

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 závažný důvod 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

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Popis problému, specifikace cíle</b>	<b>2</b>
2.1	Cíl projektu . . . . .	2
2.2	Využití . . . . .	2
<b>3</b>	<b>Analýza</b>	<b>4</b>
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 . . . . .	5
3.2.3	Studentova berlička . . . . .	5
3.2.4	Edubase . . . . .	5
3.2.5	Závěr . . . . .	5
3.3	Formuláře . . . . .	5
3.4	Otázky . . . . .	6
3.5	Uživatelské role . . . . .	7
3.6	Případy užití . . . . .	8
<b>4</b>	<b>Návrh</b>	<b>9</b>
4.1	Architektura systému . . . . .	9
4.2	Meta model . . . . .	9
4.3	Datová vrstva . . . . .	9
<b>5</b>	<b>Implementace</b>	<b>11</b>
5.1	Datová vrstva . . . . .	11
5.2	Bezpečnost . . . . .	11
5.2.1	Bezpečnost přenášených dat . . . . .	11
5.2.2	Bezpečnost interakce . . . . .	12

<b>6</b>	<b>Testování</b>	<b>14</b>
6.1	Unit testy . . . . .	14
6.2	Selenium testy . . . . .	14
6.3	Výkonové testy . . . . .	14
<b>7</b>	<b>Nasazení</b>	<b>15</b>
<b>8</b>	<b>Použité technologie a frameworky</b>	<b>16</b>
8.1	Entity Framework . . . . .	16
8.2	Unity Application Block . . . . .	16
8.3	Porovnání ASP.NET MVC a ASP.NET Web Forms . . . . .	16
<b>9</b>	<b>Závěr</b>	<b>17</b>
9.1	Možná rozšíření . . . . .	17
9.1.1	Otázky . . . . .	17
<b>A</b>	<b>Seznam použitých pojmů</b>	<b>19</b>
<b>B</b>	<b>Seznam použitých zkratk</b>	<b>20</b>
<b>C</b>	<b>UML diagramy</b>	<b>21</b>
<b>D</b>	<b>Instalační a uživatelská příručka</b>	<b>22</b>
<b>E</b>	<b>Obsah příloženého CD</b>	<b>23</b>

# Seznam obrázků

3.1	Hlavní stránka Google Dokumentů . . . . .	6
3.2	Ukázka Google Formuláře: tvorba vlevo, výsledek vpravo . . . . .	7
3.3	Diagram užití . . . . .	8
4.1	Meta model . . . . .	9
4.2	Design model datové vrstvy . . . . .	10
4.3	Stavový diagram formuláře . . . . .	10
5.1	Entity Data Model . . . . .	13



# Seznam tabulek

3.1	Možnosti chování formulářů . . . . .	6
-----	--------------------------------------	---



# Seznam zdrojových kódů

5.1	Použití atributu <code>AuthorizeUserType</code> . . . . .	12
5.2	Ověřování v business vrstvě . . . . .	12





# 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 Formulářů<sup>1</sup> 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.

---

<sup>1</sup>Název Google Formulář zde 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 nadřízenému 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

### 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. Každá otázka může sestávat z několika variací

FP4. Náhodný výběr variace otázky při generování testu

FP5. Neomezený počet možností daný test opakovat dokud nebude zodpovězen správně

FP6. 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

FP7. Rozšíření o více typů testů (obecně *formulářů*)

FP8. Možnost výběru z více typů otázek

FP9. Lektor může ohodnotit přijatý formulář

FP10. Podpora exportu výsledků formulářů do Google Formulářů

FP11. 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, takříkajíc 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.

### 3.2.2 Moodle

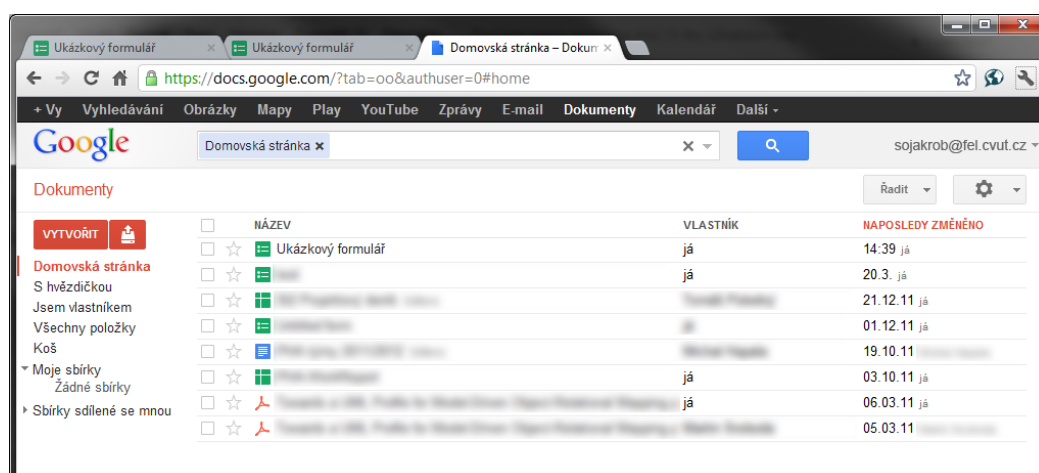
### 3.2.3 Studentova berlička

### 3.2.4 Edubase

### 3.2.5 Závěr

## 3.3 Formuláře

Základní množinu formulářů systému tvoří tyto tři typy:



Obrázek 3.1: Hlavní stránka Google Dokumentů

**Naučný test** Je to netradiční typ testu vycházející z požadavků zadavatele. Ke každé otázce je možné přidat vysvětlení správné odpovědi (FP1), které je vyplňujícímu studentovi zobrazeno po označení odpovědi, kterou považuje za správnou. Test tedy plní dvě funkce, vypovídá o současných znalostech studenta a připravenost na možný zkouškový test, a zároveň ho opravuje a směřuje správnou cestou. Test je možné vyplnit libovolněkrát. Díky alternativním otázkám (FP3) není pokaždé stejný.

**Zkouškový test** Toto je běžný test pro ostré otestování studentových znalostí, jak ho známe ze všech vzdělávacích institucí. Může mít navíc určen například časový limit pro dokončení či možnost více pokusů pro správné vyplnění.

**Dotazník** Plní funkci běžného dotazníku. Každý ho smí vyplnit pouze jedenkrát.

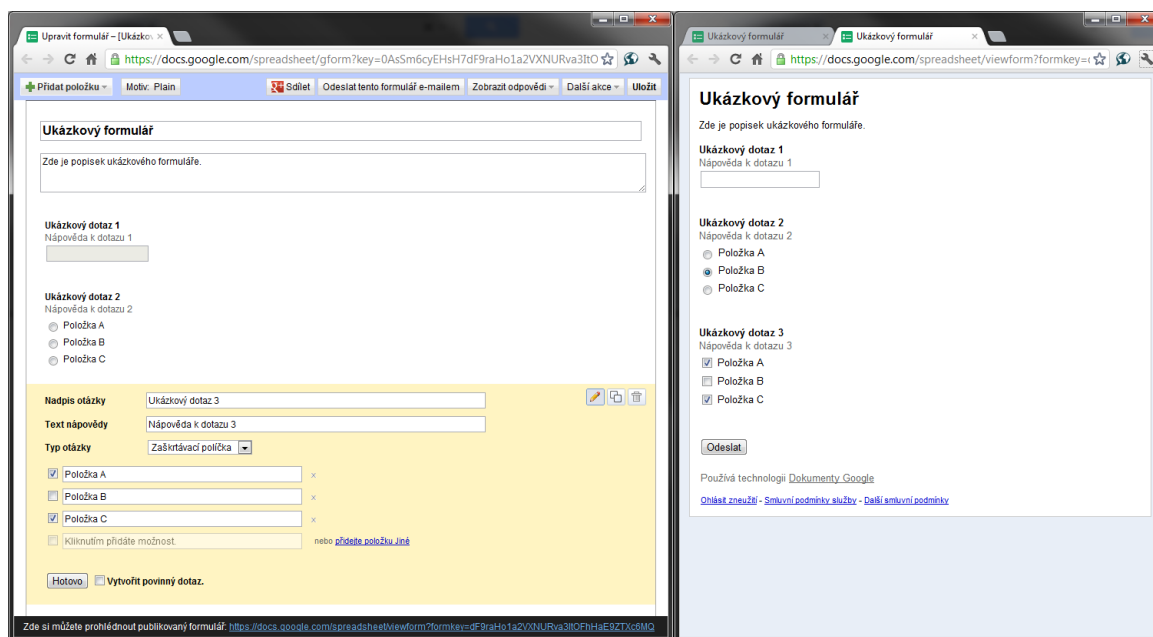
Následující tabulka 3.1 ukazuje možnosti nastavení chování jednotlivých typů formulářů.

	Naučný test	Zkušební test	Dotazník
Čas na vyplnění	✓	✓	✗
Zamíchání otázek	✓	✓	✗
Zamíchání nabízených odpovědí	✓	✓	✗
Vysvětlení správné odpovědi	✓	✗	✗
Možnost vyplnit libovolněkrát	✓	✓	✗

Tabulka 3.1: Možnosti chování formulářů

## 3.4 Otázky

Množina typů jednotlivých otázek je sestavena z potřeby zadavatele, čímž je otázka výběrová, a základních potřeb univerzálního e-learningového systému. Množina tedy obsahuje 5



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

typů otázek, z čehož 3 typy jsou základní. Možné další otázky vhodné pro rozšíření systému jsou nastíněny v sekci 9.1.1.

### Výběrová otázka

**Jedna správná odpověď** Právě jedna z množiny nabízených odpovědí je správná.

**Více správných odpovědí** Žádná nebo více z nabízených odpovědí je správná.

### Textová otázka

**Krátká** Od vyplňujícího studenta se očekává jednoslovná či jednovětná odpověď.

**Dlouhá** Zde se očekává vyčerpávající odpověď, tzn. jedna a více vět.

**Rozsahová otázka** Odpověď je třeba vybrat z definovaného rozsahu, např. známka 1–5, hodnocení 0–10. Tento typ otázky je hlavně pro potřebu dotazníku.

## 3.5 Uživatelské role

Uživatelské role potřebné v systému jasně plynou již ze zadání, jsou to tyto tři role:

**Student** Představuje účastníka kurzu z domény zadavatele a obecně studenta využívajícího e-learningového systému. Z hlediska práv je na nejnižší úrovni, tzn. všechny jeho akce se vztahují pouze k němu samotnému. Mezi akce patří například vyplňování testů dostupné pro jeho skupinu a prohlížení výsledků svých testů (více viz. sekce 3.6).

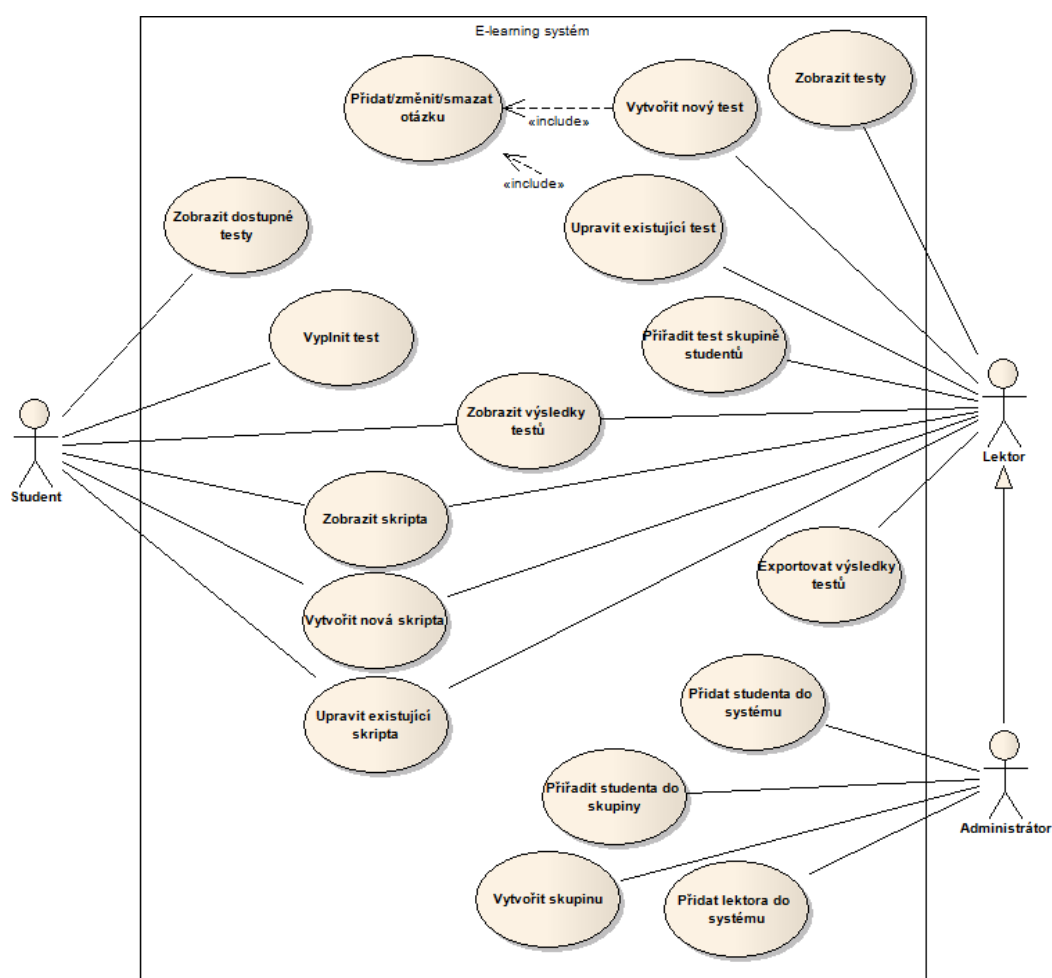


**Lektor** Lektorem je koordinátor kurzu z domény zadavatele, vedoucí jedné či více skupin (neboli kurzů). Obecně se tedy jedná o učitele studentů, který má na starost tvorbu testů pro jeho skupiny a jejich následné ohodnocení.

**Administrátor** Uživatelská role známá asi ze všech systémů. Je to správce celého portálu a jako takový má práva všech lektorů a k tomu dostupnou funkcionalitu pro administraci aplikace včetně přidávání uživatelů a tvorby skupin.

### 3.6 Případy užití

Na diagramu 3.3 jsou zachyceny základní operace prováděné uživateli v daných rolích zastoupených aktéry případů užití.



Obrázek 3.3: Diagram užití

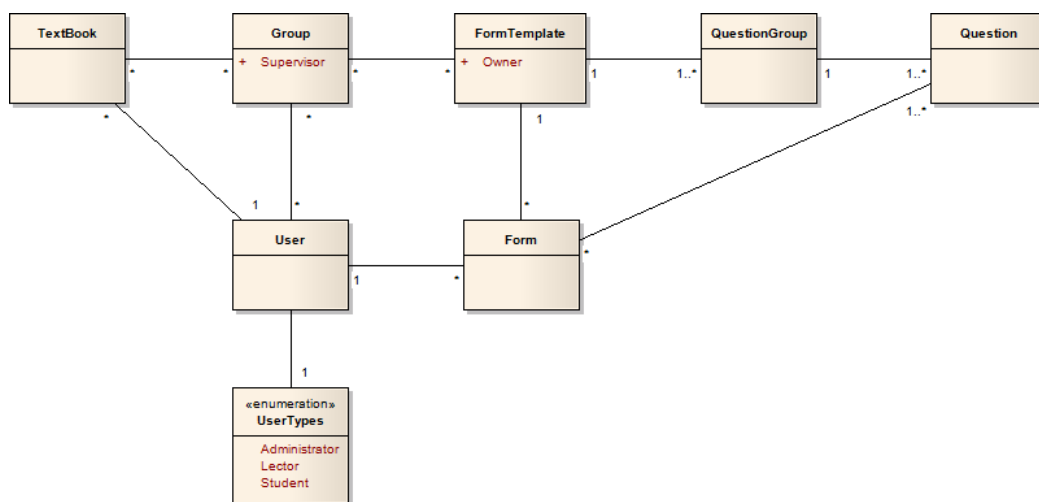


# Kapitola 4

## Návrh

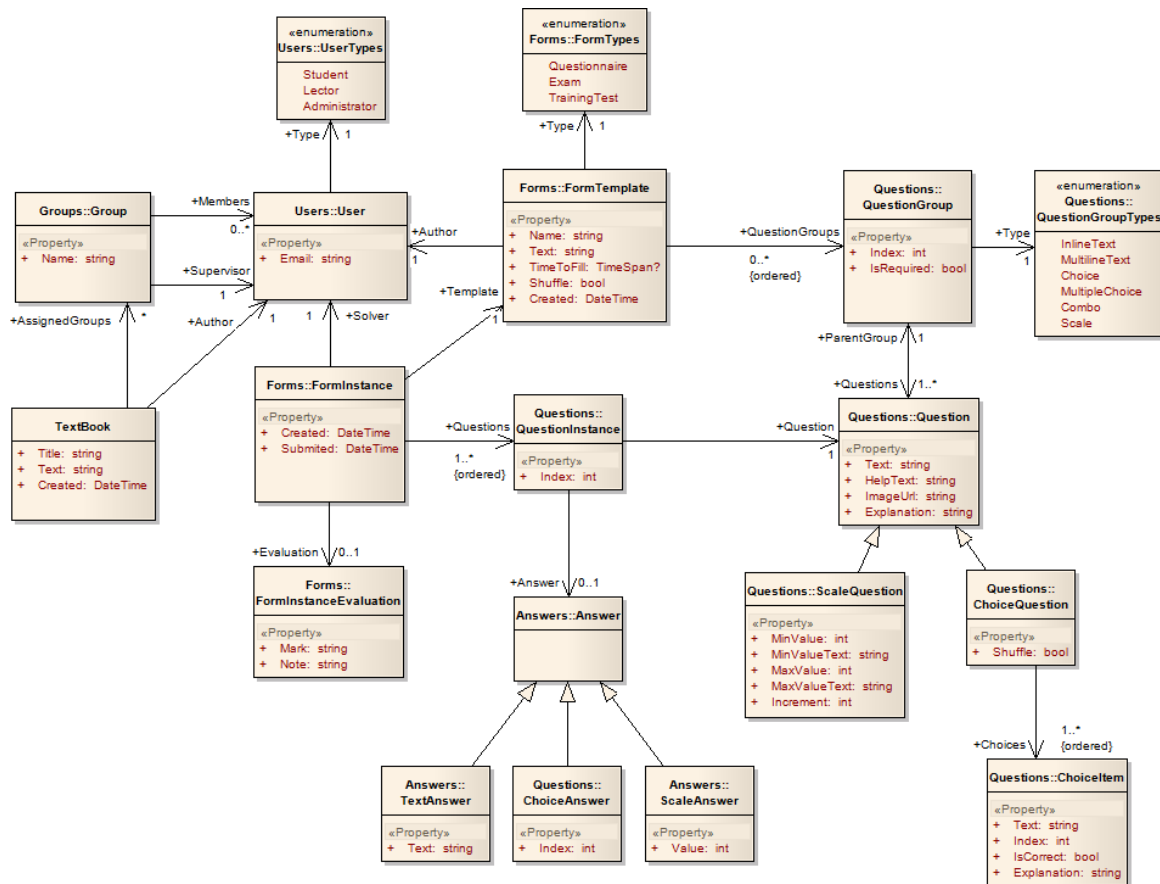
### 4.1 Architektura systému

### 4.2 Meta model

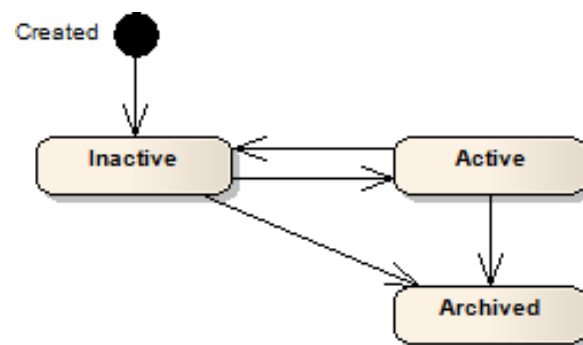


Obrázek 4.1: Meta model

### 4.3 Datová vrstva



Obrázek 4.2: Design model datové vrstvy



Obrázek 4.3: Stavový diagram formuláře

## Kapitola 5

# Implementace

### 5.1 Datová vrstva

### 5.2 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.

#### 5.2.1 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 s využitím serverového certifikátu veškerou komunikaci šifruje.

### 5.2.2 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()
{
    ...
}
```

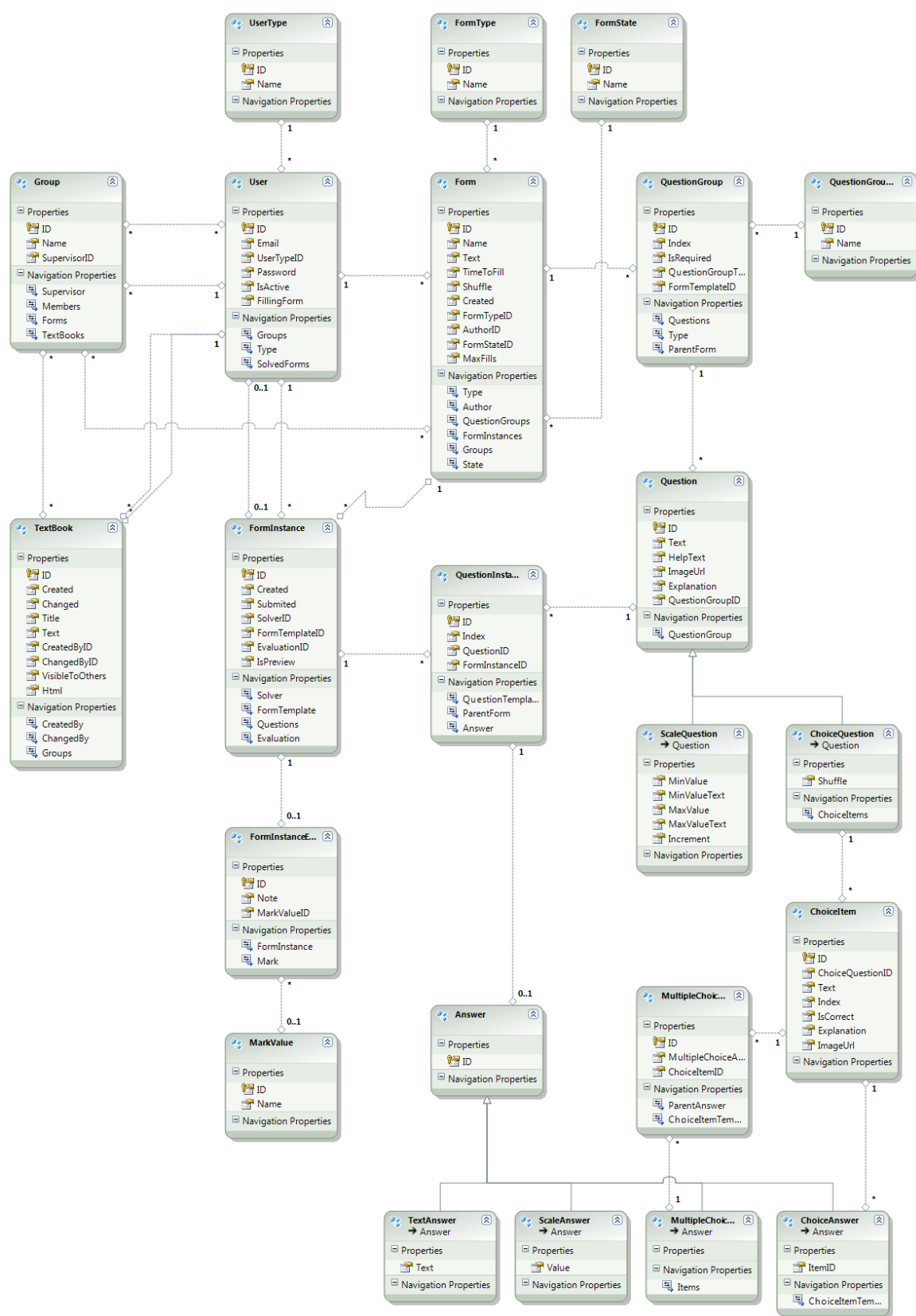
Zdrojový kód 5.1: Použití atributu `AuthorizeUserType`

Druhým stupněm je kontrola oprávnění v samotné business 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 5.2: Ověřování v business vrstvě



Obrázek 5.1: Entity Data Model





## Kapitola 6

# Testování

### 6.1 Unit testy

### 6.2 Selenium testy

### 6.3 Výkonové testy



## Kapitola 7

## Nasazení



## Kapitola 8

# Použité technologie a frameworky

### 8.1 Entity Framework

### 8.2 Unity Application Block

### 8.3 Porovnání ASP.NET MVC a ASP.NET Web Forms



## Kapitola 9

### Závěr

#### 9.1 Možná rozšíření

##### 9.1.1 Otázky

Možnost přiřadit k textu otázky obrázek.





# 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 zkratk

**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