

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

BOZP portál - modul pro správu školení

Bc. Kamil Falta

Vedoucí práce: Ing. Jiří Chludil

25. června 2015

Poděkování

Především bych chtěl poděkovat vedoucímu práce Jiřímu Chludilovi, který mi při pravidelných konzultacích dával cenné rady. Také bych chtěl poděkovat rodině, která mi poskytlala potřebnou podporu během psaní této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 25. června 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Kamil Falta. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Falta, Kamil. *BOZP portál - modul pro správu školení*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato práce se zabývá vylepšením stávající webové aplikace pro podporu BOZP. Soustředil jsem se na část aplikace, která se zabývá správou školení. Nejprve jsem zjistil nové požadavky na aplikaci a pak jsem provedl analýzu aplikace, abych zjistil, jak nové požadavky do aplikace zapracovat. Z výsledků analýzy jsem vytvořil návrh a poté jsem navržené řešení realizoval. Vývoj probíhal ve frameworku Nette. Nakonec jsem hotovou aplikaci podrobil testům. Cílem práce je poskytnout lepší nástroj pro řešení problémů spojených se správou BOZP.

Klíčová slova BOZP, správa školení, Nette

Abstract

This thesis deals with upgrade of current version of BOZP management application. I focused on part of application that deals with training courses management. I started with requirements gathering and then I analyze current version so I could figure out how to implement gathered requirements. Base on result of analysis I designed solution. I implemented designed solution using Nette framework. Then I conducted testing of final. Purpose of this thesis is to create better tool for BOZP management.

Keywords BOZP, training course management, Nette

Obsah

Úvod	1
1 Rozbor zadání	3
2 Analýza	5
2.1 Revize požadavků zadavatele	5
2.2 Uživatelské testování původního řešení	10
2.3 Analýza původního řešení	15
3 Návrh	19
3.1 Případy užití	19
3.2 Uživatelské rozhraní	33
3.3 Architektura aplikace	37
3.4 Návrhový model tříd	37
4 Realizace	39
4.1 Technologie a knihovny	39
4.2 Nástroje	42
4.3 Implementační detaily	42
4.4 Instalace	44
5 Testování	49
5.1 Selenium	49
5.2 Uživatelské testování	49
Závěr	59
Možnosti dalšího rozvoje	59
Literatura	61
A Seznam použitých zkratk	63

B	Tabulky a obrázky	65
C	Obsah příloženého CD	69

Seznam obrázků

3.1	Diagram případů užití pro správu uživatelů	21
3.2	Diagram případů užití pro správu školení	25
3.3	Diagram případů užití pro správu akcí	27
3.4	Diagram případů užití pro správu aplikace	31
3.5	Návrh obrazovky Přehled	33
3.6	Ukázka komponenty panel	34
3.7	Ukázka poznámky u akce s dlouhých trváním	34
3.8	Ukázka FAQ	35
3.9	Ukázka dotazu na potvrzení akce	35
3.10	Ukázka nápovědy u formuláře	36
3.11	Diagram MVP [2]	37
3.12	Diagram tříd modelu aplikace	38
5.1	Ukázka 1 z uživatelského testování	53
B.1	Návrh obrazovky Moji studenti	66
B.2	Ukázka obrazovky Přehled	66
B.3	Ukázka obrazovky Moji studenti	67
B.4	Ukázka 2 z uživatelského testování	67

Seznam tabulek

2.1	Typy studia	6
2.2	Optimální hodnoty úvodního testování	12
2.3	Výsledky úvodního testování	12
2.4	Dotazník úvodního testování - otázka 1	13
2.5	Dotazník úvodního testování - otázka 2	13
2.6	Dotazník úvodního testování - otázka 3	13
2.7	Dotazník úvodního testování - otázka 4	14
2.8	Dotazník úvodního testování - otázka 5	14
2.9	Dotazník úvodního testování - otázka 6	14
2.10	Nové systémové role	17
2.11	Základní uživatelské skupiny	18
5.1	Dotazník závěrečného testování - otázka 1	54
5.2	Dotazník závěrečného testování - otázka 2	55
5.3	Dotazník závěrečného testování - otázka 3	55
5.4	Dotazník závěrečného testování - otázka 4	55
5.5	Dotazník závěrečného testování - otázka 5	56
B.1	Obrazovky aplikace	65

Úvod

V této práci se budu zabývat vylepšením aplikace pro podporu řešení situací spojených s bezpečností a ochranou při práci (BOZP) na ČVUT FIT. Budu se zabývat částí aplikace, která řeší správu školení BOZP. Například se jedná o řešení vypisování školení pro studenty a zaměstnance fakulty.

Portál BOZP byl v ostrém provozu již druhým rokem a za tu dobu vzniklo několik návrhů na vylepšení modul pro správu školení. S některými bylo počítáno i při původním návrhu, který se uskutečnil v rámci bakalářské práce Martina Humeníka [8], ale z různých důvodů se je nepodařilo do původní aplikace implementovat.

V první kapitole budu rozebírat jednotlivé body zadání. Vysvětlím, co bude účelem každého bodu zadání, a popíšu, jak ho chci naplnit. Následuje analýza, jejíž součástí bude sběr nových požadavků, uživatelské testování původní aplikace a analýza kódů původní aplikace. Návrh vytvořím na základě informací získaných v analýze. Budu definovat případy užití, které využiji pro návrh uživatelského rozhraní a modelu tříd.

V kapitole Realizace popíšu technologie použité při implementaci, přiblížím zajímavé implementační detaily a budu se zabývat nasazením aplikace na server. V poslední kapitole se budu věnovat testování. Popíšu použité automatické testy a vyhodnotím výsledky uživatelských testů.

Rozbor zadání

1.0.1 Analyzujte stávající řešení správy školení, identifikujte nedostatky a získejte aktuální požadavky od pracovníků BOZP.

Portál BOZP už je druhým rokem v ostrém provozu. Za tyto dva roky byla zjištěna řada nedostatků a vznikla potřeba pro nové funkce. Budu vycházet z požadavků, které ve své bakalářské práci specifikoval Marin Humeník [8], které přepracuji a vytvořím nové aktuální požadavky. Aktuální požadavky na část portálu pro zaměstnance a BOZP pracovníky, získám od správce portálu, jímž je vedoucí práce Jiří Chludil. Také zjistím názory běžných uživatelů pomocí uživatelských testů, které napoví další nedostatky původního řešení.

1.0.2 Navrhněte nové uživatelsky přívětivé rozhraní

Ještě před začátkem prací na analýze původního řešení bylo jasné, že bude třeba navrhnout nové uživatelské rozhraní. Stávající rozhraní je bezpochyby matoucí. Největší zásluhu na nepovedeném rozhraní má pravděpodobně způsob vzniku portálu. Během postupného vývoje přibývalo do portálu tolik funkcí, že se původně navržená navigace stala nepřehlednou. Výsledkem této části zadání bude nové uživatelské rozhraní pro celou aplikaci.

1.0.3 Implementujte autentizaci pomocí Shibboleth

Shibboleth je v současné době začíná používat na všech fakultních webech jako single sign-on řešení. Poskytuje tak řešení pro příjemné přecházení mezi fakultními weby bez nutnosti neustále zadávat stejné heslo. Původní řešení využívá pro přihlašování fakultní LDAP server a v rámci udržení jednoty fakultních systémů implementují přihlašování přes Shibboleth do portálu BOZP. Cílem toho bodu zadání je pouze připravit aplikaci na použití Shibbolethu a popsat jeho zprovoznění. Instalace Shibbolethu a zprovoznění provádět nebudu z bezpečnostních důvodů.

1.0.4 Navrhnete úpravu funkcionality pro stávající a podporu nových uživatelských rolí

Navrhnou, jak realizovat funkce, které byly identifikovány v analytické části práce. Při vytváření návrhu vytvořím případy užití a upravený model tříd a upravím model databáze. Vytvořený návrh bude sloužit jako základ pro implementaci.

1.0.5 Implementujte změny do portálu BOZP

Změny navržené v předchozích částech budu implementovat. Také popíši jaké technologie a nástroje jsem při implementaci využil. Výsledkem tedy bude funkční aplikace připravená k nasazení do produkčního prostředí.

1.0.6 Implementaci podrobte automatickým a uživatelským testům.

Abych ověřil především nové uživatelské prostředí, podrobím hotovou aplikaci uživatelským testům. Uživatelské testování bude provedeno ve speciální laboratoři pro uživatelské testování *Children usability lab* (CHUL). Bude probíhat zhruba tak, že vytvořím několik scénářů pro různé druhy uživatelů a nechám uživatele vypracovávat úkoly popsané ve scénářích. Během jejich testování budu uživatele nahrávat. Nakonec vyhodnotím výsledky a vyvodím důsledky pro rozhraní aplikace. Pro automatické testování jsem si vybral framework Selenium, protože s ním mám již dobré zkušenosti.

Analýza

V této části jsem se zabýval analýzou části portálu, která poskytuje funkce pro správu školení. Nejprve jsme provedli revizi funkčních a nefunkčních požadavků. Poté jsem využil uživatelských testů, abych zjistil klady a zápory původního uživatelského rozhraní. Nakonec jsme provedli analýzu původního řešení, ze které jsem vycházel při návrhu aplikace.

Základem pro zrevidované požadavky byly původní požadavky specifikované v bakalářské práci Martina Humeníka [8], ve které se autor zabýval vytvořením první verze modulu správy školení pro BOZP. Sběr nových požadavků jsem provedl ve spolupráci s vedoucím práce. Výsledkem byl nový zrevidovaný seznam funkčních a nefunkčních požadavků aplikace. Pro získání dalších požadavků na uživatelské rozhraní jsem využil uživatelské testování.

V analýze původního řešení jsem zjistil, co je potřeba, nedostatky jak funkční části, tak i technické. V technická část analýzy jsem se zabýval přehledností návrhu a zda je návrh v souladu s doporučenými praktikami použitých knihoven a frameworků. Ve funkční části analýzy jsem specifikoval funkce, které je třeba dodělat nebo upravit.

2.1 Revize požadavků zadavatele

Sběr požadavků jsem provedl proto, abych pomohl vymezit, co je potřeba změnit či zachovat ve původním systému. Požadavky se rozdělují na funkční a nefunkční. Funkční požadavky popisují, jaké funkce má systém poskytovat a nefunkční popisují kriteria, která má systém dodržovat, především v rámci použitých technologií a výkonnostních podmínek.

Za dva roky ostrého provozu se některé požadavky na systém změnila a některé požadavky nebyly implementovány dostatečně důkladně. Kromě toho vznikly nové požadavky na systém. Nové požadavky vznikly jak kvůli potřebě nových funkcí, tak v rámci vzniku nových nebo rozvoje starých technologií, které jsou integrovány do portálu.

V této části jsem získal aktuální požadavky na systém. Požadavky v některých případech byly stejné nebo podobné jako požadavky na původní systém, který ve své bakalářské práci vytvářel Martin Humeník [8]. Změny v požadavcích a nové požadavky na systém jsem probíral s vedoucím práce, který je zároveň BOZP pracovník a správce portálu BOZP, a tak dokáže nejlépe posoudit, jaké v portále potřebuje funkcionality.

Ke každému požadavku jsem popsal pro koho je daná funkcionality určena v rámci systémových rolí a co bude daná funkcionality obsahovat.

2.1.1 Funkční požadavky

2.1.1.1 F1 Zobrazení a editace uživatelů

Uživatelé budou moci zobrazit svůj profil se svými osobními údaji. Uživatelé budou moci měnit pouze doplňující údaje (osobní e-mail a telefonní číslo). Ostatní informace se budou zjišťovat z fakultních systémů. V případě, že je uživatel vytvořen ručně a jeho data nejsou automaticky aktualizována, bude je spravovat BOZP pracovník. BOZP pracovník bude moci deaktivovat uživatele. Deaktivovaný uživatel nebude započítán do statistik.

2.1.1.2 F2 Zobrazení seznamu všech uživatelů

BOZP pracovník bude moci zobrazit seznam všech uživatelů. Ke každému uživateli se zobrazí školení, které absolvoval a do kdy dané školení platí. Také bude možné uživatele filtrovat podle jména, příjmení, typu studia, personálního čísla a platnosti vybraného školení.

2.1.1.3 F3 Zobrazení statistik proškolení uživatelů

Statistiky budou rozděleny na studenty a zaměstnance. Dále budou studenti rozdělení podle různých typů školení. Přístup k zobrazení statistik proškolených uživatelů budou mít vedoucí pracovníci. Typy školení budou následující:

Tabulka 2.1: Typy studia

Bakalářské prezenční
Bakalářské distanční
Bakalářské anglicky
Bakalářské individuální
Magisterské prezenční
Magisterské anglicky
Magisterské individuální
Doktorské
Ostatní

2.1.1.4 F4 Žádost o uznání školení

Uživatel bude moci zažádat o uznání školení. K žádosti bude muset být přiložen oskenovaný dokument, na jehož základě bude školení uznáno bez toho, aby musel absolvovat školení.

2.1.1.5 F5 Udělení školení uživatelům

Aplikace umožní BOZP pracovníkovi systému udělit školení uživatelům systému. Školení bude moci být uděleno na základě navštívené akce nebo individuálně na základě nahraných dokumentů, které zajišťují uživateli uznání školení bez toho, aby musel absolvovat školení.

2.1.1.6 F6 Správa akcí

BOZP pracovník bude moci vypsát akci. Bude možné je vytvořit, editovat a smazat. Akce bude moci být typu přednášky, on-line test nebo zápis zápočtu. Akce bude mít záznam o tom, jakého typu školení se týká. Další atributy akce budou název akce, počet volných míst, datum uzavření přihlašování, datum konání, místo a školitel.

2.1.1.7 F7 Zobrazení všech akcí

BOZP pracovník bude moci zobrazit seznam, který bude zobrazovat všechny akce. Bude možné filtrovat akce podle názvu akce, typu školení, typu akce.

2.1.1.8 F8 Zobrazení dostupných akcí

Uživatel si bude moci zobrazit seznam dostupných akce. V seznamu se zobrazí všechny akce, které ještě neproběhly a které jsou určeny pro uživatelskou skupinu uživatele.

2.1.1.9 F9 Přihlašování a odhlašování se na/z akcí

Uživatelé se budou moci přihlásit na akci, pokud je ve skupině, pro kterou je daná akce určena. Uživatel se bude moci přihlásit či odhlásit z akce, pokud již není přihlašování či ohlašování uzavřeno.

2.1.1.10 F10 Pohled na seznam přihlášených na akci

Uživatel bude mít možnost zobrazit si všechny akce, na které je přihlášený. Zobrazovat se budou pouze akce, které se ještě nekonaly.

2.1.1.11 F11 Správa typů školení

Typy školení určují, pro jakou skupinu uživatelů jsou školení s daným typem určeny. BOZP pracovník bude moci vytvářet, editovat a mazat typy školení.

2. ANALÝZA

Každé školení bude mít název, platnost, cílovou skupinu uživatelů a volitelnou poznámku.

2.1.1.12 F12 Volba uživatelské role

Aplikace umožní uživateli mít více rolí zároveň. Uživatel, který má více rolí, bude moci přepínat mezi svými rolemi. Změna aktivní role se projeví bez nutnosti odhlášení a přihlášení.

2.1.1.13 F13 Správa uživatelských rolí

BOZP pracovník bude mít možnost upravovat, jaké role mají ostatní uživatelé, a bude moci měnit aktivní role pro ostatní uživatele.

2.1.1.14 F14 Správa sledovaných předmětů

Sledované předměty jsou předměty, které vyžadují speciální typ školení. BOZP pracovník bude moci přidat, editovat a smazat sledovaný předmět. Sledovaný předmět bude identifikovaný kódem z fakultních systému. Příkladem takového předmětu je *Struktura a architektura počítačů* (BI-SAP), pro který je potřeba mít školení *Vyhláška 50/78 Sb, §4*.

2.1.1.15 F15 Zobrazení studentů sledovaných předmětů

Zaměstnanci fakulty, kteří učí, přednášejí nebo jsou garanti předmětu s požadavkem na školení, budou moci zobrazit seznam studentů, kteří navštěvují daný předmět. V seznam se jim zobrazí, kteří studenti jsou proškoleni a kteří nejsou.

2.1.1.16 F16 Import uživatelů z fakultních systému

Import uživatelů bude probíhat ve dvou krocích. Každý den bude probíhat import nových uživatelů a změn dat stávajících uživatelů. Kromě automatického importu bude moci BOZP pracovník provést tuto akci manuálně. Po importu nových uživatelských dat se BOZP pracovníkovi portálu zobrazí notifikace, že jsou dostupná nová data. BOZP pracovník pak bude moci aplikovat změny uživatelských dat a aktualizuje data uživatelů portálu.

2.1.1.17 F17 Import cvičení a přednášek z fakultních systémů

BOZP pracovník bude moci provést import aktuální dat ze sledovaných cvičení. Při importu se pro každé sledované cvičení vytvoří nové paralelky, které budou obsahovat aktuální informace o studentech, kteří danou paralelku navštěvují a cvičící či přednášejících, které danou paralelku vedou.

2.1.1.18 F18 Zobrazení školení uživatele

Každý uživatel bude mít možnost zobrazit stav svých školení. Uživateli se zobrazí stav všech školení, které již dříve absolvoval, a těch, které by měl absolvovat, ale ještě tak neučinil.

2.1.2 Nefunkční požadavky

2.1.2.1 N1 Využití jazyka PHP a aktuální verze Nette frameworku

Stávající aplikace využívá PHP a Nette framework, takže by bylo zbytečné začít stavět aplikaci na nové platformě. Navíc je Nette framework velice rozšířený a správce portálu s ním má široké zkušenosti, takže v případě potřeby může bez problému provést drobné úpravy a opravy. Ale požaduje se přechod na novou aktuální verzi frameworku. Dále bylo doporučeno využití Nette doplňků Nexttras\Datagrid pro zobrazení strukturovaných dat a Kdyby/Translation pro lokalizaci systému.

2.1.2.2 N2 Využití databáze MySQL

Podobně jako u požadavku N1 by bylo zbytečné a značně náročné měnit databázi. Výkonnostně je MySQL dostačující a navíc by bylo nutné zmigrovat veškerá aktuální data aplikace

2.1.2.3 N3 Modifikovatelnost aplikace bude zajištěna

Tento požadavek je velice důležitý hlavně kvůli tomu, jak je aplikace vyvíjena. Aplikace je vyvíjena po částech studenty jako součást jejich závěrečných prací a každý do ní zanesl jiné postupy. Aplikace tedy musí být připravena na neustálou změnu vývojářů a obměnu kódu.

2.1.2.4 N4 Podpora přepnutí lokalizace systému mezi českým a anglickým jazykem

Aplikace umožní přepnutí jazyka mezi českým a anglickým, protože na fakultě se vyučuje česky i anglicky. Studenti, kteří studují v angličtině, proto musí mít možnost zvolit si angličtinu i při používání aplikace.

2.1.2.5 N5 Podpora autentizace uživatelů pomocí Shibbolethu

Zajištění autentizace uživatelů pomocí Shibbolethu umožní využívat single sign-on řešení, které poskytuje fakulta pro většinu svých webů. Uživatelé tak nebudou obtěžováni neustálým dotazováním na stejné heslo při přechodu mezi weby fakulty.

2.1.2.6 N6 Zálohování data v aplikaci

Aplikace umožní zálohovat data, aby se neztratila v případě nepovedeného importu dat nebo jiných událostí, které by mohly data ohrozit.

2.1.2.7 N7 Použití ACL (Access Control List) pro řízení přístupu na základě role uživatele

Aplikace bude využívat ACL pro kontrolu přístupu. ACL umožní rolím přidat povolení a přístupy k akcím na definovaných zdrojích. Poskytuje tak přehlednou správu přístupů pro celou aplikaci.

2.1.2.8 N8 Návrh nového uživatelského rozhraní

Správce portálu nebyl spokojen s původním rozhraním. Největší problém byl s navigací. Správa portálu obsahuje poměrně velké množství funkcí, které nebyly intuitivně přístupné zejména s ohledem na menu, které se mění v závislosti na aktuální stránce.

2.2 Uživatelské testování původního řešení

Původní rozhraní správy školení bylo podrobena uživatelské testům. Tyto testy byly navrženy v rámci semestrální práce [6] z předmětu *Návrh uživatelského rozhraní* (MI-NUR). Tato práce byla zpracována v týmu, jehož členy byli Lukáš Janeček, Filip Vorel, Martin Náhlovský a Kamil Falta (autor práce).

Tyto testy jsem se rozhodl využít z toho důvodu, že v rámci mé práce jsem měl vytvořit nové uživatelské rozhraní, které podle správce portálu není v ideální podobě. To má za následek nevyužívání plných možností portálu a zbytečnou časovou náročnost používání některých funkcí.

2.2.1 Účel testu

Účelem testu bylo získat základní přehled o nedostacích původního rozhraní systému. Omezil jsem se na názory studentů na rozhraní funkcí, ke kterým mají studenti přístup, protože požadavky na zlepšení rozhraní částí, které využívají zaměstnanci a BOZP pracovníci, jsem získal od vedoucího práce.

Jelikož je součástí mé práce návrh nového uživatelského rozhraní, byly kromě modulu správy školení otestovány i ostatní části systému. Tak jsem získal základní přehled o hlavních nedostacích, kterých bych se měl vyvarovat, i ostatních částí systému. Nové rozhraní jsem pak navrhl s ohledem na poznatků.

2.2.2 Profil uživatelů testujících systém

Test byl navržen pro studentské role portálu, proto bylo důležité získat testery, které se podobají reálnému uživateli systému. Mně se podařilo do testování zapojit studenty z FIT ČVUT, tedy uživatele, pro které je portál určen.

Dále bylo vhodné, aby účastníci testu neměli předchozí zkušenosti se systémem, aby se dali považovat za nezkušené uživatele a výsledky testu vypovídaly o intuitivnosti rozhraní.

2.2.3 Metoda testování

Testování probíhalo vzdáleně. Uživatelé obdrželi pokyny pro vypracování a dotazník k vyplnění po skončení testu. Vzdálené testování bylo zvoleno, protože stačilo získat pouze hrubá data, která měla napovědět největší nedostatky původního rozhraní. Další podrobnější testování bude provedeno po zapracování nových požadavků a opravě nejzásadnějších chyb.

Testy se skládaly z jednotlivých úkolů, které uživatel, který aplikaci testoval, prováděl v daném pořadí. Každý úkol odpovídal reálnému požadavku, který by mohl provést reálný uživatel. Každý uživatel měl provést několik úkolů.

Během testů se měřil čas strávený plnění úkolů a počet kliknutí, které musel tester učinit. Po skončení testu testeři zodpověděli několik otázek, které se týkaly přehlednosti a intuitivnosti rozhraní.

2.2.4 Úkoly

1. Nahlaste závadu
 - Přihlaste se jako „novakmar“ s heslem „test“
 - Nahlaste prasklou žárovku v místnosti „T9:304“
2. Přihlaste se na školení
 - Přihlaste se na školení „Školení BOZP“
3. Vyplňte test
 - Vyplňte test „Cvičné školení 2 (cs)“
4. Zjistěte stav vašeho úrazu
5. Zjistěte, kdy jsou konzultační hodiny BOZP

2.2.5 Výsledky měření

Pro každý úkol byl stanoven optimální čas a počet kliknutí, aby bylo možné vyhodnotit výsledky testů. Optimální čas byl stanoven tak, že byl každý úkol proveden uživatelem, který měl přesný návod, jak úkol splnit. Problém byl se stanovením správného tempa vykonávání jednotlivých kroků, protože každý uživatel má jiné tempo, což se může projevit ve výsledcích měření. Nakonec bylo stanoveno tempo, které odpovídalo přibližně 5 sekundám na jednu podakci (vyhledání informace na stránce nebo kliknutí na odkaz). Optimální počet kliknutí byl více přímočarý. Stačilo pouze vzít minimální počet kliknutí k dosažení cíle úkolu.

Tabulka 2.2: Optimální hodnoty úvodního testování

Úkol	Čas v s	počet kliknutí
Nahlášená závada	40	8
Přihlášení školení	20	2
Vyplnění testu	75	15
Zajištění stavu vašeho úrazu	10	2
Kdy jsou konzultační hodiny	10	2

Tabulka 2.3: Výsledky úvodního testování

Testeři	Student 1	Student 2	Student 3	Student 4
Úkoly	Čas v s / Počet kliknutí			
Nahlášení závady	32/10	53/8	60/14	30/8
Přihlášení na školení	20/6	15/2	50/5	20/4
Vyplnění testu	44/16	28/15	120/19	50/15
Zjištění stavu vašeho úrazu	8/2	10/2	60/6	10/4
Kdy jsou konzultační hodiny BOZP	43/5	23/2	180/7	40/10

2.2.6 Dotazník

Dotazník slouží k získání informací o testovaném rozhraní, které doplňují informace získané z měření. Tyto informace mohou poskytnout větší porozumění naměřených výsledků. Navíc všichni testeři měli znalosti z vývoje webových aplikací, tudíž by byla škoda nevyužít jejich názorů. Testeři zodpovídali následující otázky:

Tabulka 2.4: Dotazník úvodního testování - otázka 1

Orientovali jste se v navigaci?	
Tester	Hodnocení (5 nejlepší)
Student 1	4
Student 2	3
Student 3	4
Student 4	3

Tabulka 2.5: Dotazník úvodního testování - otázka 2

Pochopili jste k čemu web slouží?	
Tester	Hodnocení (5 nejlepší)
Student 1	5
Student 2	5
Student 3	5
Student 4	4

Tabulka 2.6: Dotazník úvodního testování - otázka 3

Jaké jste našli závady, chyby či nelogické věci?	
Student 1	Možná mi chýbí položka novinky, respektive mi nie je jasné v akom poradí majú byť stránky. Po přihlášení mam stránku z ktorej neviem, že nejaké novinky existujú.
Student 2	Při kliknutí na Martin Novak chybí navigace. Podmenu není moc vizuálně odlišené od hlavního menu, na první pohled mi přišlo, že to je 6 položek menu stejného charakteru akorát ve 2 řádcích.
Student 3	Prokliknutí se na informace o BOZP(novinky) přes logo je hodně nenápadné. Na této stránce nemám přístup k menu a naopak z menu nemam přístup k novinkám. Občas mi chybí na podstránce šipka zpět a musím se tedy proklikávat přes prohlížeč, třeba u stránky nahlásit závadu, u profilu nebo u novinek. Taky mi tu chybí nějaký rychlý kontakt, třeba jen e-mail na podporu nebo odkaz na kontakty.
Student 4	S navigací jsem měl trochu problémy, nejprve jsem myslel, že v menu celkově existuje 6 položek a jsou uspořádány do tabulky, pak mne trochu zarazilo, co se to stalo s menu, když jsem otevřel Úrazy a až pak mi došlo, že to jsou dvě různé úrovně menu. Možná, že je to dobrý nápad, ale mně připadal dost matoucí a nestandardní, dropdown menu mi připadá pro běžného uživatele „stravitelnější“. Navíc již výše zmíněný odkaz logo na stránku, kterou jinak ani nemohu zobrazit, místo úvodní stránky po přihlášení.

2. ANALÝZA

Tabulka 2.7: Dotazník úvodního testování - otázka 4

Co se vám nelíbilo?	
Student 1	Trochu som mal chaos v tom, čo je cvičné školenie, a čo školenie, osobne by som to asi zlúčil do jednej kategórie
Student 2	Viz předchozí.
Student 3	Asi to, že nemám přístup k rychlému kontaktu a že na stránce s novinami není menu, abych se dostala zpět ke svým informacím.
Student 4	Opakoval bych se, stránky jsem ještě jednou projel a ničeho dalšího významného jsem si nevšiml.

Tabulka 2.8: Dotazník úvodního testování - otázka 5

Co se vám líbilo?	
Student 1	Myslím, že takto by aplikácia splnila svoj účel.
Student 2	Obsah stránek byl přehledný.
Student 3	Líbilo se mi jednoduché menu s dobře zvolanými popisy, usnadňuje to orientaci na stránkách.
Student 4	Logo padajícího vykřičníku na nebohého pracovníka. Jinak mi stránky přijdou ok.

Tabulka 2.9: Dotazník úvodního testování - otázka 6

Co byste vylepšili?	
Student 1	Nic
Student 2	Přidat do navigace viditelně návrat na hlavní stránku (zjištění konzultačních hodin), víc vizuálně oddělit hlavní menu od podmenu.
Student 3	Možná nějaké aktuality k vlastnímu profilu typu, že bude potřeba znova absolvovat školení BOZP, protože bude propadat. Nebo že školení na které je uživatel přihlášen se zrušilo.
Student 4	Asi to menu a přihlašování, jinak ok.

2.2.7 Vyhodnocení

Výsledky testů ukazují na několik nedostatků rozhraní. Přesto, že rozhraní pro studenty obsahuje pouze několik funkcí, někteří studenti se v aplikaci hůře orientovali.

Hlavním problémem byla navigace. Její dvě úrovně způsobovaly zmatení testerů. Rozdělení na dvě úrovně bylo provedeno po přidání modulů přístrojů, závad a zranění, protože se položky menu nevešly do jedno-úrovňového menu. Jak ale test ukázal, dvou-úrovňové menu není ideální řešení.

Další věcí, se kterou měli testeři problém, byla hlavní stránka, kde vystavené informace zapadly. Když měli zjistit konzultační hodiny, trvalo jim to i několikanásobně déle než, byla stanovena optimální doba.

Testeři hodnotili vcelku pozitivně, ale test potvrdili některé nedostatky, na které upozornil při konzultacích BOZP pracovník. Pro BOZP pracovníky je problém obzvláště matoucí navigace, protože pro správu portálu používají daleko více funkcí. Nedostatky, kterých bych se měl při návrhu nového uživatelské rozhraní vyvarovat by mohli být shrnuty do následujících bodů:

- chybí nějaký rychlý kontakt
- špatné umístění konzultačních hodin na hlavní stránce
- nepřehledné menu, tváří se jako 6 položek, ale pak se ukáže, že jde o pod menu
- na hlavní stránku se dá dostat pouze přes logo, trochu skryté

2.3 Analýza původního řešení

Při analýze původního řešení jsem se zabýval původním návrhem, zvolenými postupy, použitými komponentami cizích stran a také aplikační logikou aplikace. Tuto analýzu jsem provedl z toho důvodu, že jsem potřeboval zjistit, jakým způsobem musím upravit původní verzi aplikace. Především jsem zjišťoval, jak ovlivní aplikaci fakt, že jsem ji měl převést z verze Nette 2.0.3 na aktuální verzi Nette 2.3.0 a zda aplikační logika odpovídá aktuálním funkčním požadavkům.

2.3.1 Původní návrh modelu

Při analýze původního kódu první věc, na kterou jsem narazil, byl nevhodný návrh aplikace. Rozdělení tříd nedává moc smysl a vypadá to, že celá aplikace je rozdělena podle zobrazovaných obrazovek. Popíši to na příkladu uživatele. Existuje až 5 modelů (*ProfileModel*, *AllModel*, *BcMgrModel*, *PhdModel* a *EmployeeModel*), které získávají data uživatelů. Každý získává data pro jednu stránku a kromě dat o uživateli poskytují i jiná data. Systém navržený tímto způsobem ztěžuje správu a další rozšíření systému.

V doporučení [25] se popisuje, že model aplikace by se měl obsahovat třídy, které reprezentují různé databázové tabulky. Tyto třídy získávají data z databáze a poskytují je ostatním třídám. V tomto případě jsou změny v logice snadno dohledatelné důsledky provedené změny.

Dalším problémem aplikace, který by měl být adresován, je použití dlouhých SQL v modelu. SQL příkazy je vhodné využít pro zvýšení efektivity, ale jinak je lepší používat nástroje, které poskytují zvolený framework, v tomto případě Nette. Aplikace sice pro volání použila knihovna dibi, která se poskytuje nějaké podpůrné prostředky pro volání SQL dotazů, ale v některých případech je využita pouze pro volání SQL dotazů, které jsou přes 100 řádku dlouhé. Problém s takovýmto použitím SQL je ten, že se špatně spravuje a často je nesrozumitelné.

Některé z těchto neduhů mohly být způsobeny Nette framework, který byl pro tvorbu aplikace použit, je ve stálém vývoji a jeho možnosti se neustále rozšiřují a tak některé možnosti, které jsou ve frameworku dostupné nyní, nemusely být dostupné tehdy. Podíl na nejednostnosti návrhu mohl být způsoben také změnou týmů, které na projektu pracovaly.

Vzhledem k nedostatku modelu aplikace jsem zhodnotil, že upravovat původní model by bylo náročnější, než navrhnout celý model znovu. A tak jsem se rozhodl pro nový návrh celého modelu s využitím aktuálních nástrojů v nejnovější verzi Nette 2.3.0.

2.3.2 Využité komponenty

V další části analýzy původního řešení jsem analyzoval využité komponenty a jejich kompatibilitu s aktuální verzí Nette. První komponentou, na kterou jsem se zaměřil, byl DataGrid.

DataGrid je doplněk, který poskytuje nástroje pro jednoduchou tvorbu tabulek s vyhledáváním a řazením. Součástí zadání bylo použít místo této komponenty Extras\Datagrid. Hlavním důvodem pro nahrazení tohoto doplňku je fakt, že přestal být podporovaný a byla zrušena i stránka s jeho dokumentací. DataGrid byl v aplikaci využit celkem často, proto si jeho náhrada vyžádala docela značné změny.

Další komponentou, která měla být nahrazena, byl doplněk pro lokalizaci aplikace NetteTranslator. Stejně jako doplněk DataGrid i NetteTranslator ztratil podporu a byl nahrazen jinými, novějšími doplňky. Všechny komponenty doporučované na stránkách frameworku Nette se používají téměř stejně. Jediným rozdílem je formát, v jaké se píší překlady. Proto jsem jako náhradu vybral Kdyby/Translation, který má přehlednou syntaxi a byl doporučený vedoucím práce.

Ostatních komponent, které nebyly zastaralé, jako třeba ajax.nette.js nebo Joseki/PDFResponse, jsem zachoval nebo použil novější verzi dané komponenty. S ohledem na to, že už jsem se rozhodl předělat od nuly model aplikace

a změny použitých komponent značně zasahovaly do prezenční vrstvy aplikace, jsem se rozhodl vytvořit i celou prezenční vrstvu znovu.

2.3.3 Systémové role

Stávající aplikace využívá systémových rolí nejen pro autorizace akcí, které může provést uživatel, ale i v logickém modulu aplikace, například pro určení, které skupiny uživatelů se daný typ školení týká. Dle mého názoru je tento postup v rozporu se správným návrhem aplikace. Důsledkem tohoto přístupu je vznik systémových rolí, které mají úplně stejná přístupová práva jako jiné systémové role.

Dále jsem narazil na role, které byly do systému přidány s tím, že v budoucnu budou potřeba pro další požadované funkcionality systému. Některé z těchto požadavků poté byly zrušeny nebo ještě nebyly naplánovány. Rozhodl jsem se tedy systémové role, které nejsou aktuálně potřeba, zrušit. Vytvoření nových uživatelských rolí totiž není implementačně složité, takže bude lepší nechat to na vývojářích, kteří budu nové role potřebovat. Výsledkem bude čistší a přehlednější návrh.

Současný systém, včetně modulů správy přístrojů, správy závad a správy úrazů, obsahuje 13 systémových rolí. Role, mezi kterými se nerozlišuje, jsem se rozhodl sloučit od nových rolí. Role osoba pověřená a osoba zodpovědná jsem se rozhodl sloučit do role BOZP pracovník. Uživatelé s rolemi děkan a tajemník jsem přiřadil k roli vedoucí pracovník. Technická skupina a studijní oddělení budou zrušeny a uživatelé s těmito rolemi dostali roli zaměstnanec. Role phd a Bc.\Mgr. jsem sloučil do role student. Výsledný seznam rolí je popsán v tabulce 2.10.

Tabulka 2.10: Nové systémové role

BOZP pracovník
Vedoucí pracovník
Zaměstnanec
Student

2.3.4 Uživatelské skupiny

Jak jsem popisoval v předchozí sekci, systémové role byly použity pro určení skupiny uživatelů příslušné typu školení. Tuto závislost jsem zrušil z toho důvodu, že systémové role jsou prakticky neměnné, při přidání nebo změně systémové role je totiž nutné upravit zdrojový kód a může nastat situace, kdy je potřeba přidat typ školení pro skupinu uživatelů, která není vymezena žádnou ze současných rolí.

Rozhodl jsem se tedy vytvořit uživatelské skupiny. Každý uživatel může být členem několika uživatelských skupin. Vazba typ školení na systémové

2. ANALÝZA

role byla zrušena a nahrazena vazbou na uživatelskou skupinu nezávisle na své systémové roli.

Tabulka 2.11: Základní uživatelské skupiny

Vedoucí pracovníci
Zaměstnanci
Doktorandi
Studenti

Návrh

Během analýzy jsem dospěl k rozhodnutí, že celá aplikace bude navržena a implementována znovu od nuly s využitím nejnovějších doplňků a doporučených postupů. Důvody pro toto rozhodnutí jsem specifikoval v analytické části práce. Především se jedná o problémy s řešením kompatibility původních doplňků, nevhodným návrhem modelu aplikace a nově navrženým uživatelským rozhraním.

Návrh jsem začal definováním případů užití. Některé případy užití zůstaly nezměněné vůči původní verzi Martina Humeníka [8], ale většina jich byla více či méně upravena. Také přibyly nové případy užití, týkající se především rozšíření integrace s fakultními systémy a přidání podpory cvičení a přednášek.

Následoval návrh nového uživatelského prostředí. Z případů užití jsem sestavil strukturu stránek a jednotlivé prvky, které se na každé stránce nacházejí. Poté jsem Nielsenovou heuristickou analýzou zhodnotil návrh, abych zajistil kvalitu rozhraní.

3.1 Případy užití

Případy užití jsou zachyceny ve vizuální a textové podobě. Diagramy případů užití poskytují rychlou představu o jednotlivých funkcích systému, ale přesné postupy, rozšiřující a alternativní scénáře musejí být zachyceny v textové formě. V jazyce UML je kodifikován pouze diagram případů užití, strukturu dokumentu s textem případů užití si musíme navrhnout sami [19]. Kvůli přehlednosti jsem rozdělil případy užití do několika skupin.

3.1.1 Aktoři

Každý aktor reprezentuje skupinu uživatelů, která má přístup k různým případům užití. Přístup aktorů k jednotlivým případům užití je definovaný v UML diagramech případu užití pomocí asociace. Někteří aktoři mají přístup k pří-

3. NÁVRH

padům užití jiných aktorů. Tato vlastnost je modelována na UML digramu případu užití pomocí dědičnosti.

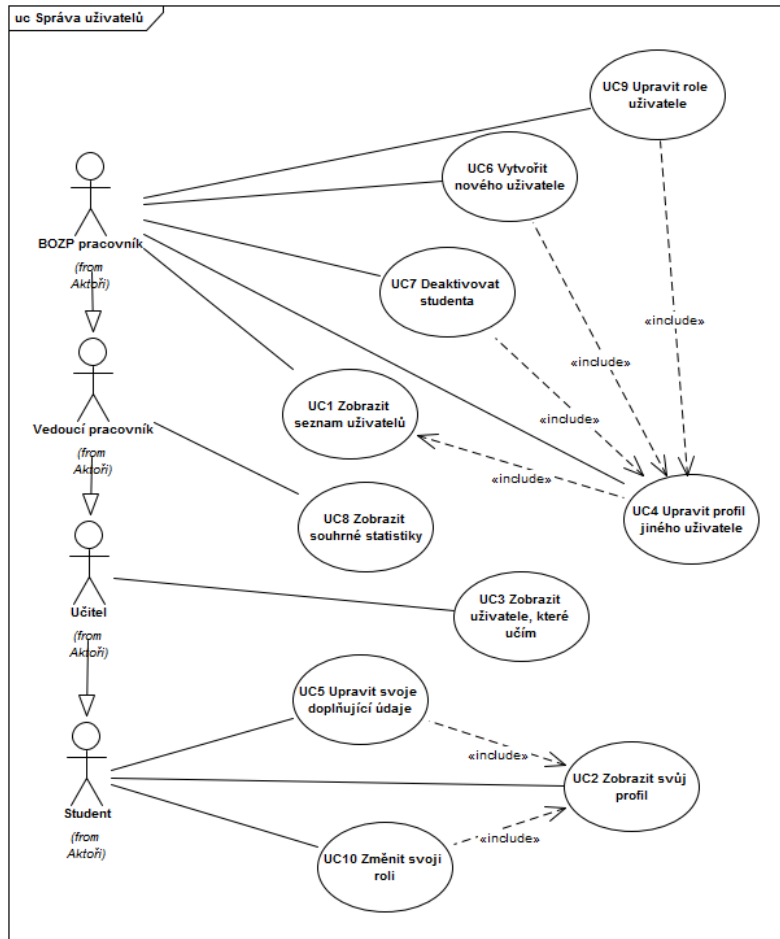
První aktorem je *BOZP pracovník*. Tento aktor zahrnuje skupinu uživatelů, kteří se starají o správu BOZP na fakultě. V novém návrhu systémových rolí odpovídá roli BOZP pracovník. Uživatel s rolí BOZP pracovník není omezen a je v hierarchii rolí nad všemi ostatními rolemi. To znamená, že bude mít přístup ke všem funkcím, ke kterým mají přístup ostatní role a navíc k funkcím spojeným se správou portálu.

Aktor *Vedoucí zaměstnanec* reprezentuje uživatele rolí vedení v novém návrhu systémových rolí. Patří sem například děkan, tajemník a vedoucí kateder. Dalším aktorem je *Učitel*. Součástí skupiny uživatelů, kterou reprezentuje tento aktor, jsou uživatelé se systémovou rolí zaměstnanec, kteří učí nebo garantují alespoň jeden předmět, ke kterému je potřeba mít zvláštní školení BOZP.

Aktor *Student* odpovídá uživatelům se systémovou rolí student. Roli student bude mít každý uživatel, který absolvuje některý ze studijních programů fakulty nebo studuje na jiné škole či fakultě a na fakultě informačních technologií pouze absolvuje nějaký předmět.

Případy užití, které se týkají funkcí, které bude provádět systém automaticky bez zásahu uživatele, budou přiřazeny aktoru *Systém*. Aktor *IS KOS* reprezentuje systém KOS, který obsahuje data o uživateli a informace, které z paralelky absolvují. Přístup k datům systému KOS bude realizován pomocí KOSapi.

3.1.2 Správa uživatelů



Obrázek 3.1: Diagram případů užití pro správu uživatelů

3.1.2.1 UC1 Zobrazit seznam uživatelů

BOZP pracovník bude moci zobrazit seznam uživatelů s možností zobrazit konec platnosti školení pro vybraný typ školení. Tento seznam bude přístupný z navigační lišty. Obrazovka bude obsahovat seznam uživatelů, ve které se zobrazí v každém řádku jméno, příjmení, osobní číslo uživatele a datum konce platnosti pro vybraný typ školení. U každého uživatele bude odkaz na jeho profil a editaci.

Bude možné filtrovat podle všech zobrazených parametrů a navíc bude dostupný rozšířený filtr, který umožní zobrazit pouze uživatele z vybrané uživatelské skupiny nebo uživatele, kteří nejsou aktivní, nebo uživatele, kteří byli přidáni ručně.

3.1.2.2 UC2 Zobrazit svůj profil

Každý přihlášený uživatel bude moci zobrazit svůj profil. Zobrazit svůj profil bude možné přímo přes navigační lištu aplikace. Uživateli se zobrazí následující informace:

- uživatelské jméno
- jméno
- příjmení
- fakultní e-mail
- osobní e-mail
- telefon
- uživatelské skupiny uživatele
- dostupné role uživatele
- aktuální platnost školení

3.1.2.3 UC3 Zobrazit uživatele, které učím

Učitel může zobrazit uživatele, kteří docházejí na cvičení nebo přednášky, které učí. Obrazovka se seznamem studentů, které učím, bude přístupná přes položku z navigační lišty. Uživateli se zobrazí hlavní filtr a tabulka uživatelů. V hlavním filtru bude možné vybrat, jaké cvičení nebo přednášku chci zobrazit. V tabulce uživatelů bude zobrazeno jméno, příjmení a personální číslo uživatele a také datum konce platnosti pro typ školení, které je potřeba pro vybrané cvičení. U každého uživatele bude odkaz na jeho profil.

3.1.2.4 UC4 Upravit profil jiného uživatele

BOZP pracovník bude moci ke každému uživateli přidávat poznámku a upravit doplňující profilové údaje. V případě, že budou vypnuty automatické aktualizace, bude moci BOZP pracovník upravovat všechny atributy uživatele. Na obrazovku s formulářem pro úpravu profilu uživatele bude vést odkaz z profilu uživatele. Pokud se uložení změn nezdařilo, zobrazí se hláška popisující nastalý problém. V opačné případě se zobrazí hláška o úspěšném uložení.

3.1.2.5 UC5 Upravit svoje doplňující údaje

Každý uživatel bude moci upravovat svoje doplňující profilové údaje. Aktualnost ostatních údajů je zaručena pomocí komunikace s jinými systémy fakulty, proto je není potřeba je měnit v této aplikaci. Změna doplňujících údajů proběhne tak, že si zobrazím svůj profil a kliknu na tlačítko upravit profil. Poté

se mi zobrazí formulář s poli pro změnu telefonního čísla a osobního e-mailu. Po požadované úpravě mohu kliknutím na tlačítko pro uložení provedených změn. Pokud se provedené změny správně uloží na server, zobrazí se hláška o úspěšné změně doplňujících údajů. Pokud se nepodaří uložit požadované změny, zobrazí se hláška popisující nastalý problém.

3.1.2.6 UC6 Vytvořit nového uživatele

Primárně se budou uživatelé a jejich data získávat ze systému KOS, ale BOZP pracovník bude mít také možnost vytvořit nového uživatele přímo v aplikaci. Přímo vytvořený uživatel bude mít vypnuté automatické aktualizace ze systému KOS. Na formulář pro vytvoření uživatele bude možné se dostat ze seznamu. Bude možné zadat následující parametry:

- uživatelské jméno
- jméno
- příjmení
- fakultní e-mail
- typ studia
- systémová role
- tituly před jménem
- tituly za jménem
- osobní e-mail
- telefon

3.1.2.7 UC7 Deaktivovat uživatele

Na profilu uživatele bude moci BOZP pracovník deaktivovat studenta. Deaktivace studenta se bude provádět v případě, že student již reálně nestuduje, ale oficiálně je stále veden jako student. Pokud je student deaktivován, zobrazí se místo tlačítka pro deaktivaci na jeho profilu tlačítko pro jeho aktivaci. Nepředpokládá se, že by v reálných případech docházelo k aktivaci dříve deaktivovaných studentů. Jedná se spíše o opatření proti chybné deaktivaci uživatele.

3.1.2.8 UC8 Zobrazit souhrnné statistiky

Souhrnné statistiky budou přístupné BOZP pracovníkům a vedoucím pracovníkům. Bude to obrazovka, na kterou bude možné se dostat z navigační lišty. Bude obsahovat tabulku pro každý typ studia a zaměstnance. Typ studia bude pro každého uživatele určen na základě studijního programu. Přehled typů studia je v tabulce 2.1. Pro každý řádek tabulky se zobrazí počet studentů, počet proškolených studentů a procento proškolených studentů pro každý typ školení.

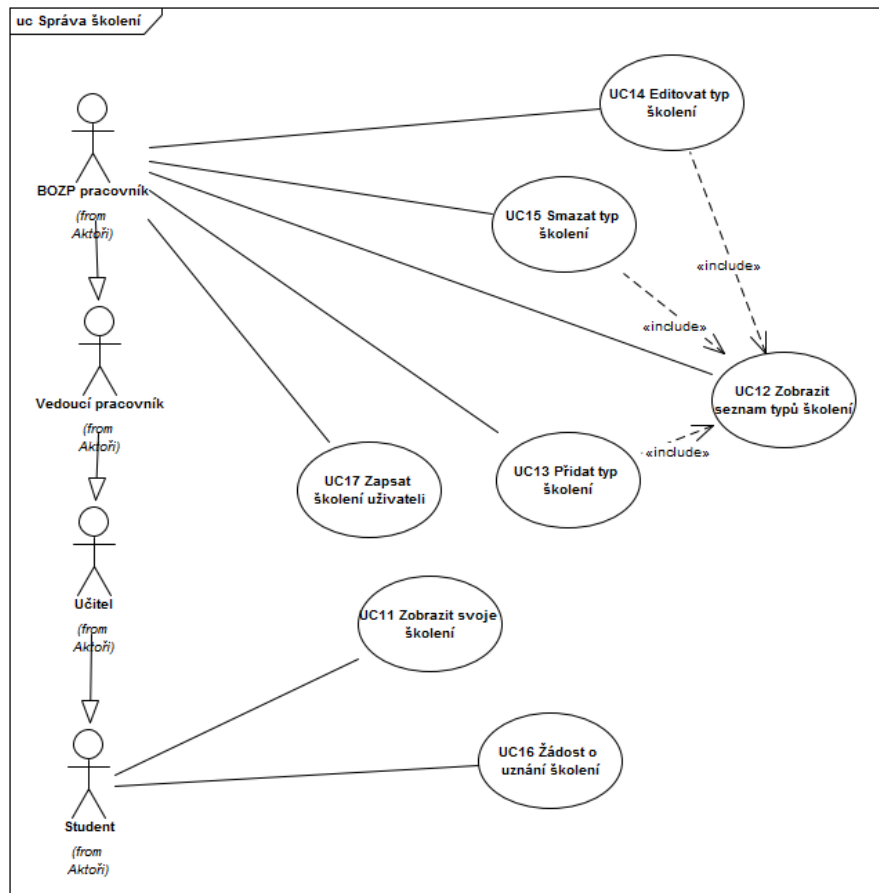
3.1.2.9 UC9 Upravit role uživatele

BOZP pracovník bude mít možnost změnit, jaké role jsou dostupné pro jiného uživatele. Z profilu uživatele bude možné zobrazit obrazovku pro správu systémových rolí. Na obrazovce se zobrazí zaškrávací pole pro každou systémovou roli a tlačítko pro uložení provedených změn. Aby uložení provedených změn proběhlo úspěšně, musí být vybrána alespoň jedna ze systémových rolí. V opačné případě se zobrazí chybová hláška.

3.1.2.10 UC10 Změnit svoji roli

Každý uživatel bude moci změnit svoji aktivní systémovou roli. Když uživatel otevře svůj profil a má dvě nebo více dostupné systémové role, zobrazí se mu tlačítko pro změnu role. Toto tlačítko vede na obrazovku, kde si uživatel bude moci pomocí zaškrávacích polí vybrat jednu z dostupných systémových rolí. Dále se zobrazí tlačítko pro uložení výběru role.

3.1.3 Správa školení



Obrázek 3.2: Diagram případů užití pro správu školení

3.1.3.1 UC11 Zobrazit svoje školení

Zobrazit všechny svoje aktuální školení bude moci každý přihlášený uživatel. Moje aktuální školení se budou zobrazovat na obrazovce Přehled. Přehled školení bude také přístupný z profilu uživatele. Moje školení se zobrazí v tabulce, která bude obsahovat název typu školení a datum platnosti. Propadlé školení a školení, které brzo vyprší, budou zvýrazněny. Při kliknutí na školení se zobrazí přehled aktuálně dostupných školení pro zvolený typ školení.

3.1.3.2 UC12 Zobrazit seznam typů školení

Seznam typu školení bude moci zobrazit pouze BOZP pracovník. Seznam typů školení bude přístupný z navigační lišty. Na obrazovce se zobrazí tabulka se

3. NÁVRH

všemi typy školení. V tabulce bude zobrazen název typu školení a skupina uživatelů, pro kterou je školení určeno.

3.1.3.3 UC13 Přidat typ školení

Přidat typ školení bude povoleno pouze správci portálu. Při zobrazení seznamu typů školení se zobrazí nad tabulkou tlačítko, které zobrazí obrazovku pro přidání typu školní. Obrazovka bude obsahovat formulář pro vytvoření typu školení a tlačítko, které uloží vytvořený typ školení. Formulář bude obsahovat textové pole pro název typu školení a výběr z možných uživatelských skupin pro určení cílové skupiny typu školení.

3.1.3.4 UC14 Editovat typ školení

BOZP pracovník bude moci upravit stávající typ školení. Když se zobrazí seznam typů školení, bude v tabulce u každého záznamu odkaz pro editaci typů školení. Při zvolení editace typů školení se zobrazí obrazovka s formulářem. Formulář bude obsahovat textové pole pro název a výběr pro skupinu uživatelů a stávajícími hodnoty předvyplněny.

3.1.3.5 UC15 Smazat typ školení

BOZP pracovník bude moci smazat typ školení. Při zobrazení seznamu typů školení bude u každého záznamu v tabulce odkaz, který umožní smazání daného typu školení. Po kliknutí na odkaz pro smazání typu školení se zobrazí dialog, který upozorní uživatele, že se chystá provést nezvratitelnou operaci a zeptá se ho, zda chce přesto pokračovat.

3.1.3.6 UC16 Žádost o uznání školení

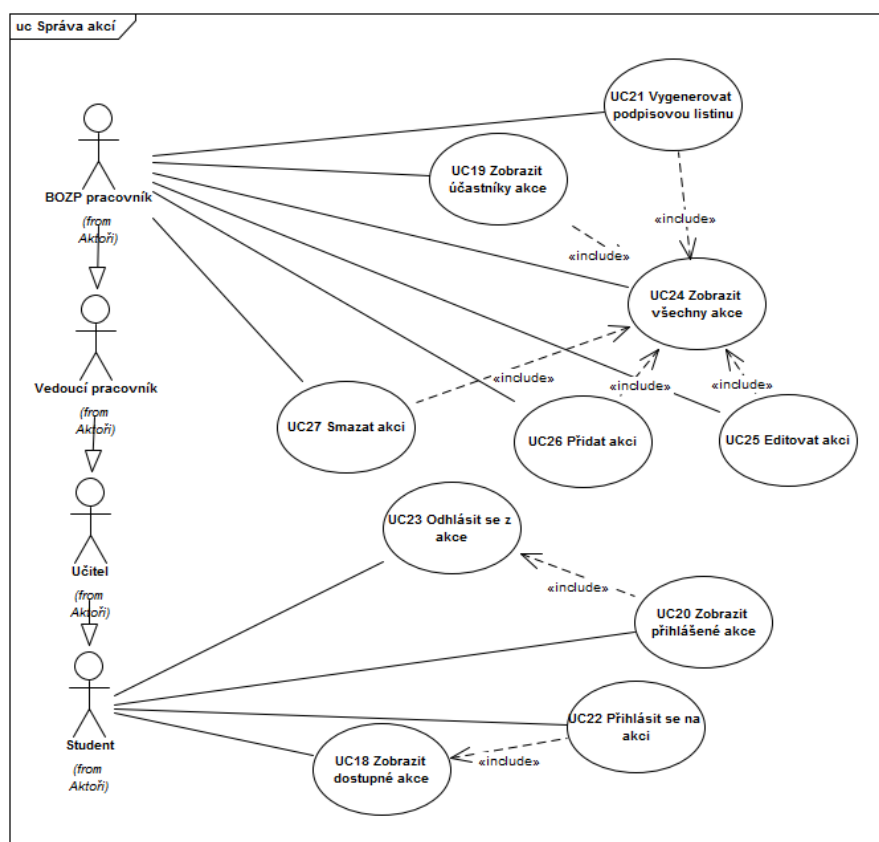
Každý uživatel bude mít možnost zažádat o uznání školení v případě, že potřebné školení získal z jiného zdroje nebo již dříve absolvoval cvičení, pro které je školení nutné. V případě, že student chce zažádat o uznání školení, zobrazí si svůj profil a klikne na tlačítko pro žádost o uznání školení. Poté se zobrazí formulář s textovým polem pro text žádosti a pole pro nahrání souboru. Nahraný soubor musí být oskenovaný dokument, který prokazuje, že nemusí požadované školení absolvovat (např. oskenovaný certifikát o proškolení nebo index se započteným předmětem).

3.1.3.7 UC17 Zapsat školení uživateli

BOZP pracovník bude moci udělovat školení ostatním uživatelům. Školení bude ve většině případu udělováno na základě absolvovaných akcí pro jeden z typů školení. V tomto případě bude školení uděleno přes obrazovku detailu akce, kde vybere BOZP pracovník z přihlášených účastníků uživatele, kteří se

na školení dostavily a udělí jim zápočet. V některých případech bude nutné, aby BOZP pracovník udělil na základě podané žádosti o uznání školení. V takovém případě se nová žádost zobrazí na přehledu BOZP pracovníka. Při kliknutí na vybranou žádost se zobrazí její detail s textovou žádostí a dokumentem, který dokazuje oprávněnost žádosti. BOZP pracovník může přidat poznámku a žádost zamítnout nebo schválit.

3.1.4 Správa akcí



Obrázek 3.3: Diagram případů užití pro správu akcí

3.1.4.1 UC18 Zobrazit dostupné akce

Každý uživatel bude moci zobrazit všechny akce, které jsou vypsané pro některou ze skupin uživatelů, do které je zařazen. Zobrazit dostupné akce bude možné z navigační lišty. Při příchodu na obrazovku se zobrazí uživateli tabulka s následujícími parametry:

- název akce
- typ akce
- typ školení (pokud je k akci přiřazen)
- datum a čas konání
- počet volných míst

Uživatelé budou mít možnost filtrovat podle názvu akce, typu školení, typu akce. Filtrování typ akce a typu školení se bude provádět výběrem z několika možností.

3.1.4.2 UC19 Zobrazit účastníky akce

BOZP pracovník bude mít možnost zobrazit účastníky vybrané akce. Akci, jejíž účastníky chce BOZP pracovník zobrazit, vybere v seznamu akcí a zvolí možnost zobrazit detail akce. Poté se mu zobrazí obrazovka s detaily akce a s tabulkou uživatelů, kteří se na akci přihlásili. Detaily akce budou obsahovat typ školení, pokud je typ školení přiřazen, typ akce, místo konání a uzávěrku odhlášení. Každý záznam v tabulce uživatelů bude obsahovat jméno, příjmení a osobní číslo.

3.1.4.3 UC20 Zobrazit přihlášené akce

Zobrazit všechny aktuální akce, na které jsem přihlášen, bude moci každý přihlášený uživatel. Moje aktuální akce se budou zobrazovat na obrazovce Přehled. Kliknutím na titulek bude možné přejít na všechny dostupné akce. V případě kliknutí na jednu z přihlášených akcí se zobrazí její detail. Každé políčko seznamu mých akcí bude obsahovat typ akce, typ školení, pokud je přiřazen, místo konání a čas konání.

3.1.4.4 UC21 Vygenerovat podpisovou listinu

BOZP pracovník bude moci vygenerovat podpisovou listinu uživatelů. Generování listiny bude možné provést na výpisu uživatelů nebo na výpisu účastníku akce zobrazeném na obrazovce detail akce. Data uživatelů vybraných pomocí zaškrtávacích polí budou naformátována podle připravené struktury do PDF dokumentu, který bude výstupem této akce.

3.1.4.5 UC22 Přihlásit se na akci

Každý uživatel se bude moci přihlásit na jakoukoliv akci, která se mu zobrazí v seznamu dostupných akcí. V seznamu dostupných akcí bude u každé akce odkaz, kterým se může uživatel přihlásit. Pokud se uživatel přihlásí na vybranou akci a přihlášení proběhne v pořádku, zobrazí se hláška o úspěšném přihlášení na akci. Pokud nastane nějaký problém a přihlášení na akci selže, zobrazí se hláška popisující problém (např. akce je již plně obsazena).

3.1.4.6 UC23 Odhlásit se z akce

Stejně jako se bude moci každý uživatel přihlásit na akci, může se i každý uživatel odhlásit z akce. Odhlášení bude možné provést pomocí seznamu mých akcí. U každého záznamu o akci bude odkaz pro odhlášení akce, pokud již není uzavřeno přihlašování a odhlašování z akce.

3.1.4.7 UC24 Zobrazit všechny akce

BOZP pracovník bude mít možnost zobrazit všechny akce. Obrazovka se všemi akcemi půjde zobrazit přímo z navigační lišty. Bude obsahovat tabulku, ve které se zobrazí akce s těmito parametry:

- název akce
- typ akce
- typ školení
- místo konání
- datum a čas konání
- počet volných míst

3.1.4.8 UC25 Editovat akci

BOZP pracovník bude mít možnost editovat všechny akce, které ještě neproběhly. V případě, že jsou už na akci přihlášení nějakí uživatelé zobrazí se varování, že uživatelé by měli být informováni o změně akce.

3.1.4.9 UC26 Přidat akci

BOZP pracovník bude mít možnost přidat novou akci. Na obrazovce se všemi akcemi se bude vyskytovat tlačítko pro vytvoření nové akce. Po stisknutí tlačítka se zobrazí formulář s následujícími poli:

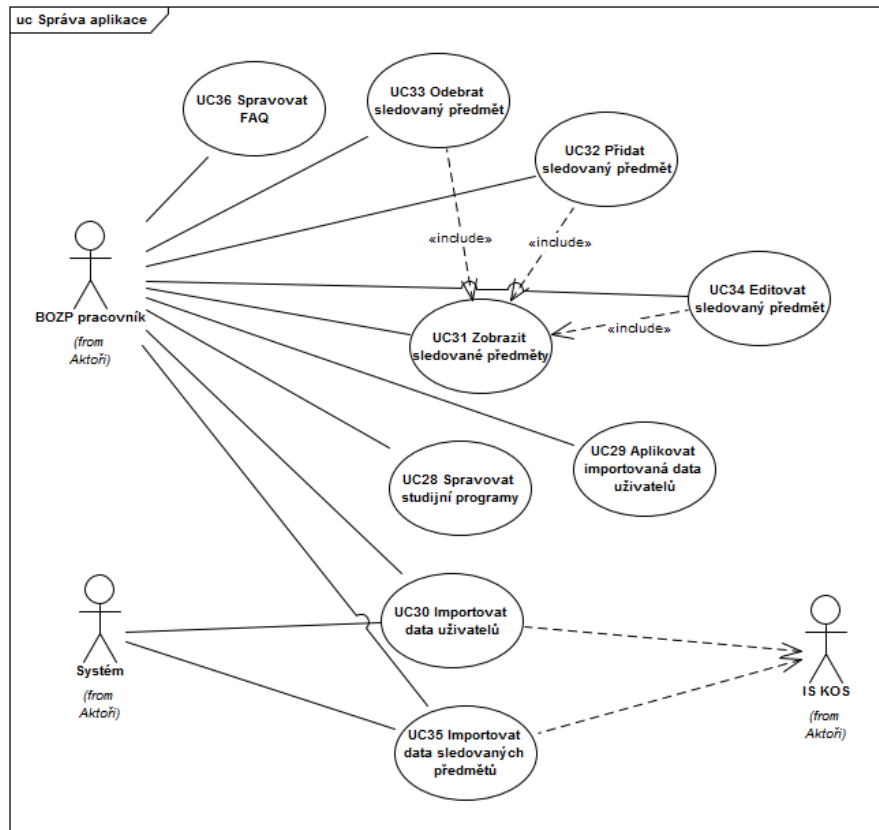
- název akce
- typ akce
- typ školení
- uzávěrka pro přihlašování
- doba konání akce
- místnost
- kapacita
- školitel
- poznámka (nepovinné)

Pokud se povede vytvořit novou akci, zobrazí se hláška o úspěšném vytvoření a uživatel je přesměrován na výpis všech akcí. Pokud nastane nějaký problém a nepodaří se akci uložit, zobrazí se chybová hláška.

3.1.4.10 UC27 Smazat akci

BOZP pracovník bude moci smazat každou akci, která ještě neproběhla. Akce půjde smazat na obrazovce pro zobrazení všech akcí. U každého záznamu bude odkaz pro smazání dané akce. Při kliknutí se zobrazí potvrzovací dialog. Pokud je smazání potvrzeno, zobrazí se hláška o úspěchu.

3.1.5 Správa aplikace



Obrázek 3.4: Diagram případů užití pro správu aplikace

3.1.5.1 UC28 Spravovat studijní programy

Přes položku menu nastavení se bude moci BOZP pracovník dostat na správu studijních programů. Zde bude možné nastavit, jaké studijní programy z KOSu jsou přiřazeny k jednotlivým typům studia. Toto nastavení ovlivní jaký typ studia se přiřadí uživatelů při aktualizaci z KOSu.

3.1.5.2 UC29 Aplikovat importovaná data uživatelů

Na obrazovce Přehled se uživateli zobrazí počet uživatelů se změněnými daty a odkaz na stránku s možností aktualizace dat uživatelů. Když vše proběhne v pořádku, zobrazí se hláška o úspěšném provedení. Pokud nastane problém, žádná změna neproběhne a zobrazí se hláška popisující nastalou chybu.

3.1.5.3 UC30 Importovat data uživatelů

O importování nových dat uživatelů se bude starat systém. Každý den stáhne data z jiných fakultních systémů a připraví je na manuální import prováděný BOZP pracovníkem.

3.1.5.4 UC31 Zobrazit sledované předměty

BOZP pracovník bude mít možnost zobrazit seznam předmětů, které jsou sledované pro účely BOZP, tedy mají požadavek na speciální školení. Obrazovka se seznamem sledovaných předmětů bude přístupná přes navigační lištu. Každý předmět bude reprezentován kódem předmětu.

3.1.5.5 UC32 Přidat sledovaný předmět

BOZP pracovník bude moci přidat předmět do seznamu sledovaných předmětů. Na obrazovce se seznamem sledovaných předmětů bude tlačítko, které zobrazí formulář s textovým polem pro kód předmětu a zaškrtačací pole pro každý typ školení.

3.1.5.6 UC33 Odebrat sledovaný předmět

BOZP pracovník bude mít možnost smazat předmět ze seznamu sledovaných předmětů. U každého předmětu na seznamu sledovaných předmětů bude tlačítko pro smazání daného záznamu. Při kliknutí na tlačítko pro smazání předmětu se zobrazí potvrzovací dialog.

3.1.5.7 UC34 Editovat sledovaný předmět

BOZP pracovník bude moci upravit kód nebo potřebné typy školení pro předmět na seznamu sledovaných předmětů. Každý záznam na seznamu sledovaných předmětů bude mít odkaz, který zobrazí formulář s textovým polem pro kód předmětu a zaškrtačací pole pro každý typ školení. Současné hodnoty budou předvyplněny.

3.1.5.8 UC35 Importovat data sledovaných předmětů

O importování nových dat předmětů se bude starat systém. Sledované předměty se budou aktualizovat jednou za semestr a nahradí data z minulých semestrů. Kromě automatické aktualizace bude moci BOZP pracovník provést ruční import nových předmětů. Ruční import předmětů bude BOZP pracovník provádět při změně sledovaných předmětů.

3.1.5.9 UC36 Spravovat FAQ

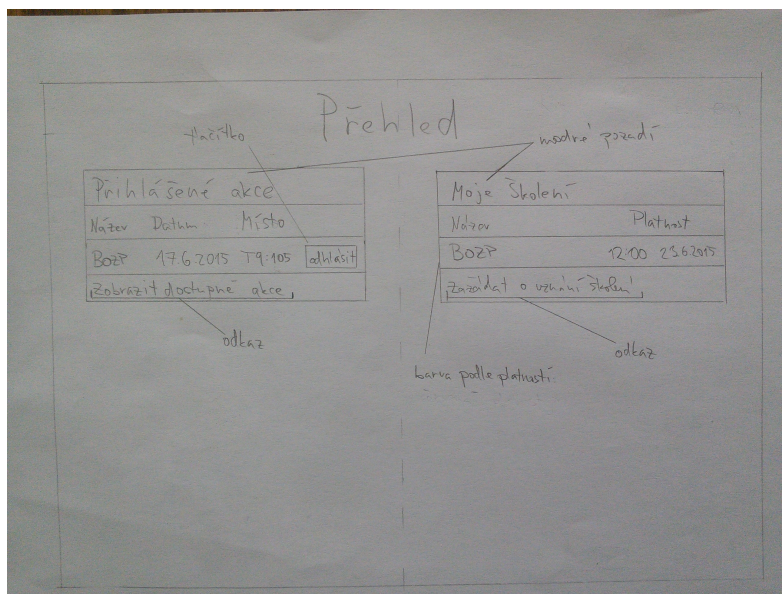
BOZP pracovníkovi se zobrazí na obrazovce FAQ tlačítko pro přidání nové otázky a odpovědi. Tlačítko bude vést na formulář, kde lze zadat jazyk, otázku

a odpověď. U každé otázky navíc bude odkaz na editaci a smazání. Při zvolení editace bude uživatel přesměrován na stejný formulář jako při vytváření, ale hodnoty budou předvyplněny. Při zvolení smazání se daná otázka a odpověď smaže.

3.2 Uživatelské rozhraní

3.2.1 Návrh rozhraní

S návrhem jsem začínal od nuly. Začal jsem definováním všech obrazovek, které bude aplikace obsahovat. Všechny definované obrazovky jsou popsány v tabulce B.1. V druhém kroku jsem vymyslel, jaké prvky bude každá obrazovka obsahovat. V této fázi jsem vytvářel návrh lo-fi prototypu pomocí tužky a papíru. Výsledkem byly obrazovky s hrubě rozmístěnými ovládací prvky. Ukázkou návrhu obrazovky Přehled si můžete prohlédnout na obrázku 3.5. Další ukázkou můžete nalézt v příloze B.1.



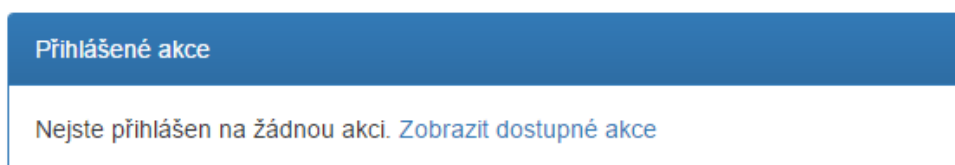
Obrázek 3.5: Návrh obrazovky Přehled

Třetím krokem bylo vytvoření hi-fi prototypu. Prototyp jsem vytvářel přímo v Nette, abych si usnadnil práci s převodem prototypů do aplikace. Výsledkem hotového hi-fi prototypu byla funkční kostra aplikace, do které stačilo přidat pouze aplikační logiku.

Rozhodl jsem se využít frameworku Bootstrap 4.1.7, který již byl použit v původním projektu pro některé části, ve větším rozsahu. Využil jsem styly Bootstrapu pro layout aplikace i základní grafické prvky jako jsou tabulky, formulářové prvky, seznamy, a další.

3. NÁVRH

Použil jsem navigační lištu z Bootstrapu, která vypadá daleko přehledněji než navigační lišta z původní aplikace. Také jsem využil komponentu *panel 3.6* pro zobrazení přihlášených akcí, uživatelských údajů a informací o školení.



Obrázek 3.6: Ukázka komponenty panel

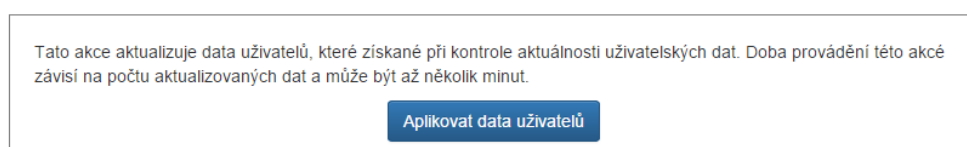
V příloze práce naleznete ukázky hotového návrhu obrazovky Přehled B.2 a obrazovky Moji studenti B.3.

3.2.2 Nielsenova heuristická analýza

Mnou navržené rozhraní jsem podrobil heuristické analýze. Heuristická analýza patří mezi nejužívanější metody testování použitelnosti interaktivních rozhraní. Tato metoda spočívá v odhalování chyb a slabých míst v rozhraní za pomoci porovnávání jeho současného stavu s pravidly (heuristikami), která jsou předem daná. Tato pravidla jsou obvykle sestavena na základě výzkumů, předchozích testování a několikaletých zkušeností. Toto testování bývá prováděno jedním a více odborníky. Nejznámější heuristická pravidla je deset bodů Jakoba Nielsena [18], kterou jsem použil na navržené rozhraní.

3.2.2.1 Viditelnost stavu systému

Doba zpracování většiny akcí je velice krátká a navíc se při jejich provádění zobrazí v prohlížeči informace o načítání nové stránky. Jediné akce s delší dobou zpracování se týkají aktualizace dat z kosu. U těchto akcí jsme v poznámce popsal, že doba jejich zpracování může být delší než je běžné. Lze to vidět na ukázce 3.7.



Obrázek 3.7: Ukázka poznámky u akce s dlouhých trváním

3.2.2.2 Propojení systému a reálného světa

Zvolené ikony reprezentují očekávané akce. Například při správě FAQ jsem u každé otázky zobrazil ozubené kolečko pro úpravu a přeškrtnuté červené kolečko pro smazání otázky, viz 3.8.

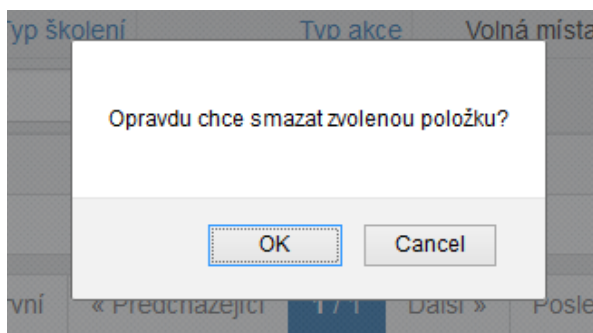
Jsem nový student magisterské etapy. Jak získám zápočet z předmětu BOZP? ⚙️ ❌

Zápočet se v magisterské etapě neuděluje, ale musíte absolvovat školení BOZP pro první ročníky a řádně vyplnit a podepsat prezenční listiny.

Obrázek 3.8: Ukázka FAQ

3.2.2.3 Uživatelská kontrola a svoboda

Všechny akce, které provádějí smazání nějakého záznamu, vyžadují potvrzení vyskakovacího okna prohlížeče, viz 3.9. U ostatních akcí poskytuje aplikace možnost provést opačnou akci. Příkladem takových akcí je přihlášení a odhlášení na akci.



Obrázek 3.9: Ukázka dotazu na potvrzení akce

3.2.2.4 Standardizace a konzistence

Tento bod byl zajištěn především použitím frameworku Bootstrapu, který poskytuje zaběhnutý vzhled pro grafické prvky. Do tohoto bodu také patří barevné rozdělení tlačítek. Běžná tlačítka jsou vyobrazena modrou barvou a tlačítka s červenou barvou reprezentují nebezpečné akce, jako je mazání.

3.2.2.5 Prevence chyb

Prevence chyb je řešena hlavně při zadávání uživatelského vstup. Každá povinná položka formuláře je označena hvězdičkou. Pro zadávání data a času je použita speciální komponenta. U méně jasných formulářů je zobrazena krátká nápověda.

3. NÁVRH

3.2.2.6 Rozpoznání namísto vzpomínání

Každý stránka obsahuje přehledný nadpis, takže uživatel vždy ví, kde se nachází. Menu je fixně zobrazené v horní části obrazovky a je vidět v každém okamžiku.

3.2.2.7 Flexibilní a efektivní použití

Běžní uživatele používají pouze pár funkcí a mají jednoduché rozhraní. Při návrhu rozhraní jsem se tedy nemusel flexibilitou a efektivností moc zabývat. Naopak při návrhu rozhraní pro BOZP pracovníky jsem udělal pár rozhodnutí, které mě nebyli ideální, což se projevilo při uživatelském testování.

3.2.2.8 Estetický a minimalistický

Navržené rozhraní je velice jednoduché a neobsahuje složité ovládací prvky. Na obrazovkách lze vidět všechny ovládací prvky bez nutnosti posunout stránku. Výjimku tvoří některé delší formuláře.

3.2.2.9 Pomoc uživatelů pochopit, poznat a vzpamatovat se z chyb

Chybové hlášky se zobrazují především, když uživatel zadává vstup do formuláře. Při chybě se zobrazí vyskakovací okno s popisem chyby a focus je nastaven na pole s chybou. Chybové hlášky jsou formulované v běžném jazyce a neobsahují chybové kódy.

3.2.2.10 Náповěda a návody

U některých akcí, u kterých by uživatel mohl mít problém s pochopením správného postupu, jsem zobrazil krátkou nápovědu. Taková nápověda je například při zadávání žádosti na uznání školení, viz 3.10.

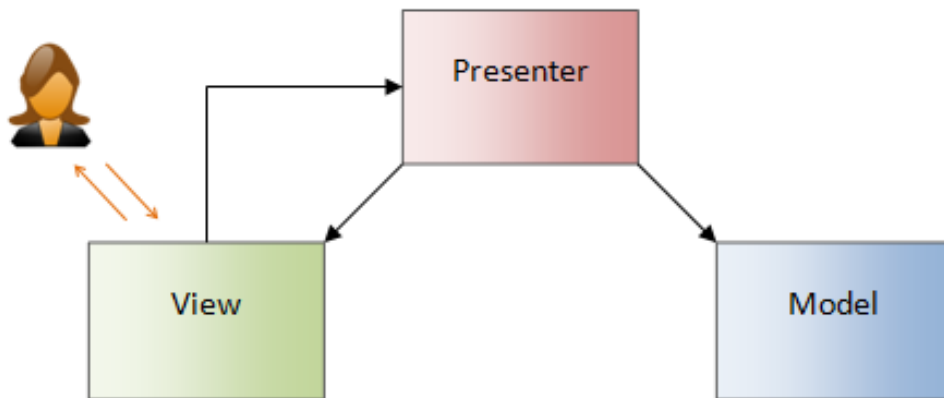
Pokud máte nárok na uznání školení na základě dřívějšího absolvování předmětu či nějakého certifikátu, popište jaké školení byste chtěli uznat a přiložte oskenovaný dokument, který dokazuje váš nárok.

Text žádost *

Obrázek 3.10: Ukázka nápovědy u formuláře

3.3 Architektura aplikace

Framework Nette využívá návrhového vzoru návrhového vzoru MVP, což je varianta více rozšířeného MVC. V tomto vzoru je aplikace rozdělena do tří částí. Model, Presenter a View jak napovídá název. Model poskytuje data a aplikační logiku pro ostatní části. Presenter obsahuje aplikační a prezentační logiku. Manipuluje s Modelem, což pomocí systému notifikací zajistí aktualizaci View, nebo ovlivňuje View přímo. View zobrazuje data poskytnutá Preseterem a deleguje uživatelské akce na Presenter [2].



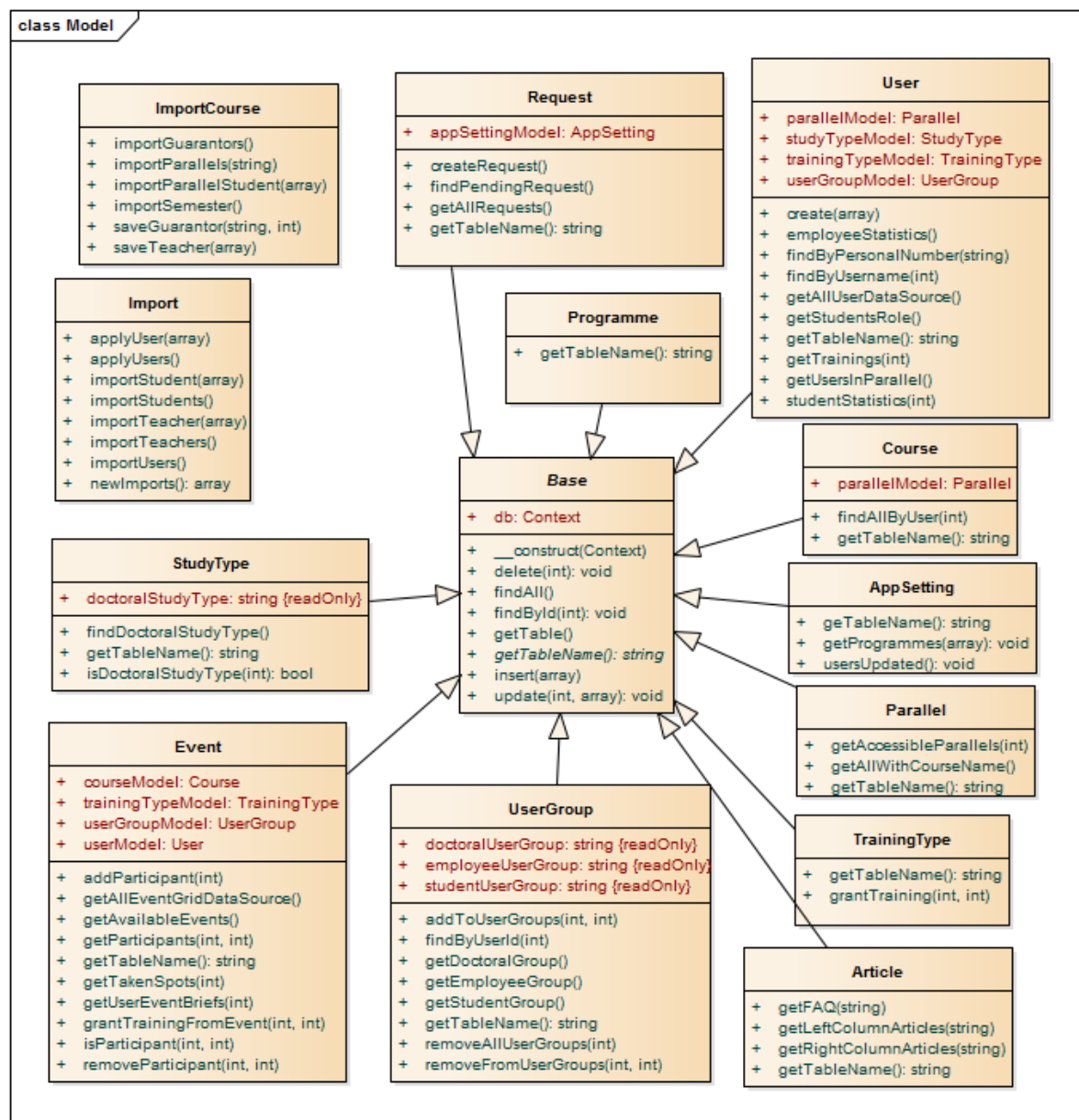
Obrázek 3.11: Diagram MVP [2]

3.4 Návrhový model tříd

Při návrhu nového modelu jsem se inspiroval doporučením z článku [25]. Výsledný návrh modelu 3.12 byl upraven pro potřeby aplikace a využití NDBT místo notORM. V mé implementaci existuje abstraktní třída *Base*, která je poskytuje svým potomkům základní CRUD operace. Potomci této třídy musí pouze poskytnout název tabulky, kterou třída reprezentuje, v metodě *getTableName*.

Výhodou toho návrhu je to, že na rozdíl od původního návrhu modelu aplikace, jednotlivé třídy reprezentují jednotlivé tabulky v databázovém modelu. Lze tedy poměrně snadno přidávat jednoduché třídy pro nové tabulky se základními CRUD funkcemi a změny jsou snáze dohledatelné.

3. NÁVRH



Obrázek 3.12: Diagram tříd modelu aplikace

Realizace

Po dokončení návrhu jsem mohl začít implementovat navržené řešení. Realizace probíhala tak, že jsem nejdříve upravil databázi podle navrženého schématu. Po úpravě databáze jsem začal implementovat model aplikace podle návrhu. Poté jsem pouze propojil logiku s hi-fi prototypem, který jsem vytvořil v rámci návrhu uživatelského rozhraní.

Pro realizaci výsledného řešení jsem použil několik technologií a knihoven třetích stran. Důvody pro jejich použití a zároveň jejich popis jsem zpracoval v následující sekci. Po popisu využitých technologií a knihoven popisují, jaké nástroje jsem při vývoji využil. Dále jsem představil implementační detaily, které si zasloužily bližší pozornost. Nakonec jsem popsal kroky potřebné k nasazení do provozu.

4.1 Technologie a knihovny

4.1.1 PHP

PHP je objektově orientovaný skriptovací jazyk, který je využíván pro tvorbu webových aplikací. Skripty běží na straně serveru, který na základě požadavků klienta vrací dynamicky generované odpovědi [22]. Použití PHP bylo součástí požadavku N1.

4.1.2 Nette

Nette framework byl zvolen v požadavku N1. Je to framework, který využívá moderních technologií ke snadné tvorbě webových aplikací v PHP [9]. Obsahuje všechny části, které by měl moderní framework pro tvorbu webů obsahovat. Poskytuje dokonalé zabezpečení, ladící nástroje, aktivní komunitu s řadou rozšíření a pracuje s moderními webovými technologiemi.

Od verze Nette 2.0.3, která byla použita v původní aplikaci, prošel framework nezanedbatelným vývojem, proto byl rozhodnuto o přechodu na no-

vější verzi. Tato verze obsahuje vylepšený databázový přístup pomocí NDBT, který umožňuje větší odstup od SQL a tím pádem poskytuje lepší modifikovatelnost. Zrovna tato vlastnost byla požadována v požadavku N4 a zároveň byla jedním z problémů původní verze portálu.

4.1.2.1 Kdyby/**Translation**

Tento doplněk poskytuje robustní překladový systém. Obsahuje implementaci rozhraní Nette\Localization\ITranslator. Tato implementace využívá pro definování překladů přehledné soubory `.neon`, které jsou Nette využity pro konfigurační soubory. Soubor, kterým jsou překlady definovány, může pak vypadat následovně [15].

menu:

```
login: 'Přihlášení'  
dashboard: 'Přehled'  
courses: 'Sledované předměty'  
training: 'Školení'  
profile: 'Profil'  
logout: 'Odhlásit'  
events: 'Správa akcí'
```

4.1.2.2 Nextras**Datagrid**

Datagrid je komponenta, která poskytuje tabulkové zobrazení, filtrování a řazení pro strukturovaná data. Hlavním cílem Nextras\DataGrid je poskytnout použitelné API pro vytváření silný Datagrid. Také poskytuje snadné formátování pro buněk Datagridu [11].

4.1.2.3 nette.ajax.js

Tato knihovna poskytuje podporu ajaxového volání. Po instalaci knihovny stačí přidat odkazu css třídu `ajax` a tato knihovna se postará, aby se provedlo ajaxové volání [3].

4.1.2.4 Joseki/**PdfResponse**

Doplněk pro Nette, který poskytuje jednoduché řešení generace PDF dokumentů. V pozadí využívá knihovnu mPDF a obaluje ji přehlednějším API [14].

4.1.2.5 Nextras**Forms**

Tento doplněk byl využit proto, že obsahuje prvek pro výběr data a času. Kromě toho obsahuje další užitečné formulářové prvky, které jsem ale v aplikaci nevyužil [11].

4.1.2.6 Nextras\SecuredLinks

Poskytuje ochranu proti CSRF pro odkazy. Vyniká především v tom, že použití je velice jednoduché. Do komentáře před každý odkaz, který chceme chránit před útoky, stačí přidat anotaci @secured [11].

4.1.3 MySQL

MySQL je open-source relační databáze. Patří mezi nejrozšířenější databázové stroje. Mezi její výhody patří rychlost a jednoduchost. Použití MySQL bylo součástí požadavku N2 [12].

4.1.4 MySQL Dump Utility

Tento nástroj slouží pro zálohování databáze nebo pro převod databáze na jiný server. Jak název napovídá, je možné tuto knihovnu použít pouze pro MySQL databáze. Generovaný soubor typicky obsahuje příkazy k vytvoření tabulky, naplnění tabulky nebo obojí [4].

4.1.5 Shibboleth

Shibboleth je open-source projekt, který poskytuje SSO (Single Sign-On) možnosti. Tato technologie se skládá ze dvou částí SP (Service Provider) a IdP (Identity Provider). IdP zajišťuje přihlášení uživatele a předává data o přihlášeném uživateli SP. SP poskytuje data ostatním aplikacím na serveru [17].

4.1.6 KOSapi

KOSapi poskytuje aplikační rozhraní (API) v podobě RESTful webových služeb, které zprostředkovává přístup k vybrané části dat v databázi KOS. Odstraňuje nutnost zpracovávání exportů, neustálou duplikaci všech dat a potíže s jejich udržováním. RESTové služby staví na osvědčených konceptech webu jakožto distribuovaného prostředí vzájemně provázaných informací.

KOSapi umožňuje a podporuje vznik školních i studentských aplikací, které pro svou činnost vyžadují online aplikační přístup k datům souvisejícím s výukou. Tvoří také jednu ze základních komponent nově vznikajícího Informačního systému ČVUT (IS ČVUT) [7].

4.1.7 Bootstrap

Bootstrap je rozšířený HTML, CSS a javascriptový framework, který podporuje responzivní návrh aplikace. Součástí bootstrapu jsou obecné styly pro základní HTML elementy a responzivní layout, přepoužitelné komponenty a javascriptové pluginy pro jQuery [13].

4.2 Nástroje

4.2.1 Git

Git je open-source distribuovaný verzovací systém. To, že je Git distribuovaný, znamená, že každý uživatel má svoji vlastní kopii repozitáře a synchronizace probíhá přes centrální server. Oproti nedistribuovaným systémům se tedy může jakákoliv lokální kopie použít jako záloha v případě poškození centrálního repozitáře.

Další výhodou je fakt, že téměř všechny akce, kromě samotné synchronizace s centrálním repozitářem, probíhají lokálně a jsou rychlejší. Také poskytuje záruku, že každý bit, který byl do Gitu nahrán, bude nezměněný pomocí nahraných dat. A v neposlední řadě poskytuje možnost připravit si každý commit do tzv. staging area [23].

4.2.2 Composer

Composer je nástroj pro správu PHP balíčku (knihoven). Jeho vznik byl inspirován správcem balíčku NPM pro javascriptový server node a bundelerem pro ruby. Composer umožňuje jednoduchou instalaci knihoven třetích stran bez problémů s dalšími závislostmi [1].

4.2.3 PHPStrom

Integrované vývojové prostředí, neboli IDE, pro PHP projekty s řadou pomocných nástrojů. Kromě nezbytného textového editoru nabízí navíc řadu dalších funkcí, které poskytují ostatní používané IDE. Mezi jeho funkce patří automatické našeptávání při psaní kódu, podpora refaktorování kódu, podpora PHPDoc, integrace verzovacích systémů, zabudovaná podpora Composeru a další užitečné funkce [5].

4.2.4 Adminer

Adminer je plně vybavený nástroj pro správu databáze napsaný v PHP. Celý nástroj je obsažen v jediném souboru, který je připraven pro nahrání na cílový server. Adminer je k dispozici pro MySQL, PostgreSQL, SQLite, MS SQL, Oracle, Firebird, SimpleDB, Elasticsearch a MongoDB [24].

4.3 Implementační detaily

4.3.1 Aktualizace dat z fakultních systémů

K aktualizaci jsem použil rozhraní KOSapi 4.1.6, které jako jediné poskytuje potřebná data. Usermap API, které poskytuje data v přehlednější podobě

bohužel neposkytuje rozvrhová data studentů a učitelů, která jsou potřeba pro získání účastníků sledovaných předmětů.

Aktualizaci jsem rozdělil do tří akcí. První akce, kontrola nových dat, získá nová data uživatelů přes KOSapi a připraví je do tabulky *imported_user*. Tato akce kontroluje pouze data změněná od poslední aktualizace a bude probíhat automaticky každý den. Pro tento účel jsem využil spojení unixového nástroje Cron a CLI routeru, který poskytuje Nette. Kromě automatických aktualizací může BOZP pracovník také provést tuto akci manuálně.

Druhá akce, aplikování nových dat, aktualizuje data uživatelů podle předpřipravených dat z tabulky *imported_user*. Tuto akci bude provádět manuálně BOZP pracovník portálu. Před každým provedením této akce se udělá záloha databáze. Aktuální počet uživatelů, kterým se změnila nějaká data, se zobrazuje na obrazovce *Přehled*.

Třetí, poslední akce, se týká aktualizace dat sledovaných předmětů. Tato akce bude probíhat automaticky jednou za semestr. BOZP pracovník může provést aktualizaci i manuálně. V případě, že bude přidán nový sledovaný předmět, musí BOZP pracovník provést manuální aktualizaci, aby se načetla data nově přidaného předmětu.

4.3.2 Zabezpečení aplikace

Nette framework poskytuje automatické zabezpečení proti různým druhům útoků. Útoky, proti kterým je Nette chráněno, jsou XSS, CSRF, útoky spojené se správou session a další útoky, které se snaží podstrčit webové stránce špatný vstup (URL attack, control codes, invalid UTF-8) [10].

XSS neboli Cross-Site Scripting využívá pro narušení stránky neošetření vstupů. Útočník může například získat pomocí vnořeného scriptu data o ostatních návštěvnících webu. Nette využívá technologii *Context-Aware Escaping*, která ošetřuje všechny uživatelské vstupy automaticky.

Cross-Site Request Forgery (CSRF) útok jiná stránka zaútočí na webovou aplikaci, ve které je uživatel přihlášen. Nette poskytuje ochranu formulářů proti CSRF jedním příkazem, který generuje autorizační token pro každý ošetřený formulář. Příkaz pro ošetření formuláře *\$form* vypadá následovně:

```
$form->addProtection();
```

Navíc jsem se rozhodl využít doplňku Nextras\securedLinks 4.1.2.6, který poskytuje zabezpečení proti CSRF pomocí autorizačních tokenů i pro jednoduché odkazy.

Mezi útoky spojené se správou session patří Session hijacking, session stealing a session fixation. Obecně útok spočívá v tom, že uživatel zcizí nebo podstrčí session ID a poté vystupuje na webu jako napadený uživatel, který se o tom vůbec nedozví. Obrana spočívá ve správné konfiguraci PHP, kterou zajišťuje Nette automaticky, a serveru.

4.4 Instalace

4.4.1 Migrace databáze

Přestože jsem vytvářel aplikaci od nuly, musel jsem zachovat stávající data. Proto jsem už při návrhu databázového modelu začínal s původní databází a pro každou změnu jsem generoval sadu změnových skriptů neboli changesetů.

Při nasazování na testovací prostředí pak stačilo zduplikovat produkční databázi a aplikovat vytvořené changesety. Po úpravě databáze jsem ještě musel aktualizovat data uživatelů pomocí importu z KOSapi, protože změny databáze zasáhly tabulku *user*. Pak jsem mohl provést testy na skutečných datech. Při migraci produkční databáze bude provedena migrace stejným způsobem.

4.4.2 Nasazení na produkční prostředí

Nasazení na produkční prostředí jsem neprováděl, protože se aplikace přesunula na zabezpečený server a já jsem nedostal potřebné bezpečnostní oprávnění. Místo toho jsem nasadil aplikace na server *Mantichora*, na kterém běžela aplikace předtím.

Nejdříve jsem nahrál zdrojové kódy na server do předpřipravené složky. Nahrání zdrojových kódů na server bylo možné provést dvěma způsoby. K prvnímu je potřeba mít zdrojové kódy se všemi potřebnými knihovnamí, které jsou na CD přiloženém k této práci. Poté je stačí překopírovat pomocí SCP protokolu na připravený server.

Druhý způsob je mít nainstalovaný na serveru Git 4.2.1, Composer 4.2.2 a práva k projektu na GitLabu, fakultním správcí Git repozitářů. Stažení zdrojových kódů lze provést pomocí Gitu. K tomu je použit příkaz:

```
git clone https://gitlab.fit.cvut.cz/faltakam/bozp3.git .
```

Poté se musí provést stažení závislostí. To se provede pomocí composeru příkazem:

```
composer install
```

V případě použití druhého způsobu lze pak jednoduše aktualizovat aplikaci při úpravě zdrojového kódu pomocí dvojice příkazů:

```
git pull origin master  
composer install
```

První příkaz aktualizuje zdrojové kódy aplikace a druhý závislosti.

Dalším rokem k nasazení aplikace je nastavení Apache HTTP Serveru. V tomto případě stačí přidat konfiguraci pro nový VirtualHost se správnými parametry.

Pokud například aplikace bude umístěna ve složce `/var/www/bozp`, může nastavení vypadat následovně:

```
<VirtualHost *>
  DocumentRoot /var/www/bozp/www
  ServerName bozp.fit.cvut.cz

  <Directory /var/www/bozp/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>
</VirtualHost>
```

Po nastavení HTTP serveru se ještě musí upravit parametry konfigurace aplikace. Konfigurační soubor `config.neon` se nachází ve složce `app|config`. Parametry, které je třeba upravit, jsou následující:

- `database.host` - adresa databázového serveru
- `database.scheme` - název databáze
- `database.username` - uživatelské jméno
- `database.password` - heslo

4.4.3 Zprovoznění Shibbolethu

Aplikace je připravena na použití Shibbolethu, ale bohužel jsem nemohl před odevzdáním práce zprovoznit autentizaci pomocí Shibbolethu, protože mi chyběla potřebná bezpečnostní oprávnění. Pro otestování, že je aplikace připravena pro Shibboleth, jsem si nainstaloval testovací verzi Shibbolethu na lokální server, který jsem využíval pro vývoj.

Aplikace bude využívat Shibboleth IdP od federace `cvutID`. Při instalaci a konfiguraci Shibboleth SP na produkčním server se bude postupovat podle návodu [21].

4.4.3.1 Instalace

V návodu [21] je doporučeno použít předpřipravenou instalaci od švýcarské federace SWITCH. Na jejich stránkách lze nalézt návod [20], který popisuje postup instalace pro různé systémy. V dalších odstavcích popisují návod instalace pro Debian 6.x (squeeze).

4. REALIZACE

Před instalací Shibboleth SP je nutné zajistit, že jsou nainstalované programy NTP, Sudo a Curl. Dále je nutné povolit SSL modul pro Apache. Nakonec se ještě musí nakonfigurovat další zdroj pro APT, protože *The Shibboleth project* poskytuje oficiální balíčky pouze pro linuxové distribuce s RPM. To lze provést následujícími příkazy:

```
sudo curl -k -O http://pkg.switch.ch/switchaai/SWITCHaai-swdistrib.asc
```

```
gpg --with-fingerprint SWITCHaai-swdistrib.asc
```

```
sudo apt-key add SWITCHaai-swdistrib.asc
```

```
echo 'deb http://pkg.switch.ch/switchaai/debian squeeze main' |  
sudo tee /etc/apt/sources.list.d/SWITCHaai-swdistrib.list >  
/dev/null
```

```
sudo apt-get update
```

Poté už lze provést instalace pomocí příkazu:

```
sudo apt-get install shibboleth
```

4.4.3.2 Konfigurace

Nejdříve je nutné získat serverový certifikát a klíč. Obojí zajistí vedoucí práce. Získané klíče umístíme do */etc/ssl*. Poté je potřeba nastavit Apache serveru s modulem SSL. Pokud předpokládáme, že je aplikace umístěna v */var/www/bozp*, musíme v souboru *etc/apache2/sites-enabled/default-ssl* udělat následující změny:

```
Redirect seeother /shibboleth  
    https://bozp.fit.cvut.cz/Shibboleth.sso/Metadata  
  
SSLCertificateFile /etc/ssl/certs/bozp.cvut.cz.pem.crt  
SSLCertificateKeyFile /etc/ssl/private/bozp.cvut.cz.pem.key  
SSLCertificateChainFile /etc/ssl/certs/tcs-ca-bundle.pem  
  
<Location /bozp>  
  
    AuthType shibboleth  
    ShibRequireSession On  
    require valid-user  
  
</Location>
```

Pro dokončení konfigurace Apache serveru musíme ještě zapnout modul Shibboleth. To provedeme příkazem:

```
a2enmod shib
```

Dalším krokem je konfigurace Shibboleth SP, která se nachází v souboru */etc/shibboleth/shibboleth2.xml*. Musíme nastavit následující údaje:

- jméno serveru/aplikace *entityID*
- session - nastaveno na 28800 sekund (8 hodin)
- timeout - nastaveno na 3600 sekund (1 hodinu)
- poskytovatel identity (IdP) pro cvutID
- zdroj pro stahování metadat cvutID
- cesty k OpenSSL certifikátům

Výsledek může vypadat následovně:

```
<ApplicationDefaults entityID="https://bozp.fit.cvut.cz/shibboleth"
    REMOTE_USER="eppn persistent-id targeted-id">

<Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
    checkAddress="false" handlerSSL="true"
    cookieProps="; path=/; secure">

<SSO entityID="https://idp2.civ.cvut.cz/idp/shibboleth"
    discoveryProtocol="SAMLDS"
    discoveryURL="https://ds.example.org/DS/WAYF">
    SAML2 SAML1
</SSO>

<MetadataProvider type="XML"
    uri="https://idp2.civ.cvut.cz/cvutid/cvutid-metadata.xml"
    backingFilePath="/etc/shibboleth/cvutid-metadata.xml"
    reloadInterval="3600">
</MetadataProvider>

<CredentialResolver type="File"
    key="/etc/ssl/private/bozp.cvut.cz.pem.key"
    certificate="/etc/ssl/certs/bozp.cvut.cz.pem.crt"/>
```

Na závěr je nutné stáhnout aktuální soubor *attribute-map.xml* z <https://gingo.civ.cvut.cz/secure/attribute-map.xml> a nahradit jím původní soubor */etc/shibboleth/attribute-map.xml*.

Testování

Testování je důležitou součástí vývoje aplikace. Přímo v zadání práce bylo specifikováno, že aplikace má být podrobena automatickým i uživatelským testům. Oba typy testů slouží k jinému účelu.

Automatické testy především slouží k nalezení implementační chyb. Pro tento účel existuje spousta různých nástrojů. Já zvolil Selenium, které odpovídalo mým požadavkům na testovací nástroj.

Na druhou stranu uživatelské testy slouží pro otestování uživatelského rozhraní. Při testech se zjistí, jaké úkoly či grafické prvky dělají uživatelům problémy. Tyto testy jsem provedl po dokončení aplikace a na základě výsledků jsem aplikaci ještě dodatečně doladil.

5.1 Selenium

Selenium je sada softwarových nástrojů, které slouží pro automatizaci internetového prohlížeče. Já jsem pro testování využil nástroj Selenium IDE. Selenium IDE je doplněk pro prohlížeč Firefox, který umožňuje jednoduše vytvářet a provádět sadu automatických testů [16].

Při vytváření testů jsem postupoval tak, že jsem si připravil výchozí databázi, zapnul jsem nahrávání v Selenium IDE a začal jsem provádět navržené testovací scénáře. Po dokončení všech scénářů jsem prošel kód skriptů vygenerovaný při nahrávání, zkontroloval jsem, zda se provádějí požadované kroky, a přidal jsem kontrolu požadovaných výsledků. Všechny vytvořené testy je možné nalézt na příloženém CD.

5.2 Uživatelské testování

Závěrečné uživatelské testování se bude skládat ze tří částí. Každá část bude reprezentovat skupinu uživatelů, která má jiná přístupová práva, tudíž jiný uživatelský zážitek. První část jsou úkoly pro uživatele se studentskou rolí,

druhá část obsahuje úkoly pro zaměstnance a třetí část testu se bude zabývat úkoly BOZP pracovníků.

5.2.1 Účel testů

Účel závěrečného uživatelského testování bylo zjistit, zda je navržené rozhraní přehledné a zda poskytuje všechny potřebné funkce ke správě BOZP. Uživatelské testování je nezbytné přesto, že jsem rozhraní podrobil heuristické analýze. Nieslenovo desatero mi sice pomohlo předejít obecným nedostatkům, kterými uživatelské rozhraní trpí, ale nemohu objektivně posoudit použitelnost rozhraní, které jsem sám navrhl.

Na základě výsledků testů bylo navrženo, jak řešit objevené problémy. V případě, že se jednalo o drobné změny nebo kritické problémy, upravil jsem je po provedení testů. Pokud by vyvstaly návrhy na úpravy složitějšího rázu, bude jejich zapracování přesunuto do dalších prací, které se budou zabývat BOZP portálem.

5.2.2 Profil uživatele

Zajistit relevantní uživatele pro uživatelské testování je důležitá součást celého testovacího procesu. Musí se získat vlastnosti průměrných budoucích uživatelů a na základě těchto vlastností zvolit testery.

Mně se podařilo sehnat pro test studentské části studenty fakulty, tedy přímo uživatele z cílové skupiny, a tím zvýšit relevanci výsledků. Pro ostatní části se mi také podařilo zajistit testery přímo z cílové skupiny uživatelů. Pro test BOZP pracovníků se mi dokonce podařilo otestovat všechny budoucí uživatele systému, kteří budou mít příslušnou roli.

5.2.3 Metoda testování

Pro závěrečné testování se mi podařilo zajistit specializovanou laboratoř pro provádění uživatelského testování (CHUL), ve které jsem mohl provést důkladnější test uživatelské rozhraní než při úvodním testování.

Při každém testu byli testeři nahráváni ze tří úhlů pomocí kamer a navíc se zaznamenával děj na obrazovce. Celkově jsem tedy měl po testu 4 různé typy záznamů, ze kterých jsem mohl posuzovat reakce testera. Každý typ záznamu poskytoval jiné informace o uživatelské zkušenosti s rozhraním.

Záznam obrazovky poskytoval aktuální testerovy akce, jak pohybuje myší, kam kliká apod. První kamera také zabírala děj na obrazovce. Druhá kamera byla umístěna na obrazovce počítače, který byl použit pro účely testování. Účel této kamery byl sledování pohybu očí. Poslední kamera byla umístěna po pravé ruce testera a zabírala spolu s obrazovkou i pohyb rukou po klávesnici.

Test jsem začal jednoduchým vysvětlením účelu aplikace. Ve většině případů nebyl s pochopením problém, protože testeři v minulosti již používali

původním verzi aplikace. Samotný test probíhal tak, že jsem já dělal moderátora a postupně diktoval testerovi úkoly, které má provádět.

Jelikož jsem dělal moderátora, nestačil jsem si dělat poznámky při průběhu testu a závěry jsem vyvozoval ze zaznamenaných nahrávek. V ideálním případě by se testů účastnil ještě někdo jiný, kdo by dělal poznámky přímo při testu, ale vzhledem k tomu, že jsem k vyhodnocení měl dostupné čtyři typy záznamů, to nebylo nezbytné.

5.2.4 Úkoly

5.2.4.1 Studentská část

1. Přihlásit se jako student
2. Zjistit, zda mám platné „Školení pro studenty“
3. Přihlásit se na akci „Školení pro studenty“
4. Zažádat o uznání školení „Vyhláška 50“
5. Zadat osobní e-mail
6. Zjistit konzultační hodiny
7. Odhlásit se

5.2.4.2 Část pro zaměstnance

1. Přihlásit se jako zaměstnanec
2. Zjistit, zda mám platné „Školení pro zaměstnance“
3. Zažádat o uznání školení „Vyhláška 50“
4. Zjistit, zda studenti ve cvičení „MI-SAP 10 (Laboratoř)“ mají splněna potřebná školení
5. Zadat osobní e-mail
6. Odhlásit se

5.2.4.3 Část pro BOZP pracovníky

1. Přihlásit se jako BOZP pracovník
2. Schválit požadavek na uznání školení uživatele „Kamil Falta“
3. Zjistit, kolik procent studentů s typem studia „bakalářské prezenční“ má splněno školení „Školení pro studenty“

5. TESTOVÁNÍ

4. Přidejte do FAQ otázku a odpověď
5. Vytvořit nový typ školení „Školení pro vedení“
6. Vytvořit akci pro vytvořený typ školení
7. Rozšířit kapacitu vytvořené akce
8. Zobrazit detail akce „Školení pro studenty“
9. Vygenerovat prezentační listu akce „Školení pro studenty“
10. Udělit školení všem studentům z akce „Školení pro studenty“
11. Odhlásit se

5.2.5 Průběh testů

Po provedení testu jsem analyzoval nahrané záznamy z testů jeden po druhém. Ze záznamů jsem se snažil zjistit, jaké úkoly dělaly uživatelům problémy a z jakého důvodu. Zda se jednalo o problém s navigací, rozmístěním ovládacích prvků nebo cokoliv jiného. Záznamy z testů lze nalézt na přiloženém CD.

5.2.5.1 Student 1

Přihlášení do systému bylo bez problémů. Stejně tak při druhém úkolu, zjištění stavu svého školení, jsem nezaznamenal žádný problém a tester bez prodlevy zjistil požadovanou informaci o svém školení.

Třetí úkol, přihlášení na akci, proběhl také v pořádku. První zádrhel nastal při podávání žádosti na uznání školení. Uživatel nemohl najít odkaz vedoucí na formulář pro zadání žádosti. Po nalezení formuláře už tester dokončil úkol bez problémů. Pátý úkol, úprava profilu, proběhla hladce v optimálním čase. Při šestém úkolu, zjištění konzultačních hodin, našel požadovanou informaci dostatečně rychle.

5.2.5.2 Student 2

Tester splnil téměř všechny úkoly v optimálním čase a s optimálním počtem kliknutí. Nejvíce se zdržel při čtvrtém úkolu, podávání žádosti o uznání školení, ale zdržení bylo způsobeno samotným vypisováním žádosti. Také se při zjišťování konzultačních hodin nejdříve podíval na obrazovku *Přehled* místo na obrazovku *Aktuality*. Jinak ale všechny úkoly proběhly bez problému a velice rychle.

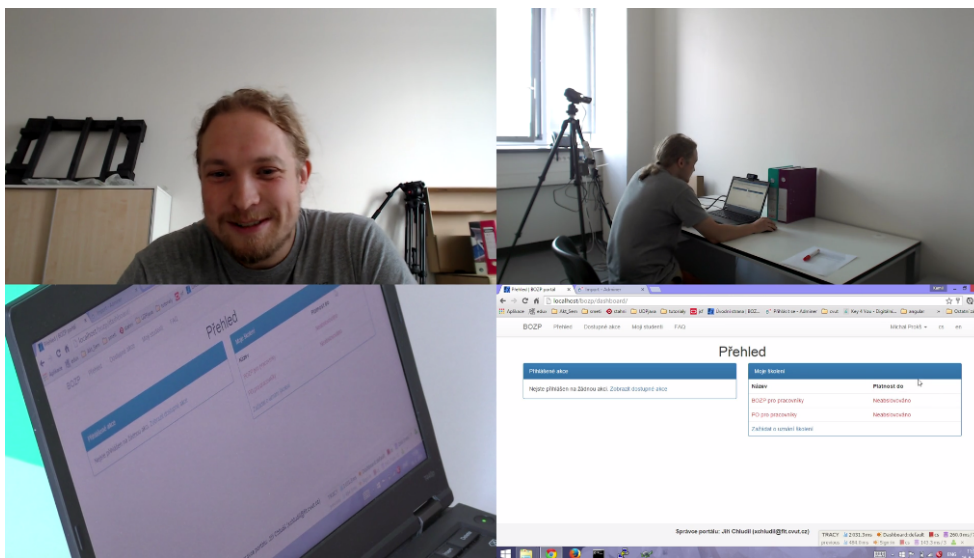
5.2.5.3 Zaměstnanec 1

Přihlášení bylo bezproblémové. Při zjišťování stavu svého školení si tester zprvu nevšiml stavu školení a chvíli tápal, ale v rozumném čase informaci zjistil. Formulář pro žádost o uznání školení našel tester ihned. Samotné vyplnění žádosti proběhlo v pořádku a dokonce při vyplňování tester použil nápovědu zobrazenou nad formulářem.

Čtvrtý úkol, zadání osobního e-mailu, proběhl bez problému. Při pátém úkolu, kdy měl zjistit kolik studentů nemá potřebné školení pro daný předmět, si tester nevšiml filtru a seznamu uživatelů po stránkách.

5.2.5.4 Zaměstnanec 2

Tester splnil všechny úkoly velice rychle s minimálním počtem kliknutí. U žádném úkolu nezaváhal a na každé stránce se zorientoval velice rychle. Zde je ukázka z tohoto uživatelského testování 5.1.



Obrázek 5.1: Ukázka 1 z uživatelského testování

5.2.5.5 Zaměstnanec 3

První a druhý úkol zvládl tester bez problému. Při třetím úkolu nastal problém pouze kvůli jazyku klávesnice, což je ale pro účel testu irelevantní. U čtvrtého úkolu se pozastavil nad tím, že nevyplněný osobní e-mail se nezobrazuje, ale během chvíle se znovu zorientoval a podařilo se mu e-mail zadat. Zbytek testu proběhl bez problému.

5.2.5.6 BOZP pracovník 1

Tester zvládl přihlášení bez problému. I druhý úkolu, schválení požadavku na uznání školení, tester zvládl bez zaváhání. Při třetím úkolu, zjištění kolik student má splněno školení, tester nemohl najít stránku se statistikami. Přidání nové otázky a odpovědi do FAQ proběhlo bez problému.

Pátý úkol, vytvoření nového typu školení, proběhl bez problému. U šestého úkolu, vytvoření nové akce, měl tester trochu problém s vyplněním data a času. Úkoly sedm, osm a devět tester zvládl bez problému. Při desátém úkolu, udělování školení studentů na akci, testerovi chyběl způsob, jak udělit zápočet hromadně.

5.2.5.7 BOZP pracovník 2

Přihlášení a schválení požadavku proběhlo bez problémů. Při třetím úkolu tester zobrazil několik špatných obrazovek, než se dostal na stránku se statistikami studentů.

Zadání nové otázky a odpovědi do FAQ proběhlo bez problému. Stejně tak vytvoření nového typu školení zvládl tester bez problému. Šestý úkol provedl tester bez většího zádrhelu, ale nevyhovovalo mu zadávání data a času. Zbytek úkolů proběhl bez problémů. Ukázkou ze záznamu tohoto testu je možné nalézt v příloze B.4.

5.2.6 Dotazník

Tabulka 5.1: Dotazník závěrečného testování - otázka 1

Přišla Vám navigace přehledná?	
Tester	Odpověď
Student 1	Spíše ano
Student 2	Spíše ano
Zaměstnanec 1	Spíše ano
Zaměstnanec 2	Ano
Zaměstnanec 3	Spíše ano
BOZP pracovník 1	Ano
BOZP pracovník 2	Spíše ano

Tabulka 5.2: Dotazník závěrečného testování - otázka 2

Myslíte, že aplikace splňuje svůj účel?	
Tester	Odpověď
Student 1	Ano
Student 2	Ano
Zaměstnanec 1	Ano
Zaměstnanec 2	Ano
Zaměstnanec 3	Nevím
BOZP pracovník 1	Ano
BOZP pracovník 2	Ano

Tabulka 5.3: Dotazník závěrečného testování - otázka 3

Přidali byste nějakou funkci?	
Tester	Odpověď
Student 1	Ne
Student 2	Ne
Zaměstnanec 1	Žádnou
Zaměstnanec 2	Ne
Zaměstnanec 3	Ne
BOZP pracovník 1	Hromadné akce a automatické doplnění nevyplněného času na 0:00 nebo jinou rozumnou hodnotu.
BOZP pracovník 2	Hromadné akce pro udělování zápočtu

Tabulka 5.4: Dotazník závěrečného testování - otázka 4

Co se Vám líbilo?	
Tester	Odpověď
Student 1	Design
Student 2	Vzhled
Zaměstnanec 1	Jednoduchost
Zaměstnanec 2	Logicky sestavené menu, vše běhalo
Zaměstnanec 3	Nevím
BOZP pracovník 1	Grafické rozhraní a rychlost aplikace.
BOZP pracovník 2	Bootstrap, nextras datagrid a že to funguje.

5. TESTOVÁNÍ

Tabulka 5.5: Dotazník závěrečného testování - otázka 5

Co se Vám nelíbilo?	
Tester	Odpověď
Student 1	Nic
Student 2	Nic
Zaměstnanec 1	Chtěl bych souvislý seznam vypsanych studentů
Zaměstnanec 2	Nekorespondoval název filtru a vyfiltrované položky
Zaměstnanec 3	Při výběru cvičení nebyla nastavena výchozí paralelka a že se jsou na profilu nevyplněné údaje skryté
BOZP pracovník 1	Nic
BOZP pracovník 2	Výběr data a času, nezaokrouhlená procenta na statistikách, moc široké formuláře, moc vysoké formuláře, u formulářů nejsou pro číselné hodnoty použity prvky pro zadávání čísel, ale textové pole

5.2.7 Vyhodnocení

Vyhodnocení závěrečného uživatelského testování jsem provedl na základě průběhu testů a dotazníku. Zaměřím se především na problémy, na které uživatelé při testech narazili, zda jsem je řešil a případně jak jsem je vyřešil. Testy studentské a zaměstnanecké části dopadly celkem dobře.

Při testu studentské části se projeví problémy pouze při podávání žádosti a při zjištění konzultačních hodin. První tester v prvním případě nemohl chvíli najít odkaz vedoucí na formulář s podáním žádosti. Druhý tester zaváhal při zjišťování konzultačních hodin, ale hned na podruhé zobrazil správnou obrazovku. Tyto situace jsem se rozhodl neřešit. Hlavní důvodem pro toto rozhodnutí byl fakt, že se tyto situace vyskytly ojedinele a podařilo se jim je vyřešit bez nápovědy. Dále také oba testeři nenapsali do dotazníku žádné připomínky a aplikace jim přišla celkem přehledná.

Testeři z řad zaměstnanců splnili každý z úkolů v dobrém čase. Během testování se neobjevily žádné problémy. Testeři pouze měli pár připomínek k některým prvkům uživatelského rozhraní.

Výtky, které testeři zaměstnanecké části napsali do dotazníků, byly více méně otázkou osobního názoru, jako v případě různých názvu filtru a hodnoty položky na obrazovce Moji studenti. V tomto případě totiž sehrála roli náhoda a při zvolení hodnoty *Neplatné* na filtru platnosti školení se vyfiltrovali pouze uživatelé, kteří školení neabsolvovali a žádní studenti, kterým školení již propadlo. Souvislý seznam, který navrhoval jeden z testerů, není zase podle mého názoru vhodný kvůli přehlednosti. Kromě těchto drobností byli s aplikací celkově spokojeni.

Další tester vytýkal, že jsou na profilu nevyplněné údaje skryté. Také se mu nelíbilo, že na obrazovce Moji studenti nebyla zobrazena výchozí paralelka.

Obě výtky jsem považoval za opodstatněné a přidal jsem opravu v dodatečných úpravách aplikace.

Test části BOZP pracovníky dopadl o něco hůře, ale to se dalo očekávat. BOZP pracovníci mají totiž daleko složitější rozhraní. Oba testeři narazili na stejné problémy. Prvním z problému bylo umístění odkazu na stránku se statistikami v podmenu. Problém jsem vyřešil v dodatečných úpravách aplikace a přesunul odkaz do hlavního menu.

Jako další nedostatek se projevil prvek pro výběr data a času, který oba testeři dokonce zmínili v dotazníku. Dal jsem jim za pravdu a v dodatečných úpravách jsem prvek pro výběr data a času vyměnil za jiný. Oba testeři si také postěžovali, že udělování zápočtu uživatelů na akci není možné provést hromadně. Znova jsem uznal požadavek za rozumný a v dodatečné úpravě aplikace jsem přidal možnost provést zápočet hromadně.

Jeden z testerů měl ještě další poznámky k některým formulářovým prvkům grafického rozhraní. Doporučil použití speciální formulářového prvku pro číselné hodnoty místo prostého textového pole. Také kritizoval šířku a výšku formulářových polí. V neposlední řadě navrhoval zaokrouhlení procent na dvě desetinná místa na stránce se statistikami. Rozhodl jsem se provést všechny požadované změny. Navíc jsou to změny docela přímočaré a nenáročné. V případě příliš vysokých formulářů využiji zúžení a naskládám prvky do dvou sloupců.

V dodatečné úpravě aplikace jsem tedy na základě uživatelských testů provedl následující opravy:

- Zobrazení nevyplněných položek na profilu
- Nastavení výchozí paralelky na obrazovce
- Přidání statistik do hlavního menu
- Změna prvku pro výběr data a času
- Podpora hromadného udělení zápočtu
- Použití číselných formulářových vstupů pro číselné hodnoty
- Zúžení formulářových políček
- Přerovnání dlouhých formulářů do dvou sloupců
- Zaokrouhlení statistik na dvě desetinná místa

Závěr

Podle zadání jsem nejprve provedl analýzu původního řešení správy školení, identifikoval jsem nedostatky a získal aktuální požadavky od pracovníka BOZP. Požadavky jsem získal od vedoucího práce, který je pracovníkem BOZP a je správcem portálu BOZP. Zanalyzoval jsem původní aplikaci a zjistil nedostatky. Také jsem využil uživatelské testování, abych objevil další nedostatky.

Při analýze jsem zjistil, že bude jednodušší aplikaci vytvořit znovu od nuly než, abych se snažil zapracovat nové požadavky do původní aplikace. Definoval jsem tedy nové případy užití. Na základě aktuální případů užití jsem navrhl nové uživatelské rozhraní. Nakonec jsem navrhl třídní model aplikační logiky.

Po dokončení návrhu jsem se vrhl na implementaci. Použité technologie jsem popsal v kapitole Realizace. Po dokončení implementace jsem aplikaci nasadil na testovací prostředí. Nasazení na produkční prostředí jsem neprováděl, protože jsem nedostal potřebné bezpečnostní oprávnění, ale budoucího nasazení se budu účastnit.

Hotovou aplikaci otestoval automatickými testy a provedl jsem i závěrečné uživatelské testování. Uživatelské testování dopadlo, až na drobné nedostatky, které byly upraveny v závěrečné úpravě aplikace, docela dobře. Výsledkem práce tak byla aplikace připravená k nasazení do produkčního prostředí.

Možnosti dalšího rozvoje

Portál BOZP se bude vyvíjet i v budoucnu. Už při tvorbě této práce byla v běhu další práce, která integruje portál BOZP s e-learningovou platformou Moodle. Kromě této práce je určitě možné, že se objeví další funkce, které by mohly ulehčit práci pracovníků BOZP při plnění jejich povinností.

Literatura

- [1] ADERMANN, Nils, Jordi BOGGIANO a další. *Composer* [online]. [cit. 2015-06-10]. Dostupné z: <<https://getcomposer.org>>.
- [2] BERNARD, Borek. *Prezentační vzory z rodiny MVC* [online]. 2009 [cit. 2015-05-19]. Dostupné z: <<http://www.zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc>>.
- [3] DOBEŠ, Vojtěch. *Nette.ajax.js* [online]. [cit. 2015-06-10]. Dostupné z: <<http://addons.nette.org/vojtech-dobes/nette-ajax-js>>.
- [4] GRUDL, David. *MySQL Dump Utility* [online]. [cit. 2015-06-10]. Dostupné z: <<https://github.com/dg/MySQL-dump>>
- [5] JETBRAINS S.R.O. *PHP Storm Features* [online]. 2000 [cit. 2015-06-10]. Dostupné z: <<https://www.jetbrains.com/phpstorm/features>>.
- [6] JANEČEK, Lukáš, Filip VOREL, Marin NÁHLOVSKÝ a Kamil FALTA. *Portál BOZP - semestrální práce z MI-NUR*. Praha, 2014.
- [7] JIRŮTKA, Jakub. Projekt KOSapi [online]. [cit. 2015-06-10]. Dostupné z: <<https://kosapi.fit.cvut.cz/projects/kosapi/wiki>>.
- [8] HUMENÍK, Martin. *BOZP portál - správa školení*. Bakalářská práce, ČVUT v Praze, Fakulta informačních technologií. 2012.
- [9] NETTE FOUNDATION. *Rychlý a pohodlný vývoj webových aplikací v PHP | Nette Framework* [online]. 2008, [cit. 2015-06-10]. Dostupné z: <<http://nette.org/>>.
- [10] NETTE FOUNDATION. Zabezpečení před zranitelnostmi [online]. [cit. 2015-06-10]. Dostupné z: <<http://doc.nette.org/cs/2.3/vulnerability-protection>>.

- [11] NEXTRAS COMMUNITY. *Nextras components* [online]. [cit. 2015-06-10]. Dostupné z: <<http://nextras.cz>>.
- [12] ORACLE CORPORATION. *MySQL :: The world's most popular open source database* [online]. [cit. 2015-06-10]. Dostupné z: <<https://www.mysql.com>>.
- [13] OTTO, Mark a Jacob THORNTON. *Bootstrap* [online]. [cit. 2015-06-10]. Dostupné z: <<http://getbootstrap.com>>.
- [14] PAULÍK, Miroslav. *Joseki/PdfResponse* [online]. [cit. 2015-06-10]. Dostupné z: <<http://addons.nette.org/joseki/pdf-response>>.
- [15] PROCHÁZKA, Filip. *Kdyby/Translation* [online]. [cit. 2015-06-10]. Dostupné z: <<http://addons.nette.org/kdyby/translation>>.
- [16] SELENIUM PROJECT. *Selenium - Web Browser Automation* [online]. [cit. 2015-06-19]. Dostupné z: <<http://www.seleniumhq.org>>.
- [17] SHIBBOLETH CONSORTIUM. *Shibboleth* [online]. [cit. 2015-06-10]. Dostupné z: <<https://shibboleth.net>>.
- [18] SNOZOVÁ, Martina. *Heuristická analýza* [online]. 2013 [cit. 2015-05-21]. Dostupné z: <<http://www.inflow.cz/heuristicka-analyza>>.
- [19] STEIN, René. *Návrh aplikací v jazyce UML* [online]. 2014 [cit. 2015-06-01]. Dostupné z: <<https://www.interval.cz/clanky/navrh-aplikaci-v-jazyce-uml-zaciname-s-pripady-uziti>>
- [20] SWITCH. *Shib SP Installation - Service Provider - Guides - SWITCHaai - SWITCH* [online]. [cit. 2015-06-20]. Dostupné z: <<https://www.switch.ch/aai/guides/sp/installation/?os=debian>>.
- [21] ŠTRUPL, Petr. *Shibboleth SP 2.5 - Configure - ČVUT IT návody - VICwiki* [online]. 2015 [cit. 2015-06-20]. Dostupné z: <<https://wiki.cvut.cz/confluence/display/CVUTNAVODY/Shibboleth+SP+2.5+-+Configure>>
- [22] THE PHP GROUP. *PHP: Hypertext Preprocessor* [online]. 2001 [cit. 2015-06-10]. Dostupné z: <<http://php.net/>>.
- [23] TORVALDS, Linus, Junio HAMANO a další. *Git* [online]. [cit. 2015-06-10]. Dostupné z: <<https://git-scm.com>>.
- [24] VRÁNA, Jakub. *Adminer - Správa databáze v jednom PHP souboru* [online]. [cit. 2015-06-10]. Dostupné z: <<http://www.adminer.org>>.
- [25] VOTRUBA, Tomáš. *Načítání modelu s NotORM, DI a inject* [online]. [cit. 2015-05-29]. Dostupné z: <<http://pla.nette.org/cs/nacitani-modelu-s-notorm-a-dependency-injection>>.

Seznam použitých zkratek

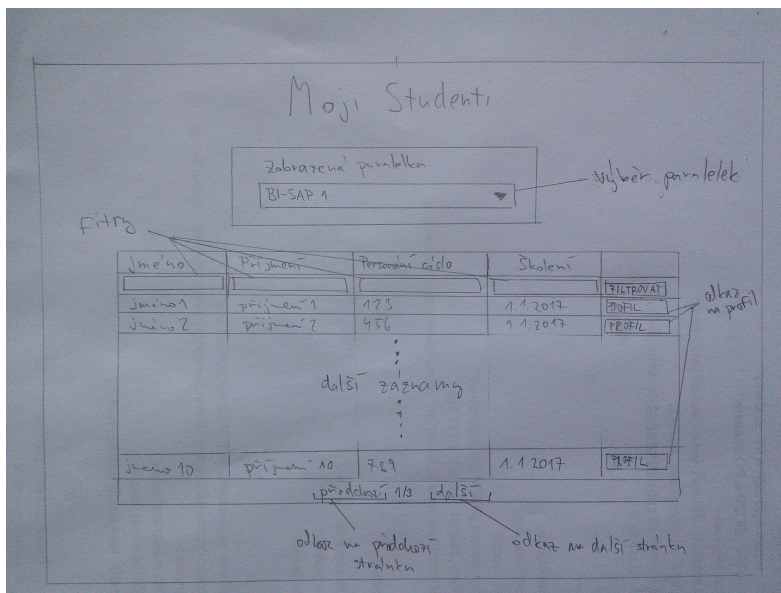
- API** Application program interface
- APT** Advanced packaging tool
- BOZP** Bezpečnost a ochrana zdraví v práci
- CLI** Command-line interface
- CRUD** Create, read, update, delete
- CSRF** Cross-site request forgery
- CHUL** Children usability lab
- ČVUT** České vysoké učení technické
- FIT** Fakulta informačních technologií
- IDE** Integrated development environment
- IdP** Identity provider
- LDAP** Lightweight directory access protocol
- NBDT** Nette database table
- REST** Representational state transfer
- RPM** RPM Package Manager
- SP** Service provider
- SSL** Secure sockets layer
- UML** Unified modeling language
- XSS** Cross-site scripting

Tabulky a obrázky

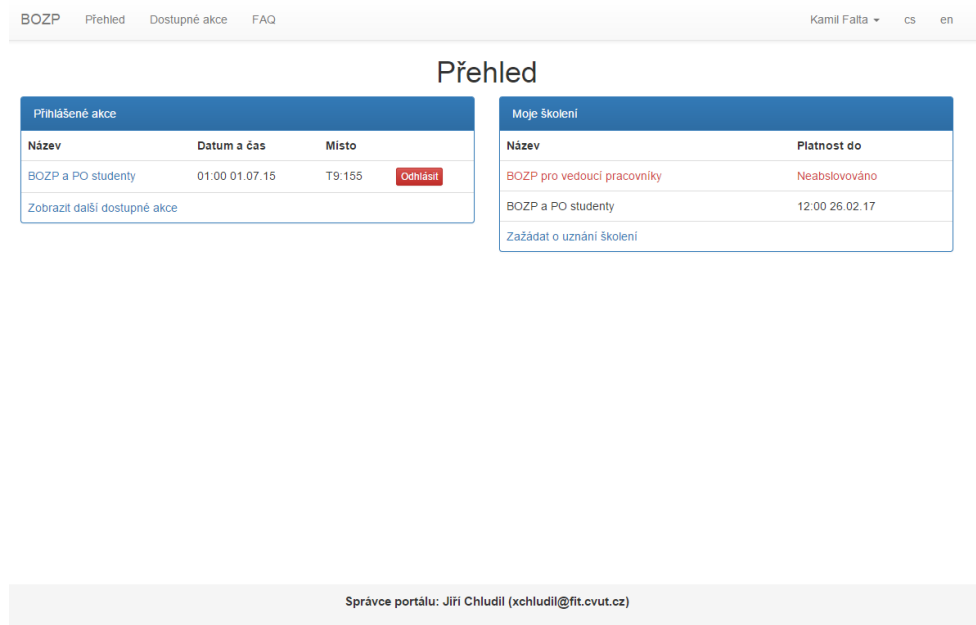
Tabulka B.1: Obrazovky aplikace

Aktuality
FAQ
Přehled
Dostupné akce
Profil
Správa sledovaných předmětů
Správa akcí
Vytvoření/editace akce
Zobrazení akce
Vytvoření/editace článku
Vytvoření požadavku
Zobrazení požadavku
Seznam požadavků
Statistiky uživatelů
Vytvoření/editace typů školení
Správa typů školení
Zobrazení typů školení
Vytvoření uživatele
Editace uživatele
Správa uživatelů
Moji studenti
Správa rolí
Změna role
Správa uživatelských skupin
Udělení školení

B. TABULKY A OBRÁZKY



Obrázek B.1: Návrh obrazovky Moji studenti



Obrázek B.2: Ukázka obrazovky Přehled

BOZP Přehled Školení Uživatelé FAQ Nastavení Jíří Chudil cs en

Moji studenti

Zobrazená paralelka

BI-SAP 6 (Laboratoř) ▼

[Zobrazit paralelku](#)

Jméno	Příjmení	Personální číslo	Typ školení(Vyhlaška 50/78 Sb. 4)	
			Všechny ▼	Filter
█	█	█	14. 11. 2015	Profil
█	█	█	11. 11. 2015	Profil
█	█	█	10. 11. 2016	Profil
█	█	█	10. 11. 2016	Profil
█	█	█	01. 12. 2016	Profil
█	█	█	27. 11. 2016	Profil
█	█	█	10. 11. 2016	Profil
█	█	█	10. 11. 2016	Profil
█	█	█	10. 11. 2016	Profil
█	█	█	13. 11. 2016	Profil

« First « Previous **1 / 3** Next » Last »

Obrázek B.3: Ukázka obrazovky Moji studenti

The image shows a user testing session. On the left, a man is seated at a desk with a laptop and a camera on a tripod. On the right, a close-up of the laptop screen displays the 'Správa typů školení' (Training Type Management) interface. The interface includes a table with columns for 'Název', 'Ověřovací skupina', and 'Platnost', along with 'Upravit' and 'Odstranit' buttons for each entry.

Název	Ověřovací skupina	Platnost	Upravit	Odstranit
BOZP pro vedoucí pracovníky	vedoucí	24 měsíců	Upravit	Odstranit
PO pro vedoucí pracovníky	vedoucí	24 měsíců	Upravit	Odstranit
BOZP pro pracovníky	zaměstnaní	24 měsíců	Upravit	Odstranit
PO pro pracovníky	zaměstnaní	24 měsíců	Upravit	Odstranit
test	Všechny skupiny	1 měsíc	Upravit	Odstranit
BOZP a PO studenti	studenti	24 měsíců	Upravit	Odstranit
Vyhlaška 50/78 Sb. 4	Všechny skupiny	24 měsíců	Upravit	Odstranit

Obrázek B.4: Ukázka 2 z uživatelského testování

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
├─ impl.....	zdrojové kódy implementace
├─ thesis	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
tests.....	soubory spojené s testováním aplikace
text	text práce
├─ DP_Falta_Kamil_2015.pdf	text práce ve formátu PDF