. Naše aplikace bude muset pokrýt potřeby všech tří typů.

Třemi různými typy autoškol jsou: Autoškola¹ (1.3.1), Akreditované školicí středisko, které je nazýváno jednoslovným pojmenování Profi (1.3.2) a Školení referentů nazývané Referenti (1.3.3).

1.3.1 Autoškola

Provozování autoškoly je oblast, ve které probíhá výuka a výcvik zájemců o získání nového řidičského oprávnění pro některou skupinu nebo podskupinu řidičského oprávnění, zájemců o rozšíření určité skupiny nebo podskupiny řidičského oprávnění, ale například také zájemců o kondiční výuku a výcvik.

1.3.2 Akreditované školicí středisko (Profi)

Činností Akreditovaného školicího střediska je provádění výuky a výcviku řidičů v rámci zdokonalování odborné způsobilosti řidičů formou vstupních a pravidelných školení. Řidiči (tzv. řidiči z povolání) se tohoto školení účastní v pravidelných ročních kurzech.

1.3.3 Školení referentů (Referenti)

Zde jde o oblast, kdy má zaměstnavatel zaměstnance, kteří při své práci musejí řídit vozidlo. Je pak povinností zaměstnavatele dbát na řádné proškolení těchto zaměstnanců v oblasti pravidel silničního provozu a zajišťovat jim tato školení.

1.4 Hlavní požadavky na aplikaci

Na základě analýzy požadavků (viz. 1.2) bylo zjištěno několik nezbytných bodů, které musely být primárně dodrženy a vývoj a návrh celé aplikace se odvíjel právě od nich.

Prvním z hlavních požadavků na aplikaci byla její multiplatformnost² a to ve smyslu možnosti provozovat ji na různorodých zařízeních, kterými jsou především obecně PC^3 , tablet a mobilní telefon (téže označovaný jako smartphone).

¹Protože typ **Autoškola** je speciálním typem autoškoly, pakliže se budeme bavit pouze o tomto konkrétním typu autoškoly, budeme jej uvádět s prvním velkým písmenem.

²Multiplatformní software je termín, kterým se označuje takový program, který je schopný běžet na více počítačových platformách.

³Zkratka PC je z angického "Personal Computer", tedy v překladu "Osobní Počítač".

Druhým důležitým požadavkem byl přístup k aplikaci téměř odkudkoliv. Tento požadavek je samozřejmě limitován uživatelovou dostupností k internetovému připojení, které je pro distribuovaný systém nezbytné.

Třetím krokem bylo naplnění požadavků vyplývajících ze zákona 247/2000 Sb. o získávání a zdokonalování odborné způsobilosti k řízení motorových vozidel, ve znění pozdějších předpisů. Tento zákon v § 27 nařizuje provozovateli o průběhu výuky a výcviku žadatele o řidičské oprávnění a jeho účasti na hodinách výuky a výcviku vést průkaznou evidenci. Zákon připouští, že tato evidence může být vedena i v elektronické podobě. Vyhláška 167/2002 Sb., kterou se provádí zákon č. 247/2000 Sb., o získávání a zdokonalování odborné způsobilosti k řízení motorových vozidel, ve znění zákona č. 478/2001 Sb. dále upřesňuje v § 13 podmínky elektronické evidence. Konkrétně jsou to následující požadavky:

- Evidenční knihu a třídní knihu lze nahradit elektronickou evidencí výuky a výcviku.
- V elektronické evidenci výuky a výcviku se vedou údaje podle § 9 odst. 3 a § 10 odst. 2.
- Elektronickou evidenci výuky a výcviku lze vést za předpokladu, že program splňuje následující požadavky:
 - a) zapsané údaje o provedené výuce a výcviku žadatele o řidičské oprávnění jsou průkazné a vylučují možnost záměny těchto údajů mezi jednotlivými žadateli o řidičské oprávnění navzájem,
 - b) žadatel o řidičské oprávnění je označen svým jménem, příjmením, datem narození a evidenčním číslem,
 - c) zápisy v evidenci o výuce a výcviku žadatele o řidičské oprávnění lze provádět průběžně,
 - d) údaje o provedené výuce a výcviku je možno průběžně prohlížet, pořídit v tištěné podobě, popřípadě uložit na přenosná paměťová média ve formátu kompatibilním s používaným softwarem,
 - e) elektronická evidence výuky a výcviku je vybavena elektronickou pamětí s dostatečnou kapacitou pro zápis údajů o průběhu výuky a výcviku se stálostí údajů na dobu nejméně deseti let,
 - f) elektronická evidence výuky a výcviku je označena názvem a adresou autoškoly,
 - g) zapsaná data jsou zabezpečena proti vymazání, přepisu a dalším neautorizovaným zásahům.
- Jednotlivé zápisy údajů v elektronické evidenci výuky a výcviku se neopatřují podpisy žadatele o řidičské oprávnění a učitele výuky a výcviku.
- Veškeré údaje o provedené výuce a výcviku vedené v elektronické evidenci výuky a výcviku se ukládají v autoškole po dobu pěti let po posledním provedeném zápisu.

• Po celou dobu uložení musí program elektronické evidence výuky a výcviku zajišťovat možnost tiskového výstupu.

Neméně důležitá je rovněž dostatečná úroveň kontroly zadávaných dat, jako je validace vstupů od uživatele zahrnující kontrolu vyplnění povinných údajů a jejich správnost (např. kontrola formátu data apod.). Navíc byl kladen důraz i na logickou kontrolu zadávaných dat, jako je například kontrola potřebného věku žáka (pro danou skupinu výuky) při ukončování jeho výcviku nebo kontrola křížení kilometrů při zadávání jízd vozidla⁴ atd.

1.5 Výběr platformy pro implementaci aplikace

Z dříve zmíněných požadavků na aplikaci se jeví jako nejlepší volba webové aplikace. Jako programovací, skriptovací a kódovací jazyky byly zvoleny:

- PHP (rekurzivní název PHP: HYPERTEXT PREPROCESSOR): Je skriptovací programovací jazyk, určený především pro programování webových aplikací.
- HTML (název HyperText Markup Language): Je značkovací jazyk pro tvorbu (respektive kódování) webových stránek.
- CSS (název CASCADING STYLE SHEETS, česky KASKÁDOVÉ STYLY): Je jazyk pro popis vzhledu HTML elementů na webových stránkách.
- Javascript (zkráceně JS): Je skriptovací jazyk, který se vkládá přímo do HTML kódu stránky a umožňuje tvorbu tzv. dynamických webových aplikací.

2 Použité knihovny

Jako hlavní programovací jazyk aplikace byl zvolen jazyk PHP (viz 1.5). Dále bylo využito několik pomocných knihoven pro tento jazyk⁵.

2.1 Nette Framework

Jedná se o moderní framework⁶ postavený na jazyce PHP. Původně byl tento projekt vytvořen českým programátorem **Davidem Grudlem**. Nyní je vyvíjen

⁴Při zapisování jízd se kromě zvolení vozidla také zadává stav tachometru a to před jízdou i po jízdě. Je tedy potřeba, aby aplikace kontrolovala tyto zadané údaje a nepovolila uživateli uložit zadávanou jízdu v případě, že se zadaný stav tachometru se kříží se stavem tachometru jiné jízdy tohoto vozidla.

⁵Využití knihoven třetích stran je pro větší projekty důležité. Převážně jsou využívány větším množstvím programátorů a tím pádem dostatečně otestovány. Při programování větší aplikace je také práce s čitým jazykem PHP velmi nepraktická a proto je vhodné využít některého více strukturovaného řešení.

⁶Framewrok je softwarová struktura napomáhající při programování a usnadňující vývoj velkých aplikací.

skupinou dobrovolníků, kteří přispívají vylepšeními. O vydávání nových verzí framworku se stále stará původní tvůrce Nette David Grudl. Současně (rok 2015) se jedná o 3. nepopulárnější PHP framework [2].

Hlavními důvody pro volbu právě Nette Frameworku jsou:

- Jedná se o český framework s převážně českou a slovenskou aktivní komunitou
- Moderní framework s aktivním vývojem
- Dokonalé zabezpečení (více v kapitole 2.1.7)
- Dobrý návrh systému s důrazem na dobré programovací návyky

2.1.1 Architektura frameworku

Nette využívá objektového návrhu systému, který je dobře otestován praxí a v dnešní době velmi populární. Framework je postaven na architektuře nazvané jako MVC (z Model-View-Controller) a jedná se řešení oddělující přístup k datům (Model) od vykreslení dat (View) a od samotné logiky apliace (Controller⁷).

Architektura MVC zajišťuje přehledný kód a umožňuje jednoduché rozšíření aplikace v budoucnu. Je tedy samozřejmostí, že jsou všechny aspekty této architektury v aplikaci dodržovány.

V následujících třech kapitolách si rozebereme všechny části MVC modelu a začleníme je do architektury našeho frameworku.

2.1.2 Presenter (Controller)

Presenter je třída definující sekci webu (např. Nástěnka). Obsahuje jednotlivé akce (např. výpis úkolů na nástěnce, přidání nového úkolu apod.), kde většinou každá akce má přiřazen vlastní View. Dále může obsahovat tzv. handly, což jsou pouze signály⁸ předávané presenteru, tedy například signál *smazat úkol s ID 5*. Presenter se stará o vytvoření potřebných formulářů a komponent (viz. 2.1.5) a také o obstarání dat z Modelu a předání dat k vykreslení do View.

```
1 class TestPresenter extends \Nette\Application\UI\Presenter {
2   public function actionSeznam() {
3    $this->template->seznam = [1, 2, 3];
4   }
5 }
```

Zdrojový kód 1: Presenter s akcí a předáním parametrů do View

 $^{^7\}mathrm{Controller}$ je v Nette nazýván jako Presenter. Jedná se však pouze o jiné pojmenování téhož.

⁸Jak již název napovídá, handly tedy nemají vlastní stránku, která by se vykreslila. Po jejich zavolání dojde na straně serveru k provedení nějaké činnosti.

2.1.3 View

Za View je označována ta část, kterou je možné zobrazit ve webovém prohlížeči a je tedy odesílána uživateli. Zde se využívá jazyk HTML společně s CSS a Javascriptem.

V Nette se tato část píše do souborů nazývajících se **Šablony** s příponou latte (označováno tedy často jako Latte). Tento šablonovací systém je jednou z velkých předností Nette Frameworku. Jedná se o klasické HTML s výpisem PHP kódu pomocí maker⁹. Latte nám výrazným způsobem ulehčuje sestavení a údržbu HTML kódu a zvyšuje přehlednost. Navíc nám také zajišťuje ochranu vypisovaných dat (viz. 2.1.7). Porovnání mezi Latte a klasický zápisem je znázorněno v kódech 2 a 3.

```
1
  <thead>
2
   3
4
   ID
   Jméno
5
   Příjmení
6
7
   8
  </thead>
  9
10
    id \} 
11
   {$zak->jmeno}
12
   {$zak->prijmeni}
13
14
   15
  16
```

Zdrojový kód 2: Ukázka latte souboru a jeho výhod

Kromě maker disponuje šablonovací systém filtry. Jedná se o pomocníky při výpisu proměnných, které lze dokonce i řetězit. Pokud potřebujeme například vypsat pouze část textu, pak nám stačí zapsat **{\$text|striptags|truncate:200}**, kde filtr **striptags** zajistí odstranění html tagů z textu a následně se aplikuje filtr **truncate**, který zkrátí délku řetězce na (námi zvolených) 200 znaků s tím, že se navíc pokusí zachovat pouze celá slova.

Latte také využívá šikovného systému bloků a dědičnosti, takže si typicky nadefinujeme jeden Latte soubor jako tzv. **layout**, což je část stránky, která je na všech podstránkách stejná a v jednom místě pomocí **{include content}** označíme pozici, kam se vloží kód unikátní pro danou podstránku.

Latte soubor je nakonec přeložen do běžného PHP souboru a pro maximální

⁹Makrem je označována speciální funkce, která je provedena před samotným spuštěním vykonání kódu. Makro tak upraví zdrojový kód souboru a až ten je poté vykonán. Typicky se makro využívá pro ulehčení zápisu některých zdlouhavých kódů.

```
1
2
   <thead>
3
     <t.r>
      ID
4
      Jméno
5
      Příjmení
6
7
     8
    </thead>
    9
  <?php $iterations = 0; foreach ($zaci as $zak) { ?> 
10
      <?php echo Nette\Templating\Helpers::escapeHtml($zak->id,
11
         ENT_NOQUOTES) ?>
12
      <?php echo Nette\Templating\Helpers::escapeHtml($zak->jmeno
         , ENT NOQUOTES) ?>
      <?php echo Nette\Templating\Helpers::escapeHtml($zak->
13
         prijmeni, ENT NOQUOTES) ?>
     14
  <?php $iterations++; } ?>
15
   16
  17
```

Zdrojový kód 3: Ukázka přeloženého latte souboru do PHP

možný výkon je uložen do cache¹⁰.

2.1.4 Model

Obecně je model místo, které slouží pro obstarání dat. Jde především o to, aby se kód potřebný pro získání dat (třeba z databáze) nenacházel všude možně po aplikaci, ale byl centralizovaný. V našem případě to znamená, aby se veškeré dotazy do databáze nacházely na úrovni modelů a výsledky dotazů (data) byla dostupná jen pomocí veřejně přístupných metod modelu¹¹.

Jako Model naší aplikace byla zvolena knihovna LeanMapper, která je podrobně rozepsána v kapitole 2.2.

2.1.5 Komponenty

Komponenty jsou předpřipravené Latte šablony, které se mohou na více částech webu opakovat. V takovém případě nemusíme tyto kusy kódu kopírovat, ale na požadovaném místě stránky vykreslíme danou komponentu (samozřejmě

¹⁰Cache je termín užívaný pro dočasné uložení souboru, který se typicky nemění při každém načtení stránky, na disk serveru. Ve výsledku se pak nemusí stále dokola spouštět výpočet nějaké složité procedury, ale použije se již dříve vypočítaný výsledek. Souboru lze nastavit životnost, po které se automaticky znovu přepočítá. Toto přepočítání lze vynutit i v PHP kódu, což je potřeba při provedení akce, která změnila výsledek uložený v cache.

¹¹Všechny zmíněné vlastnosti modelu samozřejmě přispívání přehlednějšímu a jednoduššímu kódu, který je ve větší aplikaci tohoto rozsahu nezbytný.

jí předáme data, které se mají v komponentě vykreslit). V presenteru pak přes továrnu komponent¹² (viz. zdrojový kód 5) vytvoříme instanci dané komponenty (viz. zdrojový kód 4). Použití komponenty v šabloně pak může vypadat nějak takto: {control detailZaka \$zak} (voláme tedy vykreslení komponenty s názvem detailZaka a předáváme ji objekt žáka k vykreslení).

```
1
   namespace Control;
2
  class DetailZaka extends BaseControl {
3
4
    public function render($zak) {
5
      $template = $this->template;
6
      $template->setFile(__DIR__ . "/detailZaka.latte");
7
      $template->zak = $zak;
8
      $template->render();
9
10
     }
11
12 }
```

Zdrojový kód 4: Vytvoření třídy komponenty

```
1 //Metoda nacházející se v presenteru
2 protected function createComponentDetailZaka($name) {
3 return new \Control\DetailZaka($this, $name);
4 }
```

Zdrojový kód 5: Vytvoření instance komponenty přes továrnu

2.1.6 Moduly

Nette disponuje i takzvanými moduly, které slouží pro rozdělení aplikace do několika částí. Web pak může být rozdělen na moduly jako je běžně přístupný web (tzv. Frontend) a administrátorská část webu (tzv. Backend).

2.1.7 Neprůstřelný framework

Nette se zaměřuje na zajištění maximálního možného zabezpečení. Vyznává princip takzvaného "less code, more security", tedy méně kódu a více bezpečí.

Framework se snaží na všech možných zranitelných místech dosáhnout maximálního zabezpečení. To si můžeme představit třeba na výpisu proměnné do

 $^{^{12}}$ Továrna komponent zajišťuje, že při opakovaném využití komponenty se vytvoří pouze jedna její instance. Šetří se tak čas a paměť, protože více instancí téhož není potřeba.

HTML kódu stránky. Její hodnota se ve výchozím stavu vypíše zaescapovaná¹³ (zabezpečená). Pokud chceme vypsat proměnnou nezabezpečeně, musíme to explicitně uvést.

Framework za nás také řeší spoustu problémů a bezpečnostních děr nacházejících se v jazyce PHP. Například session¹⁴ se vyznačuje řadou bezpečnostních problémů. Nette proto obstarává implementaci session po svém a to takovým způsobem, že je opět zajištěno maximální zabezpečení.

Formuláře jsou jedním z hlavních prvků, kterými uživatel komunikuje s aplikací. Je tedy velmi důležité dobře ošetřit vstupy zadávané uživatelem. Nette dělá většinu práce za nás a automaticky provede základní ověření a ošetření údajů z formuláře. Mezi základní vlastnosti patří odstranění mezer ze začátku a z konce zadaných řetězců. Ověření, zda hodnota vybraná z nabídky (např. elementu select) se v nabídce nachází atd.

V případě, že při ověření zadaných dat dojde k chybě, nedojde k vykonání kódu po úspěšném odeslání formuláře a uživateli se vypíše seznam chyb.

```
1 $form = new \Nette\Application\UI\Form;
2 $form->addText("email", "Email")
3 ->addRule(\Nette\Application\UI\Form::EMAIL, "Zadaná hodnota musí
být email");
```

Zdrojový kód 6: Přidání formulářového prvku s omezením na zadání pouze validního emailu

2.2 Lean Mapper

Jedná se o ORM¹⁵ pro jazyk PHP postavené na knihovně **dibi** (viz. 2.3). Pro výběr Lean Mapperu bylo hned několik důvodů.

2.2.1 Rychlost Lean Mapperu

Jak ukazuje tabulka 1, Lean Mapper je současně pravděpodobně nejrychlejším databázovým modelem (jednoduše nasaditelným do frameworku Nette). Tabulka obsahuje porovnání rychlostí několika databázových modelů při provádění stejných úkonů. Zkratka **NDB** znamená "Nette Database", což je nativní databázová

¹³Zaescapovaná znamená, že se ošetří výpis znaků, které mají v jazyce HTML speciální význam. Ošetření proběhne takovým způsobem, aby se nevyvolal speciální efekt onoho znaku, ale aby došlo pouze k jeho textovému vypsání. Nedojde tak k rozpadení vzhledu stránky nebo dokonce napadení stránky útoky jako například XSS.

¹⁴Jedná se o funkcionalitu umožňující uložení dat ke konkrétnímu uživateli během jeho pobytu na stránce v jinak bezstavovém protokolu HTTP. Tímto způsobem je pak zajištěno třeba uložení informace, že je uživatel přihlášen.

¹⁵ORM (název OBJECT RELATION MODEL, česky OBJEKTOVĚ RELAČNÍ MODEL) je technika, která se využívá na mapování (neboli překlad) výsledků získaných z databáze na objekty a zpátky.

Vrstva	Čas	Paměť
NDB bez cache	0.39 s	9 MB
NDB s cache	0.39 s	8 MB
Doctrine2 bez cache	1.35 s	19 MB
Doctrine2 s cache	1.31 s	18 MB
NotORM bez cache	0.91 s	5 MB
NotORM s cache	0.92 s	4 MB
Lean Mapper	0.30 s	9 MB

Tabulka 1: Porovnání rychlostí databázových modelů [4]

vrstva v aplikaci Nette, která původně vychází z projektu **NotORM**. V zobrazené tabulce jsou navíc skutečnými ORM pouze **Doctrine2** a **Lean Mapper**.

2.2.2 Řešení problému s n+1 dotazy

Většina databázových modelů trpí jedním velkým nešvarem. Pokud vybereme z databáze **n** záznamů, které potom procházíme a u každého řádku získáváme hodnotu z jiné tabulky, na kterou se odkazujeme pomocí ID, pak se pro každý procházený řádek provede individuální dotaz do databáze, který si vybere pouze jeden konkrétní záznam z odkazované tabulky. Ve výsledku je stav takový, že jsem provedli nejprve jeden dotaz, který nám vrátil **n** záznamů a poté pro každý z **n** záznamů jsme provedli jeden individuální dotaz, který vrátil **1** řádek. Celkově tedy proběhlo **n+1** dotazů, což je velmi neefektivní.

Lean Mapper tento problém řeší takovým způsobem, že při položení prvního dotazu vracejícího **n** řádků si pamatuje, které řádky načetl a poté při prvním pokusu o získání řádku z tabulky na kterou daný záznam odkazuje, provede do databáze dotaz, který naráz získá všechny potřebné záznamy z odkazované tabulky. Poté už jen vrací požadovaný řádek bez potřeby dotazovat se do databáze. Efektivně tak redukuje problém $\mathbf{n+1}$ dotazů na pouhé $\mathbf{2}$ dotazy.

2.2.3 Jednoduchý zápis entit

Za entity jsou označovány právě objekty, na které jsou mapovány výsledky dotazů do databáze. Obecně neplatí, že jedna entita odpovídá jedné tabulce v databázi, avšak pro naše účely to není podstatné a v celé naší aplikaci platí pravidlo, že poměr entity k tabulce v databázi je vždy 1:1.

Lean Mapper usnadňuje vytváření těchto entity pomocí tzv. anotací¹⁶ což je velmi elegantní řešení šetřící především čas programátorovi a dále také zvyšující přehlednost entity. Přístup k hodnotám entita zprostředkovává pomocí tzv. **ma**-

 $^{^{16}{\}rm Anotace}$ je speciální typ komentáře nacházející před třídou či metodou a je dostupný pomocí reflexe.

gických metod¹⁷ a to konkrétně pomocí <u>get</u>, <u>set</u> a <u>call</u>. Dále také Lean Mapper ověřuje správnost datového typu zadané proměnné a například i zda se hodnota nachází ve výčtu přípustných hodnot a to pomocí klíčového slova **m:enum** (viz. zdrojový kód 7). Pokud dojde k problému při přiřazování hodnoty, Lean Mapper vyhodí výjimku. Dále je také značným způsobem usnadněn zápis odkazování se na jiné entity (někdy nazýváno traverzování), tedy relace mezi entitami a to pomocí klíčových slov **m:hasOne** (odkazujeme se na jednu nebo žádnou entitu), **m:hasMany** (pomocí spojovací tabulky se odkazujeme na několik entit), **m:belongsToOne** (na tuto entitu se odkazuje několik entity).

Entity lze samozřejmě zapisovat klasickým stylem, tedy pomocí přístupových metod. Avšak jak ukazuje rozdíl entity zapsané pomocí anotací (zdrojový kód 7) a entity zapsané bez použití anotací (obrázek 2), pak první volba je určitě přehlednější a mnohem pohodlnější pro zápis.

```
namespace Model\Entity;
1
2
   / * *
3
4
    * @property-read int $id
    * @property-read int|null $ucet_id
5
    * @property Ucet|null $ucet m:hasOne
 6
    * @property boolean $hromadna = true
7
8
    * @property string $title
    * @property string $text
9
    * @property string $typ m:enum(self::TYP_*)
10
11
    * @property \DibiDateTime|null $zobrazovat_do
12
    * @property \DibiDateTime $pridano
13
    * @property Ucitel[] $ucitele m:hasMany
14
15
    */
  class Zprava extends BaseEntity {
16
17
    const TYP DEFAULT = "default",
18
        TYP PRIMARY = "primary";
19
20
21
  }
```

Zdrojový kód 7: Zápis entity s pomocí anotací

2.2.4 Bezpečné obstarávání výsledků

V Lean Mapperu je doporučeno pravidlo, že metody repozitáře vracejí entitu nebo kolekci¹⁸ entit. Je tak zaručeno, že z datového úložiště nebudou získána i

¹⁷Magické metody jsou takové metody, které jsou zavolány v případě, že nebyla nalezena požadovaná metoda nebo členská proměnná.

 $^{^{18}}$ Kolekci si můžeme přestavit jako skupinu dat. Může se jednat o jednoduché pole nebo v některých případech o objekt zajišťující pohodlnější práci s kolekcí.

```
3
      namespace Model\Entity;
5 🗇 class Zprava extends BaseEntity {
0
           protected function initDefaults() {
              parent::initDefaults();
8
                $this->setHromanda(true);
10 [-]
11 [-]
12 [-]
13
          public function getId() { return $this->row->id; }
           public function getUcet_id() { return $this->row->ucet_id; }
           public function getUcet() {
                $row = $this->row->referenced('ucet', 'ucet_id', null);
14 E
15
16 -
17 E
18 E
                if ($row === null) { return null; }
                return new Ucet ($row);
           public function setUcet(Ucet $ucet) { $this->row->ucet id = $ucet; return $this; }
           public function getHromadna() { return $this->row->hromadna; }
19
           public function setHromadna($hromanda) { $this->row->hromadna = $hromanda; return $this; }
20 E
           public function getTitle() { return $this->row->title; }
           public function setTitle($title) { $this->row->title = $title; return $this; }
22
           public function getText() { return $this->row->text; }
23 🖨
24 📮
           public function setText($text) { $this->row->text = $text; return $this; }
public function getTyp() { return $this->row->typ; }
25 E
26 E
27 E
           public function setTyp($typ) { $this->row->typ = $typ; return $this; }
           public function getZobrazovat_do() { return $this->row->zobrazovat_do; }
public function setZobrazovat_do(\DibiDateTime $date = null) { $this->row->zobrazovat_do = $date; return $this; }
28 🕂
29 🖵
30 📮
           public function getPridano() { return $this->row->pridano;
           public function setPridano(\DibiDateTime $date = null) { $this->row->pridano = $date; return $this; }
           public function getUcitele() {
31
32
               $rows = $this->row->referencing('zprava_ucitel', 'zprava_id');
$value = [];
33
                foreach ($rows as $row) {
34
35
                    $valueRow = $row->referenced('ucitel', 'ucitel_id');
if ($valueRow !== null) { $value[] = new Ucitel($valueRow); }
36
37
38
                return $this->createCollection($value);
           3
39
```

Obrázek 2: Zápis entity bez pomoci anotací (redukované formátování)

data, která nejsou danému uživateli přístupná.¹⁹

2.3 dibi

Dibi je databázová abstrakční knihovna pro jazyk PHP. Umožňuje nám jednoduchým způsobem psát SQL dotazy a zároveň nás chrání před tzv. "SQL injection", což je forma útoku používaná k napadení a neoprávněnému přístupu do databáze.

Tato knihovna je vyzrálá a současně označená za hotovou. Znamená to tedy, že již není potřeba dalších vylepšení a není očekávána další verze.

V aplikaci je zapotřebí kvůli použité knihovně Lean Mapper (2.2).

```
1 //Proměnná $db je instance třídy DibiConnection
2 $db->select("*")
3                                 ->from("[zaci]")
4                               ->where("[jmeno] = ?", $jmeno);
```

Zdrojový kód 8: SQL dotaz zapsaný pomocí knihovny dibi

¹⁹Toto pravidlo bylo úmyslně porušeno v případech, kdyby bylo potřeba dosáhnout co nejrychlejšího poskytnutí výsledků (více se této problematice věnuje kapitola 4.2).

2.4 mPDF

MPDF je knihovna v jazyce PHP, která dokáže vygenerovat PDF soubor ze zdroje ve formátu HTML. Jelikož je v aplikaci potřeba spousta tiskových výstupů, stává se tím tato knihovna nezbytnou součástí pro tuto činnost.

2.5 Výčet knihoven mimo programovací jazyk PHP

2.5.1 Bootstrap

Jedná se zejména o CSS knihovnu[6], jejímž hlavní úkolem je zaručení toho, aby se stránka zobrazila v různých webových prohlížečích stejně. Použitím Bootsrapu tak odpadá značné množství problémů se odlišným zobrazením stránky v různých webových prohlížečích. Dále také zprostředkovává možnost, jakou lze velmi jednoduchým způsobem nakódovat responsivní²⁰ webovou stránku (zdrojový kód 9). Využití v aplikaci je zobrazeno na obrázcích 3 a 4. Bootstrap obsahuje také několik šikovných předpřipravených prvků, jako je vyskakovací okno, navigační menu, vysouvací menu apod.

```
<div class="row">
1
    <div class="col-lg-4 col-sm-6">
2
      Tento div bude na velké obrazovce roztažen přes jednu třetinu
 3
      stránky, na menší obrazovce bude roztažen přes půlku stránky a
 4
      na nejmenší obrazovce bude na šířku celé stránky.
 5
 6
    </div>
    <div class="col-lg-8 col-sm-6">
7
      Tento div bude na velké obrazovce roztažen dvě třetiny stránky,
8
      na menší obrazovce bude roztažen přes půlku stránky (bude tak do-
9
      plňovat zbylou šířku za předchozím divem) a na nejmenší obrazovce
10
11
      bude na šířku celé stránky.
     </div>
12
    <div class="col-lg-12 hidden-xs">
13
      Tento div bude vždy rozprostřen přes celou šířku stránky a
14
      na nejmenší obrazovce bude skrytý.
15
     </div>
16
17 </div>
```

Zdrojový kód 9: Ukázka tvorby responsivního designu s pomocí Bootstrapu

2.5.2 jQuery

Další využitou knihovou je jQuery [7], což je Javascriptiová knihovna, která usnadňuje práci se samotným Javascriptem.

²⁰Responsivní stránka je taková, která se přizpůsobí šířce displeje, na kterém je zobrazovaná. Tím pádem jsou prvky webové stránky jinak uspořádány, když jsou zobrazeny na klasickém počítači a když jsou zobrazeny na mobilním telefonu. Tím je docíleno toho, že bude stránka přehledná i na zařízeních s nižším rozlišením.

Menu Nástěnka Ev	.kniha Kurzy Jizo	ly Zkoušky Inventář	Tisk Statistiky	Admin Nastaveni			🛲 profici	Pirko Jirka
Upravit Odebrat	Zobrazit Přidat	Upravit Odebrat	Zobrazit Přic	dat Upravit Odebrat	Restartovat heslo	Učitel v autoškolách	Zavřit	Obnovit
dia	U	lčebnv		Učitelė				
Učitel Nov	ák Petr							Zavřít
Osobní informa E-mail:	ICE petr.novak@email.cz	Další informace Přihlašovací jméno: petrno	v	Oprávnění v autošl profíci	kole	Výchozí opr učitele	ávnění to	hoto
Pohlaví: muž Rodné číslo: 700101/1237		Občanský průkaz: 12345 Řidičský průkaz: 12345	6798 679	Typ účtu: admin		Typ účtu:		admin
Zdravotní způsobilost do: Ulice: Část obce: PSČ: Telefon 1: Telefon 2:	30.11.2015 Uničovská 5 Šternberk 78501 +420777555333	Číslo PP2: Vydání PPZ: Pracovní poměr: hlavní Datum nástupu: Datum odchodu: Přídán do aplikace: 08.04. Aktivní ano	2015 15:17:58	Oprávnění pro úpravy:	Vše	Oprávnění pro úpra	avy:	Vše
MZdy Platnost od data Výc	cvik skupiny A Výcvik	skupiny B Výcvik skupin	y C Výcvik skupiny	D Výcvík skupiny T Prosto	oj Teorie Dovo	A Diená Akce 🕫	utoškoly ^{profici}	
Přidat mzdu								
Platnost od	Výcvik SK	A	Výcvik SK B	Výcvik SK C				
Výcvik SK D	Výcvik Sł	кт	Prostoj	Teorie				
Dovolená	Uložit							

Obrázek 3: Stránka detailu učitele zobrazená na zařízení s běžným rozlišením.

3 Vývoj aplikace

3.1 Vzhled

Jako vhled aplikace byl vybrán tzv. Ribbon. Jedná se o prvek vyvinutý společností Microsoft pro svůj balík kancelářských aplikací "Microsoft Office". Do naší aplikace se dobře hodí a spousta zaměstnanců autoškol je s tímto vzhledem sžita²¹. Práce s aplikací jim tak přijde jednodušší a příjemnější.

3.1.1 Styly pro tisk

Aplikace obsahuje také speciální CSS soubor se styly, které se aplikují při tisku stránky. Tím je dosaženo toho, že je při tisku skryto Ribbon menu a všechny ostatní – pro tiskový výstup zbytečně – elementy stránky.

3.2 Autentizace a oprávnění

Autentizace uživatele probíhá jeho přihlášením a to zadáním uživatelského jména nebo emailu a hesla. Po úspěšném ověření přihlašovacích údajů je uživatel přihlášen a přesměrován na Nástěnku (3.5.3).

Každý uživatel má přidělenou určitou roli, která se může lišit pro každou autoškolu. Administrátor tedy uživateli zvolí výchozí roli a v některé autoškole mu může tuto roli změnit.

Všechny role jsou popsány v následujících podsekcích.

 $^{^{21}{\}rm V}$ autoškole pracuje i značné množství starších osob, kteří nemají rádi inovativní design a především potřebu učit se a sžívat s něčím novým.



Obrázek 4: Stránka detailu učitele zobrazená na zařízení se středním rozlišením (typický je například tablet) vlevo a na zařízení s nejnižším rozlišením (např. mobilní telefon) vpravo.

3.2.1 Superadministrátor

Jedná se o unikátní roli, která je speciálním případem role Admin (3.2.2). To znamená, že kterýkoliv uživatel s rolí Admin se může stát Superadministrátorem. U jednoho účtu však může být pouze jeden.

Uživatel s touto rolí má maximální možné oprávnění v rámci celého účtu a může tedy navíc provádět tyto specifické operace:

- Administrace²² autoškol
- Administrace učitelů a přiřazování učitelů do autoškol
- Administrace vozidel a přiřazování vozidel do autoškol

3.2.2 Admin

Uživatel v této roli má oprávnění k plnému řízení celé autoškoly, tedy ke všem možným funkcím pro danou autoškolu²³.

3.2.3 Učitel

Toto je nejběžnější role, kterou má většina uživatelů. Je zapotřebí uživateli s touto rolí určit množinu oprávnění. **Superadministrátor** tak určí, v kterých sekcích (viz. 3.5) bude mít uživatel (v roli **Učitel**) oprávnění k editace záznamů.

 $^{^{22}\}mathrm{Mluvíme}$ li o administraci, pak máme na mysli akce přidání, úpravy a smazání apod.

 $^{^{23}{\}rm Mezi}$ tato oprávnění se však nevztahuje možnost administrace vozidel a učitelů, kteří spadají pod celý účet.

3.3 Účty

Program obsahuje účty, které slouží jako obálka pro autoškoly a uživatele spadající pod stejného majitele. Správcem aplikace (viz. 3.4.2) je tedy vytvořen účet a k němu přidán jeden (první) uživatel²⁴ v roli **Admin**, aby se z něj mohl stát **Superadministrátor**.

Nově přidaný uživatel se pak přihlásí do standardní části webu (3.4.1) a zde již může ke svému účtu přidávat další uživatele s rolí běžného učitele či administrátora, kteří se sami mohou posléze přihlásit a provádět akce, ke kterým mají oprávnění. Dále pak lze přidávat autoškoly a vozidla, které je možné k jednotlivým autoškolám přiřadit (viz. 3.5.9). Výběrem některé (uživateli dostupné) autoškoly se pak tento uživatel pohybuje ve zvolené autoškole a provádí úpravy vztahující se k této autoškole (viz. 3.5).

3.4 Moduly – části webu

Jelikož je celá aplikace postavena na Nette Frameworku (2.1), ve kterém se tyto části označují jako "Module", budeme se bavit o modulech.

3.4.1 Modul pro učitele

Jedná se o největší část aplikace (hlavní důvod její tvorby), do které se přihlašují uživatelé a pracují s agendou. Tomuto tématu je tedy věnována celá sekce 3.5.

3.4.2 Modul administrace aplikace

Do této části má přístup pouze správce celé aplikace. Zde může přidávat nové účty a k nim jednotlivé uživatele²⁵. U uživatelů má také možnost změnit Superadministrátora, restartovat některému uživateli heslo, nebo se přihlásit za kteréhokoliv uživatele²⁶.

3.4.3 Modul pro žáky

Pokud je tato funkce v nastavení autoškoly povolena, pak mají žáci (této autoškoly) možnost přihlásit se a kontrolovat přehled jejich průběhu výcviku (v případě žáků autoškoly typu Profi se také mohou přihlásit do blížících se kurzů) nebo upravit osobní údaje.

 $^{^{24}\}mathrm{Každ}\acute{y}$ takovýto uživatel je zároveň učitelem a tyto pojmy tak splývají.

 $^{^{25}}$ Toto je nezbytné pro vytvoření uživatele s rolí Superadministrátor, který má oprávnění pro spravování celého účtu.

 $^{^{26}\}mathrm{P}$ řihlášení za uživatele je přidáno z důvodu, aby mohl pověřený člověk pomoci vyřešit problém.

3.4.4 Modul pro veřejnost

Tato část není dostupná jako celek, pouze se zde nacházejí stránky, které jsou veřejně přístupné (tedy bez potřeby přihlášení). Všechny stránky v této sekci mají povoleno vkládání přes "iframe"²⁷.

Nachází se zde stránky jako je například formulář pro zaregistrování žadatele o řidičské oprávnění. Tímto způsobem si může majitel autoškoly vložit formulář s přihláškou na svojí webovou stránku. Žadatel, který vyplní tento formulář, se automaticky zařadí do aplikace do sekce Sklad (3.5.11) a na emailovou adresu mu přijde potvrzující zpráva, kde se v příloze nachází předvyplněná přihláška a potvrzení zdravotní způsobilosti ve formátu PDF. Superadministrátor účtu, ke kterému se žadatel přihlásil, pak jednoduše přidělí uživatele do některé z autoškol. Tím je ušetřena spousta času jak žadateli, který pouze vyplnil přihlášku pomocí webového formuláře a automaticky mu byla předvyplněna písemná přihláška, tak uživateli aplikace, který nemusel opisovat žadatelovy osobní údaje z papírové (či jiné) formy přihlášky do naší aplikace.

3.5 Sekce modulu pro učitele

Tato část je dominantní v celé aplikaci a obsahuje většinu funkcionality. Jedná se o jádro a podstatu agendy. Budeme-li se tedy bavit o "sekci webu" a nebude specifikováno jinak, bude tím mít na mysli právě jednu z těchto sekcí.

Do aplikace přidáváme autoškoly, se kterými po sléze dále pracujeme. Přidávané autoškoly mají jeden ze tří již dříve zmíněných typů (viz. 1.3), kterými jsou **Autoškola** (1.3.1), **Profi** (1.3.2) a **Referenti** (1.3.3). Protože se jednotlivé funkce (ve všech sekcích) liší v závislosti na typu autoškoly, ve které se pohybujeme, budeme rozlišovat:

- funkce pouze pro typ Autoškola budou označovány^A
- funkce pouze pro typ Profi^P
- funkce pouze pro typ Referenti^R
- funkce pouze pro typ Autoškola a Profi^{AP}, případně lze ekvivalentně vyjádřit jako funkci platící mimo typ Referenti^{!R}

3.5.1 Obecné vlastnosti sekcí

Ve většině sekcí – vyjma Nástěnky (3.5.3) a Nastavení (3.5.12) – je potřeba mít zvolenou některou z autoškol, do které má přihlášený uživatel přístup, jako aktivní. To je nezbytné proto, aby bylo možno manipulovat s daty pouze zvolené autoškoly.

 $^{^{27}}$ Iframe neboli "vkládané okno" je metoda pro zobrazování webové stránky uvnitř jiné webové stránky. Z důvodu možných útoků na tuto technologii je ve výchozím stavu zakázána, proto je potřeba ji explicitně povolit (více o tomto problému v kapitole 4.5).

Sekce **Firmy**, **Žáci**, **Kurzy**, **Jízdy**, **Zkoušky** a **Inventář** také obsahují na jejich úvodní stránce seznam záznamů této sekce. Tyto seznamy jsou pomocí javascriptové knihovny **DataTables** zpřehledněny a obohaceny o možnost řazení záznamů dle libovolného sloupce či fulltextového vyhledávání. Díky rozsáhlé škále možností této knihovny bylo možné zajistit, aby byly všechny typy záznamů řazeny správně (například datum, které je vypisováno v českém formátu). V případě onoho data tedy primárně dle roku, poté dle měsíce, nakonec dle dne. Ve výchozím stavu by totiž řazení proběhlo textově zleva doprava, což je u data v českém formátu nedostačující. Dále lze také zajistit, že fulltextové vyhledávání vyhledává nejen z množiny zobrazených dat, ale také podle speciálně předaných dat, která by byla pro výpis do tabulky příliš obsáhlá. Toto se využívá například v sekcích **Kurzy**, **Jízdy** a **Zkoušky**, kde při vyhledání jména žáka se nám zobrazí ty kurzy, jízdy či zkoušky, ve kterých se takový žák nachází (viz. 3.5.6).

3.5.2 Hlavní menu

Za hlavní menu je označována nejhornější lišta aplikace, která obsahuje tlačítko "Menu", seznam uživateli dostupných sekcí (pro navigaci po aplikaci), název aktuální autoškoly a jméno přihlášeného uživatele.

Po kliknutí na zmíněné tlačítko "Menu" dojde k rozbalení nabídky, ve které je možno jednoduchým způsobem změnit aktuální autoškolu a několik dalších možností, které jsou pouze zkratkou k tlačítkům v sekci **Nastavení**. Více tedy o těchto možnostech v příslušné kapitole 3.5.12.

Hlavní menu i s rozbalenou nabídkou je zobrazeno na obrázku 5.

Menu	Nástěnka	Ev.kniha	Kurzy	Jízdy	Zkoušky	Inventář	Tisk	Statistiky	Nastavení	🖨 Autoškola	Test Testik 🕶
🖨 Auto	školy										
🚚 Ško středi	olící ska										
🚨 Ško refere	entů					_					
🖨 Nas	tavení										
🛛 Odhl	ásit se										

Obrázek 5: Hlavní menu s otevřenou nabídkou.

3.5.3 Nástěnka

Zde se učitel ocitne automaticky po přihlášení do aplikace. V této sekci se může dozvědět o nový zprávách od uživatelů s rolí Admin²⁸. Dále zde nalezne seznam upozornění²⁹, který obsahuje informace o blížícím se konci platností a propadnuté:

²⁸Pouze uživatelé s rolí Admin totiž mohou přidávat nové zprávy na nástěnku.

²⁹Každé z upozornění je zobrazené jen v případě, že má přihlášený uživatel pro danou sekci oprávnění úpravy.

- Zdravotní způsobilosti učitelů³⁰
- Platnosti STK³¹ vozidel
- Nezahájené kurzy³²

Uživatel má možnost si také přidat vlastní upozornění, která budou viditelná pouze pro něj.

Na obrázku 1 na straně 9 je zobrazena nástěnka s několika upozorněními.

3.5.4 Firmy

Tato sekce se nachází pouze v autoškole typu **Referenti**. Zobrazuje seznam firem, pod které lze poté přidávat žáky a kurzy.

Nachází se zde menu s tlačítky pro administraci (tedy přidání, úpravu a odebrání) firem a jejich přehledný seznam, kde je po kliknutí na libovolný záznam zobrazen detail firmy.

Detail firmy zobrazuje podrobný výpis všech jejich údajů a také seznam žáků nacházejících se v této firmě, kde každý vypsaný žák obsahuje odkaz na zobrazení jeho detailu.

3.5.5 Žáci

Zde se nalézají všichni žáci náležící do aktuální autoškoly.

Menu obsahuje:

- tlačítka pro administraci žáků
- přehled celkového dluhu³³
- pokročilé filtrování umožňuje například nevypisování již hotových žáků, kteří nejsou pro každodenní práci s aplikací nezbytní.
- filtrování rozsahu
 R umožňuje vyfiltrovat pouze přítomné či nepřítomné žáky ve zvoleném období

V případě, že je vybrán některý žák, v menu se dále aktivují tyto možnosti:

• tlačítko pro zobrazení přehledu jízd tohoto žáka

 $^{^{30}}$ Majitel autoškoly musí dbát na platnost zdravotních způsobilostí svých zaměstnanců, která je pro výkon učitele v autoškole nezbytná.

³¹STK (Stanice Technické Kontroly) je místo, kde dochází k technické prohlídce silničních vozidel, které podléhají všechna silniční vozidla evidovaná v České Republice a opatřená registrační značkou s výjimkou vozidel kategorie R.

 $^{^{32}}$ Při založení kurzu je požadováno také datum jeho zahájení. Proto je uživatel upozorňován při nezahájení kurzu (v rámci aplikace) po datu jeho skutečného zahájení.

³³Celkový dluh vidí pouze administrátoři autoškoly a jedná se o součet všech dluhů žáků v aktuální autoškole.

- tlačítko ukončit výcvik^A (pokud je žákův výcvik již ukončen, toto tlačítko je nahrazeno tlačítkem "znovu zahájit" pro možnost opětovného zahájení výcviku žáka)
- tlačítko vyřadit žáka^P
 - jedná se o stav, kdy je žák do jisté míry deaktivován
 - -vyřazený jedinec se pak nezobrazuje v seznamu žáků pro přidání do kurzu
 - v případě, že není vynuceno zobrazování vyřazovaných žáků (pomocí nastavení filtru), se nezobrazuje v seznamu žáků
 - nejedná se o stejnou akci, jako je odebrání žáka, protože v případě vyřazení lze žáka posléze zařadit zpět
- tlačítko vyřadit žáka z autoškoly^{A34} (za předpokladu, že s tímto žákem nebylo manipulováno, tedy pokud nebyl zařazen do kurzu apod.)
- tlačítko odebrat žáka (jedná se o uživatelem nevratnou akci, ačkoliv samotný záznam o žákovi v databázi z důvodu zachování referenční integrity zůstane)
- tlačítko pro zobrazení telefonních čísel žáků v seznamu (viz. 1.2.5)

Vypisovaný seznam obsahuje přehled filtrovaných žáků s jejich základními údaji a velikostí případného dluhu.

Detail žáka obsahuje:

- kompletní výpis všech osobních údajů
- cena kurzu A35 a tlačítko pro její úpravu
- doplňující ceny^{A36}
- seznam kurzů, ve kterých se žák nachází^{RP}
- platby (přidávané platby lze přiřadit k jednotlivým kurzům P)
- zkoušky^{!R} (zobrazují se pouze v případě, že je přítomna alespoň jedna zkouška)
- tiskové výstupy (jedná se o tiskové výstupy vztahující se pouze k aktuálnímu žákovi a pro každý typ autoškoly jsou specifické)

 $^{^{34}}$ Po vyřazení se žák přidá do skladu (3.5.11) žáků a je ho tedy možné znovu přiřadit do kterékoliv autoškoly.

³⁵Cena je dostupná pouze od chvíle, kdy je žák zařazen do kurzu.

 $^{^{36}}$ Jedná se o možnost, jakou lze žákovi naúčtovat prostoj (nedostavení se na jízdu) nebo poplatek za doplňkové jízdy apod.

3.5.6 Kurzy

Tato sekce – jak již název napovídá – se stará o administraci a přehled všech kurzů autoškoly.

Kurz se postupně nachází v jednom ze tří stavů, kterými jsou:

- rozpracován
 - jedná se o stav, ve kterém se kurz nachází ve chvíli, kdy je vytvořen
 - -v tom
to stavu lze kurzu libovolně upravovat a přidávat nebo odebírat z něj žáky
 - -v případě autoškoly typu Referenti musí během této fáze dojít k "odeslání kurzu na úřad" 37
 - $\ast\,$ při prvním odeslání se kurzu přidělí unikátní číslo a není již možné parametry kurzu upravovat
 - * po odeslání je stále možné upravovat množinu žáků kurzu, ovšem před jeho zahájením by mělo být opět odesláno nyní tzv. "Opravné přihlášení"
- zahájen
 - při zahájení se kurzu přidělí unikátní číslo (pokud ho již neměl přiřazeno)
 - všem žákům kurzu je vygenerováno unikátní evidenční číslo
- ukončen
 - zpravidla jsou žáci kurzu ukončováni postupně (po splnění docházky k teoretickému výcviku, odjezdění požadovaného počtu hodin jízd a zaplacení celé sumy za výcvik)
 - pro ukončení kurzu musí být ukončeni také všichni jeho žáci, proto při ukončování posledního neukončeného žáka kurzu dojde k ukončení samotného kurzu automaticky
 - v případě vynucení ukončení kurzu jsou zároveň ukončení i všichni jeho neukončení žáci

Kurz také obsahuje učební plán. Jedná se o seznam lekcí teoretické výuky. Protože se často struktura učebního plánu opakuje, je možné ji uložit jako šablonu a posléze tuto předpřipravenou šablonu použít u kteréhokoliv kurzu. Po vytvoření učebního plánu je žákům zaznamenávána docházka na jednotlivé lekce. Žák má celkovou účast splněnou pouze ve chvíli, kdy byl přítomen na všech lekcích.

Menu obsahuje:

 $^{^{37}}$ Jedná se o odeslání emailu na krajský úřad obsahující tiskový výstup "Přihlášení". V tomto tiskovém výstupu se nachází informace o kurzu a seznam žáků tohoto kurzu.

- tlačítka pro administraci kurzů
- tlačítko pro změnu stavu kurzu (chování tohoto tlačítka závisí na aktuálním stavu kurzu)
- tlačítko pro přihlášení kurzu na krajský úřad P
- tlačítka pro přidání a odebrání žáků z kurzu
- tlačítko pro úpravu učebního plánu
- tlačítko pro úpravu docházky
- filtr pro možnost výběru, zda zobrazovat či nezobrazovat ukončené kurz³⁸
- tlačítko pro zobrazení telefonních čísel žáků v zobrazeném kurzu (viz. 1.2.5)³⁹

Detail kurzu obsahuje:

- kompletní výpis všech parametrů kurzu
- seznam žáků v kurzu a tlačítko pro export seznamu do formátu CSV⁴⁰
- seznam přihlášení na krajský úřad P
- tiskové výstupy vztahující se k aktuálnímu kuru (jsou specifické pro každý typ autoškoly)

3.5.7 Jízdy

Sekce jízd se zabývá především jejich organizací, tedy možnostmi jak přidat, upravit nebo odebrat jízdu.

Seznam na hlavní stránce sekce zobrazuje stručný přehled jízd seskupený podle dnů. Vypisují se tedy pouze dva sloupce, kterými jsou datum a počet jízd toho dne. Od výpisu jednotlivých jízd bylo ustoupeno, protože jich mohou být v jeden den desítky a při velkém počtu jízd by se stal seznam velmi obsáhlý a nepřehledný a navíc by se stránka načítala poměrně dlouhou dobu. Díky chytrému fulltextovému vyhledávači však lze v tomto seznamu hledat také podle jmen žáků. Při vyhledání jména žáka se pak zobrazí pouze dny, ve kterých má daný žák zapsánu jízdu. Dále díky filtru v menu lze zobrazený seznam omezit jen na zobrazení jízd určitého vozidla, roku či měsíce (všechny tyto tři parametry lze samozřejmě také kombinovat).

 $^{^{38}}$ Zobrazení ukončených kurzů zpravidla není potřebné pro běžnou práci s aplikací. Bylo by tedy neefektivní ukončené kurzy v seznamu vždy vypisovat.

³⁹Rychlý přístup k telefonním číslům žáků v určitém kurzu je potřebný zejména v případech, kdy dojde ke změně času teoretické výuky. Kontaktovat žáky o změně je pak velmi snadné.

⁴⁰Soubor ve formátu CSV (COMMA SEPARATED VALUES) je formát souboru, ve kterém jsou jednotlivé položky odděleny čárkou (znak čárky může být nahrazen i jiným znakem, jako například středníkem).

3.5.8 Zkoušky

Tato sekce se velmi podobá předchozí sekci **Jízdy**.

Seznam obsahuje opět pouze dva sloupce a těmi jsou datum zkoušky a počet žáků na zkoušce, kde fulltextové vyhledávání znovu zvládne vyhledávat i podle jmen žáků.

V detailu zkoušky se nachází seznam jejich žáků s formulářem pro možnost úpravy výsledku zkoušky nebo odebrání žáka ze zkoušky.

Menu obsahuje:

- Administraci zkoušek
- Přidání žáků na zkoušku
- Tiskové výstupy pro Přihlášení žadatelů na zkoušku, Seznam žadatelů⁴¹ a možnost exportu zkoušky do formátu XML.

3.5.9 Inventář

Inventář lze definovat jako obálku pro 3 odlišné podsekce, které jsou však pro přehlednost aplikace zahrnuty pod jedinou záložku.

Podsekcemi tedy jsou **Vozidla**, **Učebny** a **Učitelé**, kde chování sekce **Učebny** je mírně odlišné. Zatímco přidání učebny platí pouze pro aktuální autoškolu a při změně autoškoly musíme případně stejnou učebnu znovu přidat, vozidlo či učitele přidáváme k celému účtu a poté pouze zvolíme, ve kterých autoškolách mají být dostupné. Tímto způsobem je docíleno kontroly křížení kilometrů vozidla a výuky učitelů při zadávání jízd a to i napříč všemi autoškolami (spadajícími pod jeden účet). Navíc údajů pro vyplnění je u vozidla a učitele (na rozdíl od učebny) větší množství a je tak ušetřen čas s přidáváním stejného vozidla či učitele do více autoškol.

U vozidel a učitelů se administrátorům v seznamu vypisují všechny záznamy (i ty nepřiřazené do aktuální autoškoly), ovšem vozidla a učitelé náležící do aktuální autoškoly jsou zobrazena tučně. Obecně však tyto sekce fungují základním způsobem, tedy stejně jako sekce **Firmy** (3.5.4). Uvedeme si tedy vždy pouze specifické funkce pro každou sekci.

Vozidla

- Změna přiřazení vozidla do autoškol
- Zobrazení tiskového výstupu Kniha jízd

 $^{^{41}}$ Jedná se o tiskový výstup, do kterého se posléze vyplňuje výsledek zkoušky.

Učitelé

- Změna přiřazení učitele do autoškol a nastavení specifického oprávnění pro každou autoškolu
- Restartování hesla
- Přidání a úprava mezd slouží pro počítání výkonu učitelů, kdy je možné spočítat mzdu dle odjetých jízd a odučené teorie

3.5.10 Tisk

V této sekci se nachází veškeré tiskové výstupy, které lze v aktuální autoškole provést. Většina tiskových výstupů je vytvořena v jazyce HTML⁴² a do formátu PDF jsou převedeny pomocí knihovny **mPDF** (viz. 2.4). Z předchozího vyplývá, že je tiskové výstupy možné zobrazit přímo, jako HTML stránku ve webovém prohlížeči nebo převést do formátu PDF, který lze stáhnout. Aplikace skutečně disponuje možností výběru, zda budou tiskové výstupy zobrazovány přímo v aplikaci jako část stránky a případě vytisknuty pomocí webového prohlížeče⁴³ (např. stisknutím klávesové zkratky **Ctrl+P**), nebo rovnou převedeny do formátu PDF a posléze zobrazeny v prohlížeči v tomto formátu nebo staženy na disk počítače. Situace se samozřejmě mění v případě, kdy je potřeba daný tiskový výstup odeslat jako přílohu emailu. Tehdy je skutečně nezbytné, aby byl tiskový výstup převeden rovnou do formátu PDF.

3.5.11 Sklad

V této sekci se nacházejí žáci nerozřazení do autoškol. Jedná se o žáky, kteří byli z autoškoly buď vyřazeni nebo se registrovali prostřednictvím formuláře. Superadministrátor si poté vybere, do které autoškoly (pokud vůbec) je zařadí. Před zařazením má také možnost zobrazit detail žáka a případně jej i upravit nebo odebrat.

3.5.12 Nastavení

Zde se nachází prakticky vše ostatní.

Úvodní stránka sekce obsahuje seznam autoškol, do kterých má přihlášený uživatel přístup, kde si může změnit právě aktivní autoškolu. Při kliknutí na některou z nich dojde k zobrazení jejího detailu. Dále se zde nacházejí odkazy pro registraci, přes které se mohou lidé do autoškoly registrovat.

Menu obsahuje tlačítka pro:

⁴²Toto pravidlo neplatí pro tiskové výstupy, které z povahy věci nelze tímto způsobem vytvořit. Jedná se například o přihlášku do autoškoly, kde je již pevně zadaný PDF soubor, do kterého se pouze doplní žadatelovy údaje.

 $^{^{43}}$ Zaručení přehledného vytištění tiskového výstupu zobrazeného na stránce je dosaženo pomocí již zmíněných stylů pro tisk (viz. 3.1.1)

- Přidání nové autoškoly
- Úpravu hlavičky (jedná se o úpravu údajů aktivní autoškoly)
- Úpravu osobních údajů
- Změnu hesla
- Změnu místního nastavení
 44 toto nastavení obsahuje:
 - Zapnutí ajaxových požadavků⁴⁵
 - Zapnutí animací 46
 - -Zapnutí posunování Enterem 47
- Odhlášení

4 Implementační aspekty

V této kapitole stručně shrňme některé komplikace a řešení na klíč, se kterými jsme se během implementace setkali.

4.1 Porovnání dvou nesetříděných polí

V aplikaci bylo zapotřebí ověřit, zda dvě nesetříděná pole obsahují stejné prvky. Možností řešení tohoto problému bylo několik, od porovnávání za pomoci dvou zanořených cyklů s algoritmem se složitostí $\theta(n^2)$, přes prvotní setřídění obou polí a poté souběžné procházení obou polí s porovnáním prvků s již příznivější

 $^{^{44}\}mathrm{M}$ ístní nastavení je ukládáno na straně klienta a to konkrétně v jeho prohlížeči.

⁴⁵Při využívání ajaxových dotazů nedochází k opakovanému načítání celé stránky. Přes javascript se tak odešle standardní požadavek, ale odpovědí není klasické HTML požadované stránky, nýbrž odpověď ve formátu JSON. Tato odpověď se poté opět javascriptem zpracuje a aktualizuje pouze změněné části stránky. Tímto způsobem dojde ke zrychlení celé aplikace, kdy se při pohybu po aplikaci načítají pouze změněná data a nemusí se tak obnovovat celá stránka. Problém samozřejmě může nastat při načítání nějaké obsáhlé stránky, kdy pomalejší zařízení (typicky mobilní telefony) nemusí toto načítání objemných dat pře Javascript zvládat a výsledkem může být pomalejší načítání než bez ajaxových požadavků. Z tohoto důvodu je možnost ajaxové požadavky vypnout.

⁴⁶Pokud je zapnutý ajax, pak při načtení stránky dojde k jejímu zobrazení pomocí animací. V případě zobrazení či skrytí detailu (např. detailu žáka) dojde k takzvanému efektu "fade", což je něco jako dýchnutí. Při zobrazení jiných podstránek (např. stánky pro přidání nového záznamu) dojde k příjezdu této stránky směrem zleva. Při skrývání takové stránky zase dojde k odjetí směrem doprava. Pokud by měl uživatel méně výkonné zařízení nebo by se mu jednoduše tyto animace nelíbily, je možné je tímto nastavením vypnout.

⁴⁷Po zapnutí funkce se pak při stisknutí klávesy Enter během vyplňování formuláře neprovede jeho odeslání, což je standardní chování, ale pouze se přesuneme (tedy změníme "focus") do dalšího pole formuláře (viz. 1.2.2).

složitostí $\theta(n * log(n))$. To vše by však vyžadovalo značný kus kódu s vyšší pravděpodobností výskytu chyby. Proto bylo využito tohoto algebraického předpisu:

 $\alpha = \beta \Leftrightarrow |\alpha| = |\beta| = |\alpha \cap \beta|$

Pak stačilo toto pravidlo převést do jazyka PHP a vzniknul jednoduchý jednořádkový zápis porovnání dvou polí (zdrojový kód 10), který je využit v souboru app/model/Entities/PrihlaseniKraj.php na řádku 27.

Zdrojový kód 10: Porovnání obsahu dvou nesetříděných polí.

4.2 Zajištění rychlejšího výpisu některých částí webu

Téměř každá sekce webu obsahuje přehledný výpis jejích dat (žáci, kurzy...). Tento výpis navíc probíhá při přístupu do sekce a uživatel se na stránkách tohoto typu pohybuje většinu času používání aplikace. Je tedy zásadní, aby výpis těchto seznamů probíhal co nejrychleji a nezdržoval uživatele.

Jde o části aplikace, ve kterých dochází k výpisu velkého množství záznamů a kde i výhody jinak velmi přínosného Lean Mapperu (2.2) selhávají. Proto bylo potřeba využít alternativního řešení.

Nevýhodou bezpečného vracení výsledku Lean Mapperu (viz. 2.2.4) je, že se nejprve musí z databáze obstarat všechna data. Poté jsou jednotlivě převáděna na entity a nakonec uložena do kolekce, která je vrácena jako výsledek. V takovém případě, pokud si obstaráme data, která poté ihned vypíšeme, musí Lean Mapper projít všechny fáze předcházející vrácení dat a my poté tyto data ihned procházíme a údaje vypisujeme. Dochází tak ke zbytečnému zdržení před a zároveň během samotného výpisu dat, které výrazně roste⁴⁸ vzhledem k množství získaných dat. V těchto situacích tedy bylo přistoupeno k ručnímu zapsání poměrně složitějších SQL dotazů (vracejících všechna potřebná data) a výsledek je vrácen jako instance třídy **LeanMapper\Fluent**, který implementuje rozhraní **ITraversable**. Takový objekt je pak možné vložit přímo jako argument do cyklu **foreach** a jednoduchým způsobem takto projít všechny výsledky dotazu.

 $^{^{48}}$ Začíná se narážet na hranice možností algoritmu využitého pro řešení problému s n+1 dotazy (2.2.2). Pokud je získáno například 1000 entit žáků, u kterých poté získáváme všechny jejich platby, Lean Mapper načte platby všech 1000 žáků najednou a vrací nám ty platby, které patří k žákovi, pro kterého je právě získáváme. To ale znamená, že v případě, že má každý žák řekněme 5 plateb, pak musí Lean Mapper při každém získávání plateb jednotlivého žáka projít všech 5000 plateb, ze kterých vybere jen ty, které k danému žákovi patří a ty vrátí jako výsledek. Znamená to, že při výpisu všech žáků se musí projít 5000 * 1000 = 5000000 plateb.

4.3 Zabezpečení přístupu k datům

Při vývoji aplikace vyvstal problém s implementací mechanismu, jakým bude zajištěna ochrana přístupu pouze k datům, ke kterým má daný uživatel v danou chvíli oprávnění.

Uveďme si například případ úpravy platby žáka. Pokud chceme některou platbu žáka upravit, odkazujeme se na ni pomocí jejího ID. Aplikace ale musí zajistit, abychom takovýmto způsobem nemohli měnit platby žáků nacházejících se mimo aktuální autoškolu či dokonce mimo aktuální účet.

Bylo využito stromové struktury v logickém větvení databázových tabulek a to takovým způsobem, že v kořenu stromu se nachází **účet**, pod který spadají **autoškoly**, **vozidla** a **učitelé**. Pod jednotlivé autoškoly spadají **žáci**, **kurzy** a **zkoušky**. K žákům přísluší **platby**, **doplňující ceny**, **jízdy**, apod. V případě našeho příkladu spadá platba pod některého žáka. Ten je zařazen v autoškole, která je přiřazena k některému účtu, ke kterému spadá zmiňovaný uživatel (manipulující s platbou) a který má povolen přístup do autoškoly, ve které je žák s upravovanou platbou.

Požadovaného efektu bezpečnosti bylo docíleno přidáním dodatečného parametru **\$parent** (sloužící jako rodič či majitel) do všech veřejných metod repozitářů vracejících citlivá data. Jmenovaný parametr musí být instancí třídy **Model\Entity\BaseEntity** a jedná se tak o objekt, který jsme již bezpečně získali z databáze. Při vyhledávání dat v databázi se tak omezujeme pouze na hledání v datech jednoho konkrétního předka (objekt \$parent). Při hledání v repozitáři **Model\Repository\PlatbaRepository** je potřebné předat jako parametr **\$parent** instanci třídy **Model\Entity\Zak**⁴⁹, jinak by vygenerovaný dotaz do databáze neměl smysl a zkolaboval by na problému dotazu obsahujícího neexistující sloupec. Vygenerovaný SQL dotaz do databáze (zdrojový kód 11) je tímto způsobem zabezpečený. V případě, že bychom se pokusili přistoupit k platbě, která existuje, ale spadá pod jiného žáka, dotaz vrátí nulový počet žáků a tedy by se platba se zadaným ID tvářila jako neexistující. To skutečně odpovídá faktu, že v kontextu aktuálního žáka existují pouze platby právě tohoto žáka.

```
1 SELECT 'platba'.*
2 FROM 'platba'
3 WHERE 'zak_id' = 1536 AND 'platba'.'id' = 3524
4 LIMIT 1
```

Zdrojový kód 11: Ukázka automaticky zabezpečeného dotazu do databáze.

Pro úplné doplnění příkladu je dobré dokresli obstarání všech rodičů (od

⁴⁹Zmiňovaný repozitář **Model****Repository****PlatbaRepository** je uzpůsoben tak, že lze jako parametr **\$parent** předat i instanci třídy **Model****Entity****Autoskola**. V takovém případě je u SQL dotazu připojena tabulka žáků za účelem autentizace platby vůči předané instanci entity autoškoly.

rodiče platby až po rodiče autoškoly), které proběhlo tímto způsobem⁵⁰:

- Pro získání platby jsme použili jako rodiče žáka, kterému platba patří.
- Pro získání žáka (potřebného pro získání platby) jsme jako rodiče použili autoškolu, ve které se aktuálně nacházíme.
- Aktuálně zvolenou autoškolu jsme ověřili vůči účtu, ve kterém se nachází jak samotná autoškola, tak právě přihlášený uživatel. Dále jsme také museli ověřit oprávnění přístupu onoho uživatele do autoškoly.

4.4 Bezpečné odebírání dat

Bezpečné odebírání nazývané jako "soft delete" je technika, která nám zajišťuje, že nedojde ke skutečnému smazání dat, ale je pouze nastaven příznak, že se jedná o smazaný záznam. Implementace této funkce a ošetření nechtěné práce s takto smazanými záznamy se nachází na úrovni aplikace.

Tato technika je používána v případě, kdy nechceme odebíraný záznam fyzicky smazat z databáze, což může být žádáno z několika důvodů:

- zachování integrity a existence cizích klíčů⁵¹
- ochrana před nechtěným odebraním

V naší aplikaci je této techniky využíváno z obou výše zmíněných důvodů. Konkrétně je "soft delete" aplikován na tabulkách **autoškola**, **vozidlo**, **učitel**, **učebna** a **žák**. Hlavním důvodem je především ono zachování existence cizích klíčů, kdy při skutečném odebírání vozidla by musely být odebrány také všechny jízdy tohoto vozidla (není možné, aby jízda neměla přiřazeno žádné vozidlo). Tím pádem by žákům zmizely veškeré jízdy s tímto vozidle a učitelům by tak zmizela část jejich výkonu a tím pádem i část jejich mezd.

Tato funkcionalita je implementována tak, že repozitář každé z takto zabezpečených tabulek je potomkem třídy **Model\Repository\BaseSoftRepository**, která zaručuje, že při odebírání je pouze nastaven příznak označující smazaný záznam. Při dotazech nad zabezpečeným repozitářem jsou automaticky všechny dotazy do databáze opatřeny ochranou před získáním odebraných záznamů (zdrojový kód 12).

 $^{^{50}\}mathrm{V}$ aplikaci probíhaly ve skutečnosti kroky v obráceném pořadí.

⁵¹Při fyzickém odstranění záznamu z databáze se musí vyřešit, co se stane se všemi záznamy, které na odstraňovaný záznam odkazují. Obecně platí několik technik pro řešení této situace: CASCADE – je metoda, která zajistí, že při odebírání záznamu jsou také odebrány všechny záznamy, které na odebíraný záznam odkazují; RESTRICT – je metoda, kdy zabráníme odebrání v případě, že se na odebíraný záznam odkazují některé záznamy z jiných tabulek; SET NULL – všem záznamům, které odkazují na odstraňovaný záznam se tento odkaz změní na hodnotu NULL; NONE – toto je standardní chování databázových enginů, které neposkytují podporu cizích klíčů a jedná se metodu, kdy dojde k odebrání záznamu, zatímco záznamy, které na odstraňovaný řádek odkazovaly, nyní ukazují na neexistující záznam a tím pádem je narušena integrita dat.

```
1 SELECT 'autoskola'.*
```

```
2 FROM `autoskola`
```

```
3 WHERE 'autoskola'.'odebrano' IS NULL AND 'ucet_id' = 2
```

Zdrojový kód 12: Ukázka dotazu získávajícího všechny autoškoly účtu, které nejsou označeny jako odstraněné.

4.5 Vkládání stránek přes iframe

Jak již bylo zmíněno (v sekci 3.4.4), byla potřeba umožnit některým stránkám aplikace, aby je bylo možno zobrazovat přes iframe (vložit jako část jiné webové stránky).

To je ve výchozím nastavení prohlížečů zakázáno z důvodu možného útoku nazývaného "clickjacking". Útok spočívá v tom, že se na zdánlivě bezpečnou webovou stránku vloží přes iframe jiná webová stránka. Tomu je nastavena průhlednost a je napozicován tak, aby překrýval původní obsah stránky. Uživatel se tak pohybuje po původní webové stránce, ale při kliknutí ve skutečnosti kliká do průhledného iframu, kde tímto spustí útočníkem požadovanou činnost. Firma Microsoft proto přišla s inovací (a to konkrétně s příchodem prohlížeče Internet Explorer 8[8]) ve formě nové HTTP hlavičky nazvané **X-FRAME-OPTIONS**, pomocí které lze nastavit chování webové stránky v případě, že je zobrazena jako iframe. Některé prohlížeče šly s ochranou ještě dále a začaly zakazovat i stránky, které tuto hlavičku vůbec neobsahují. V případě, že tedy chceme vkládání naší stránky přes iframe povolit, musíme tuto hlavičku explicitně nastavit (zdrojový kód 13).

Hodnotu této hlavičky lze nastavit na:

- DENY zakazuje jakékoliv vkládání přes iframe
- SAMEORIGIN umožňuje vložit iframe pouze pokud je vkládán na webové stránce nacházející se na stejné doméně (např. v rámci naší aplikace)
- ALLOW_FROM uri povoluje vložení stránky pouze na specifické URL adrese
- jakákoliv jiná hodnota povoluje vložení stránky přes iframe (toto je naše požadované chování a jako klíčové slovo používáme GOFORIT)

```
namespace VerejnostModule;
1
2
3
  class BasePresenter extends \Core\Presenters\BasePresenter {
4
    public function startup() {
5
     parent::startup();
6
     $this->getHttpResponse()->setHeader("X-Frame-Options", "GOFORIT")
7
    }
8
9
  }
```

Zdrojový kód 13: Definice předka všech presenterů v modulu pro veřejnost, který automaticky povoluje používat stránky jako iframy.

5 Uživatelská příručka

Vyvinutý informační systém slouží k vedení autoškoly v "online podobě". Obsahuje export všech potřebných údajů pro případné převedení agendy do tištěné formy. Aplikace také předpokládá uživatelovu znalost s vedením autoškoly. Jak jsme si již dříve uvedli, uživatel se může v aplikaci nacházet pod několik různými oprávněními, kterými jsou **Učitel**, **Administrátor** a **Superadministrátor**. Učitel má nejmenší úroveň oprávnění a v libovolné sekci mu může být odepřena možnost úprav. Administrátor má veškeré oprávnění v rámci dané autoškoly. Superadministrátor má nejvyšší možné oprávnění a to v rámci celého účtu (zaštiťujícího všechny autoškoly) a v rámci jednoho účtu musí být právě jeden (zpravidla se může jednat o majitele autoškol).

Popisovat funkce programu budeme tedy pouze z pohledu **Superadministrátora**, protože veškeré ostatní úrovně oprávnění mohou provádět stejně nebo méně operací.

V následujících sekcích si tedy popíšeme kroky potřebné pro vytvoření účtu a přidání prvního uživatele, který bude **Superadministrátorem**.

5.1 Přidání účtu

K přidání účtu se potřebujeme přihlásit do modulu pro správce aplikace. Do modulu se dostaneme pomocí tlačítka "Správa aplikace" na hlavní stránce webu. Přihlašovací jméno je **admin** a heslo je rovněž **admin**. Po přihlášení se nám zobrazí hlavní stránka aplikace, kde klikneme na tlačítko "Účty". Načte se nám stránka se seznamem účtů, která také obsahuje tlačítko "Přidat účet", kterým se nám zobrazí formulář pro vytvoření nového účtu.

Nově vytvořený účet se nám zařadí do seznamu účtů a kliknutím na tlačítko "Detail" zobrazíme detail účtu. Zde pomocí tlačítka "Přidat učitele" přidáme učitele. Ve formuláři musíme zvolit roli uživatele **Admin**, aby se mohl vytvářený uživatel stát **Superadministrátorem**.

Tímto máme vytvořený účet a v něm jednoho uživatele. Nyní tedy můžeme přejít do hlavní sekce aplikace, kde se za vytvořeného uživatele přihlásíme.

5.2 Přihlášení do aplikace

Přihlášení provedeme na úvodní stránce aplikace. Pro přihlášení potřebujeme znát přihlašovací jméno nebo email existujícího uživatele. V případě, že ještě nemáme vytvořeného žádného uživatele, pak musíme nejprve provést krok "Přidání účtu" (viz. 5.1). Při prvním přihlášení ještě nemá uživatel vytvořeno žádné heslo a proto jej ani nezadáváme (přihlásíme se tedy pouze pomocí uživatelského jména nebo emailu). Ihned po přihlášení jsme však vyzváni ke zvolení nového hesla. Zvolíme si tedy nové heslo a po úspěšném uložení se ocitneme v sekci **Nastavení** (3.5.12).

5.3 Vytvoření nové autoškoly

Vytvoření autoškoly provedeme pomocí tlačítka "Přidat novou" v sekci **Nasta-vení** nebo pomocí "Menu" -> "Nastavení" -> "Přidat k účtu další autoškolu".

Po vyplnění formuláře se vytvoří nová autoškola. Tuto nově vytvořenou autoškolu aktivujeme a to pomocí tlačítka "Aktivovat" vedle názvu autoškoly v sekci Nastavení.

Aktivací autoškoly získáme přístup do sekcí **Žáci**, **Kurzy**, **Jízdy**, **Zkoušky**, **Inventář** a **Tisk**. Dále se nám také zobrazí název aktivní autoškoly v horní liště (vedle jména přihlášeného uživatele) a případně se nám změní barva pozadí (pokud jsme pro autoškolu nastavili specifickou barvu pozadí).

5.4 Přidání učebny, vozidla nebo učitele do autoškoly

Všechny tyto akce se provádí v sekci **Inventář**, kde v menu klikneme na příslušné tlačítko "Přidat". Pro práci s aplikací je potřeba mít přidanou alespoň jednu položku od každého. Jednoho učitele již máme (jsme to my) a stačí tedy přidat učebnu a vozidlo.

Důležitou poznámkou zde je, že pokud posléze přidáme další autoškolu a budeme k ní chtít přiřadit učitele nebo vozidlo, které již v aplikaci máme, neměli bychom tyto záznamy znovu vytvářet. V takovémto případě v sekci **Inventář** zobrazíme seznam vozidel či učitelů a tlačítkem "Vozidla/Učitelé v autoškolách" nastavíme seznam aktivních vozidel či učitelů v aktuální autoškole.

5.5 Výcvik žáků

Výcvik žáků (z pohledu naší aplikace) probíhá ve všech třech typech autoškol lehce odlišným způsobem. Popíšeme si proto průběh výcviku v autoškole typu **Autoškola**, který je nejkomplexnější a do značné míry zahrnuje i průběh výcviku ostatních typů autoškol. Výcvik žáka tedy probíhá takto:

- 1. vytvoříme kurz
- 2. do vytvořeného kurzu přidáme žáky a nastavíme jim cenu výcviku
- 3. kurzu nastavíme učební plán (lekce pro žáky)
- 4. zahájíme kurz
- 5. probíhá kurz a žákům se přidávají platby, zapisuje se docházka na lekce a přidávají se jízdy
- 6. po dokončení výcviku žáka mu jeho výcvik v kurzu ukončíme
- 7. vytvoříme zkoušku
- 8. přidáme na zkoušku žáky
- 9. po proběhnutí zkoušky uložíme její výsledek

Upřesnění některých kroků je popsáno v následujících kapitolách.

5.6 Přidání žáků do kurzu

Při přidání žáka do kurzu musí být v kurzu vyučovány všechny skupiny výcviku, které žák žádá. Navíc v případě autoškoly typu **Autoškola** musí korespondovat druh výcviku zvolený u žáka s druhem výcviku kurzu.

5.7 Stavy kurzu

Při vytvoření se kurz nachází ve stavu **rozpracován**. V tomto stavu lze kurz libovolně upravovat a přidávat či ubírat z něj žáky.

V případě autoškoly typu Referenti musí být kurz odeslán na krajský úřad jako tiskový výstup "Přihlášení". Při odeslání na kraj je kurzu přiřazeno unikátní číslo a již ho nelze upravit, pouze lze přidávat a odebírat žáky. Případné provedené změny v seznamu žáků musí být před zahájením kurzu opět odeslány na krajský úřad nyní jako "Opravné přihlášení".

Po kliknutí na "Zahájit kurz" se kurz přepne do stavu **zahájen**. V tomto kroku se kurzu přidělí číslo kurzu (pokud ho již neměl přiděleno) a všem žákům kurzu je přiděleno evidenční číslo. Tento krok je nevratný, protože by mohla vzniknout mezera v číslech kurzu nebo v evidenčních číslech žáků.

Ukončování výcviku se zpravidla provádí jednotlivý žákům zvlášť a to ve chvíli, kdy dokončí teoretický i praktický výcvik. Při ukončení výcviku poslednímu neukončenému žákovi kurzu se automaticky ukončí také celý kurz. Kurz lze také ukončit naráz, kdy dojde k automatickému ukončení výcviku všem jeho neukončeným žákům.

5.8 Přidání žáka na zkoušku

Sekce **Zkoušky** je využívána především v autoškolách typu **Autoškola**. Žáka na zkoušku můžeme přidat až ve chvíli, kdy má ukončený výcvik v kurzu. Po přidání žáků na zkoušku a po reálném proběhnutí zkoušky uložíme její výsledek. Žáci, kteří splní všechny části zkoušky (případě v průběhu více zkoušek), jsou v aplikaci označeni jako "hotoví".

5.9 Specifická chování různých typů autoškol

Průběh výcviku žáků probíhá v každém typu autoškoly odlišně. Upřesníme tedy rozdíly ve všech třech typech autoškol.

5.9.1 Autoškola

V tomto typu autoškoly žák prochází postupně několika stavy, kterými jsou:

- Nezařazen žák se nachází v autoškole, ale nepatří do žádného kurzu
- Zařazen žák byl přidán do kurzu, který je ve stavu rozpracován
- Zahájen žák se nachází v kurzu, který je ve stavu **zahájen**
- Výcvik ukončen žákovi byl ukončen výcvik v kurzu, ale zatím nesplnil všechny části zkoušky
- Hotový žák již splnil zkoušky

Každý žák v tomto typu autoškoly se může nacházet pouze v jednom kurzu.

5.9.2 Profi

Tento typ autoškoly je používán pro řidiče z povolání, kteří musí být každý rok přeškoleni. Tím pádem mohou být žáci opakovaně přidáváni do kurzů.

5.9.3 Referenti

Tento typ je velmi podobný předchozímu typu autoškoly (**Profi**) s tím rozdílem, že v případě Referentů posílá zaměstnavatel své zaměstnance na povinné školení. Tím pádem se žáci i kurzy takové autoškoly vztahují k jedné firmě. Výcvik tedy hradí zaměstnavatel a platí za celý výcvikový kurz, nikoliv za jednotlivé zaměstnance.

Závěr

Vyvinutá aplikace je plnohodnotnou náhradou písemného vedení autoškoly. Bylo dbáno na specifické požadavky uživatelů autoškoly, ve které byla provedena analýza. Aplikace se snaží maximálním možným způsobem ušetřit čas jak uživatelům aplikace, tak žadatelům o výcvik a to především tak, že poskytuje veškeré tiskové výstupy potřebné pro obě zmíněné strany.

Naše aplikace navíc obsahuje značnou řadu výhod oproti písemnému vedení autoškoly. Všechna data lze efektivně řadit a filtrovat, takže vždy rychle nalezneme požadovaný záznam. Aplikace upozorňuje uživatele na blížící se události jako jsou vypršení platnosti STK vozidla či vypršení zdravotní způsobilosti některého ze zaměstnanců. Tato upozornění jsou pro chod autoškoly kritická, protože na vozidle s propadlou STK nesmí být odjety žádné kilometry a učitel s propadlou zdravotní způsobilostí nesmí provádět žádný výcvik. Systém také kontroluje uživatele a zamezuje možnosti zanesení nechtěných chyb kontrolou uživatelových vstupů. V maximální možné míře upozorňuje na snadno přehlédnutelné chyby, jako je ukončení výcviku žáka, který zatím nesplňuje věkový limit pro získání řidičského oprávnění pro požadovanou skupinu apod.

Systém také obsahuje několik nadstandardních funkcí, které ještě více usnadňují vedení autoškoly. Tím je například statistický přehled výkonu učitelů, který může být užitečným nástrojem i pro účetní autoškoly, kteří si jednoduše zjistí měsíční (či jakýkoliv jiný) přehled výkonu učitele a to jak praktické tak teoretické výuky.

Při vývoji aplikace bylo dbáno na přehlednost kódu a jednoduchou rozšiřitelnost systému. V případě nových požadavků na funkce, které by vyvstaly každodenním užíváním této aplikace, bude takovéto rozšíření jednoduchým úkonem.

A Příloha: Návod ke zprovoznění aplikace

A.1 Požadavky na zprovoznění aplikace

Vyvinutý software je webovou aplikací, z čehož vyplývají následující požadavky na jeho zprovoznění:

- webový server podporující PHP ve verzi 5.4 nebo vyšší
- databázový server MySQL ve verzi 5.5 nebo vyšší
- správné nastavení webového serveru nutné pro spuštění aplikace v Nette Frameworku
 - -požadavky na nastavení webového serveru se nacházejí na
http://doc.nette.org/cs/2.2/requirements
 - -v PHP je také nutné povolit rozšíření $\mathbf{mcrypt},$ které je potřeba pro generování hashů hesel

Ze strany uživatele je pro práci s aplikací požadováno použití aktuálního webového prohlížeče. Jedná se o prohlížeč ve verzi, kterou jeho vydavatel neoznačil za zastaralou. Doporučené je využít prohlížeče Mozilla Firefox nebo Google Chrome.

A.2 Instalace aplikace

- 1. Do určené složky webového serveru je potřeba překopírovat obsah archívu ve složce bin/ na přiloženém CD.
- 2. Na databázovém serveru je potřeba vytvořit novou databázi a do ní naimportovat strukturu tabulek nacházející se v souboru bin/db-structure.sql na přiloženém CD.
- 3. Dále je potřeba nakonfigurovat připojení aplikace k databázi. Konfigurační soubor se nachází v již překopírované struktuře aplikace z bodu 1. Kon-krétně se jedná o soubor app/config/config.local.neon, ve kterém je potřeba nastavit řádky username, password a database.
- Je potřeba zajistit, aby měl webový server možnost vytvářet soubory a složky v adresářích app/log a app/temp.

A.3 Spuštění aplikace

A.3.1 Administrace aplikace

Do administrace aplikace se dostaneme kliknutím na tlačítko "Administrace webu" na hlavní stránce webu. Přihlašovací údaje jsou:

- přihlašovací jméno: admin
- heslo: admin

A.3.2 Předpřipravená data

Na přiloženém CD se nachází soubor data/db-data.sql obsahující testovací data, která je možné naimportovat do databáze. Za předpřipraveného uživatele se pak do aplikace přihlásíme na hlavní stránce webu s použitím přihlašovacích údajů:

- přihlašovací jméno: **admin**
- heslo: admin

A.3.3 Testovací verze aplikace

Funkční testovací verze aplikace obsahující předpřipravená data se nachází na URL adrese http://bpdv.eu/.

B Obsah přiloženého CD

bin/

Kompletní adresářová struktura webové aplikace (v ZIP archivu) pro zkopírování na webový server. Dále se ve složce nachází soubor **db-structure.sql** obsahující strukturu databáze.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové texty webové aplikace se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory adresářové struktury pro zkopírování na webový server.

readme.txt

Instrukce pro nasazení webové aplikace na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Navíc CD obsahuje:

data/

Složka obsahující soubor **db-data.sql**, ve kterém se nacházejí ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce. Soubor je potřeba naimportovat do databáze, která již obsahuje strukturu databáze (nacházející se v sobuboru /**bin/db-structure.sql**).

Literatura

- [1] Nette Framework [online]. 2008-2014 [cit. 2015-04-04]. Dostupné z: http://nette.org/
- [2] SKVORC, Bruno. Best PHP Framework for 2015. In: Sitepoint [online]. 2000 - 2015 [cit. 2015-04-04]. Dostupné z: http://www.sitepoint.com/best-phpframework-2015-sitepoint-survey-results/
- [3] Lean Mapper [online]. 2013 [cit. 2014-04-04]. Dostupné z: http://www.leanmapper.com/
- [4] GRUDL, David. Lean Mapper tenké ORM nad dibi. In: *Dibi fórum* [on-line]. 2013 [cit. 2014-04-04]. Dostupné z: http://forum.dibiphp.com/cs/14592-lean-mapper-tenke-orm-nad-dibi?p=5#p106562
- [5] *MPDF* [online]. 2005 2013 [cit. 2015-04-04]. Dostupné z: http://www.mpdf1.com/mpdf/
- [6] Bootstrap [online]. 2014 [cit. 2015-04-04]. Dostupné z: http://getbootstrap.com/
- [7] JQuery [online]. 2015 [cit. 2015-04-04]. Dostupné z: http://jquery.com/
- [8] LAWRENCE, IE8 VII: De-Eric. Security Part ClickJacking fenses. In: IEBlog [online]. 2009 [cit. 2015-04-13]. Dostupné \mathbf{z} : http://blogs.msdn.com/b/ie/archive/2009/01/27/ie8-security-part-viiclickjacking-defenses.aspx