

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

E-learningový systém se zameřením na testování

Robert Soják

Vedoucí práce: Ing. Ondřej Macek

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Softwarové inženýrství

10. dubna 2012

Poděkování

Zde můžete napsat své poděkování, pokud chcete a máte ke komu děkovat.

Prohlášení

Prohláším, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažných důvodů proti uložení tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne

Abstract

Translation of Czech abstract into English.

Abstrakt

Abstrakt práce by měl velmi stručně vystihovat její podstatu. Tedy čím se práce zabývá a co je jejím výsledkem/přířnosem.
Očekávají se cca 1 – 2 odstavce, maximálně pět stránek.

Obsah

1	Úvod	1
2	Popis problému, specifikace cíle	2
2.1	Cíl projektu	2
2.2	Využití	2
3	Analýza a návrh řešení	4
3.1	Požadavky na systém	4
3.1.1	Požadavky zadavatele	4
3.1.1.1	Funkční požadavky	4
3.1.1.2	Nefunkční požadavky	4
3.1.2	Rozšiřující požadavky	4
3.1.2.1	Funkční požadavky	4
3.1.2.2	Nefunkční požadavky	5
3.2	Rešerše existujících řešení	5
3.2.1	Google Formuláře	5
3.2.2	Moodle	6
3.2.3	Studentova berlička	6
3.2.4	Edubase	6
3.2.5	Závěr	6
3.3	Diagram užití	6
3.4	Meta model	6
3.5	Datová vrstva	6
3.6	Architektura systému	6
4	Implementace	8
4.1	Datová vrstva	8
4.2	Bezpečnost	8
4.2.1	Bezpečnost přenášených dat	8
4.2.2	Bezpečnost interakce	9
5	Testování	10
5.1	Unit testy	10
5.2	Selenium testy	10
5.3	Výkonové testy	10

6	Použité technologie a frameworky	11
6.1	Entity Framework	11
6.2	Unity Application Block	11
6.3	Porovnání ASP.NET MVC a ASP.NET Web Forms	11
7	Závěr	12
A	Seznam použitých pojmů	14
B	Seznam použitých zkratk	15
C	UML diagramy	16
D	Instalace a uživatelská příručka	17
E	Obsah přílohy CD	18

Seznam obrázků

3.1	Hlavní stránka Google Dokumentů	5
3.2	Ukázka Google Formuláře: tvorba vlevo, výsledek vpravo	5
3.3	Diagram užití	6
3.4	Meta model	6
3.5	Design model datové vrstvy	6
3.6	Stavový diagram formuláře	7
4.1	Entity Data Model	8

Seznam tabulek

Seznam zdrojových kódů

4.1	Použití atributu <code>AuthorizeUserType</code>	9
4.2	Ověřování v bussiness vrstvě	9

Kapitola 1

Úvod

Tento projekt e-learningového systému vznikl na základě poptávky neziskové organizace. Má sloužit jako nástroj pro naučné testování účastníků (dále *studentů*) kurzů, které organizace pořádá.

V současné době organizace využívá pro testování Google Formulářů¹ s testovými otázkami, kde student vybírá právě jednu odpověď z několika možných. Skrze generovanou tabulku odpovědí jednotlivých studentů je sice koordinátor kurzu (dále *lektor*) schopen vyhodnotit správnost odpovědí, ale to je vše, co mu aplikace Formulářů umožňuje. Výsledky je nucen ručně vyhodnotit a danému studentovi nastínit správné odpovědi nebo alespoň odkázat na související kapitolu v učebním materiálu (dále *skripta*).

Pro kategorizaci studentů a příslušných lektorů je v organizaci využito Google Skupin.

¹Názvem Google Formulář značím formulář textového procesoru z rodiny Google Dokumentů

Kapitola 2

Popis problému, specifikace cíle

Jak již z výše nastíněného popisu vyplývá, organizace potřebuje aplikaci pro tvorbu, vyplnění, okamžité vyhodnocení a okomentování testů.

Lektor kurzu vytvoří šablonu testu s otázkami o několika možných odpovědích. Ke každé otázce by měl mít možnost napsat slovní komentář, ve kterém může nastínit, proč je daná odpověď správně nebo špatně a případně odkázat na studijní materiál, který danou problematiku vysvětluje. Každá otázka k sobě může vázat otázky alternativní, které test inovují při vypňování stejného testu vícekrát.

Student by měl mít vždy přehled o dostupných testech, které jsou určené pro skupiny, do nichž je zařazen. Po konkrétním výběru se pro něj test vygeneruje z dostupných otázek a jejich alternativ. Vždy, když testovaný označí svou odpověď, zobrazení se vysvětlení správné odpovědi a dojde k jejímu viditelnému zvýraznění. Po vyplnění celého testu či pouze jeho části má student možnost odeslat svůj výsledek nadřiznému lektorovi. Ten má ve výsledku přehled, jak na tom jeho svěřenci se znalostmi jsou.

2.1 Cíl projektu

Cílem projektu je vyvinout portál pro podporu výuky. Hlavní funkcionalitou by měla být tvorba a vyplňování naučných testů určených skupinám studentů. Systém by měl také podporovat tvorbu a správu elektronických skript.

Jakožto bakalářská práce by měl projekt také ukázat mou schopnost samostatně řešit daný problém. Chtěl bych předvést nabyté znalosti a schopnost využít moderní technologie při jeho realizaci.

2.2 Využití

Výsledný systém by měl být nasazen do zmíněné poptávající neziskové organizace.

Má ale i širší využití, protože dává za vznik univerzálnímu e-learningovému portálu. Systém naučných testů je zde svým způsobem originální a odlišuje portál od ostatních aplikací tohoto typu.

Já osobně bych chtěl portál využít jako nástroj pro podporu výuky studentů informatiky na gymnáziu, kde působím jako učitel. Očekávám od něj efektivní pomoc při tvorbě a známkování testů a sdílení studijních materiálů.

Může být přínosem i pro další předměty jako jednoduše ovladatelný nástroj pro tvorbu tištěných testů či dotazníků, které jsou povětšinou těžkopádně tvořeny v textovém procesoru. Nastavil by se tím i jednotný vzhled, který bývá test od testu odlišný.

Kapitola 3

Analýza a návrh řešení

3.1 Požadavky na systém

Požadavky na výsledný e-learningový systém sestávají ze základních požadavků zadávající organizace a rozšiřujících požadavků, které byly sestaveny na základě obecných požadavků na systém takového typu a inspirací zmíněných v rešerši v následující sekci 3.2. Díky těmto rozšiřujícím požadavkům by měl vzniknout zmíněný komplexnější univerzální e-learningový systém.

3.1.1 Požadavky zadavatele

3.1.1.1 Funkční požadavky

- FP1. Zobrazení komentáře k právě zodpovězené otázce (cca odstavcový)
- FP2. Zobrazení výsledku testu a možnost jeho odeslání lektorovi kurzu
- FP3. Náhodný výběr variace otázky při generování testu
- FP4. Neomezený počet možností daný test opakovat dokud nebude zodpovězen správně
- FP5. Možnost kategorizace studentů a příslušných lektorů

3.1.1.2 Nefunkční požadavky

- NP1. Jednoduché ovládání, pochopitelné pro běžného uživatele i bez zaškolení

3.1.2 Rozšiřující požadavky

3.1.2.1 Funkční požadavky

- FP6. Rozšíření o více typů testů (obecně *formulářů*)
- FP7. Možnost výběru z více typů otázek

FP8. Lektor může ohodnotit přijatý formulář

FP9. Podpora exportu výsledků formulářů do Google Formulářů

FP10. Možnost tvorby elektronických skript

3.1.2.2 Nefunkční požadavky

NP2. Výsledný systém bude webová aplikace

3.2 Rešerše existujících řešení

Rešerši existujících řešení e-learningových a tvorbou formulářů zabývajících se aplikací jsem neprováděl za účelem nalezení kandidáta na rozšíření o požadovanou funkcionalitu. Rešerši jsem dělal s cílem prozkoumat dostupná řešení, zjistit co poskytují za služby a třeba je využít pro inspiraci. Projekt byl již od začátku zamýšlen pro vznik od začátku, takřka jíc na zelené louce. Jedním z důvodů je funkcionalita speciálního naučného testu. Dalším důvodem je požadavek na jednoduché ovládání a zkušenost uživatelů ze zadávající společnosti s prostředím Google Formulářů. Hlavním důvodem je ale myslím moje snaha zapojit do vývoje znalosti softwarového inženýrství a touha vytvořit si systém vlastní. Nechtěl jsem jen vybrat z předpřipraveného řešení s hotovou architekturou a přibalit k ní plugin pro práci s naučnými testy.

3.2.1 Google Formuláře

Jako Google Formulář v této bakalářské práci označuji formulář textového procesoru z balíku Google Dokumenty[1]. Pomocí poskytovaného nástroje je uživateli umožněno vytvořit vlastní formulář z několika možných formulářových prvků. Hotový formulář lze poté zveřejnit (zcela nebo jen pro určitý okruh uživatelů) a nechat uživateli vyplnit. Odpovědi jsou automaticky zapisovány do tabulky, kde jsou spolu s dalšími informacemi (např. datum a čas vyplnění) připravené pro další analýzu.

Základní myšlenka tohoto projektu vychází právě z Google Formulářů, proto zde jejich rozebrání věnuji více prostoru.

Obrázek 3.1: Hlavní stránka Google Dokumentů

Obrázek 3.2: Ukázka Google Formuláře: tvorba vlevo, výsledek vpravo

3.2.2 Moodle

3.2.3 Studentova berlička

3.2.4 Edubase

3.2.5 Závěr

3.3 Diagram užití

Na diagramu [3.3](#) jsou zachyceny základní operace prováděné uživateli v daných rolích.

Obrázek 3.3: Diagram užití

3.4 Meta model

Obrázek 3.4: Meta model

3.5 Datová vrstva

Obrázek 3.5: Design model datové vrstvy

3.6 Architektura systému

Obrázek 3.6: Stavový diagram formuláře

Kapitola 4

Implementace

4.1 Datová vrstva

Obrázek 4.1: Entity Data Model

4.2 Bezpečnost

Na bezpečnost této webové aplikace se dívám ze dvou směrů, a to z hlediska bezpečnosti přenášených dat mezi klientem a serverem a z hlediska bezpečnosti v interakci uživatele s aplikací. Jsou zde možná i další hlediska, například komunikace webového serveru s databázovým serverem, ale tu zde myslím není třeba rozebírat. Pro jednoduchost předpokládám, že aplikace i databáze běží na totožném serveru.

4.2.1 Bezpečnost přenášených dat

Jedná se o bezpečnost přenosu veškerých dat, která se přenášejí po síti (ať už po LAN či po internetu), když uživatel interaguje s aplikací. Nejdůležitější je asi moment, kdy se uživatel do aplikace přihlašuje. Odesílá své uživatelské jméno a heslo, díky čemuž je systémem identifikován a autorizován k provádění jednotlivých akcí.

Když přihlašovací údaje zachytí útočník (útokem známým jako MITM), získává tím identitu pravého uživatele a přístup ke všem jeho datům v systému. To dělá tento útok nejnebezpečnějším ze všech. Když se navíc útočníkovi podaří získat přihlašovací údaje administrátora systému, dostane se rázem k veškerým datům v systému. Navíc pro export do Google Formulářů musí uživatel zadat údaje ke svému Google účtu, takže v případě odposlechnutí dojde k narušení bezpečnosti i ve zcela jiném systému než je tento.

I když se ale útočníkovi nepodaří zachytit přihlašovací údaje, stále má možnost se hodně věcí dozvědět. Při zachycení komunikace vidí všechny informace, které si uživatel v danou chvíli prohlíží či zadává.

Naštěstí se dá útokům lehce zabránit, a to zavedením komunikace přes HTTPS namísto HTTP, což se řeší na úrovni webového serveru. Při komunikaci přes HTTPS je využito protokolu SSL, který za pomoci serverového certifikátu veškerou komunikaci šifruje.

4.2.2 Bezpečnost interakce

Zde se jedná o bezpečnost interakce uživatele s rozhraním aplikace, tzn. zobrazování informací, přidávání a úprava dat, to vše podmíněné stupněm oprávnění uživatele a jeho přiřazení do skupin.

V aplikaci je implementováno „dvoustupňové“ zabezpečení. Prvním stupněm je kontrola oprávnění uživatele na úrovni prezentační vrstvy. Přístup na jednotlivé stránky je regulován pomocí zavedeného atributu `AuthorizeUserType`, který jsem připojil ke všem metodám controllerů. Předáním parametru `UserType` je určen nejnižší stupeň oprávnění pro přístup na danou url zastupovanou názvem metody. Při neoprávněném pokusu o přístup na url je atributem vyhozena výjimka typu `PermissionException`. Běžně by se uživatel na takovou url vůbec dostat neměl, protože stejná politika je implementována ve views, čímž adresa takové url není uživateli dostupná.

```
[AuthorizeUserType(UserType = UserTypes.Lector)]
public ActionResult Create()
{
    ...
}
```

Zdrojový kód 4.1: Použití atributu `AuthorizeUserType`

Druhým stupněm je kontrola oprávnění v samotné bussiness vrstvě. Všechny operace nad daty zde obsahují ověřují práva na vykonání pomocí `UserPermissions`, testují jestli požadovanou položku uživatel vytvořil či patří do jeho skupiny atd. Pokud práva nemá, vznikne opět výjimka `PermissionException`.

```
public Form GetForm(int id)
{
    var result = GetSingle(f => f.ID == id);
    if (result == null)
    {
        if (Context.Form.Count(f => f.ID == id) > 0)
            throw new PermissionException("Form_Get");
        else
            throw new ArgumentException("Form not found");
    }

    return result;
}
```

Zdrojový kód 4.2: Ověřování v bussiness vrstvě

Kapitola 5

Testování

5.1 Unit testy

5.2 Selenium testy

5.3 Výkonové testy

Kapitola 6

Použité technologie a frameworky

6.1 Entity Framework

6.2 Unity Application Block

6.3 Porovnání ASP.NET MVC a ASP.NET Web Forms

Kapitola 7

ZÁVĚR

- Zhodnocení splnění cílů DP/BP a vlastního přínosu práce (pří formulaci je třeba vzít v potaz zadání práce).
- Diskuse dalšího možného pokračování práce.

Literatura

- [1] web:GoogleDocs. Google Dokumenty — hlavní stránka.
<http://docs.google.com/>.

Příloha A

Seznam použitých pojmů

E-learning

Vzdělávání za pomoci moderních elektronických informačních a komunikačních technologií

⋮

Příloha B

Seznam použitých zkratek

LAN Local Area Network

MITM Man In The Middle

HTTP HyperText Transfer Protocol

HTTPS HyperText Transfer Protocol Secure

⋮

Příloha C

UML diagramy

Příloha D

Instalační a uživatelská příručka

Příloha E

Obsah přílohy E